



# **UNIVERSIDAD TÉCNICA DEL NORTE**

## **FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**TEMA:**

**DISEÑO DE UN SISTEMA DE CONTROL DE ACCESO PARA LAS AULAS DE  
LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS UTILIZANDO  
DISPOSITIVOS ELECTRÓNICOS Y TECNOLOGÍA INALÁMBRICA EN UNA  
RED WPAN.**

**AUTOR: ANTONIO BOLÍVAR GUERRERO ASEICHA**

**DIRECTOR: Ing. OMAR OÑA.**

**Ibarra-Ecuador**

**Abril del 2016**



**UNIVERSIDAD TÉCNICA DEL NORTE  
BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN  
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

**1. IDENTIFICACIÓN DE LA OBRA**

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

<b>DATOS DE CONTACTO</b>			
<b>CÉDULA DE IDENTIDAD:</b>	1003788401		
<b>APELLIDOS Y NOMBRES:</b>	Guerrero Aseicha Antonio Bolívar		
<b>DIRECCIÓN:</b>	Miguel Sánchez y Tobías Mena		
<b>EMAIL:</b>	<a href="mailto:tonny_g520@hotmail.com">tonny_g520@hotmail.com</a>		
<b>TELÉFONO FIJO:</b>	062585315	<b>TELÉFONO MÓVIL:</b>	0981162999

<b>DATOS DE LA OBRA</b>	
<b>TÍTULO:</b>	“DISEÑO DE UN SISTEMA DE CONTROL DE ACCESO PARA LAS AULAS DE LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS UTILIZANDO DISPOSITIVOS ELECTRÓNICOS Y TECNOLOGÍA INALÁMBRICA EN UNA RED WPAN”
<b>AUTOR (ES):</b>	Guerrero Aseicha Antonio Bolívar
<b>FECHA: AAAAMMDD</b>	2016/04/04
SOLO PARA TRABAJOS DE GRADO	
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSGRADO</b>
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniero en Electrónica y Redes de Comunicación
<b>ASESOR /DIRECTOR:</b>	Ing. Omar Oña

## 2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Antonio Bolívar Guerrero Aseicha, con cédula de identidad Nro.1003788401, en calidad de autor (es) y titular (es) de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 04 días del mes de Abril de 2016

### EL AUTOR:

(Firma).....  
Nombre: Guerrero Aseicha Antonio B.  
C.C.: 1003788401

(Firma).....  
Ing. Betty Chávez  
Encargada Biblioteca



## UNIVERSIDAD TÉCNICA DEL NORTE

### CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Guerrero Aseicha Antonio Bolívar, con cédula de identidad Nro. 1003788401, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado: **"DISEÑO DE UN SISTEMA DE CONTROL DE ACCESO PARA LAS AULAS DE LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS UTILIZANDO DISPOSITIVOS ELECTRÓNICOS Y TECNOLOGÍA INALÁMBRICA EN UNA RED WPAN"**, que ha sido desarrollado para optar por el título de Ingeniero en Electrónica y Redes de Comunicación en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 04 días del mes de Abril de 2016

(Firma).....

Nombre: Guerrero Aseicha Antonio Bolívar

C.C.: 1003788401





**UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CERTIFICACIÓN DEL DIRECTOR**

Certifico que bajo mi dirección el trabajo de grado: **“DISEÑO DE UN SISTEMA DE CONTROL DE ACCESO PARA LAS AULAS DE LA FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS UTILIZANDO DISPOSITIVOS ELECTRÓNICOS Y TECNOLOGÍA INALÁMBRICA EN UNA RED WPAN”**, fue desarrollado en su totalidad por el señor egresado Guerrero Aseicha Antonio Bolívar, previo a la obtención del título de Ingeniero en Electrónica y Redes de Comunicación.

Certifico que ha sido dirigido en todas sus partes, cumpliendo con todas las disposiciones emitidas por la Universidad Técnica del Norte en lo referente a la elaboración del trabajo de grado.

Por lo expuesto:

Autorizo su presentación ante los organismos competentes para sustentación del mismo.

Ibarra, a los 04 días del mes de Abril de 2016.

(Firma).....  
Ing. Omar Oña.  
**DIRECTOR DE TESIS**



**UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CONSTANCIA**

Yo, Guerrero Aseicha Antonio Bolívar, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado, ni calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de a presente declaración cedo los derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica del Norte, según lo establecido por la ley de Propiedad intelectual, por su reglamento y por la normatividad institucional vigente.

Ibarra, a los 04 días del mes de Abril del 2016.

**EL AUTOR:**

(Firma).....

Nombre: Guerrero Aseicha Antonio Bolívar  
C.C.: 1003788401



UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

*A Dios por mantenerme con la salud y la vida.*

*A mi madre Rosa Ascicha por darme la vida y todo el apoyo que me brinda siempre por su paciencia y sus consejos en momentos difíciles de mi carrera admiro su esfuerzo y fortaleza para hacer que salga adelante. Todo lo que ha hecho por mí me ha impulsado para que pueda cumplir con esta meta.*

*A mi padre Antonio Guerrero que me ha apoyado y por el esfuerzo que hace diariamente para sacar adelante a la familia.*

*A mi novia Katherine Benavides por su comprensión y apoyo indispensable estuviste a mi lado incluso en los momentos más difíciles de mi vida dándome ánimos y apoyándome en cada decisión por eso y más, quiero que sepas que te amo...!!!*



UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

*Agradesco infinitamente a mis padres que me dieron cuanto pudieron para que salga adelante e hicieron posible la culminación de mi carrera a mi hermana Carolina por su apoyo en momentos de necesidad.*

*Mis más sinceros agradecimientos al Ingeniero Omar Oña director del presente proyecto de titulación, por su paciencia y colaboración durante la realización del mismo. Agradesco a todas las personas que me apoyaron durante mi carrera y en la realización del presente proyecto de titulación en especial a mi compañero de trabajo Geovanny Romero quien me brindó su apoyo incondicional en momentos difíciles durante el proceso de este proyecto.*

*Agradesco a la "Universidad Técnica del Norte" por todas las experiencias adquiridas durante el proceso de enseñanza además por haberme permitido la realización de este proyecto de titulación en las instalaciones de la Facultad de Ingeniería en Ciencias Aplicadas.*

# ÍNDICE GENERAL

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	II
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	¡ERROR! MARCADOR NO DEFINIDO.
CERTIFICACIÓN DEL DIRECTOR.....	¡ERROR! MARCADOR NO DEFINIDO.
CONSTANCIA .....	¡ERROR! MARCADOR NO DEFINIDO.
DEDICATORIA .....	VII
AGRADECIMIENTO.....	VIII
RESUMEN .....	XXXII
ABSTRACT.....	XXXIII
CAPÍTULO I.....	1
ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS .....	1
1.1 INTRODUCCIÓN.....	1
1.2 ANÁLISIS DE LA SITUACIÓN ACTUAL .....	1
1.3 PLANTEAMIENTO DEL PROBLEMA.....	2
1.3.1 <i>Solución para los inconvenientes presentes en la facultad</i> .....	4
1.4 OBJETIVOS.....	6
1.4.1 <i>Objetivo general</i> .....	6
1.4.2 <i>Objetivos específicos</i> .....	7
1.5 ALCANCE .....	7
1.6 JUSTIFICACIÓN.....	10
CAPÍTULO II.....	12
ESTUDIO DE DISPOSITIVOS ELECTRÓNICOS Y TECNOLOGIAS INALÁMBRICAS .....	12
2.1 INTRODUCCIÓN.....	12

2.2	ESTUDIO DE DISPOSITIVOS ELECTRÓNICOS .....	12
2.2.1	<i>Microcontroladores.</i> .....	13
2.2.1.1	Fabricantes de microcontroladores. ....	13
2.2.1.1.1	Microchip. ....	14
2.2.1.1.2	Intel. ....	15
2.2.1.1.3	Motorola. ....	15
2.2.1.1.4	Atmel Corporation. ....	16
2.2.1.2	Recursos comunes de los microcontroladores. ....	17
2.2.1.2.1	Unidad Central de Procesamiento o CPU. ....	17
2.2.1.2.2	Memoria. ....	18
2.2.1.2.3	Líneas de entrada y salida. (E/S) .....	19
2.2.1.2.4	Reloj Principal. ....	19
2.2.1.2.5	Periféricos. ....	20
2.2.1.3	Recursos especiales de los microcontroladores. ....	20
2.2.1.3.1	Temporizadores o "timers" .....	20
2.2.1.3.2	Perro guardián o "watchdog". ....	21
2.2.1.3.3	Power up timer. ....	21
2.2.1.3.4	Protección ante fallo de alimentación o "brownout". ....	21
2.2.1.3.5	Estado de reposo o de bajo consumo. ....	21
2.2.1.4	Características para la selección del microcontrolador. ....	22
2.2.1.4.1	Aplicación del microcontrolador. ....	22
2.2.1.4.1.1	Procesamiento de datos. ....	22
2.2.1.4.1.2	Pines de Entradas/Salidas .....	23
2.2.1.4.1.3	Consumo de energía .....	23
2.2.1.4.1.4	Capacidad de memoria del microcontrolador .....	23
2.2.1.4.2	Lenguaje de programación. ....	24
2.2.1.4.3	Costo del microcontrolador. ....	24
2.2.1.4.4	Diseño de la placa .....	24
2.2.1.5	Arquitectura de un microcontrolador y tipos de procesadores. ....	25
2.2.1.5.1	Arquitectura Von Neumann. ....	25
2.2.1.5.2	Arquitectura Harvard. ....	26
2.2.1.5.3	Procesador segmentado. ....	26

2.2.1.5.4	Procesador según el número de instrucciones.....	27
2.2.1.6	Comparación de microcontroladores de diferentes fabricantes.....	28
2.2.1.6.1	Microcontrolador de microchip.....	29
2.2.1.6.1.1	Microcontrolador PIC18F4550.....	30
2.2.1.6.1.2	Principales características del PIC18F4550.....	31
2.2.1.6.2	Microcontrolador de Atmel.....	32
2.2.1.6.2.1	Microcontrolador ATmega164P.....	33
2.2.1.6.2.2	Principales características del ATmega164P.....	34
2.2.1.6.3	Microcontrolador de Intel.....	35
2.2.1.6.3.1	Microcontrolador Intel 8051.....	35
2.2.1.6.3.2	Principales características del Intel 8051.....	36
2.2.1.7	Selección del microcontroladores adecuado.....	36
2.2.2	<i>Elementos adicionales del hardware</i> .....	39
2.2.2.1	Relé.....	39
2.2.2.2	Transistores.....	40
2.2.2.2.1	Transistor de unión bipolar. (BJT).....	41
2.2.2.3	Resistencia.....	42
2.2.2.4	Potenciómetro.....	44
2.2.2.5	Diodos.....	45
2.2.2.6	Regulador de voltaje.....	46
2.2.2.7	Condensadores.....	46
2.2.2.8	Pantalla de cristal líquido (LCD).....	47
2.2.2.9	Teclado matricial.....	48
2.2.2.10	Disipadores de calor.....	49
2.2.2.11	Transformador.....	50
2.2.3	<i>Módulos inalámbricos</i> .....	52
2.2.3.1	Estándares inalámbricos.....	52
2.2.3.1.1	Estándares abiertos y cerrados.....	53
2.2.3.2	IEEE 802.15.4 (ZIGBEE).....	54
2.2.3.2.1	Pila de protocolos del estándar 802.15.4.....	54
2.2.3.2.1.1	Capas PHY y MAC: IEEE 802.15.4.....	55
2.2.3.2.1.2	Capas NWK y APL: Zigbee.....	57



2.2.3.2.2	Características de 802.15.4 .....	59
2.2.3.2.3	Topología de red en 802.15.4. ....	59
2.2.3.2.4	Ventajas y desventajas de Zigbee. ....	61
2.2.3.3	IEEE 802.11 (WI-FI).....	62
2.2.3.3.1	Pila de protocolos del estándar 802.11.....	63
2.2.3.3.2	Los distintos estándares Wi-Fi. ....	64
2.2.3.3.3	Características de 802.11. ....	67
2.2.3.3.4	Topología de red en 802.11. ....	68
2.2.3.3.5	Ventajas y desventajas de Wi-Fi.....	70
2.2.3.4	IEEE 802.15.1 (BLUETOOTH).....	71
2.2.3.4.1	Pila de protocolos del estándar 802.15.1.....	72
2.2.3.4.2	Características de 802.15.1. ....	73
2.2.3.4.3	Normas Bluetooth.....	74
2.2.3.4.4	Topología de red en 802.15.1. ....	74
2.2.3.4.4.1	Piconet .....	74
2.2.3.4.4.2	Scatternet.....	75
2.2.3.4.5	Ventajas y desventajas de Bluetooth .....	76
2.2.4	<i>Selección de la tecnología inalámbrica adecuada.</i> .....	77
<b>CAPÍTULO III.....</b>		<b>81</b>
<b>DISEÑO DEL HARDWARE .....</b>		<b>81</b>
3.1	INTRODUCCIÓN.....	81
3.2	DISEÑO DEL SISTEMA ELÉCTRICO PARA EL CONTROL DE ACCESO A LAS AULAS.....	81
3.2.1	<i>Descripción del sistema de control de acceso a las aulas</i> .....	82
3.3	DIAGRAMA DE BLOQUE .....	83
3.4	DIAGRAMA DE CONEXIÓN FÍSICA DE LOS CIRCUITOS TRANSMISOR Y RECEPTOR.....	85
3.5	TOPOLOGÍA DE RED .....	87
3.5.1	<i>Ubicación de los diferentes dispositivos en la estructura física de las instalaciones de la facultad.</i> .....	88
3.5.1.1	Plano de la planta baja .....	89
3.5.1.2	Plano del primer piso .....	90

3.5.1.3	Plano del segundo piso .....	90
3.5.1.4	Plano del tercer piso .....	91
3.5.1.5	Vista frontal de la facultad .....	92
3.6	ALIMENTACIÓN DE ENERGÍA PARA LA INTERFAZ ELECTRÓNICA.....	93
3.6.1	<i>Requisitos de alimentación del circuito transmisor</i> .....	94
3.6.2	<i>Requisitos de alimentación del circuito receptor</i> .....	94
3.6.3	<i>Desarrollo de la fuente de alimentación del circuito receptor</i> .....	95
3.6.3.1	Transformador de red .....	96
3.6.3.2	Convertidor de alterna a continua o rectificador (AC/DC) .....	98
3.6.3.2.1	Análisis del semiciclo positivo.....	100
3.6.3.2.2	Análisis del semiciclo negativo.....	101
3.6.3.3	Convertidor de continua a continua o filtro (CD/CD) .....	103
3.6.3.4	Estabilizador o regulador de voltaje.....	106
3.6.3.5	Sistema de backup de energía eléctrica .....	110
3.7	DIAGRAMA DE FLUJO .....	110
3.7.1	<i>Diagrama de flujo del circuito transmisor</i> .....	111
3.7.2	<i>Diagrama de flujo del circuito receptor</i> .....	112
3.8	TRAMAS DE ENVÍO Y RECEPCIÓN.....	125
3.9	PROGRAMACIÓN DEL MICROCONTROLADOR .....	126
3.10	FUNCIONAMIENTO DEL SISTEMA ELÉCTRICO.....	127
3.10.1	<i>Recepción y transmisión de datos</i> .....	127
3.10.2	<i>Visualización de los datos</i> .....	129
3.10.3	<i>Mecanismo de apertura de la puerta</i> .....	131
3.10.4	<i>Seguridad del aula (Autenticación)</i> .....	133
3.10.5	<i>Información del tiempo restante en el aula (Alarma)</i> .....	134
3.10.6	<i>Interrupción de la hora de clase y utilización del aula por un conserje (Pulsadores)</i> .....	135
3.10.7	<i>Indicadores de estado del circuito</i> .....	139
3.11	DISPOSITIVOS O ELEMENTOS A UTILIZAR EN EL CIRCUITO TRANSMISOR Y RECEPTOR .....	142
3.11.1	<i>Dispositivos del circuito transmisor</i> .....	142
3.11.2	<i>Dispositivos del circuito receptor</i> .....	143

3.12	DISEÑO DE LOS CIRCUITOS TRANSMISOR Y RECEPTOR .....	145
3.13	DISEÑO DE LA PLACA DE LOS CIRCUITOS TRANSMISOR Y RECEPTOR EN EAGLE .....	147
3.14	CIRCUITOS TRANSMISOR Y RECEPTOR .....	148
<b>CAPÍTULO IV .....</b>		<b>150</b>
<b>DISEÑO DEL SOFTWARE .....</b>		<b>150</b>
4.1	INTRODUCCIÓN.....	150
4.2	DISEÑO DE LA BASE DE DATOS DEL SISTEMA DE INGRESO A LAS AULAS .....	151
4.2.1	<i>Diseño conceptual.....</i>	<i>152</i>
4.2.1.1	Requerimientos para la base de datos del proyecto .....	153
4.2.2	<i>Elección del Sistema de Gestor de Base de Datos (SGBD).....</i>	<i>154</i>
4.2.2.1	Modelos del Sistema de Gestor de Base de Datos (SGBD) .....	155
4.2.3	<i>Diseño lógico.....</i>	<i>156</i>
4.2.4	<i>Diseño físico.....</i>	<i>158</i>
4.3	DISEÑO DE LA INTERFAZ GRÁFICA DEL SISTEMA DE INGRESO A LAS AULAS .....	159
4.3.1	<i>Prioridades de los modos de ingreso al sistema de control .....</i>	<i>162</i>
4.3.1.1	Prioridad del modo Administrador.....	162
4.3.1.2	Prioridad del modo Usuario. ....	162
4.3.1.3	Prioridad del modo Eventual.....	163
4.3.2	<i>Proceso del diseño de las interfaces según su prioridad.....</i>	<i>163</i>
4.3.2.1	Proceso para la creación de un nuevo docente. ....	163
4.3.2.2	Proceso para la búsqueda, modificación y eliminación de un docente.....	166
4.3.2.3	Proceso para Ingresar o modificar parámetros de las diferentes aulas y materias.....	167
4.3.2.4	Proceso para crear horarios de clases.....	168
4.3.2.5	Proceso para modificar los datos personales .....	170
4.3.2.6	Proceso para utilizar un aula asignada o reservada .....	171
4.3.2.7	Proceso para utilizar un aula disponible.....	173
4.3.2.8	Proceso para ver un mapa de aulas.....	174
4.4	PROGRAMACIÓN DE LA INTERFAZ GRÁFICA .....	175
4.5	MANUALES DEL SISTEMA DE CONTROL DE ACCESO.....	176

<b>CAPÍTULO V .....</b>	<b>177</b>
<b>PRUEBAS Y CORRECCIÓN DE ERRORES .....</b>	<b>177</b>
5.1 INTRODUCCIÓN.....	177
5.2 INTERFAZ ELECTRÓNICA ENCARGADA DE LA ADQUISICIÓN Y PROCESAMIENTO DE DATOS.....	177
5.2.1 <i>Corrección de errores de la interfaz electrónica encargada de la adquisición y procesamiento de datos</i> 180	
5.3 PRUEBAS DE CONECTIVIDAD EN LA RED .....	183
5.3.1 <i>Configuración de los módulos XBEE.....</i>	183
5.3.2 <i>Verificación de envío y recepción de datos.....</i>	185
5.4 INTERFAZ GRÁFICA ENCARGADA DEL ALMACENAMIENTO Y PROCESAMIENTO DE DATOS .....	188
5.5 AMPLIACIÓN DE COBERTURA DEL SISTEMA DE CONTROL DE ACCESO .....	189
<b>CAPÍTULO VI .....</b>	<b>191</b>
<b>ANÁLISIS ECONÓMICO Y FACTIBILIDAD .....</b>	<b>191</b>
6.1 INTRODUCCIÓN.....	191
6.2 COSTOS.....	191
6.2.1 <i>Materia Prima.....</i>	192
6.2.2 <i>Mano de Obra Directa .....</i>	195
6.2.3 <i>Costos Indirectos de Fabricación (CIF) .....</i>	196
6.2.4 <i>Estado de costos de producción.....</i>	197
6.3 INVERSIÓN .....	198
6.4 FACTIBILIDAD .....	198
6.5 INDICADORES DE DESEMPEÑO .....	199
6.5.1 <i>Indicador de Eficacia.....</i>	199
6.5.2 <i>Indicador de Eficiencia .....</i>	199
6.5.3 <i>Indicador de Economía .....</i>	200
<b>CAPÍTULO VII .....</b>	<b>202</b>
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>202</b>

7.1	INTRODUCCIÓN.....	202
7.2	CONCLUSIONES.....	202
7.3	RECOMENDACIONES .....	204
7.4	BIBLIOGRAFÍA:.....	206
<b>ANEXOS .....</b>		<b>208</b>
<b>ANEXO A.....</b>		<b>209</b>
7.5	DATASHEETS DE LOS DISPOSITIVOS ELECTRÓNICOS MÁS RELEVANTES.....	209
<b>ANEXO B .....</b>		<b>222</b>
7.6	CÓDIGO DE PROGRAMA DEL MICROCONTROLADOR PIC 18F4550 PARA UN CIRCUITO RECEPTOR DE UNA AULA DETERMINADA DE LA FACULTAD. ....	222
<b>ANEXO C .....</b>		<b>242</b>
7.7	PROCESO DE ELABORACIÓN DE LOS CIRCUITOS ELECTRÓNICOS DEL SISTEMA DE CONTROL DE ACCESO PARA LAS AULAS DE LA FACULTAD .....	242
7.7.1	<i>Circuitos electrónicos</i> .....	243
7.7.1.1	Impresión del circuito .....	243
7.7.1.2	Recortar el circuito impreso.....	244
7.7.1.3	Medición de la baquelita con el circuito impreso .....	244
7.7.1.4	Recortar la baquelita.....	245
7.7.1.5	Pulir la baquelita .....	246
7.7.1.6	Ubicar y fijar el circuito en la baquelita.....	247
7.7.1.7	Impregnar la pista del circuito en la baquelita .....	249
7.7.1.8	Desmontar el papel de acetato .....	250
7.7.1.9	Separar el cobre adicional del circuito .....	251
7.7.1.10	Secary Lijar el circuito.....	252
7.7.1.11	Perforar el circuito .....	253
7.7.1.12	Montar los elementos electrónicos en el circuito.....	254
7.7.1.13	Montar el circuito en la carcasa .....	256
<b>ANEXO D.....</b>		<b>259</b>

7.8	CÓDIGO DE PROGRAMA LA INTERFAZ GRÁFICA CREADA EN MICROSOFT VISUAL STUDIO C# .....	259
<b>ANEXO E</b>	.....	<b>314</b>
7.9	MANUALES DEL SISTEMA DE CONTROL DE ACCESO (SOFTWARE) PARA LAS AULAS DE LA FACULTAD	314
7.9.1	<i>MANUAL DEL ADMINISTRADOR</i> .....	315
7.9.1.1	INTRODUCCIÓN .....	315
7.9.1.2	MODOS DE INGRESO .....	315
7.9.1.3	FORMULARIOS DEL SISTEMA .....	316
7.9.1.3.1	Formulario de autenticación .....	316
7.9.1.3.2	Formulario de procesos .....	318
7.9.1.3.2.1	Secciones del menú selectivo .....	321
7.9.1.3.3	Formulario de reportes .....	332
7.9.1.4	Proceso de utilización del sistema del administrador .....	333
7.9.1.4.1	Pasos a seguir en el formulario de autenticación. ....	334
7.9.1.4.2	Pasos a seguir en el formulario procesos .....	336
7.9.1.4.2.1	Nuevo Docente. ....	336
7.9.1.4.2.2	Buscar Modificar y Eliminar Docente. ....	339
7.9.1.4.2.3	Crear o Modificar Materia, Aula o Carrera. ....	346
7.9.1.4.2.4	Crear Horario. ....	355
7.9.1.4.2.5	Modificar Datos Personales. ....	369
7.9.1.4.2.6	Utilizar Aula Asignada o Reservada .....	371
7.9.1.4.2.7	Reservar Aula Disponible. ....	376
7.9.1.4.2.8	Ver Aulas .....	378
7.9.1.5	Recomendaciones: .....	379
7.9.2	<i>MANUAL DE USUARIO</i> .....	380
7.9.2.1	INTRODUCCIÓN .....	380
7.9.2.2	FORMULARIOS DEL SISTEMA .....	380
7.9.2.2.1	Formulario de autenticación .....	380
7.9.2.2.2	Formulario de procesos .....	382
7.9.2.2.2.1	Secciones del menú selectivo .....	385
7.9.2.3	Proceso de utilización del sistema del usuario .....	390

7.9.2.3.1	Pasos a seguir en el formulario de autenticación .....	390
7.9.2.3.2	Pasos a seguir en el formulario procesos.....	392
7.9.2.3.2.1	Modificar Datos Personales.....	392
7.9.2.3.2.2	Utilizar Aula Asignada o Reservada.....	394
7.9.2.3.2.3	Reservar Aula Disponible.....	399
7.9.2.3.2.4	Ver Aulas.....	401
7.9.2.4	Recomendaciones:.....	402
<b>7.9.3</b>	<b>MANUAL DE USUARIO EVENTUAL .....</b>	<b>403</b>
7.9.3.1	INTRODUCCIÓN.....	403
7.9.3.2	FORMULARIOS DEL SISTEMA .....	403
7.9.3.2.1	Formulario de autenticación.....	403
7.9.3.2.2	Formulario de procesos .....	405
7.9.3.2.2.1	Secciones del menú selectivo .....	407
7.9.3.3	Proceso de utilización del sistema del usuario eventual.....	412
7.9.3.3.1	Pasos a seguir en el formulario de autenticación.....	412
7.9.3.3.2	Pasos a seguir en el formulario procesos.....	414
7.9.3.3.2.1	Utilizar Aula Asignada o Reservada.....	414
7.9.3.3.2.2	Reservar Aula Disponible.....	418
7.9.3.3.2.3	Ver Aulas.....	420
7.9.3.4	Recomendaciones:.....	421
<b>ANEXO F</b>	<b>.....</b>	<b>423</b>
<b>7.10</b>	<b>MANUALES DEL SISTEMA DE CONTROL DE ACCESO (HARDWARE) DE LAS AULAS DE LA FACULTAD PARA DOCENTES Y CONSERJES .....</b>	<b>423</b>
<b>7.10.1</b>	<b>MANUAL DEL HARDWARE PARA DOCENTES .....</b>	<b>424</b>
7.10.1.1	Introducción.....	424
7.10.1.2	Elementos del hardware del sistema .....	424
7.10.1.2.1	Dispositivo transmisor.....	424
7.10.1.2.2	Dispositivo receptor.....	427
7.10.1.3	Pasos a seguir en el dispositivo receptor del sistema .....	430
7.10.1.4	Alerta de finalización.....	434
7.10.1.5	Finalización del tiempo en un aula.....	434



7.10.2	MANUAL DEL HARDWARE PARA CONSERJES.....	438
7.10.2.1	Introducción .....	438
7.10.2.2	Elementos del hardware del sistema .....	438
7.10.2.2.1	Dispositivo transmisor. ....	438
7.10.2.2.2	Dispositivo receptor.....	439
7.10.2.3	Pasos a seguir en el dispositivo receptor del sistema .....	442
7.10.2.4	Alerta de finalización.....	445
7.10.2.5	Finalización del tiempo en un aula .....	446

## ÍNDICE DE FIGURAS

Figura 1. Diagrama general del proyecto.....	6
Figura 2. Microcontroladores PIC de Microchip.....	14
Figura 3. Microcontrolador y microprocesador de Intel.....	15
Figura 4. Microcontrolador de Motorola.....	16
Figura 5. Microcontroladores Atmel.....	16
Figura 6. Estructura interna del microcontrolador.....	17
Figura 7. Arquitectura Von Neumann.....	25
Figura 8. Arquitectura Harvard.....	26
Figura 9. Procesador segmentado o pipeline.....	27
Figura 10. Familia de microcontroladores Microchip.....	30
Figura 11. Microcontrolador PIC18F4550.....	31
Figura 12. Familia de microcontroladores Atmel.....	33
Figura 13. Microcontrolador ATmega164P.....	34
Figura 14. Microcontrolador intel 8051.....	35
Figura 15. Estructura física y simbología de un relé.....	40
Figura 16. Estructura de un transistor BJT.....	41
Figura 17. Estructura física y simbología de un transistor BJT en sus dos configuraciones.....	42
Figura 18. Estructura física y simbología de una resistencia.....	43
Figura 19. Código de colores para la selección de una resistencia.....	43
Figura 20. Estructura física y simbología de un potenciómetro.....	44
Figura 21. Estructura física y simbología del diodo.....	45
Figura 22. Estructura física y simbología de un regulador de voltaje.....	46
Figura 23. Estructura física y simbología del condensador.....	47
Figura 24. Estructura física de la LCD.....	48
Figura 25. Estructura física del teclado matricial.....	48
Figura 26. Estructura física del disipador de calor.....	49
Figura 27. Estructura interna de un transformador.....	51
Figura 28. Estructura física y simbología del transformador.....	51

Figura 29. Pila de protocolos de Zigbee. ....	55
Figura 30. Canales del estándar IEEE 802.15.4.....	56
Figura 31. Topología de red del estándar 802.15.4. ....	60
Figura 32. Pila de protocolos de Wi-Fi.....	64
Figura 33. Topología de red Ad-Hoc o IBSS. ....	69
Figura 34. Topología de red Infraestructura o EBSS. ....	70
Figura 35. Pila de protocolos de Bluetooth. ....	72
Figura 36. Topología Piconet de una red Bluetooth.....	75
Figura 37. Topología Scatternet de una red Bluetooth .....	76
Figura 38. Diagrama de bloque del circuito transmisor. ....	84
Figura 39. Diagrama de bloque del circuito receptor. ....	84
Figura 40. Diagrama de conexión del circuito transmisor.....	85
Figura 41. Diagrama de conexión del circuito receptor. ....	86
Figura 42. Topología de red utilizando dispositivos XBEE. ....	88
Figura 43. Planos de la planta baja de la facultad con el módulo transmisor.....	89
Figura 44. Planos del primer piso de la facultad. ....	90
Figura 45. Planos del segundo piso de la facultad con los módulos receptores y router.....	91
Figura 46. Planos del tercer piso de la facultad con los módulos receptores y router. ....	92
Figura 47. Comunicación entre modulos routers y transmisor. ....	93
Figura 48. Transformador eléctrico en una fuente de alimentación.....	96
Figura 49. Transformador reductor y curva característica. ....	98
Figura 50. Rectificador de onda completa tipo puente de diodos en una fuente de alimentación. .	99
Figura 51. Análisis del rectificador en el semiciclo positivo y curva característica. ....	101
Figura 52. Análisis del rectificador en el semiciclo negativo y curva característica.....	102
Figura 53. Rectificación de onda completa en los dos semiciclos.....	103
Figura 54. Filtro capacitivo en una fuente de alimentación.....	104
Figura 55. Filtro de salida con el capacitor .....	105
Figura 56. Regulador de voltaje en una fuente de alimentación.....	106
Figura 57. Estabilización del voltaje en una fuente de alimentación. ....	107
Figura 58. Fuente de alimentación del circuito receptor. ....	109

Figura 59. Sistema de backup de energía. ....	110
Figura 60. Diagrama de flujo de la transmisión y recepción de datos del circuito transmisor. ....	111
Figura 61. Diagrama de flujo de la transmisión y recepción de datos del circuito receptor. ....	112
Figura 62. Diagrama de flujo del programa principal del microcontrolador (PIC18F4550).....	114
Figura 63. Diagrama de flujo de la subrutina para inicializar valores. ....	115
Figura 64. Diagrama de flujo de la subrutina interrupción. ....	116
Figura 65. Diagrama de flujo de la subrutina para el menú dela LCD .....	117
Figura 66. Diagrama de flujo de la subrutina clave de ingreso.....	118
Figura 67. Diagrama de flujo de la subrutina escribir clave .....	119
Figura 68. Diagrama de flujo de la subrutina horas .....	120
Figura 69. Diagrama de flujo de la subrutina cadena a enviar .....	122
Figura 70. Diagrama de flujo de la subrutina escritura de la clave del conserje .....	123
Figura 71. Diagrama de flujo de la subrutina confirmación de datos del conserje .....	124
Figura 72. Formato general de las tramas .....	125
Figura 73. Diagrama de conexión para el cálculo de las resistencias R1 y R2.....	128
Figura 74. Diagrama de conexión física entre el microcontrolador y el módulo XBEE Pro.....	129
Figura 75. Diagrama de conexión física entre el microcontrolador y la LCD.....	130
Figura 76. Diagrama de conexión para el cálculo de la resistencia R4.....	132
Figura 77. Diagrama de conexión física entre el microcontrolador, el relé y la cerradura.....	133
Figura 78. Diagrama de conexión física entre el microcontrolador y el teclado. ....	134
Figura 79. Diagrama de conexión física entre el microcontrolador y el parlante. ....	135
Figura 80. Diagrama de conexión para el cálculo de la resistencia R6.....	137
Figura 81. Diagrama de conexión física entre el microcontrolador y el pulsador .....	139
Figura 82. Diagrama de conexión para el cálculo de la resistencia R7.....	140
Figura 83. Diagrama de conexión física entre el microcontrolador y el indicador Led. ....	141
Figura 84. Diseño del circuito transmisor.....	145
Figura 85. Diseño del circuito receptor. ....	146
Figura 86. Pista del circuito transmisor. ....	147
Figura 87. Pista del circuito receptor.....	147
Figura 88. Circuito transmisor. ....	148

Figura 89. Circuito receptor.....	148
Figura 90. Diseño del modelo conceptual de la base de datos. ....	154
Figura 91. Logo del software SQL Server 2008.....	155
Figura 92. Diseño físico de la base de datos .....	159
Figura 93. Interfaz gráfica del modo Administrador. ....	160
Figura 94. Interfaz gráfica del modo Usuario. ....	161
Figura 95. Interfaz gráfica del modo Eventual. ....	161
Figura 96. Diagrama de flujo del método validación de cédula .....	164
Figura 97. Interfaz gráfica de la creación de un nuevo docente .....	165
Figura 98. Interfaz gráfica de la búsqueda, modificación y eliminación de un docente.....	166
Figura 99. Interfaz gráfica de la creación y modificación de un aula materia o carrera .....	168
Figura 100. Interfaz gráfica de la creación del horario de clases .....	169
Figura 101. Interfaz gráfica de la modificación de datos personales.....	170
Figura 102. Diagrama de flujo del envío de datos desde la interfaz gráfica .....	172
Figura 103. Interfaz gráfica de la utilización del aula .....	173
Figura 104. Interfaz gráfica de la reserva del aula.....	174
Figura 105. Interfaz gráfica del estado de las aulas de la facultad.....	175
Figura 106. Pruebas eléctricas del circuito transmisor. ....	178
Figura 107. Pruebas eléctricas del circuito receptor. ....	178
Figura 108. Errores eléctricos del circuito receptor.....	179
Figura 109. Pruebas eléctricas del circuito receptor modificada la fuente de alimentación. ....	182
Figura 110. Puente de los pines TX y RX del módulo XBEE correspondiente al Aula 201. ....	185
Figura 111. Puente de los pines TX y RX del módulo XBEE correspondiente al Aula 301. ....	185
Figura 112. Pruebas de convergencia entre módulos XBEE en la red.....	186
Figura 113. Reporte generado por el sistema de control de acceso. ....	188
Figura 114. Ampliación de cobertura en la red. ....	190
Figura 115. Impresión del circuito transmisor. ....	243
Figura 116. Impresión del circuito receptor.....	243
Figura 117. Recorte del circuito transmisor. ....	244
Figura 118. Recorte del circuito receptor. ....	244

Figura 119. Medición del circuito transmisor.....	245
Figura 120. Medición del circuito receptor. ....	245
Figura 121. Recorte de la baquelita del circuito transmisor.....	246
Figura 122. Recorte de la baquelita del circuito receptor. ....	246
Figura 123. Lijado de la baquelita del circuito transmisor. ....	247
Figura 124. Lijado de la baquelita del circuito receptor. ....	247
Figura 125. Ubicación de la pista impresa en la baquelita del circuito transmisor. ....	248
Figura 126. Fijación de la pista impresa a la baquelita del circuito transmisor.....	248
Figura 127. Ubicación de la pista impresa en la baquelita del circuito receptor. ....	248
Figura 128. Fijación de la pista impresa a la baquelita del circuito receptor. ....	249
Figura 129. Impregnación del circuito transmisor en la baquelita.....	249
Figura 130. Impregnación del circuito receptor en la baquelita. ....	250
Figura 131. Desmontaje del papel de las placas en los diferentes circuitos. ....	250
Figura 132. Separación del cobre adicional de la pista en los circuitos transmisor. ....	251
Figura 133. Separación del cobre adicional de la pista en los circuitos receptor. ....	251
Figura 134. Secado del circuito transmisor.....	252
Figura 135. Lijado del circuito transmisor. ....	252
Figura 136. Secado del circuito receptor. ....	252
Figura 137. Lijado del circuito receptor. ....	253
Figura 138. Perforación del circuito transmisor.....	253
Figura 139. Perforación del circuito receptor. ....	254
Figura 140. Ubicación de los elementos electrónicos en la placa del circuito transmisor.....	254
Figura 141. Ubicación de los elementos electrónicos en la placa del circuito receptor.....	255
Figura 142. Circuito transmisor terminado. ....	255
Figura 143. Circuito receptor terminado.....	256
Figura 144. Ubicación del circuito transmisor en su respectiva carcasa. ....	256
Figura 145. Aseguramiento del circuito transmisor.....	257
Figura 146. Ubicación del circuito receptor en su respectiva carcasa.....	257
Figura 147. Aseguramiento del circuito receptor. ....	258
Figura 148. Formulario de autenticación del administrador.....	317

Figura 149. Identificación de las secciones en el formulario de autenticación del administrador.	318
Figura 150. Formulario de procesos del administrador.	319
Figura 151. Identificación de los menús en el formulario de procesos del administrador.	321
Figura 152. Identificación de las secciones en el formulario de procesos del administrador del submenú nuevo docente.	322
Figura 153. Identificación de las secciones en el formulario de procesos del administrador del submenú buscar, modificar y eliminar docente.	323
Figura 154. Identificación de las secciones en el formulario de procesos del administrador del submenú crear, modificar materia, aula o carrera.	325
Figura 155. Identificación de las secciones en el formulario de procesos del administrador del submenú crear horario.	326
Figura 156. Identificación de las secciones en el formulario de procesos del administrador del submenú modificar datos personales.	327
Figura 157. Identificación de las secciones en el formulario de procesos del administrador del submenú utilizar aula asignada o reservada.	329
Figura 158. Identificación de las secciones en el formulario de procesos del administrador del submenú reservar aula disponible.	330
Figura 159. Identificación de las secciones en el formulario de procesos del administrador del submenú ver aula.	331
Figura 160. Formulario de reportes del administrador.	332
.Figura 161. Identificación del menú en el formulario de reportes del administrador	333
Figura 162. Pasos a seguir en el formulario autenticación del administrador.	335
Figura 163. Mensaje de confirmación del ingreso al sistema del usuario administrador.	335
Figura 164. Pasos para validar el número de cedula del docente.	337
Figura 165. Confirmación de la validación del número de cédula del docente.	337
Figura 166. Pasos para la creación de un docente en el formulario procesos del administrador.	338
Figura 167. Confirmación de los datos guardados en el sistema del nuevo docente.	339
Figura 168. Pasos para la búsqueda de un docente en el formulario procesos del administrador.	341
Figura 169. Pasos para modificar datos del docente seleccionado.	343



Figura 170. Confirmación de los datos modificados del docente. ....	343
Figura 171. Pasos para eliminar a un docente seleccionado. ....	344
Figura 172. Mensaje de precaución antes de eliminar a un docente. ....	345
Figura 173. Confirmación del docente eliminado del sistema. ....	345
Figura 174. Pasos para crear un aula en el sistema.....	347
Figura 175. Confirmación de la creación de una nueva aula.....	347
Figura 176. Pasos para modificar los datos de un aula.....	348
Figura 177. Confirmación de la modificación de los datos del aula.....	349
Figura 178. Pasos para crear una materia en el sistema. ....	350
Figura 179. Confirmación de la creación de una nueva materia. ....	350
Figura 180. Pasos para modificar los datos de una materia.....	351
Figura 181. Confirmación de la modificación de los datos de la materia.....	352
Figura 182. Pasos para crear una carrera en el sistema.....	353
Figura 183. Confirmación de la creación de una nueva carrera. ....	353
Figura 184. Pasos para crear una carrera en el sistema.....	354
Figura 185. Confirmación de la modificación de los datos de la carrera. ....	355
Figura 186. Pasos para seleccionar un periodo académico a crear en el sistema. ....	357
Figura 187. Sugerencia antes de crear un periodo académico nuevo. ....	357
Figura 188. Sugerencia antes de continuar con la modificación del periodo académico. ....	358
Figura 189. Confirmación de la modificación y creación de un nuevo periodo académico. ....	358
Figura 190. Creación de un horario en un periodo académico en el sistema. ....	360
Figura 191. Confirmación de la creación de un nuevo horario académico.....	360
Figura 192. Pasos para seleccionar un periodo académico a modificar en el sistema. ....	362
Figura 193. Pasos para cargar los datos de un periodo académico a modificar en el sistema. ...	363
Figura 194. Modificación de un horario en un periodo académico en el sistema.....	364
Figura 195. Confirmación de la modificación del horario académico en el sistema.....	364
Figura 196. Pasos para seleccionar un periodo académico a duplicar en el sistema.....	366
Figura 197. Confirmación de la duplicación del horario académico en el sistema.....	366
Figura 198. Pasos para cargar los datos de un periodo académico duplicado en el sistema.....	367

Figura 199. Modificación de la duplicación de un horario en un periodo académico en el sistema. .....	368
Figura 200. Confirmación de la modificación del horario académico duplicado en el sistema.....	368
Figura 201. Pasos para modificar los datos personales del administrador en el sistema. ....	370
Figura 202. Confirmación de la modificación de los datos personales del administrador. ....	370
Figura 203. Formulario de clases por medio de un horario. ....	372
Figura 204. Formulario de clases por medio de una reserva. ....	373
Figura 205. Formulario de no tener clases. ....	374
Figura 206. Pasos para utilizar el aula asignada o reservada para el docente. ....	375
Figura 207. Confirmación de los datos guardados en el reporte del sistema.....	375
Figura 208. Confirmación del estado del aula actual. ....	376
Figura 209. Pasos para realizar una reserva de un aula en el sistema.....	377
Figura 210. Sugerencia para reservar el aula.....	378
Figura 211. Confirmación de la reserva del aula realizada correctamente. ....	378
Figura 212. Identificación del estado de las aulas de la facultad.....	379
Figura 213. Formulario de autenticación del usuario. ....	381
Figura 214. Identificación de las secciones en el formulario de autenticación del usuario. ....	382
Figura 215. Formulario de procesos del usuario.....	383
Figura 216. Identificación de los menús en el formulario de procesos del usuario. ....	384
Figura 217. Identificación de las secciones en el formulario de procesos del usuario del submenú modificar datos personales. ....	385
Figura 218. Identificación de las secciones en el formulario de procesos del usuario del submenú utilizar aula asignada o reservada.....	386
Figura 219. Identificación de las secciones en el formulario de procesos del usuario del submenú reservar aula disponible. ....	388
Figura 220. Identificación de las secciones en el formulario de procesos del usuario del submenú ver aula.....	389
Figura 221. Pasos a seguir en el formulario autenticación del usuario. ....	391
Figura 222. Mensaje de confirmación del ingreso al sistema del usuario. ....	391
Figura 223. Pasos para modificar los datos personales del usuario en el sistema. ....	393

Figura 224. Confirmación de la modificación de los datos personales del usuario. ....	394
Figura 225. Formulario de clases por medio de un horario. ....	395
Figura 226. Formulario de clases por medio de una reserva. ....	396
Figura 227. Formulario de no tener clases. ....	397
Figura 228. Pasos para utilizar el aula asignada o reservada para el docente. ....	398
Figura 229. Confirmación de los datos guardados en el reporte del sistema.....	398
Figura 230. Confirmación del estado del aula actual. ....	399
Figura 231. Pasos para realizar una reserva de un aula en el sistema.....	400
Figura 232. Sugerencia para reservar el aula. ....	401
Figura 233. Confirmación de la reserva del aula realizada correctamente. ....	401
Figura 234. Identificación del estado de las aulas de la facultad.....	402
Figura 235. Formulario de autenticación del docente eventual. ....	404
Figura 236. Identificación de las secciones en el formulario de autenticación del docente eventual. .....	405
Figura 237. Formulario de procesos del docente eventual. ....	406
Figura 238. Identificación de los menús en el formulario de procesos del docente eventual. ....	407
Figura 239. Identificación de las secciones en el formulario de procesos del docente eventual del submenú utilizar aula asignada o reservada.....	408
Figura 240. Identificación de las secciones en el formulario de procesos del docente eventual del submenú reservar aula disponible. ....	410
Figura 241. Identificación de las secciones en el formulario de procesos del docente eventual del submenú ver aula.....	411
Figura 242. Pasos a seguir en el formulario autenticación del docente eventual.....	413
Figura 243. Mensaje de confirmación del ingreso al sistema del docente eventual.....	413
Figura 244. Formulario de clases por medio de una reserva. ....	415
Figura 245. Formulario de no tener clases. ....	416
Figura 246. Pasos para utilizar el aula reservada para el docente eventual. ....	417
Figura 247. Confirmación de los datos guardados en el reporte del sistema.....	417
Figura 248. Confirmación del estado del aula actual. ....	418
Figura 249. Pasos para realizar una reserva de un aula en el sistema.....	419

Figura 250. Sugerencia para reservar el aula. ....	420
Figura 251. Confirmación de la reserva del aula realizada correctamente. ....	420
Figura 252. Identificación del estado de las aulas de la facultad. ....	421
Figura 253. Dispositivo transmisor. ....	425
Figura 254. Parte frontal del dispositivo transmisor. ....	425
Figura 255. Parte lateral superior del dispositivo transmisor. ....	425
Figura 256. Parte lateral inferior del dispositivo transmisor. ....	426
Figura 257. Conexión del dispositivo transmisor en el computador del sistema. ....	426
Figura 258. Dispositivo receptor. ....	427
Figura 259. Parte frontal del dispositivo receptor. ....	428
Figura 260. Parte lateral izquierda del dispositivo receptor. ....	429
Figura 261. Parte lateral derecha del dispositivo receptor. ....	429
Figura 262. Parte lateral inferior del dispositivo receptor. ....	430
Figura 263. Mensaje de bienvenida en el dispositivo receptor. ....	430
Figura 264. Menú de inicio del dispositivo receptor. ....	431
Figura 265. Visualización del pedido de clave en el dispositivo receptor. ....	431
Figura 266. Proceso de digitalización de la clave en el dispositivo receptor. ....	431
Figura 267. Identificación de los pasos a seguir en el teclado del dispositivo receptor. ....	432
Figura 268. Confirmación de la clave en el dispositivo receptor. ....	432
Figura 269. Apertura de la puerta en el dispositivo receptor. ....	432
Figura 270. Información del número de horas y el tiempo transcurrido en el dispositivo receptor. ....	433
Figura 271. Mensaje de error y número de intentos en el dispositivo receptor. ....	433
Figura 272. Información del tiempo restante en el dispositivo receptor del aula. ....	434
Figura 273. Estado de utilización de un aula. ....	435
Figura 274. Visualización del pedido de clave de finalización en el dispositivo receptor. ....	435
Figura 275. Proceso de digitalización de la clave de finalización en el dispositivo receptor. ....	436
Figura 276. Información de espera de confirmación de datos en el dispositivo receptor. ....	436
Figura 277. Finalización del periodo de clases en el aula del dispositivo receptor. ....	437
Figura 278. Mensaje de error de clave en el dispositivo receptor. ....	437

Figura 279. Dispositivo receptor.....	439
Figura 280. Parte frontal del dispositivo receptor.....	440
Figura 281. Parte lateral izquierda del dispositivo receptor. ....	441
Figura 282. Parte lateral derecha del dispositivo receptor.....	441
Figura 283. Parte lateral inferior del dispositivo receptor.....	442
Figura 284. Mensaje de bienvenida en el dispositivo receptor. ....	442
Figura 285. Visualización del pedido de clave en el dispositivo receptor. ....	443
Figura 286. Proceso de digitalización de la clave en el dispositivo receptor. ....	443
Figura 287. Identificación de los pasos a seguir en el dispositivo receptor. ....	443
Figura 288. Confirmación de la clave del conserje en el dispositivo receptor. ....	444
Figura 289. Información del número de horas y el tiempo transcurrido del conserje en el dispositivo receptor. ....	445
Figura 290. Datos ingresados falsos en el dispositivo receptor.....	445
Figura 291. Información del tiempo restante en el dispositivo receptor del aula.....	446
Figura 292. Estado de utilización de un aula por un conserje. ....	447
Figura 293. Visualización del pedido de clave de finalización por el conserje. ....	447
Figura 294. Proceso de digitalización de la clave del conserje de finalización. ....	448
Figura 295. Información de espera de confirmación de datos del conserje. ....	448
Figura 296. Finalización del periodo de limpieza en un aula. ....	448
Figura 297. Mensaje de error de la clave del conserje. ....	449

## ÍNDICE DE TABLAS

Tabla 1. Comparación de microcontroladores de diferente fabricante. -----	37
Tabla 2. Bandas de frecuencia de Zigbee. -----	56
Tabla 3. Estándares de 802.11 -----	65
Tabla 4. Bandas de frecuencia de Wi-Fi. -----	67
Tabla 5. Potencias de transmisión de Bluetooth. -----	73
Tabla 6. Comparación de tecnologías inalámbricas. -----	78
Tabla 7. Parámetros de configuración de los módulos XBEE Pro de los diferentes circuitos. ----	183
Tabla 8. Costo del circuito transmisor. -----	192
Tabla 9. Costo del circuito receptor. -----	193
Tabla 10. Costo del circuito receptor parte mecánica. -----	194
Tabla 11. Costo del circuito router. -----	194
Tabla 12. Inversión de los equipos electrónicos. -----	195
Tabla 13. Costo de la mano de obra directa. -----	196
Tabla 14. Costos indirectos de fabricación. -----	196
Tabla 15. Estado de costos de producción. -----	197
Tabla 16. Indicadores de gestión y desempeño. -----	200

## RESUMEN

El presente trabajo de titulación se basa en proporcionar accesibilidad, y comodidad tanto a los docentes como a estudiantes, además de seguridad incluyendo a los equipos instalados en cada una de las aulas de la Facultad de Ingeniería en Ciencias Aplicadas (FICA) de la Universidad Técnica del Norte mediante un control de acceso diseñado en un software y hardware, el cual cuenta con parámetros necesarios para solucionar ciertos problemas presentados actualmente en dicha facultad. Inicialmente se analizó las necesidades con mayor prioridad en la facultad para brindar un mejor acceso a las aulas, tanto a docentes como estudiantes; para ello, se hizo un estudio teórico de los dispositivos electrónicos, desde un diodo led hasta la utilización de microcontroladores y módulos inalámbricos, que fueron de mucha importancia al momento de diseñar el hardware (circuito electrónico) que permite acceder a una aula electrónicamente. De igual forma, tomando en cuenta la topología física de la facultad, se procedió a realizar el estudio de las tecnologías inalámbricas que permita la convergencia en una red de datos para comunicar las diferentes aulas a un software (computadora o servidor). Este software cuenta con un programa diseñado en Microsoft Visual Studio C# 2010 el mismo que fue programado para satisfacer las necesidades de un docente o funcionario que labora en la facultad y una base de datos diseñada en SQL Server 2008 la cual permite guardar toda la información necesaria de las acciones que se realicen en un aula determinada. Posteriormente se realizó las pruebas correspondientes a estos dos elementos (software y hardware) en una topología de red totalmente convergente y finalmente se analizó los beneficios económicos y de factibilidad que ostenta este proyecto como aporte tecnológico para dicha institución.

## **ABSTRACT**

This work of degree is to provide accessibility, convenience and security to both teachers and equipment installed in each classroom of the Applied Science Engineering (FICA) of Técnica del Norte University by a controlling access designed into a software and hardware, which has necessary parameters to solve certain currently problems presented in this engineering. Initially the needs were analyzed with higher priority to provide better access to the classrooms, for both teachers and students; this, it was made a theoretical study of electronic devices, from one LED to the using of microcontrollers and wireless modules which were very important when designing the hardware (electronic circuit) that allows access to a classroom electronically. Similarly, taking into account the physical topology of the engineering it was preceded to the study of the wireless technologies that allow convergence in a data network to communicate different classrooms with software (computer or server). This software has a program designed in Microsoft Visual Studio C # 2010, the same was scheduled to identify the needs of a teacher or staff member that serves on the engineering and a database designed in SQL Server 2008 which lets save all the information needed to share a particular classroom. Then, the evidence for these two elements (software and hardware) in a full converged network topology was performed and finally the economic feasibility and benefits that this project holds as technological contribution for this institution was analyzed.



# **CAPÍTULO I**

## **ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS**

### **1.1 Introducción**

En este capítulo se identifican los inconvenientes que se presentan en la Facultad de Ingeniería en Ciencias Aplicadas (FICA) en cuanto al acceso a las aulas por parte de los estudiantes y docentes para luego proponer una solución haciendo uso de la domótica, brindando comodidad y fiabilidad tanto a los docentes y estudiantes, además de brindar seguridad a los equipos instalados en las aulas y poder contribuir con el proceso de acreditación institucional que se lleva a cabo.

### **1.2 Análisis de la situación actual**

A la Facultad de Ingeniería en Ciencias Aplicadas asisten aproximadamente 1450 estudiantes entre hombres y mujeres. Para atender a esta población estudiantil laboran 21 empleados, 30 profesores titulares y 59 contratados.

Conforme pasan los años la facultad crece de una manera considerable, es por ello que cada vez se necesita adoptar medidas para facilitar el ingreso a las aulas y dar seguridad a los equipos que se encuentran implementados en cada una de las diferentes aulas.

Actualmente en la Universidad Técnica del Norte se encuentra en un proceso de evaluación y acreditación internacional debido a que el gobierno exige a las universidades del país adoptar las medidas necesarias para poder ser una universidad de calidad, es por ello que este proyecto se fundamenta en el indicador de infraestructura tecnológica (E), específicamente con el indicador TIC (E.2) e Innovación Tecnológica (E.2.2), que exige el Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior (CEAACES) en su Modelo Institucional de Pregrado mediante la evaluación global a las Instituciones de Educación Superior (IES).

### **1.3 Planteamiento del problema**

En la Facultad de Ingeniería en Ciencias Aplicadas(FICA) existen muchos inconvenientes en cuanto al acceso a las aulas, tales como:

- ✓ La ausencia ymanipulación excesiva de las cerraduras en las puertas
- ✓ El tiempo de espera para ingresar a las aulas
- ✓ La obstrucción del paso por parte de las rejas implementadas

- ✓ Falta de distribución y organización de las aulas
- ✓ La puntualidad para el inicio de clases.
- ✓ Disponibilidad del aula para los docentes.

Este problema incluye a estudiantes y docentes, debido a que, actualmente para ingresar a las aulas se debe hacer por medio de llaves proporcionadas por una persona encargada de distribuirlas, lo cual, ha surgido problemas como los mencionados anteriormente.

Uno de estos problemas se ha dado en el tiempo para ingresar a las aulas ya que, por varias ocasiones los docentes tienen que esperar al encargado del distributivo de aulas para que les facilite las llaves y poder ingresar, e incluso este problema se da también por parte de los docentes que, se olvidan de entregar las llaves y no se puede utilizar el aula por parte de otro docente, por lo que se ha tenido que localizar al docente que tiene las llaves para poder utilizar dicha aula, además cuando el encargado del distributivo de aulas no se encuentra y las aulas permanecen cerradas, esto conlleva a mucho tiempo de espera y perjudica a la enseñanza que se lleva a cabo en las aulas.

Las cerraduras aisladas completamente, tuvieron que ser retiradas de las puertas, ya sea por la mala manipulación o el excesivo uso de las mismas lo que hace que no se las puedan cerrar completamente y permanezcan abiertas todo el tiempo. Otro de los problemas es la estética, la presencia de rejas para

proporcionar seguridad en las aulas, lo que aporta un mala imagen como institución académica, al ser una facultad que se ha ganado prestigio en toda la universidad y mucho más teniendo carreras que podrían ayudar a crecer tecnológicamente. Las rejas implementadas en las puertas de la facultad proporcionan seguridad pero a pesar de dar mala imagen obstruyen el paso, lo que es molesto para todas las personas que transitan a diario por los pasillos de la facultad.

Estos inconvenientes pueden perjudicar en el proceso de evaluación y acreditación que se lleva a cabo, ya que actualmente no cumpliría con el indicador de infraestructura tecnológica anteriormente mencionado.

### **1.3.1 Solución para los inconvenientes presentes en la facultad**

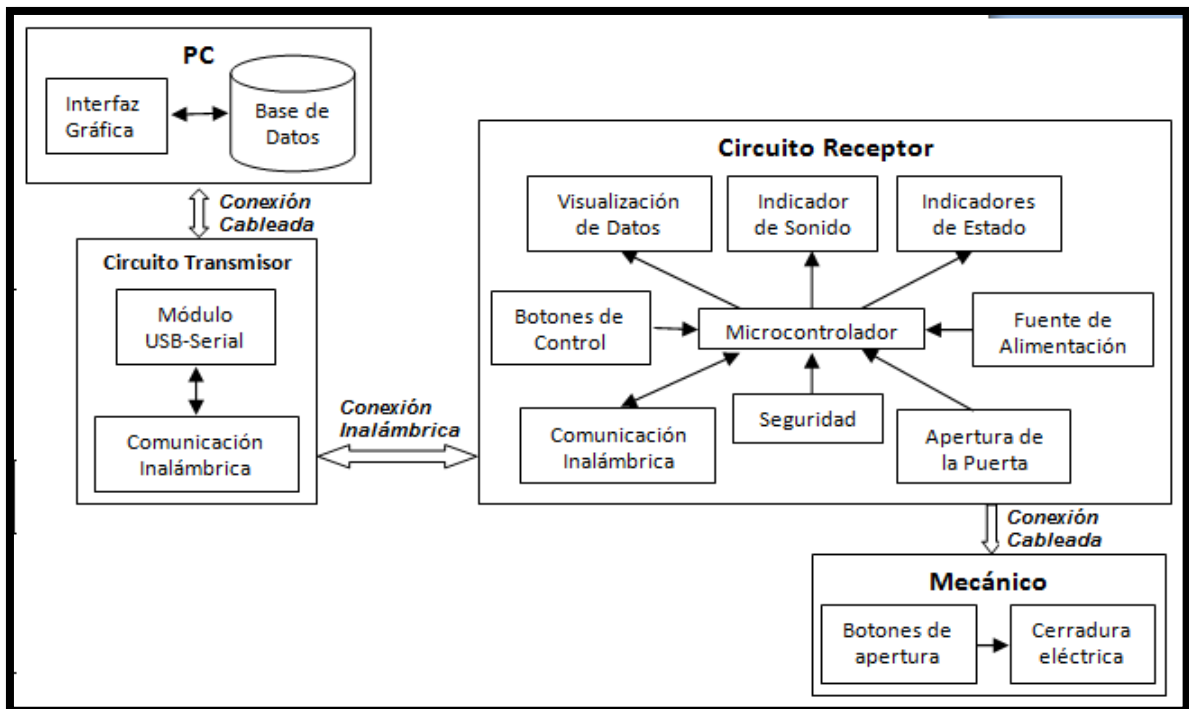
Para que estos problemas se puedan solucionar se ha tomado en cuenta que el ingreso a las aulas sea controlado por un sistema de acceso inalámbrico que será utilizado por los docentes, los cuales, sean quienes autoricen el ingreso a las aulas para no estar dependiendo de las llaves, poder eliminar las rejas que existen actualmente en la facultad y contribuir con la educación de calidad, dando comodidad a todos los estudiantes y docentes solucionando este problema, aportando con un factor importante para la acreditación y ser un ejemplo de superación en innovación tecnológica para las demás facultades.

Este sistema permitirá tener un mayor control de acceso a las aulas por lo que se podrá identificar al docente que utilice un aula determinada, e incluso para la limpieza de las mismas se llevará el control para los conserjes que se encuentran realizando dicha función en las instalaciones, por lo que este método permite restringir a cualquier persona que intente ingresar, y no se le dé acceso ya que sólo personas registradas en el sistema y, mediante un modo de ingreso tendrán ciertos privilegios de acceso por lo que el sistema es más seguro.

En cuanto a la asistencia por parte de los docentes el sistema permitirá llevar un registro de la información, para controlar el tiempo que un docente utiliza el aula, e incluso en caso de existir aulas disponibles estas puedan ser utilizadas por otros docentes, siempre y cuando éstas no interfieran con el horario establecido.

Lo dicho hasta el momento se resume en la elaboración y construcción de un software (Interfaz gráfica) para el control de acceso por medio de una computadora (PC) y un hardware (Circuitos electrónicos: Transmisor y Receptor) que realizarán las funciones de apertura y control en las puertas de la facultad, respectivamente cabe mencionar que el control se lo realiza por medio de un circuito programador denominado *microcontrolador*, lo cual se puede verificar en el diagrama de la siguiente figura.

Figura 1. Diagrama general del proyecto.



Fuente: El Autor.

## 1.4 Objetivos

### 1.4.1 Objetivo general

Diseñar un sistema de control de acceso en las aulas de la Facultad de Ingeniería en Ciencias Aplicadas utilizando dispositivos electrónicos en una red WPAN para brindar seguridad y comodidad al usuario.

### **1.4.2 Objetivos específicos**

- Analizar las características de los dispositivos inalámbricos mediante el conocimiento de tecnologías orientadas especialmente al protocolo Zigbee con los cuales el proyecto entrará en funcionamiento.
- Analizar los parámetros necesarios de los microcontroladores de la familia PIC18F y otros dispositivos electrónicos en base al estudio de sus características y funcionamiento para el montaje del proyecto.
- Realizar el esquema de la circuitería tomando en cuenta la topología de red WPAN para comunicar los dispositivos inalámbricos con la interfaz de usuario.
- Desarrollar la aplicación en el software “Microsoft Visual C#” realizando la correspondiente programación que permitirá la ejecución del sistema de control.
- Diseñar una base de datos utilizando “SQL Server” para almacenar el registro de los diferentes accesos que se realicen en las aulas.
- Verificar el proceso de funcionamiento mediante pruebas realizadas previamente tanto en hardware como en software.
- Realizar un análisis económico para poder dimensionar el beneficio del proyecto.

### **1.5 Alcance**

Para el desarrollo de este proyecto se realizará un estudio de los dispositivos de hardware que contendrá el proyecto para su correcto

funcionamiento en base a un análisis entre tecnologías inalámbricas WPAN haciendo énfasis en las principales características que estos deben tener para que se acoplen al proyecto de una mejor manera.

Se contará con un microcontrolador de la familia PIC18F, el cual permitirá manipular la puerta de una determinada aula, dicho microcontrolador es un circuito programable que contiene todos los elementos necesarios para desarrollar y controlar una tarea determinada; el mismo que se conectará a diferentes tipos de elementos electrónicos ya que cuenta con varios puertos de entrada/salida, un consumo de energía relativamente bajo para su correcto funcionamiento y lo más importante una comunicación Serial, USB, I2C entre otras las cuales permitirá conectar el dispositivo con otros microcontroladores o módulos, en este caso los módulos inalámbricos XBEE.

Los módulos XBEE permitirán establecer una comunicación inalámbrica entre el microcontrolador de la familia PIC18F y una aplicación que se ejecutará en una PC previamente diseñada para poder enviar datos desde la planta baja hacia el segundo y tercer piso de la facultad donde estarán ubicados estos módulos para eliminar el cableado vertical evitando así perforaciones innecesarias en la edificación.

Por otro lado se contará con cerraduras eléctricas que funcionarán como medio de apertura para las aulas donde la entrada deberá ser limitada y



controlada remotamente desde un solo sitio. La cerradura eléctrica contará con llaves en caso de que la energía eléctrica falle éstas entrarán en funcionamiento como medida de respaldo para poder continuar dando clases normalmente.

De igual forma se procederá a realizar una aplicación en un software para que los docentes puedan acceder a las aulas y así hacer uso de ellas. Debido a que la Universidad Técnica del Norte cuenta con la licencia correspondiente del software "Microsoft Visual Studio" se ha decidido trabajar con éste para el diseño de la aplicación que estará instalada en una PC ubicada en la planta baja de la facultad, esta PC se comunicará con un módulo XBEE (transmisor). Dicha aplicación contendrá funciones de autenticación (usuario y contraseña) del docente que hará uso de una determinada aula al que se le asigne durante su respectivo horario correspondiente al distributivo de aulas que se aplica actualmente en la facultad.

También se dispondrá de una base de datos realizada en "SQL Server" la cual contendrá la información de los docentes que hayan accedido a las aulas ya sea en su horario normal o en caso de recuperación de clases u otros parámetros en el que el docente se fundamente explicando el motivo por el cual requiere de una determinada aula en un horario diferente al asignado.

Finalmente se realizará un análisis económico del proyecto para especificar el beneficio que dicho proyecto aportará en la facultad tomando la

mejor decisión orientado en las cerraduras como medio de apertura como en los elementos de hardware que se utilizará para el diseño del mismo.

Al finalizar esta tesis, se contará con un diseño de un sistema de control para el acceso a las aulas, cuyo sistema brindará comodidad y seguridad a los usuarios de la facultad.

## **1.6 Justificación**

Debido a que en la LEY ORGÁNICA DE EDUCACIÓN SUPERIOR, mediante los organismos públicos que rigen el Sistema de Educación Superior tales como: El Consejo de Educación Superior (CES) y El Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior (CEAACES), exige a todas las universidades del país someterse en forma obligatoria a la evaluación interna y externa, acreditación, clasificación académica y al aseguramiento de la calidad.

Dicho proceso conlleva a evaluar diferentes parámetros importantes tales como: Académica, Eficiencia Académica, Investigación, Organización e Infraestructura en las universidades principalmente en las diferentes facultades.

Por ello, la Facultad de Ingeniería en Ciencias Aplicadas (FICA), se ha visto en la obligación de mejorar la infraestructura con procesos tecnológicos

innovadores para el mejoramiento de aulas. Ya que este parámetro es muy importante en la calificación final del proceso de acreditación institucional y, como se puede observar, la facultad carece de innovación tecnológica y presenta los problemas ya mencionados. Se ha decidido orientarse principalmente al mejoramiento del acceso a las aulas debido a que este ha venido siendo un gran problema durante mucho tiempo.

La finalidad de este proyecto es aportar con el indicador (E) de infraestructura tecnológica, específicamente con el indicador E.2 (TIC) y E.2.2 (Innovación Tecnológica), que exige el CEAACES para la acreditación con la prestación de seguridad y acceso a las aulas para eliminar, de manera definitiva, las llaves y rejas.

## **CAPÍTULO II**

### **ESTUDIO DE DISPOSITIVOS ELECTRÓNICOS Y TECNOLOGIAS INALÁMBRICAS**

#### **2.1 Introducción**

Este capítulo contiene el estudio de los dispositivos electrónicos que ayudarán a controlar el acceso a las aulas, analizando diferentes tipos, en un máximo de tres tecnologías ya sea en la parte de las comunicaciones inalámbricas (WPAN) como de los microcontroladores, para su posterior comparación y seleccionar el más adecuado tomando en cuenta ciertos parámetros como el costo, tamaño, servicios, velocidad de transmisión, consumo de energía, frecuencias de trabajo entre otros.

#### **2.2 Estudio de dispositivos electrónicos**

A continuación se mencionan cada uno de los diferentes tipos de dispositivos electrónicos como el microcontrolador, los módulos inalámbricos, el actuador para la apertura de las puertas entre otros, que serán parte del correcto funcionamiento del circuito electrónico de este proyecto, seleccionando el más adecuado.

Para la selección del microcontrolador y los módulos inalámbricos se estudian tres alternativas, ya que en la actualidad existen diferentes tipos y marcas comerciales que se dedican a la distribución de estos elementos, cada uno con diferentes características, las cuales serán tomadas en cuenta al momento de seleccionarlos.

### **2.2.1 Microcontroladores.**

Un microcontrolador, es básicamente un circuito integrado formado por semiconductores (transistores) interconectados entre sí, para facilitar el funcionamiento de un determinado proceso que se desee realizar de manera secuencial y automática.

Dicho de otra forma un microcontrolador es un circuito programable que está formado por componentes como memorias, procesador, periféricos de entrada y salida entre otros, los cuales se encargan de procesar la información y cumplir con una tarea determinada.

#### **2.2.1.1 Fabricantes de microcontroladores.**

En el mundo de la electrónica existen varias marcas reconocidas de microcontroladores, cada una con su respectiva característica, pero su estructura básica es la misma en todas las diferentes familias de cada uno de ellos. Entre los fabricantes más reconocidos se mencionan:

- Microchip
- Intel
- Motorola
- Atmel, etc.

### **2.2.1.1.1 Microchip.**

De acuerdo a lo indicado en: <http://www.microchip.com/pagehandler/en-us/aboutus/home.html> “Microchip Technology Inc. es un proveedor líder de microcontroladores y semiconductores analógicos, proporcionando el desarrollo de productos de bajo riesgo, el costo total del sistema y el tiempo de lanzamiento al mercado de miles de aplicaciones de diversos clientes en todo el mundo. Con sede en Chandler, Arizona, Microchip ofrece soporte técnico sobresaliente junto con la entrega y calidad confiable”. Su Producto más popular son los microcontroladores denominados PIC.

Figura 2. Microcontroladores PIC de Microchip.

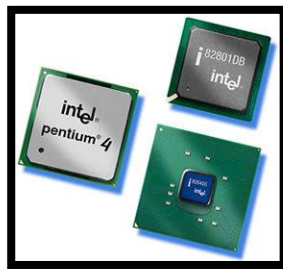


Fuente:[http://1.bp.blogspot.com/\\_XRZz93k6ZNY/SToUsJhYArI/AAAAAAAAAFQ/BHFw6CVnr1A/s400/micro.jpg](http://1.bp.blogspot.com/_XRZz93k6ZNY/SToUsJhYArI/AAAAAAAAAFQ/BHFw6CVnr1A/s400/micro.jpg)

### **2.2.1.1.2 Intel.**

De acuerdo a lo indicado en: <https://microcontroladoresesv.wordpress.com/empresas-fabricantes-de-microcontroladores/> “Intel es una empresa dedicada a la fabricación de microcontroladores y microprocesadores, aunque no trabajaba sola obtuvo un logro. En abril de 1974 pone en el mercado el microprocesador bajo el nombre 8080”.

Figura 3. Microcontrolador y microprocesador de Intel.



Fuente:[http://3.bp.blogspot.com/\\_lje0Hp\\_8mc8/SuzlRDrN\\_I/AAAAAAAAACY/LNP2ZJSfsQc/s320/57988445\\_9404e99848\\_o.jpg](http://3.bp.blogspot.com/_lje0Hp_8mc8/SuzlRDrN_I/AAAAAAAAACY/LNP2ZJSfsQc/s320/57988445_9404e99848_o.jpg)

### **2.2.1.1.3 Motorola.**

De acuerdo a lo indicado en: <https://microcontroladoresesv.wordpress.com/empresas-fabricantes-de-microcontroladores/> “Es una empresa dedicada a fabricar microprocesadores y microcontroladores entre otros productos, su mayor logro en la industria fue poner al mercado un microprocesador de 8 bits, llamado 6800. Motorola fue la primera compañía en construir otros periféricos como el microcontrolador 6820 y el 6850”.

Figura 4. Microcontrolador de Motorola.



Fuente:<http://www.forcedperfect.net/hardware/other/techstep/images/techstep-mc68hc24fn.jpg>

#### **2.2.1.1.4 Atmel Corporation.**

De acuerdo a lo indicado en: <https://microcontroladoresesv.wordpress.com/empresas-fabricantes-de-microcontroladores/> “Atmel es una compañía de semiconductores, fundada en 1984. Su línea de productos incluye microcontroladores, dispositivos de radiofrecuencia, diferentes tipos de memorias entre otros dispositivos electrónicos”.

Figura 5. Microcontroladores Atmel.



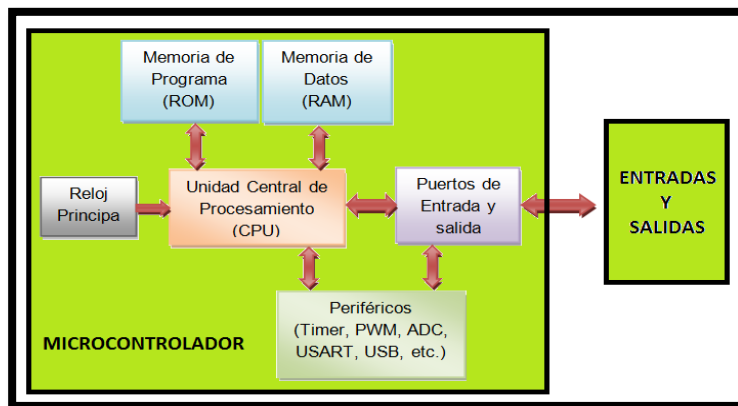
Fuente:[http://3.bp.blogspot.com/-ze2NnavG-WU/URfzsa\\_Z6jI/AAAAAAAAAU/2xdmGi4VfLo/s1600/microcontroladores\\_AVR.jpg](http://3.bp.blogspot.com/-ze2NnavG-WU/URfzsa_Z6jI/AAAAAAAAAU/2xdmGi4VfLo/s1600/microcontroladores_AVR.jpg)



### 2.2.1.2 Recursos comunes de los microcontroladores.

Un microcontrolador al estar integrado en un chip, su estructura fundamental y sus características básicas son muy parecidas, sin embargo cada fabricante busca enfatizar los recursos más eficientes para las diferentes aplicaciones que se empleen. En la siguiente figura se indica la estructura interna de un microcontrolador.

Figura 6. Estructura interna del microcontrolador



Fuente: El Autor.

#### 2.2.1.2.1 Unidad Central de Procesamiento o CPU.

La Unidad Central de Procesamiento (CPU) se encarga del procesamiento de la información que se recibe desde el exterior como también de la información que se envía hacia el exterior.

Es el elemento más importante del microcontrolador ya que es el que se encarga de direccionar la memoria de instrucciones, recibir el código de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento de los resultados. (Anibal, 2012)

#### **2.2.1.2.2 Memoria.**

Las memorias se utilizan o se emplean para dos funciones muy importantes como son:

- Almacenamiento de las instrucciones que forman el programa
- Almacenamiento de los datos y variables del programa.

Existen dos tipos de memorias:

- Memoria de Acceso Aleatorio (RAM): Es una memoria volátil, es decir, que guarda la información temporalmente mientras se encuentra alimentada por una fuente de tensión y es de lectura y escritura.
- Memoria de Solo Lectura (ROM): Es una memoria de valor fijo, es decir, que la información que se guarda será de forma definitiva y permite solo la lectura y no la escritura de la información y es independiente a la fuente de alimentación.

### **2.2.1.2.3 Líneas de entrada y salida. (E/S)**

También denominados puertos, los cuales comunican al CPU interno del microcontrolador con dispositivos externos tales como: leds, interruptores, display entre otros dispositivos que realizan una función determinada. Los puertos pueden ser configurados como (E/S) o de control dependiendo de la función que el microcontrolador realice.

### **2.2.1.2.4 Reloj Principal.**

Es un circuito que todos los microcontroladores poseen internamente, el cual genera una onda cuadrada de alta frecuencia, generalmente de 4 Megahercios (MHz), que controla los impulsos de reloj y permite la sincronización de las operaciones del sistema.

Este circuito puede ser configurado internamente o en ocasiones es necesario disponer de elementos externos, en casos de requerir un mayor tiempo de respuesta de un proceso, para seleccionar y estabilizar la frecuencia de trabajo. Estos elementos pueden ser un cristal de cuarzo junto a elementos pasivos como dos capacitores o resistencias según sea la configuración que se emplee.

#### **2.2.1.2.5 Periféricos.**

Son circuitos auxiliares que se encuentran internamente en el microcontrolador que realizan una determinada función, los cuales son: Conversores Análogo Digitales (ADC), Temporizadores, Moduladores por Ancho de Pulso (PWM), Puertas de comunicación serial, Bus Universal en Serie (USB), etc.

#### **2.2.1.3 Recursos especiales de los microcontroladores.**

Son módulos incorporados en un determinado microcontrolador, conocidos como fusibles del microcontrolador, que sirven de soporte para un mejor control de los periféricos los cuales se mencionan a continuación.

##### **2.2.1.3.1 Temporizadores o “timers”.**

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores). Para la medida de tiempo se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o disminuyendo al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a cero, momento en el que se produce un aviso. (Collaguazo, 2010)

#### **2.2.1.3.2 Perro guardián o “watchdog”.**

Consiste en un temporizador que produce un reinicio automático del microcontrolador cada cierto tiempo cuando dicho temporizador se desborda y pasa por cero.

#### **2.2.1.3.3 Power up timer.**

Mantiene al microcontrolador en un estado de reinicio hasta que la fuente de alimentación se estabilice, una vez que la fuente de alimentación se encuentre estable, se libera al microcontrolador de este estado de reinicio.

#### **2.2.1.3.4 Protección ante fallo de alimentación o “brownout”.**

Se trata de un circuito que ocasiona un reinicio del microcontrolador cuando el voltaje de alimentación es inferior al voltaje mínimo por un determinado tiempo, este voltaje mínimo debe ser inferior al proporcionado por la hoja de datos (datasheet) del microcontrolador; cuando se produce el reinicio, este se mantiene hasta que el voltaje de alimentación supere al voltaje mínimo, y el microcontrolador entra a funcionar normalmente.

#### **2.2.1.3.5 Estado de reposo o de bajo consumo.**

Los microcontroladores en varias ocasiones se encuentran sin realizar ninguna función por lo que el ahorro de energía es fundamental a la hora de

realizar un trabajo, en los microcontroladores existe una instrucción especial (Sleep en los PIC) la cual consiste en, poner al microcontrolador en un estado de reposo, bajo consumo de energía, en el momento en que no se encuentren realizando ninguna función, al activarse una interrupción accionada por algún dispositivo externo esperado, el microcontrolador se “despierta” y reanuda su trabajo.

#### **2.2.1.4 Características para la selección del microcontrolador.**

Al escoger algún tipo o fabricante de microcontroladores se debe tomar en cuenta algunos parámetros esenciales dependiendo de la aplicación que se desee realizar, tales como los mencionados a continuación.

##### ***2.2.1.4.1 Aplicación del microcontrolador.***

Antes de seleccionar un microcontrolador se debe tomar en cuenta cuál es el requerimiento de la aplicación que se desea realizar, para ello hay que tener en cuenta los siguientes parámetros.

###### ***2.2.1.4.1.1 Procesamiento de datos.***

Existen microcontroladores que pueden procesar datos de diferente tamaño, como son los de la gama de 8, 16 o 32 bits dependiendo del tipo de procesamiento de datos que se desee para la aplicación.

Si es necesario que el microcontrolador realice cálculos críticos en un tiempo limitado, es aconsejable seleccionar un dispositivo suficientemente rápido tomando en cuenta la precisión de los datos a manejar: si no es suficiente con un microcontrolador de 8 bits, puede ser necesario utilizar microcontroladores de 16 o 32 bits. (Torres & Aguirre, 2010).

#### *2.2.1.4.1.2 Pines de Entradas/Salidas*

Dependiendo de la aplicación y el número de dispositivos externos que se desee acoplar con el microcontrolador, se selecciona el número de puertos, para ello es aconsejable dibujar un diagrama de bloques de forma que sea más sencillo identificar la cantidad de señales a controlarse por el microcontrolador.

#### *2.2.1.4.1.3 Consumo de energía*

La alimentación del microcontrolador a escoger es vital por lo que al diseñar aplicaciones que necesiten un mayor consumo de energía y funcionen con baterías, es necesario que el microcontrolador pueda funcionar en un estado de bajo consumo para el ahorro de energía y poder asegurar el correcto funcionamiento.

#### *2.2.1.4.1.4 Capacidad de memoria del microcontrolador*

El tamaño de la memoria es muy necesario tomar en cuenta, según el código de la aplicación y las diferentes librerías que sean necesarias incorporar

en la aplicación a realizar, por lo general los fabricantes ofrecen diferentes tamaños de memoria ya sea esta ROM, RAM, etc.

#### ***2.2.1.4.2 Lenguaje de programación.***

Se refiere a la forma de cómo se programa al microcontrolador ya sea esta en un lenguaje de programación básico (ensamblador) o en un lenguaje de programación de alto nivel (microC, PICC Compiler, Codevision etc.).

#### ***2.2.1.4.3 Costo del microcontrolador.***

Al existir una gran variedad de microcontroladores de diferentes fabricantes en el mercado, cada fabricante los promociona a diferente precio según las características que tenga el microcontrolador. Hay que tomar en cuenta que el seleccionar un microcontrolador de bajo costo, éste puede aumentar el costo de los elementos adicionales que se incorporen al mismo, es por ello que es aconsejable analizar las prestaciones del microcontrolador para así reducir el costo total de la aplicación.

#### ***2.2.1.4.4 Diseño de la placa***

Se refiere al tamaño del microcontrolador y los dispositivos adicionales los cuales forman parte de la aplicación (circuito final). Al igual que en el costo, hay que tomar en cuenta que si se selecciona un microcontrolador de reducido



tamaño (microcontroladores del menor número de pines) este puede aumentar el número de dispositivos externos para la placa por lo que puede aumentar el tamaño final. Este parámetro también es dependiente de la aplicación que se realice.

### **2.2.1.5 Arquitectura de un microcontrolador y tipos de procesadores.**

Según la arquitectura interna de la memoria de un microcontrolador se puede clasificar considerando como el CPU accede a los datos e instrucciones, en 2 tipos:

#### **2.2.1.5.1 Arquitectura Von Neumann.**

Fue desarrollada por Jon Von Neumann, se caracteriza por tener una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. La CPU se conecta a través de un sistema de buses (direcciones, datos y control). Esta arquitectura es limitada cuando se demanda rapidez. (Collaguazo, 2010)

Figura 7. Arquitectura Von Neumann.

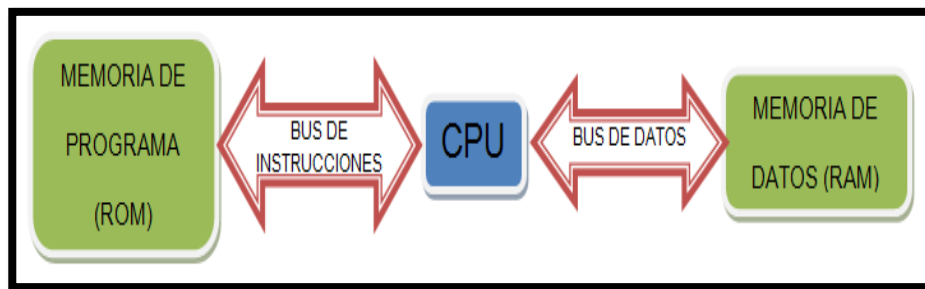


Fuente: El autor.

### **2.2.1.5.2 Arquitectura Harvard.**

Fue desarrollado en Harvard, por Howard Aiken, esta arquitectura se caracteriza por tener dos memorias independientes una que contiene sólo instrucciones y otra, que contiene sólo datos. Ambas, disponen de sus respectivos sistemas de buses para el acceso y es posible realizar operaciones de acceso simultáneamente en ambas memorias. (Collaguazo, 2010)

Figura 8. Arquitectura Harvard.



Fuente: El autor.

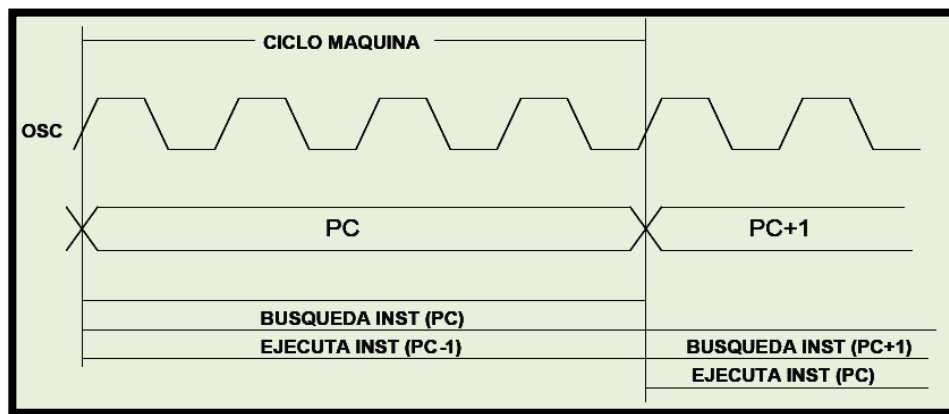
### **2.2.1.5.3 Procesador segmentado.**

(Collaguazo, 2010) “Realiza simultáneamente dos procesos:

- La ejecución de la instrucción y
- La búsqueda del código de la siguiente instrucción.

De esta manera, se puede ejecutar una instrucción en un solo ciclo máquina<sup>1</sup>. Este sistema está acompañado de una estructura Harvard, la cual permite que las instrucciones se ejecuten en un solo ciclo máquina, salvo en el salto de programa, tal como se indica en la figura siguiente.”

Figura 9. Procesador segmentado o pipeline.



Fuente: <http://es.scribd.com/doc/244728217/MICROCONTROLADORES-pdf#scribd>.

#### 2.2.1.5.4 Procesador según el número de instrucciones.

(Collaguazo, 2010) “Las CPUs atendiendo al tipo de instrucciones que utilizarán pueden clasificarse fundamentalmente en:

- *CISC (Complex Instruction Set Computer)*: Son procesadores con un juego de instrucciones complejas. Su repertorio de instrucciones es elevado y

<sup>1</sup>Ciclo máquina: Es el periodo que tarda la unidad central de proceso (CPU) en ejecutar una instrucción de lenguaje máquina.

algunas de ellas son muy sofisticadas y potentes. Su problema es que requieren de muchos ciclos de reloj para ejecutar las instrucciones.

- *RISC (Reduced Instruction Set Computer)*: Son microcontroladores con un repertorio de instrucciones reducido. Las instrucciones son muy simples y suelen ejecutarse en un solo ciclo máquina. Los procesadores RISC suelen tener una estructura Pipe Line y ejecutar casi todas las instrucciones en el mismo tiempo.
- *SISC (Specific Instruction Set Computer)*: Estos procesadores poseen un juego de instrucciones específico para cada aplicación. Están destinadas a aplicaciones muy cortas.”

#### **2.2.1.6 Comparación de microcontroladores de diferentes fabricantes.**

Los fabricantes de microcontroladores mencionados anteriormente son los más reconocidos globalmente por lo que se ha decidido estudiar un microcontrolador, el más representativo o “popular” a nivel comercial, de cada uno de ellos. Hay que tomar en cuenta que cada fabricante clasifica a los microcontroladores por “familias de microcontroladores” ya que cada “familia” tiene características similares, variando en algunos parámetros entre sí.

### **2.2.1.6.1 Microcontrolador de microchip.**

Microchip fabrica un sinnúmero de microcontroladores denominados PIC, pero el nombre verdadero del microcontrolador es PICmicro (Peripheral Interface Controller).

(Sánchez, 2013)“Su primer antecesor fue creado en 1975 por la compañía General Instruments. Este chip denominado PIC1650 fue diseñado para propósitos completamente diferentes.

Diez años más tarde, al añadir una memoria EEPROM<sup>2</sup>, este circuito se convirtió en un verdadero microcontrolador PIC. Hace unos pocos años la compañía Microchip Technology fabricó más de cinco billones de muestras.

En la siguiente figura se mencionan algunas de las familias de los microcontroladores creadas por Microchip con sus principales características.”

---

<sup>2</sup> EEPROM: Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente.

Figura 10. Familia de microcontroladores Microchip.

FAMILIA	ROM(Kbytes)	RAM(Kbytes)	PINES	Frecuencia de reloj. [MHz]	Entradas A/D	Resolución del convertidor A/D	Comparadores	Temporizadores de 8/16 bits	Comunicación Serial	Salidas PWM	Otros
<b>Arquitectura de la gama baja de 8 bits, palabra de instrucción de 12 bits</b>											
PIC10FXXX	0.375 - 0.75	16 - 24	06 - 08	4 - 8	0 - 2	8	0 - 1	1x8	-	-	-
PIC12FXXX	0.75 - 1.5	25 - 38	8	4 - 8	0 - 3	8	0 - 1	1x8	-	-	EEPROM
PIC16FXXX	0.75 - 3	25 - 134	14 - 44	20	0 - 3	8	0 - 2	1x8	-	-	EEPROM
PIC16HVXXX	1.5	25	18 - 20	20	-	-	-	1x8	-	-	VDD -15V
<b>Arquitectura de la gama media de 8 bits, palabra de instrucción de 14 bits</b>											
PIC12FXXX	1.75 - 3.5	64 - 128	8	20	0 - 4	10	1	((1 - 2)x8) - (1x16)	-	0 - 1	EEPROM
PIC12HVXXX	1.75	64	8	20	0 - 4	10	1	((1 - 2)x8) - (1x16)	-	0 - 1	-
PIC16FXXX	1.75 - 14	64 - 368	14 - 64	20	0 - 13	8 ó 10	0 - 2	((1 - 2)x8) - (1x16)	USART, I2C, SPI	0 - 3	-
PIC16HVXXX	1.75 - 3.5	64 - 128	14 - 20	20	0 - 12	10	2	(2x8) - (1x16)	USART, I2C, SPI	-	-
<b>Arquitectura de la gama alta de 8 bits, palabra de instrucción de 16 bits</b>											
PIC18FXXX	4 - 128	256 - 3936	18 - 80	32 - 48	4 - 16	10 ó 12	0 - 3	((0 - 2)x8) - ((2 - 3)x16)	USB2.0, CAN2.0, USART, I2C, SPI	0 - 5	-
PIC18FXXJXX	8 - 128	1024 - 3936	28 - 100	40 - 48	10 - 16	10	2	((0 - 2)x8) - ((2 - 3)x16)	USB2.0, ETHERNET, USART, I2C, SPI	2 - 5	-
PIC18FXXXXX	8 - 64	768 - 3936	28 - 44	64	10 - 13	10	2	(1x8) - (3x16)	USART, I2C, SPI	2	-

Fuente: <http://microcontroladoresesv.wordpress.com/microcontroladores-pic-y-sus-variedades/tabla/>

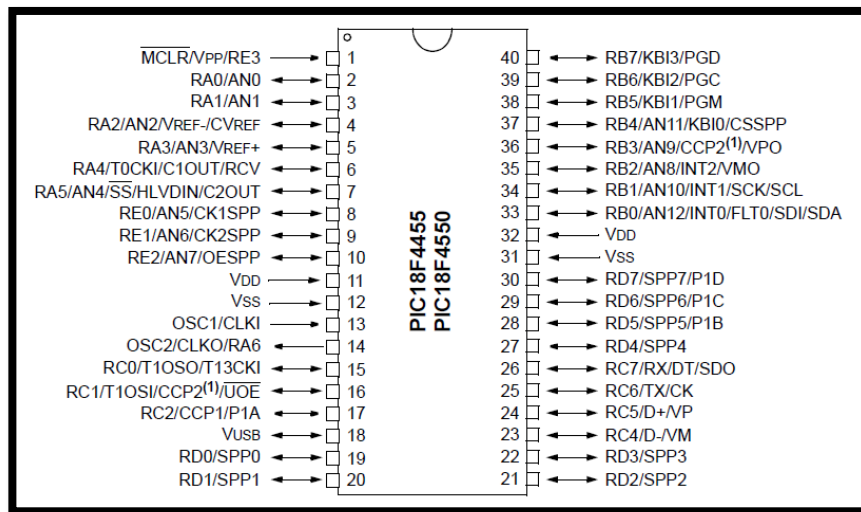
Para tomar como referencia a uno de los microcontroladores de Microchip más populares en el mercado de la electrónica y por tener algunas de las características que es necesario para este proyecto, se tomará como referencia al PIC18F4550.

### 2.2.1.6.1.1 Microcontrolador PIC18F4550.

Este microcontrolador tiene un encapsulado tipo DIP<sup>3</sup> de 40 pines, cada uno de ellos realiza diferentes funciones, dependiendo de lo que se desee, en la siguiente figura se indica las funciones del mismo.

<sup>3</sup> DIP: Es una forma de encapsulamiento común en la construcción de circuitos integrados.

Figura 11. Microcontrolador PIC18F4550.



Fuente:<http://ww1.microchip.com/downloads/en/DeviceDoc/40044D.pdf>

### 2.2.1.6.1.2 Principales características del PIC18F4550.

- CPU De alto rendimiento con un procesador RISC
- Opciones de oscilador externo e interno
- Modo de ahorro de energía en modo sueño
- Resistencias programables en los puertos
- Puerto Serial USART Programables
- Puerto USB Programable.
- Alta durabilidad de la memoria Flash /EEPROM:
  - 2048 bytes de memoria Flash
  - 224 bytes de memoria RAM
  - 128 bytes de memoria EEPROM
  - 40 años de retención de datos

- Amplio rango de funcionamiento de tensión (2.0 a 5.5 Voltios)
- Características de baja potencia:
  - Corriente en espera:
    - 100 nanoAmperios a 2.0 Voltios, típico
  - Corriente de funcionamiento:
    - 12 microAmperios ( $\mu\text{A}$ ) a 32 kilohercios (kHz), 2,0 Voltios, típica

#### **2.2.1.6.2 Microcontrolador de Atmel**

(Maroné) “Los AVR son una familia de microcontroladores con un procesador tipo RISC. La arquitectura de los AVR fue concebida por dos estudiantes en el Norwegian Institute of Technology, y posteriormente refinada y desarrollada en Atmel Norway, la empresa subsidiaria de Atmel, fundada por los dos arquitectos del chip. “

En la siguiente figura se mencionan algunos de los microcontroladores creados por Atmel con sus principales características.



Figura 12. Familia de microcontroladores Atmel.

DISPOSITIVO	MEMORIA			I/O Max. Pins	FREC. Max. MHz	Vcc Rango V	TIMERS			SERIALES				ANALOGICOS	
	Flash Kbytes	EEPROM Kbytes	SRAM Bytes				de 16-bit	de 8-bit	PWM canales	SPI	USART	TWI	ISP	10-bit A/D canales	Compa- rador
ATmega48	4	0.25	512	23	20	1.8-5.5	1	2	6	1+USART	1	Si	Si	6 de 8	Si
ATmega48 Automotive	4	0.25	512	23	16	2.7-5.5	1	2	6	1+USART	1	Si	Si	8	Si
ATmega48P	4	0.25	512	23	20	1.8-5.5	1	2	6	1+USART	1	Si	Si	8	Si
ATmega8	8	0.5	1024	23	16	2.7-5.5	1	2	3	1	1	Si	Si	6 de 8	Si
ATmega88 Automotive	8	0.5	1024	23	16	2.7-5.5	1	2	6	1+USART	1	Si	Si	8	Si
ATmega88	8	0.5	1024	23	20	1.8-5.5	1	2	6	1+USART	1	Si	Si	6 de 8	Si
ATmega88P	8	0.5	1024	23	20	1.8-5.5	1	2	6	1+USART	1	Si	Si	8	Si
ATmega16	16	0.5	1024	32	16	2.7-5.5	1	2	4	1	1	Si	Si	8	Si
ATmega16P Automotive	16	0.5	1024	32	16	2.7-5.5	1	2	6	1+USART	2	Si	Si	8	Si
ATmega164P	16	0.5	1024	32	20	1.8-5.5	1	2	6	1+USART	2	Si	Si	8	Si
ATmega165	16	0.5	1024	54	16	1.8-5.5	1	2	4	1+USI	1	Si	Si	8	Si
ATmega165P	16	0.5	1024	54	16	1.8-5.5	1	2	4	1+USI	1	Si	Si	8	Si
ATmega32	32	1	2048	32	16	2.7-5.5	1	2	4	1	1	Si	Si	8	Si
ATmega324P Automotive	32	1	2048	32	16	2.7-5.5	1	2	6	1+USART	2	Si	Si	8	Si
ATmega324P	32	1	2048	32	20	1.8-5.5	1	2	6	1+USART	2	Si	Si	8	Si
ATmega325	32	1	2048	54	16	1.8-5.5	1	2	4	1+USI	1	Si	Si	8	Si
ATmega64	64	2	4096	54	16	2.7-5.5	2	2	8	1	2	Si	Si	8	Si
ATmega644P Automotive	64	2	4096	32	16	2.7-5.5	1	2	6	1+USART	2	Si	Si	8	Si
ATmega640	64	4	8192	86	16	1.8-5.5	4	2	16	1+USART	4	Si	Si	16	Si
ATmega644	64	2	4096	32	20	1.8-5.5	1	2	6	1+USART	1	Si	Si	8	Si
ATmega128	128	4	4096	53	16	2.7-5.5	2	2	8	1	2	Si	Si	8	Si
ATmega1280	128	4	8192	86	16	1.8-5.5	4	2	16	1+USART	4	Si	Si	16	Si
ATmega1281	128	4	8192	54	16	1.8-5.5	4	2	9	1+USART	2	Si	Si	8	Si
ATmega2560	256	4	8192	86	16	1.8-5.5	4	2	16	1+USART	4	Si	Si	16	Si
ATmega2561	256	4	8192	54	16	1.8-5.5	4	2	9	1+USART	2	Si	Si	8	Si

Fuente:<http://image.slidesharecdn.com/07familiaatmega-090921042621-phpapp02/95/07-familia-atmega-7-1024.jpg?cb=1315170709>

Uno de los microcontroladores AVR más utilizados en diferentes aplicaciones es el ATmega164P, en este caso se conocerán las principales características del mismo.

#### 2.2.1.6.2.1 Microcontrolador ATmega164P.

Este microcontrolador tiene un encapsulado tipo DIP de 40 pines, cada uno de ellos realiza diferentes funciones, dependiendo de lo que se desee, en la siguiente figura se indican las funciones del mismo.

Figura 13. Microcontrolador ATmega164P.

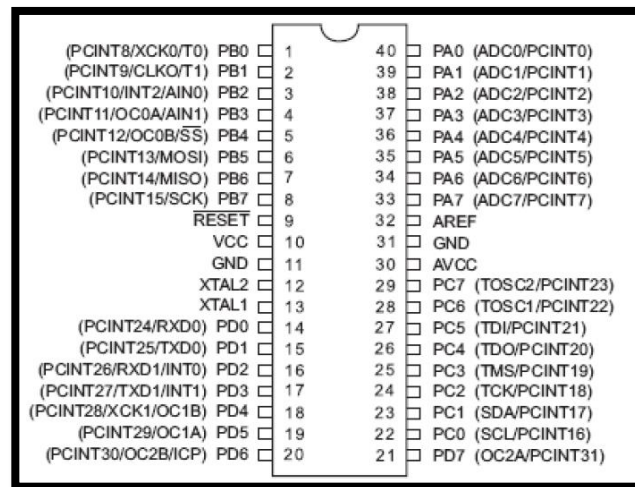


Figura: <http://www.atmel.com/images/doc7674.pdf>

#### 2.2.1.6.2.2 Principales características del ATmega164P.

- Arquitectura Avanzada RISC
- Memorias de programa y de datos no volátiles de alta duración
  - 16/32/44 K bytes de FLASH
  - 512B/1K/2K bytes de EEPROM
- Características de los periféricos
  - 6 Canales para PWM
  - ADC de 10 bits y 8 canales
  - Dos puertos Seriales USART Programables
- Velocidad de Funcionamiento
  - ATMEGA 164P/324P/644PV: (0 – 10MHz a 2.7 – 5.5V) – (20MHz a 4.5– 5.5V)

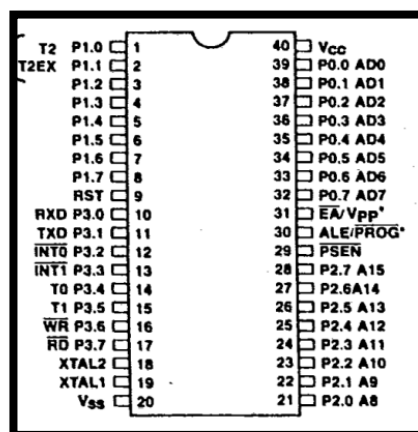
### 2.2.1.6.3 Microcontrolador de Intel.

Los microcontroladores Intel son una serie de chip único de arquitectura Harvard, con el conjunto de instrucciones CISC, que fueron desarrollados para el uso en sistemas embebidos<sup>4</sup>. Versiones originales de Intel eran populares en la década de 1980 y principios de 1990 y mejorados derivados compatibles binarios siguen siendo populares hoy en día. La denominación oficial de Intel para la familia de microcontroladores 8051 es MCS 51. (Sánchez, 2013)

#### 2.2.1.6.3.1 Microcontrolador Intel 8051.

El Intel 8051 es un microcontrolador desarrollado por Intel en 1980, es un microcontrolador muy popular.

Figura 14. Microcontrolador intel 8051.



Fuente: [http://pdf.datasheetcatalog.net/datasheets2/10/1046235\\_1.pdf](http://pdf.datasheetcatalog.net/datasheets2/10/1046235_1.pdf)

<sup>4</sup> Sistemas Embebidos: Es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas.

### 2.2.1.6.3.2 Principales características del Intel 8051.

- ALU<sup>5</sup> de 8 bits y acumuladores, registros de 8 bits, bus de datos de 8 bits y dirección de bus/contador de programa/datos.
- Procesador con 17 instrucciones, 1 bit de acumuladores, 32 registros y un máximo de 144 variables de RAM direccionable de 1 bit especiales.
- Interrupción rápida con la opción de registro de cambio de banco.
- Interrupciones y subprocesos con prioridad seleccionable.
- 128 bytes de memoria RAM en el chip.
- 4 Kbyte de ROM en el chip, con un espacio de direcciones de 16 bits.
- Cuatro puertos de entrada/salida bidireccional byte.
- Dos contadores/temporizadores de 16 bits.
- Voltaje de entrada de -0.5 V a 21.5 V. Potencia de 1.5W.
- Modo ahorro de energía.

### 2.2.1.7 Selección del microcontroladores adecuado.

La selección del microcontrolador se realizará en base a los parámetros señalados anteriormente, teniendo en cuenta a los microcontroladores más populares y representativos de cada fabricante, que se encuentran hoy en día disponibles en el mercado de la electrónica, y, que pueden ser acoplados a este proyecto según las necesidades presentadas al inicio de este capítulo.

---

<sup>5</sup> Unidad Aritmética Lógica (ALU): Es un circuito digital que calcula operaciones aritméticas (suma, resta, multiplicación, etc.) y operaciones lógicas (si, y, o, no), entre dos números.

Tabla 1. Comparación de microcontroladores de diferente fabricante.

PARÁMETROS		PIC18F4550	ATmega164P	Intel 8051
<b>Arquitectura</b>		RISC	RISC	CISC
<b>Procesamiento de datos</b>		8 bits	8 bits	8 bits
<b>Ancho de palabra</b>		8 bits	8 bits	16 bits
<b>Memoria</b>	RAM	2048 bytes	4000 bytes	128 bytes
	FLASH	32000 bytes	44000 bytes	4000 bytes
	EEPROM	256 bytes	2000 bytes	-
<b>Tensión de alimentación</b>	Max	5.5 V	5.5 V	21.5 V
	Min	2.0 V	4.5 V	-0.5 V
<b>Puertos de Entrada/Salida (Pines)</b>		5 (40 Pines)	4 (40 pines)	4 (40 pines)
<b>Comunicación</b>	USART	SI	SI	SI
	USB	SI	NO	NO
<b>Lenguaje de Programación</b>		Fácil	Fácil	Complejo
<b>Tamaño</b>		Mediano	Mediano	Mediano
<b>Costo</b>		8,00	10,50	8,50

Fuente: El Autor.

Como se puede observar en la tabla anterior, las características de los microcontroladores escogidos presentan pocas diferencias, pero se decidió escoger el PIC18f4550 por las siguientes razones:

- *El costo:* El precio de este microcontrolador en el mercado es más económico en comparación con otros ya sea por una mínima cantidad pero

eso representa en un proyecto como éste y lo más importante es que se adapta al proyecto de una mejor manera en otras características.

- *Consumo de energía.* Los microcontroladores de Microchip pueden funcionar con un voltaje de 2.2 voltios (V) sin ningún problema a diferencia de los otros, que, necesariamente deben tener una alimentación relativamente alta, casi a su máxima tensión, para su correcto funcionamiento, lo que podría traer complicaciones en un futuro.
- *Memoria:* También debido a que no se manejará tanta programación en el microcontrolador, no es muy necesario disponer de tanta capacidad de memoria, por lo que si se utiliza el PIC18F4550 no afectaría en lo absoluto al proyecto.
- *Puertos Entrada/Salida:* Los microcontroladores de microchip tiene una característica importante en los puertos ya que se pueden utilizar las configuraciones de resistencias internas, lo que es muy importante para utilizar el teclado y reducir la utilización de resistencias externas, por ende tener una placa más pequeña.
- *Lenguaje de Programación:* Otro de los factores que hay que tomar en cuenta es el lenguaje de programación de los microcontroladores y en qué programa se los realizará, es decir, qué programa se utilizará para escribir el código de programación en el microcontrolador según su lenguaje, este parámetro es muy importante ya que no todos los microcontroladores se pueden programar en un solo lenguaje de programación o con un solo programa, cada fabricante de microcontrolador tiene su propio software para realizar esta tarea, en este caso se tomó la decisión de utilizar este microcontrolador por la facilidad de programar en un lenguaje de alto

nivelal reducir el código de programación y la facilidad de utilizar el programa PICC Compile en el cual se realizará la programación del mismo.

## **2.2.2 Elementos adicionales del hardware**

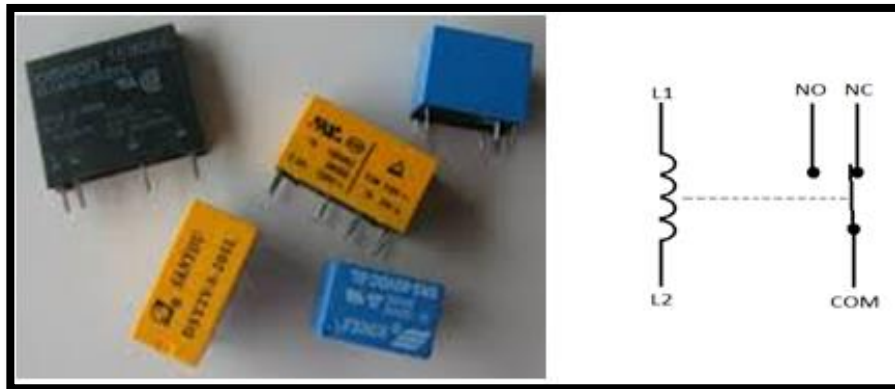
Son dispositivos electrónicos que sirven para el correcto funcionamiento del hardware o circuito electrónico del proyecto. Estos elementos servirán ya sea para la alimentación de todo el circuito como para la visualización de datos que se pueden ingresar manualmente o provenientes desde una comunicación externa, indicadores del estado del hardware y la apertura de la puerta según sea su función.

### **2.2.2.1 Relé.**

Un relé o relevador es un sistema electromecánico que funciona como un interruptor el cual es accionado mediante un electroimán. Está construido por las siguientes partes, tal como se muestra en la siguiente figura.

- Un circuito electroimán (Control)
- Un circuito de contactores (Funcional)

Figura 15. Estructura física y simbología de un relé.



Fuente:<http://st-elf.electronicafacil.net/tutoriales/147/rele1b.gif>

De acuerdo a lo indicado en: <http://www.electronicafacil.net/tutoriales/EI-rele.php> “Su funcionamiento se basa en el fenómeno electromagnético. Cuando la corriente atraviesa la bobina, produce un campo magnético que magnetiza un núcleo de hierro dulce (ferrita). Este atrae al inducido que obliga a los contactos a tocarse. Cuando la corriente se desconecta vuelven a separarse”.

### 2.2.2.2 Transistores.

El transistor es un dispositivo electrónico semiconductor utilizado para producir una señal de salida en respuesta a otra señal de entrada, el cual cumple las funciones de amplificador, oscilador, rectificador o conmutador. Existen diferentes tipos de transistores los cuales se puede utilizar dependiendo de la aplicación que se realice, siendo los más utilizados en el mundo de la electrónica el transistor de unión bipolar.



### 2.2.2.1 Transistor de unión bipolar. (BJT)

De acuerdo a lo indicado en: <http://www.slideshare.net/Naren05/transistor-y-tipos-de-transistores-13676180> “Es construido con materiales de Germanio, Silicio o Arseniuro de galio. Consta de tres regiones semiconductoras distintas llamadas, Emisor, Base y Colector. La Base es una región muy delgada de un tipo de semiconductor intercalada entre dos regiones de tipo opuesto al suyo. El semiconductor emisor está mucho más dopado que el de la base y el colector”. Este transistor consta de dos diferentes configuraciones o tipos los cuales son:

- NPN: En un transistor npn el emisor y el colector son semiconductores de tipo n y la base es un semiconductor de tipo p
- PNP: En un transistor pnp la base es un semiconductor de tipo n mientras que en el emisor y el colector son semiconductores de tipo p.

En la siguiente figura se muestra la construcción de un transistor BJT en sus dos configuraciones con el tipo de material de cada uno.

Figura 16. Estructura de un transistor BJT.

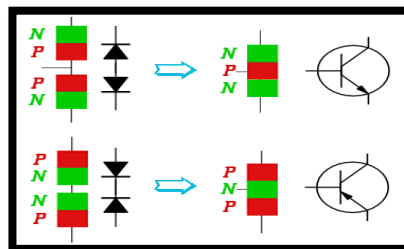
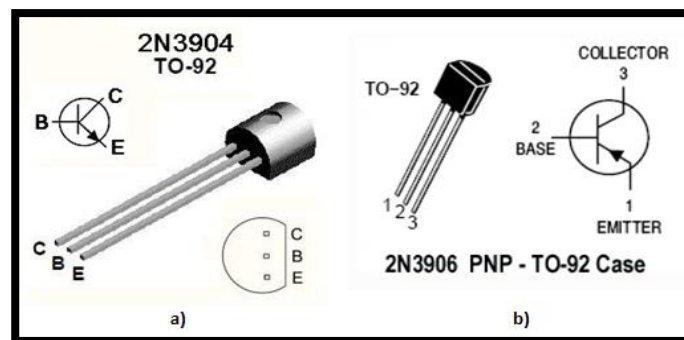


Figura: <http://guardiolajavi.wordpress.com/2013/03/27/tecno-tapa-el-transistor-bipolar/>

En la siguiente figura se muestran los diferentes tipos de transistores que se encuentran hoy en día en el mercado, siendo los más utilizados en el mundo de la electrónica el 2N3904 y 2N3906, pertenecientes a los BJT del tipo NPN y PNP respectivamente, los mismos que se utilizan en este proyecto.

Figura 17. Estructura física y simbología de un transistor BJT en sus dos configuraciones.



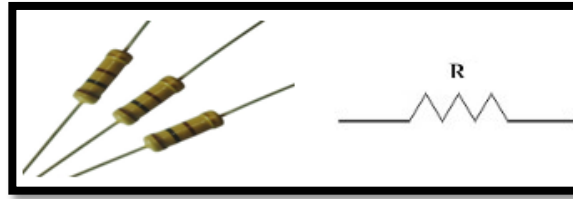
Fuentes: a) <http://upload.wikimedia.org/wikipedia/commons/e/ea/2n3904.jpg>

b) <http://www.galigear.com.au/2N3906-PNP-TO-92-Transistor-for-sale>

### 2.2.2.3 Resistencia.

Es un elemento electrónico importante en todo circuito eléctrico ya que su función principal es obtener una caída de tensión que es proporcional al flujo de corriente eléctrica, es decir, permite reducir ya sea la corriente o el voltaje a la entrada de un determinado dispositivo electrónico al que se le aplique. En el mercado comercial existen diversos tipos de resistencias, siendo las más comunes y utilizadas las resistencias de película de carbón.

Figura 18. Estructura física y simbología de una resistencia.



Fuente: [http://www.resistorinchina.es/big\\_img.html?etw\\_path=http://www.resistorinchina.es/7-3-carbon-film-fixed-resistor.html&big\\_etw\\_img=products/7-3-1b.jpg](http://www.resistorinchina.es/big_img.html?etw_path=http://www.resistorinchina.es/7-3-carbon-film-fixed-resistor.html&big_etw_img=products/7-3-1b.jpg)

Para la integración de una resistencia en un circuito eléctrico ésta se calcula mediante la ley de Ohm y su valor viene dado por el código de colores tal como se indica en la figura siguiente.

Figura 19. Código de colores para la selección de una resistencia.

COLOR	1 <sup>ST</sup> BAND	2 <sup>ND</sup> BAND	3 <sup>RD</sup> BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8		± 0.05%
White	9	9	9		
Gold				0.1Ω	± 5% (J)
Silver				0.01Ω	± 10% (K)

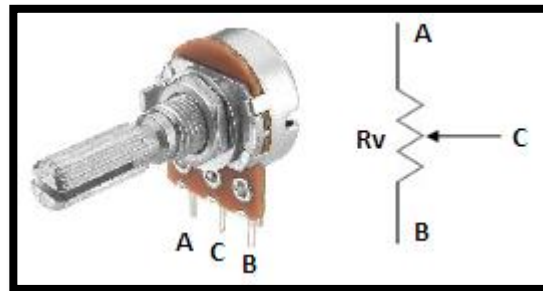
Fuente: <http://www.digikey.com/web%20export/mkt/general/mkt/resistor-color-chart.jpg>

#### 2.2.2.4 Potenciómetro.

Los potenciómetros, también denominados resistencias variables son un caso especial de resistencias ya que permiten obtener valores desde cero (sin resistencia) hasta el valor máximo de resistencia con el que se identifica este elemento.

Este tiene la función de controlar la corriente o la tensión que circula por este elemento. Su aplicación es diversa siendo más utilizadas en la intensidad luminosa de pantallas, el volumen de los parlantes, ecualizadores, mezcladoras de audio, etc.

Figura 20. Estructura física y simbología de un potenciómetro.



Fuente: [http://www.cetronic.es/sqlcommerce/ficheros/dk\\_93/productos/451220016-1.jpg](http://www.cetronic.es/sqlcommerce/ficheros/dk_93/productos/451220016-1.jpg)

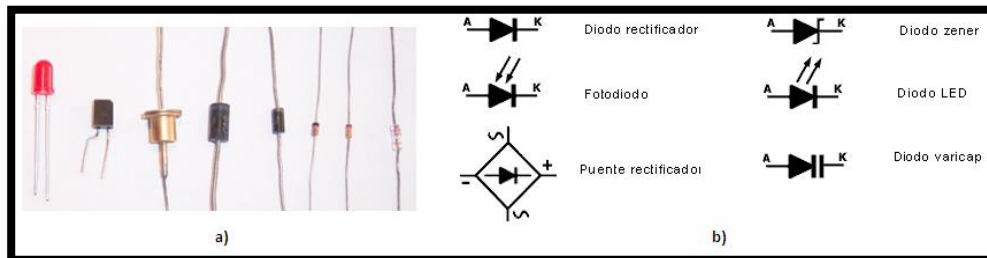
### 2.2.2.5 Diodos.

Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido. Las principales aplicaciones de los diodos son:

- Rectificador de media onda
- Rectificador de onda completa
- Doblador de tensión
- Estabilizador de tensión (Zener)
- Luminosidad (Led)
- Multiplicador y divisor de tensión

Existen diversos tipos de diodos, en la siguiente figura se indica algunos de ellos ya que cada uno realiza una función diferente.

Figura 21. Estructura física y simbología del diodo.



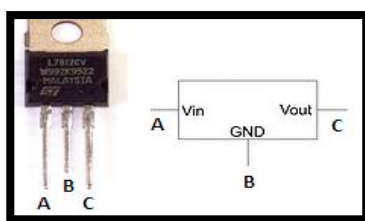
Fuente: a) [http://www.portaleso.com/usuarios/Toni/web\\_electronica\\_3/imagenes/diodos.jpg](http://www.portaleso.com/usuarios/Toni/web_electronica_3/imagenes/diodos.jpg)

b) [http://www.geocities.ws/jjrc\\_79/electronica/fundamentos\\_/simbologia/diodos.gif](http://www.geocities.ws/jjrc_79/electronica/fundamentos_/simbologia/diodos.gif)

### 2.2.2.6 Regulador de voltaje.

De acuerdo a lo indicado en: <http://artefactos.leame.com/reguladores-de-tension-78xx-y-79xx/> “Los reguladores lineales de tensión, también llamados reguladores de voltaje, son circuitos integrados diseñados para entregar una tensión constante y estable. Estos dispositivos están presentes en la gran mayoría de fuentes de alimentación, pues proporcionan una estabilidad y protección sin apenas necesidad de componentes externos, haciendo que sean muy económicos”.

Figura 22. Estructura física y simbología de un regulador de voltaje.

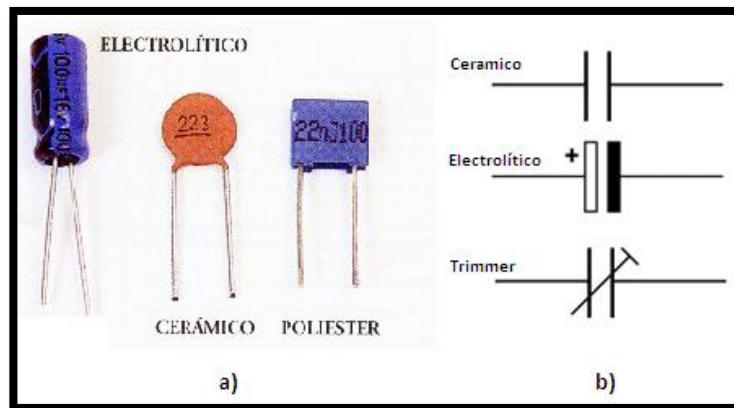


Fuente: <http://www.comohacerturobot.com/Electronica/regulador2.jpg>

### 2.2.2.7 Condensadores.

Es un componente electrónico cuya función es la de almacenar carga eléctrica y su aplicación más importante es la de corregir el factor de potencia. En el mercado existen diversos tipos de condensadores, siendo los más comunes y utilizables los condensadores cerámicos y electrolíticos.

Figura 23. Estructura física y simbología del condensador.



Fuente: a) <http://www.comohacerturobot.com/Electronica/tiposcondensa.jpg>

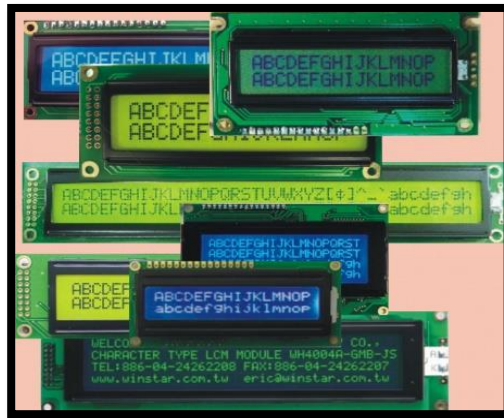
b) <http://3.bp.blogspot.com/-xcbz1iVhvA0/U-TzmDATzel/AAAAAAAAAv8/cobtDiyOVHE/s1600/simbolo.gif>

### 2.2.2.8 Pantalla de cristal líquido (LCD).

La pantalla de cristal líquido es un dispositivo de visualización gráfica para la presentación de caracteres alfanuméricos que se requiere en un sistema de control. Este dispositivo tiene características especiales en comparación con otros medios de visualización como es el display de siete segmentos, éste no puede presentar caracteres alfanuméricos y tiene un consumo de corriente relativamente alto mientras que la LCD es todo lo contrario.

Las LCD se pueden encontrar en el mercado de diferentes presentaciones que identifican el número de filas y columnas como es la de 2x16, 2x20, 4x20, 4x40, etc.

Figura 24. Estructura física de la LCD.



Fuente: <http://www.trastejant.es/tutoriales/arduino/img/lcld/Tiposdelcd.jpg>

### 2.2.2.9 Teclado matricial.

Un teclado matricial es básicamente construido en forma de una matriz de  $n$  filas por  $m$  columnas, dando como resultado al ser presionado un caracter específico, tomando como referencia la posición de la fila y la columna. En el mercado existen diversos tipos de teclados, siendo los más utilizados en el mundo de la electrónica el teclado matricial de 4x4.

Figura 25. Estructura física del teclado matricial.



Fuente: <http://www.tiendaelectronica.com.ve/1122-1378-large/teclado-matricial-4x4.jpg>



### 2.2.2.10 Disipadores de calor.

Un disipador de calor es un componente metálico, éste puede ser de aluminio, plata, cobre, oro, hierro o acero, pero generalmente es de aluminio el cual se utiliza para reducir la acumulación de calor en los elementos electrónicos como transistores, reguladores, circuitos integrados, etc. Evitando así el calentamiento de los mismos y su futuro daño.

Un disipador extrae el calor del elemento que se encuentra refrigerando y lo evacua al exterior, generalmente hacia el aire. El disipador de calor se encuentra en diferentes equipos electrónicos como televisores, radios y elementos de alimentación como fuentes reguladas, etc. Según el elemento a refrigerar, éste tendrá diferente forma y tamaño.

Figura 26. Estructura física del disipador de calor.



Fuente: [http://i00.i.aliimg.com/photo/v0/383674144/Aluminum\\_heatsink.jpg](http://i00.i.aliimg.com/photo/v0/383674144/Aluminum_heatsink.jpg)

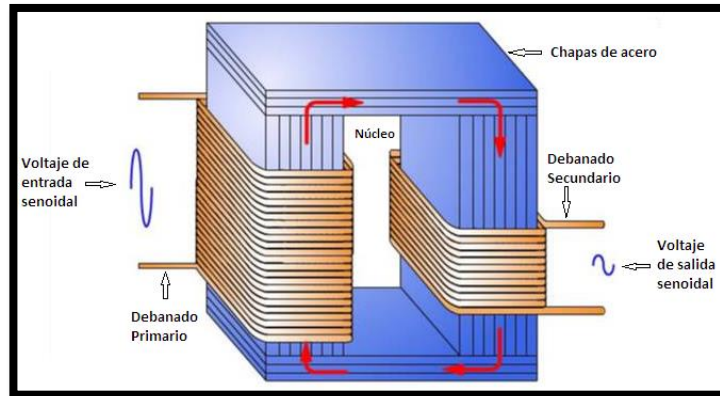
### **2.2.2.11 Transformador.**

De acuerdo a lo indicado en: <http://ddtorres.webs.ull.es/Docencia/Intalaciones/Electrifica/Tema%203.htm> “Los transformadores son dispositivos electromagnéticos estáticos que permiten, partiendo de una tensión alterna conectada a su entrada, obtener otra tensión alterna mayor o menor que la anterior en la salida del transformador. Permitiendo así proporcionar una tensión adecuada a las características de los receptores.

También son fundamentales para el transporte de energía eléctrica a largas distancias a tensiones altas, con mínimas pérdidas y conductores de secciones moderadas.

Constan esencialmente de un circuito magnético cerrado sobre el que se arrollan dos bobinados, de forma que ambos bobinados están atravesados por el mismo flujo magnético. El circuito magnético está constituido por chapas de acero de poco espesor apiladas, para evitar las corrientes parásitas. El bobinado donde se conecta la corriente de entrada se denomina primario, y el bobinado donde se conecta la carga útil, se denomina secundario. La corriente alterna que circula por el bobinado primario magnetiza el núcleo de forma alternativa. El bobinado secundario está así atravesado por un flujo magnético variable de forma aproximadamente sinodal y esta variación de flujo engendra una tensión alterna en dicho bobinado”.

Figura 27. Estructura interna de un transformador.



Fuente: [http://3.bp.blogspot.com/-WHmAUMEvVK0/UKcZN-snJZI/AAAAAAAAAHY/pF-\\_Gv9niPg/s1600/Imagen14.jpg](http://3.bp.blogspot.com/-WHmAUMEvVK0/UKcZN-snJZI/AAAAAAAAAHY/pF-_Gv9niPg/s1600/Imagen14.jpg)

Los transformadores existen de diversos tamaños, esto depende de la carga que se desee alimentar, en el mundo de la electrónica no son tan grandes pero sí pueden variar su tamaño dependiendo del voltaje de salida que se desee en un circuito electrónico.

Figura 28. Estructura física y simbología del transformador.



Fuente: <http://3.bp.blogspot.com/-P-qUn79qTIw/TbR8Ufw07DI/AAAAAAAAABA/jpVyBKcQWTg/s320/Transformadores-eletricos.jpg>

### **2.2.3 Módulos inalámbricos**

Los módulos inalámbricos son dispositivos que permiten agregar conectividad sin la necesidad de utilizar el cableado a la red y expandirla. Dichos módulos son diseñados por diferentes fabricantes, al igual que en los microcontroladores, pero, cada uno de ellos pertenece a un diferente estándar inalámbrico que especifican ciertas características de los mismos.

Estos estándares son referenciados en la IEEE 802.x, donde x especifica la tecnología con la cual se manejará la comunicación.

#### **2.2.3.1 Estándares inalámbricos.**

(Torres & Aguirre, 2010) “En 1978 la US-NSPAC (Comité consultivo de la política nacional de los estándares) definió “estándar” como: “Un sistema de reglas prescrito, condiciones o requerimientos que atañen a las definiciones de los términos; clasificación de los componentes; especificación de materiales, prestaciones u operaciones; delimitación de procedimientos; o medidas de la cantidad y calidad en la descripción de materiales, productos, sistemas, servicios o prácticas.

Los estándares son usados para garantizar seguridad, calidad, y consistencia en los equipos. Un equipo que sigue un estándar específico implica

la posibilidad de interoperabilidad con otros productos y de no estar “atado” a un vendedor único”.

#### **2.2.3.1.1 Estándares abiertos y cerrados.**

Se pueden dividir los estándares entre abiertos y cerrados (exclusivos de un fabricante o vendedor).

(Torres & Aguirre, 2010) “Un estándar abierto está disponible públicamente, mientras que uno cerrado no. Los estándares cerrados están disponibles solo bajo términos muy restrictivos establecidos en un contrato con la organización que posee el copyright de la especificación.

Un ejemplo de estándar abierto es HTML mientras que el formato de un documento de Microsoft Office es cerrado. Un estándar abierto aumenta la compatibilidad entre el hardware, software o sistemas, puesto que el estándar puede ser implementado por cualquiera. Un estándar abierto no implica necesariamente que sea exento de pago de derechos o de licencias”.

A continuación se analizan tres diferentes tecnologías que utilizan diferentes estándares dando énfasis en las principales características de cada uno de ellos.

### **2.2.3.2 IEEE 802.15.4 (ZIGBEE).**

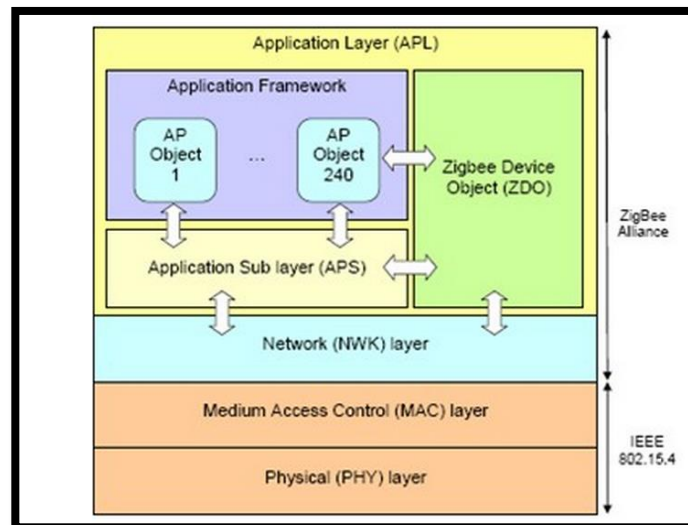
(Saucedo, 2012) “Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE 802.15.4. Creado por Zigbee Alliance, una organización, teóricamente sin ánimo de lucro, de más de 200 grandes empresas (destacan Mitsubishi, Honeywell, Philips, Motorola, entre otros), muchas de ellas fabricantes de semiconductores.

Este protocolo está siendo proyectado para permitir comunicación inalámbrica confiable, con bajo consumo de energía y bajas tasas de transmisión para aplicaciones como: sensores, juguetes interactivos, insignias inteligentes, controles remotos y la automatización del hogar, esta última aplicación, permite que las aplicaciones domóticas desarrolladas por los fabricantes sean completamente interoperables entre sí, garantizando así al cliente final fiabilidad, control, seguridad y comodidad”.

#### ***2.2.3.2.1 Pila de protocolos del estándar 802.15.4.***

(Revista de la Asociación de Ingenieros del ICAI, 2012)“Zigbee utiliza una pila de comunicaciones, similar al modelo de referencia OSI (Open System Interconnection), formada por varias capas independientes entre sí y con funciones específicas”.

Figura 29. Pila de protocolos de Zigbee.



Fuente: <http://slideplayer.es/slide/169350/>

Las capas más bajas de dicha pila (PHY y MAC) están definidas en el estándar IEEE 802.15.4; mientras que el resto de capas (NWK y APL), las define la alianza Zigbee.

#### 2.2.3.2.1.1 Capas PHY y MAC: IEEE 802.15.4.

El estándar IEEE 802.15.4 define las capas de nivel físico (PHY) y de nivel de enlace (MAC) de la pila empleada por Zigbee.

#### Capa PHY.

La capa de nivel físico se encarga de transmitir y recibir tramas de bits sobre el medio físico (el aire), modulando los bits a transmitir y demodulando la señal recibida. El dispositivo puede funcionar en distintas bandas de frecuencia

con distinta tasa binaria de transmisión de información y distinto número de canales, con un alcance global entre 10 y 75 metros.

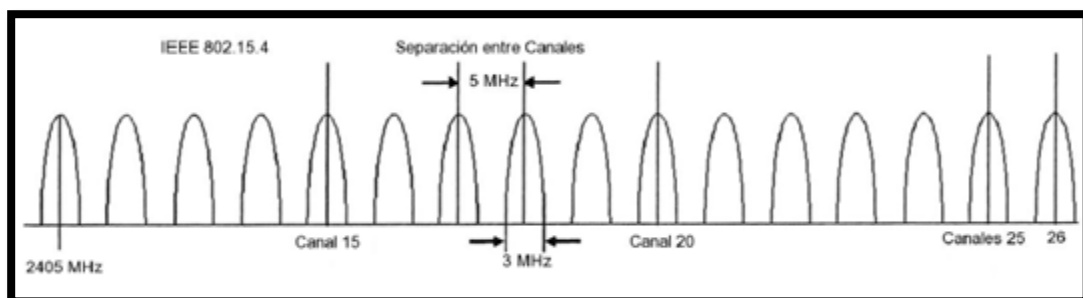
Tabla 2. Bandas de frecuencia de Zigbee.

Bandas de frecuencias	Transmisión	Canales	Modulación
868 - 868,8 MHz	20 Kbps	1	BPSK
902 - 928 MHz	40 Kbps	10	BPSK
2.4 - 2.483,5 MHz	250 Kbps	16	O-QPSK

Fuente: El autor.

El estándar, en la frecuencia 2.4GHz especifica 16 canales, denominados del 11 al 26 y está separado cada 5MHz, desde 2405MHz hasta 2480MHz.

Figura 30. Canales del estándar IEEE 802.15.4



Fuente: <http://goo.gl/yq5Gtl>



El estándar IEEE 802.15.4 utiliza la técnica DSSS (Direct Sequence Spread Spectrum) para transmitir la información a través del medio, y es robusta ante las distintas recepciones por multi-trayecto. El alcance depende de la potencia de emisión del dispositivo, el estándar especifica que cada dispositivo debe ser capaz de transmitir al menos a 1mW (mili Vatio), así como el tipo de antenas utilizadas (cerámicas, dipolos).

#### *Capa MAC.*

Respecto a la capa de nivel de enlace (MAC) definida en el estándar IEEE 802.15.4, su finalidad es controlar y gestionar el acceso al medio, para lo que utiliza un protocolo CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance); así como sincronizar tramas y controlar el flujo. También es el encargado de reducir al máximo el gasto energético del dispositivo, para alargar la vida útil de la batería.

El ahorro energético en Zigbee se realiza gracias a que los dispositivos entran en un periodo de inactividad (sleep) después del tiempo de transmisión o recepción. Este estado de sleep puede afectar a todos los dispositivos de la red Zigbee o sólo a los dispositivos finales, en función del modo en el que trabajan.

#### *2.2.3.2.1.2 Capas NWK y APL: Zigbee.*

La alianza Zigbee define la capa de red (NWK) y la capa de aplicación (APL) de la pila empleada por Zigbee.

## Capa NWK.

La capa de nivel de red (NWK) se encarga de la topología de la red, añadiendo o eliminando dispositivos de la misma; asignando direcciones de red a los dispositivos; y redireccionando las tramas de información hacia el destinatario por el camino más adecuado.

También se encarga de garantizar la fiabilidad y calidad de los datos recibidos en el nodo, mediante el control y la corrección de los errores. Éstos pueden ser provocados por una mala comunicación de radio, la congestión de la red, colisión entre paquetes transmitidos, o fallos propios del nodo.

## Capa APL.

La capa del nivel de aplicación (APL) es la responsable de ejecutar las aplicaciones del sistema, y se subdivide en tres partes:

- *La subcapa ZDO (Zigbee Device Object)*, que define el papel que desempeña el dispositivo en la red (Coordinador, Router o Dispositivo Final);
- *Los objetos de aplicación*, definidos por cada fabricante, que se emplean para definir diferentes perfiles Zigbee.
- *La subcapa de soporte (APS)*, que es una interfaz entre la capa del nivel de red (NWK) y la capa del nivel de aplicación (APL), mediante el uso de un conjunto de servicios que se utilizan colectivamente con la subcapa ZDO y los objetos de aplicación.

#### **2.2.3.2.2 Características de 802.15.4.**

- Doble capa física (2.4 GHz y 868/915 MHz)
- Velocidad de datos 250 Kbps (2.4 GHz), 40 Kbps (915 MHz) y 20 Kbps (868 MHz)
- Optimización de la energía
- El método de acceso al canal es CSMA/CA
- Bajo uso de la batería, ésta puede durar varios meses o años
- Múltiples topologías (Malla, punto a punto, estrella)
- Se puede crear una red con 65535 dispositivos conectados entre sí
- Tiene un rango de 10 a 75 m. Dependiendo del ambiente de trabajo y el equipo que se utilice. Existen dispositivos que pueden llegar hasta 1 Km de distancia.

#### **2.2.3.2.3 Topología de red en 802.15.4.**

Según el grado de funcionalidad del dispositivo, el estándar define:

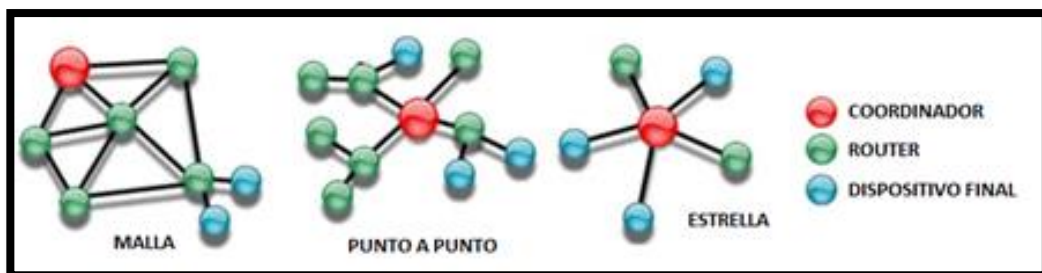
- *Los FFD (Full-Function Device)*, que son dispositivos que implementan un modelo de comunicaciones completo, que le permite comunicarse con cualquier otro dispositivo de la Red.
- *Los RFD (Reduced-Function Device)*, que son dispositivos con unas prestaciones limitadas para la comunicación, que sólo le permiten comunicarse con un FFD.

Las redes que se pueden formar albergan tres tipos de nodos:

- *Coordinador.* Es un dispositivo FFD que se encarga de crear la red de nodos y de gestionar su comunicación. En cada red existe un único coordinador.
- *Router.* Es un dispositivo FFD que se encarga de encaminar la información entre nodos que están muy separados en la red.
- *Dispositivo final.* Es un dispositivo RFD que puede transmitir o recibir información, pero, que no puede realizar labores de enrutamiento. Necesariamente debe estar comunicado con un Coordinador o con un Router.

Cada red Zigbee tiene un identificador de red único, lo que permite que varias redes Zigbee puedan compartir un mismo canal de comunicaciones, y pueden configurarse según distintas topologías: estrella, árbol o malla. En todas hay un Coordinador y uno o varios dispositivos Routers y/o finales.

Figura 31. Topología de red del estándar 802.15.4.



Fuente: [http://www.taskblog.com.br/wp-content/uploads/2013/04/ZigBee\\_interna.jpg](http://www.taskblog.com.br/wp-content/uploads/2013/04/ZigBee_interna.jpg)

Estas topologías permiten la comunicación entre dos dispositivos que se encuentran fuera de su rango de transmisión, siempre y cuando se encuentren conectados por nodos intermedios. La topología más interesante es la topología de malla. Ésta permite que si, en un momento dado, un nodo del camino falle y no sea capaz de transmitir o recibir información, pueda seguir la comunicación entre todos los demás nodos debido a que se rehacen todos los caminos. La administración de los caminos es tarea del coordinador.

#### ***2.2.3.2.4 Ventajas y desventajas de Zigbee.***

Zigbee al igual que todas las tecnologías de comunicación tiene su lado positivo y sus inconvenientes:

##### *Ventajas.*

- Ideal para conexiones punto a punto y punto a multipunto
- Diseñado para el direccionamiento de información y el refrescamiento de la red.
- Opera en la banda libre de ISM 2.4 GHz para conexiones inalámbricas.
- Óptimo para redes de baja tasa de transferencia de datos.
- Reduce tiempos de espera en el envío y recepción de paquetes.
- Detección de Energía (ED).
- Bajo ciclo de trabajo - Proporciona larga duración de la batería.
- Soporte para múltiples topologías de red: Estática, dinámica, estrella y malla.

- Hasta 65.535 nodos en una red.
- 128-bit AES de cifrado - Provee conexiones seguras entre dispositivos.
- Son más baratos y de construcción más sencilla.
- Zigbee tiene un bajo nivel de radiación y, por tanto, se puede utilizar en el sector médico.
- Rango de 10 m a 75m.

#### *Desventajas.*

- La tasa de transferencia es muy baja.
- Solo manipula textos pequeños comparados con otras tecnologías.
- Zigbee trabaja de manera que no puede ser compatible con Bluetooth en todos sus aspectos porque no llegan a tener las mismas tasas de transferencia, ni la misma capacidad de soporte para nodos.
- Tiene menor cobertura porque pertenece a redes inalámbricas de tipo WPAN.

#### **2.2.3.3 IEEE 802.11 (WI-FI).**

(Jara Werchau & Nazar, 2009) “La especificación IEEE 802.11 (ISO/IEC 8802-11) es un estándar internacional que define las características de una red de área local inalámbrica (WLAN). Wi-Fi (que significa "Fidelidad inalámbrica") es el nombre de la certificación otorgada por la Wi-Fi Alliance, anteriormente WECA (Wireless Ethernet Compatibility Alliance), grupo que garantiza la compatibilidad entre dispositivos que utilizan el estándar 802.11.

Con Wi-Fi se pueden crear redes de área local inalámbricas de alta velocidad, siempre y cuando el equipo que se vaya a conectar no esté muy alejado del punto de acceso.

En la práctica, Wi-Fi admite ordenadores portátiles, equipos de escritorio, asistentes digitales personales (PDA), o cualquier otro tipo de dispositivo de alta velocidad con propiedades de conexión también de alta velocidad (11 Mbps o superior) dentro de un radio mayor a 10 metros en ambientes cerrados (de 20 a 50 metros en general) o dentro de un radio mayor de 100 metros al aire libre.

#### ***2.2.3.3.1 Pila de protocolos del estándar 802.11.***

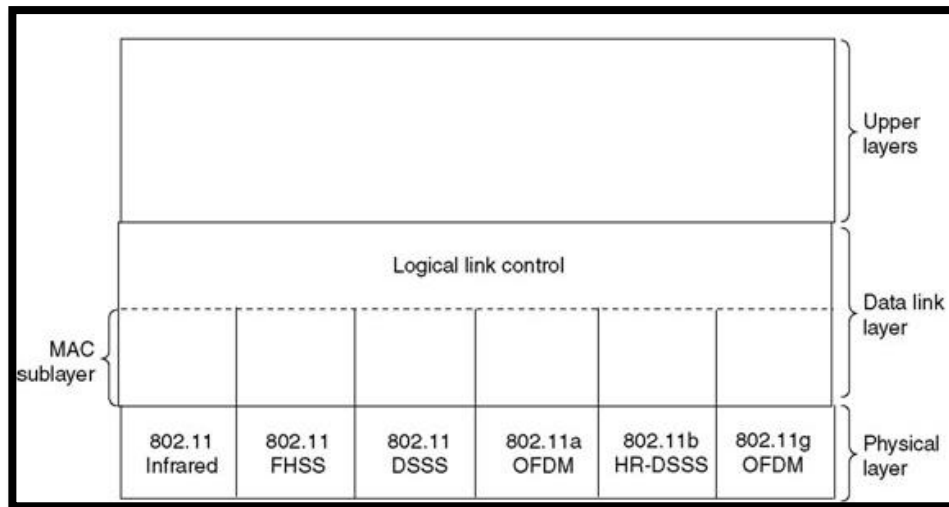
(Kioskea, 2014) El estándar 802.11 establece los niveles inferiores del modelo OSI para las conexiones inalámbricas que utilizan ondas electromagnéticas, por ejemplo:

- La capa física (a veces abreviada capa "PHY") ofrece tres tipos de codificación de información.
- La capa de enlace de datos compuesta por dos subcapas: control de enlace lógico (LLC) y control de acceso al medio (MAC).

La capa física define la modulación de las ondas de radio y las características de señalización para la transmisión de datos mientras que la capa de enlace de datos define la interfaz entre el bus del equipo y la capa física, en

particular un método de acceso parecido al utilizado en el estándar Ethernet, y las reglas para la comunicación entre las estaciones de la red. En realidad, el estándar 802.11 tiene tres capas físicas que establecen modos de transmisión alternativos:

Figura 32. Pila de protocolos de Wi-Fi.



Fuente: <http://player.slideplayer.es/7/1701168/data/images/img18.jpg>

Cualquier protocolo de nivel superior puede utilizarse en una red inalámbrica Wi-Fi de la misma manera que puede utilizarse en una red Ethernet.

### 2.2.3.3.2 Los distintos estándares Wi-Fi.

El estándar 802.11 en realidad es el primer estándar y permite un ancho de banda de 1 a 2 Mbps. El estándar original se ha modificado para optimizar el ancho de banda (incluidos los estándares 802.11a, 802.11b y 802.11g, denominados estándares físicos 802.11), o para especificar componentes de



mejor manera con el fin de garantizar mayor seguridad o compatibilidad. La tabla a continuación muestra las distintas modificaciones del estándar 802.11 y sus significados:

Tabla 3. Estándares de 802.11

Nombre del estándar	Nombre	Descripción
<b>802.11a</b>	Wifi5	El estándar 802.11 (llamado WiFi 5) admite un ancho de banda superior (el rendimiento total máximo es de 54 Mbps aunque en la práctica es de 30 Mbps). El estándar 802.11a provee ocho canales de radio en la banda de frecuencia de 5 GHz.
<b>802.11b</b>	Wifi	El estándar 802.11 es el más utilizado actualmente. Ofrece un rendimiento total máximo de 11 Mbps (6 Mbps en la práctica) y tiene un alcance de hasta 300 metros en un espacio abierto. Utiliza el rango de frecuencia de 2,4 GHz con tres canales de radio disponibles.
<b>802.11c</b>	Combinación del 802.11 y el 802.1d	El estándar combinado 802.11c no ofrece ningún interés para el público general. Es solamente una versión modificada del estándar 802.1d que permite combinar el 802.1d con dispositivos compatibles 802.11 (en el nivel de enlace de datos).
<b>802.11d</b>	Internacionalización	El estándar 802.11d es un complemento del estándar 802.11 que está pensado para permitir el uso internacional de las redes 802.11 locales. Permite que distintos dispositivos intercambien información en rangos de frecuencia según lo que se permite en el país de origen del dispositivo.
<b>802.11e</b>	Mejora de la calidad del servicio	El estándar 802.11e está destinado a mejorar la calidad del servicio en el nivel de la capa de enlace de datos. El objetivo del estándar es definir los requisitos de diferentes paquetes en cuanto al ancho de banda y al retardo de transmisión para permitir mejores transmisiones de audio y vídeo.

<b>802.11f</b>	Itinerancia	El 802.11f es una recomendación para proveedores de puntos de acceso que permite que los productos sean más compatibles. Utiliza el protocolo IAPP que le permite a un usuario itinerante cambiarse claramente de un punto de acceso a otro mientras está en movimiento sin importar qué marcas de puntos de acceso se usan en la infraestructura de la red. También se conoce a esta propiedad simplemente como itinerancia.
<b>802.11g</b>		El estándar 802.11g ofrece un ancho de banda elevado (con un rendimiento total máximo de 54 Mbps pero de 30 Mbps en la práctica) en el rango de frecuencia de 2,4 GHz. El estándar 802.11g es compatible con el estándar anterior, el 802.11b, lo que significa que los dispositivos que admiten el estándar 802.11g también pueden funcionar con el 802.11b.
<b>802.11h</b>		El estándar 802.11h tiene por objeto unir el estándar 802.11 con el estándar europeo (HiperLAN 2, de ahí la h de 802.11h) y cumplir con las regulaciones europeas relacionadas con el uso de las frecuencias y el rendimiento energético.
<b>802.11i</b>		El estándar 802.11i está destinado a mejorar la seguridad en la transferencia de datos (al administrar y distribuir claves, y al implementar el cifrado y la autenticación). Este estándar se basa en el AES (estándar de cifrado avanzado) y puede cifrar transmisiones que se ejecutan en las tecnologías 802.11a, 802.11b y 802.11g.
<b>802.11r</b>		El estándar 802.11r se elaboró para que pueda usar señales infrarrojas. Este estándar se ha vuelto tecnológicamente obsoleto.
<b>802.11j</b>		El estándar 802.11j es para la regulación japonesa lo que el 802.11h es para la regulación europea.

Fuente: <http://es.kioskea.net/contents/789-introduccion-a-wi-fi-802-11-o-wifi>

### 2.2.3.3.3 Características de 802.11.

Los estándares 802.11a, 802.11b y 802.11g, llamados "estándares físicos", son modificaciones del estándar 802.11 y operan de modos diferentes, lo que les permite alcanzar distintas velocidades en la transferencia de datos según sus rangos.

Tabla 4. Bandas de frecuencia de Wi-Fi.

Estándar	Frecuencia	Velocidad	Rango
802.11a	5 GHz	54 Mbps	10 m
802.11b	2,4 GHz	11 Mbps	100 m
802.11g	2,4 GHz	54 Mbps	100 m

Fuente: El autor.

El estándar define dos técnicas de transmisión:

- La primera de señal de espectro expandido por secuencia directa (DSSS)
- La segunda con espectro expandido por salto de frecuencia (FHSS), el cual definió el funcionamiento y la interoperabilidad entre redes inalámbricas.

El método de acceso al medio es mediante monitoreo, pero, sin detección de colisiones, CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).

Las modulaciones que se utilizan son DBPSK (Differential Binary Phase Shift Keying) y DQPSK (Differential Quadrature Phase Shift Keying) para velocidades de transmisión de 1 y 2 Mbps respectivamente.

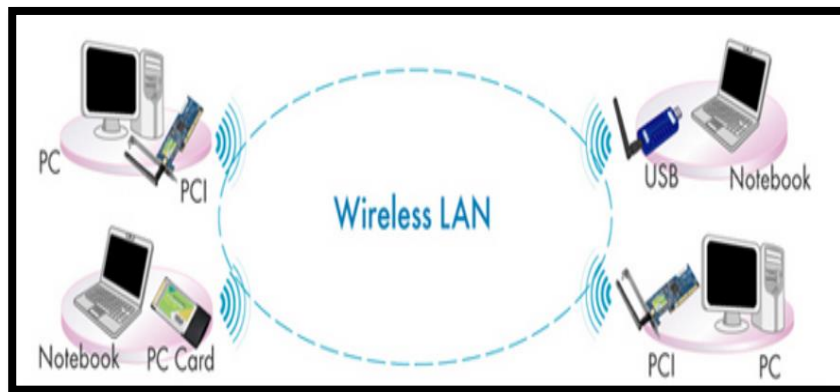
Solamente permite tres canales no interferentes espaciados 25MHz en la banda de 2.4GHz, a pesar de que en el estándar se definen 14 canales de operación en esa banda. En los productos comerciales actuales, la potencia nominal de transmisión es 100mW.

#### ***2.2.3.3.4 Topología de red en 802.11.***

(Jara Werchau & Nazar, 2009) El estándar IEEE 802.11 define el concepto de Conjunto Básico de Servicio (BSS, Basic Service Set) que consiste en dos o más nodos inalámbricos o estaciones que se reconocen una a la otra y pueden transmitir información entre ellos. Un BSS puede intercambiar información de dos modos diferentes:

1. Cada nodo se comunica con el otro en forma directa y sin ninguna coordinación. También llamado Ad-Hoc o IBSS (Independent Basic Service Set). Este modo solo permite la transmisión entre los nodos inalámbricos y no resuelve el problema de extender una LAN cableada.

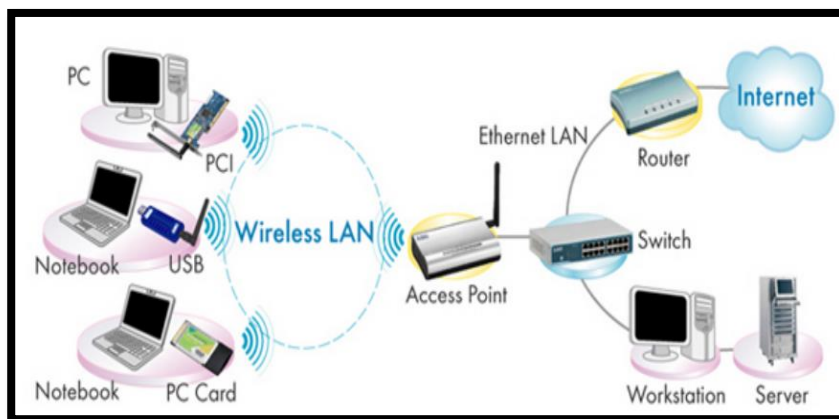
Figura 33. Topología de red Ad-Hoc o IBSS.



Fuente: [http://www.edutecne.utn.edu.ar/monografias/standard\\_802\\_11.pdf](http://www.edutecne.utn.edu.ar/monografias/standard_802_11.pdf)

2. Existe un elemento llamado comúnmente AP (Punto de Access) que coordina la transmisión entre los nodos inalámbricos. Este modo es llamado modo Infraestructura, y permite vincular la red inalámbrica con la red cableada ya que el AP actúa como bridge entre las dos redes. La existencia de varios AP conectados a un DS (Sistema de Distribución) que puede ser una LAN cableada es lo que se denomina EBSS (Extended Basic Service Set). La tecnología 802.11 permite el roaming entre los distintos AP.

Figura 34. Topología de red Infraestructura o EBSS.



Fuente: [http://www.edutecne.utn.edu.ar/monografias/standard\\_802\\_11.pdf](http://www.edutecne.utn.edu.ar/monografias/standard_802_11.pdf)

#### **2.2.3.3.5 Ventajas y desventajas de Wi-Fi.**

Wi-Fi al igual que todas las tecnologías de comunicación tiene su lado positivo y sus inconvenientes

##### *Ventajas.*

- Conectividad inalámbrica, lo que permite eliminar el cableado y dar movilidad a un equipo sin perder la comunicación en un rango determinado.
- Opera en la banda libre de ISM 2.4 GHz para conexiones inalámbricas.
- Compatibilidad con cualquier equipo de la alianza Wi-Fi.
- Permite ampliar la red por lo que su topología de red puede ser variada.
- Tiene diferentes tipos de seguridad como: AES, WEP, WAP2. etc.
- Tiene mayor cobertura por que pertenece a las redes WLAN.

### *Desventajas.*

- Falla en la conexión
- Consumo de energía elevado
- No es compatible con otras tecnologías.
- Costo elevado de los equipos.

#### **2.2.3.4 IEEE 802.15.1 (BLUETOOTH).**

(kioskea, 2014) Bluetooth es una tecnología de red de área personal inalámbrica (WPAN), que se define como un estándar global de comunicación inalámbrica, conocido también como estándar IEEE 802.15.1, la cual se utiliza para conectar dispositivos entre sí sin una conexión por cable. A diferencia de la tecnología IrDa<sup>6</sup>, los dispositivos Bluetooth no necesitan una línea de visualización directa para comunicarse. Esto hace que su uso sea más flexible y permita la comunicación entre habitaciones en espacios pequeños, lo que le hace una tecnología de corto alcance.

El objetivo de Bluetooth es transmitir voz o datos entre equipos con circuitos de radio de bajo costo, a través de un rango aproximado de entre diez y cien metros, utilizando poca energía.

La tecnología Bluetooth comprende hardware, software y requerimientos de interoperabilidad, por lo que su desarrollo ha necesitado la participación de los

---

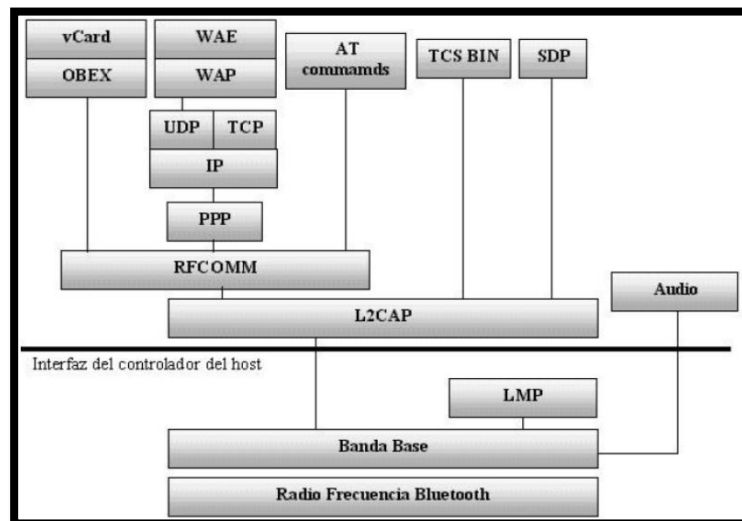
<sup>6</sup> IrDa: Define un estándar físico en la forma de transmisión y recepción de datos por rayos infrarrojos. Esta tecnología está basada en rayos luminosos que se mueven en el espectro infrarrojo

principales fabricantes de los sectores de las telecomunicaciones y la informática, además, entre sus principales características, pueden nombrarse su robustez, baja complejidad, consumo de energía y costo.

#### **2.2.3.4.1 Pila de protocolos del estándar 802.15.1.**

La pila de protocolos se puede dividir en cuatro capas lógicas: Protocolos del núcleo (Banda base, LMP, L2CAP y SDP), reemplazo de cables (RFCOMM), control de telefonía (TCS) y protocolos adaptados (PPP, UDP/TCP/IP, OBEX). A continuación se muestran los componentes básicos:

Figura 35. Pila de protocolos de Bluetooth.



Fuente: [http://www.lgama.com.mx/images/publicaciones/sitioi09\\_id80\\_bluetooth.pdf](http://www.lgama.com.mx/images/publicaciones/sitioi09_id80_bluetooth.pdf).



#### **2.2.3.4.2 Características de 802.15.1.**

Bluetooth puede transmitir velocidades de aproximadamente 1 Mbps, que corresponde a 1600 saltos por segundo en modo full dúplex, con un alcance de aproximadamente diez metros cuando se utiliza un transmisor clase II, y de un poco menos de cien metros cuando se utiliza un transmisor clase I. El estándar Bluetooth define 3 clases de transmisores, cuyo alcance varía en función de su potencia radiada:

Tabla 5. Potencias de transmisión de Bluetooth.

<b>Clase</b>	<b>Potencia (pérdida de señal)</b>	<b>Alcance</b>
<b>I</b>	100 mW (20 dBm)	100 metros
<b>II</b>	2,5 mW (4 dBm)	15-20 metros
<b>III</b>	1 mW (0 dBm)	10 metros

Fuente: <http://es.kioskea.net/contents/70-bluetooth>.

A diferencia de la tecnología IrDa, la principal competencia, que utiliza radiación de luz para enviar datos, Bluetooth utiliza ondas de radio (en la banda de frecuencia de 2.4 GHz) para comunicarse. Como consecuencia, los dispositivos Bluetooth no necesitan estar visualmente comunicados para intercambiar datos. Esto significa que los dos dispositivos pueden comunicarse incluso si se encuentran separados por un muro; y lo mejor de todo es que los dispositivos Bluetooth pueden detectarse entre sí sin la participación del usuario, siempre y cuando uno se encuentre dentro del alcance del otro.

#### **2.2.3.4.3 Normas Bluetooth.**

El estándar Bluetooth se divide en múltiples normas:

- IEEE 802.15.1 define Bluetooth 1.x, que puede alcanzar velocidades de 1 Mbps;
- IEEE 802.15.2 recomienda prácticas para utilizar la banda de frecuencia de 2.4 GHz. Sin embargo, este estándar todavía no se ha aprobado.
- IEEE 802.15.3 es un estándar que actualmente se está desarrollando, que ofrecerá velocidad de banda ancha (20 Mbps) con Bluetooth.
- IEEE 802.15.4 es un estándar que actualmente se está desarrollando para el uso con aplicaciones Bluetooth de baja velocidad.

#### **2.2.3.4.4 Topología de red en 802.15.1.**

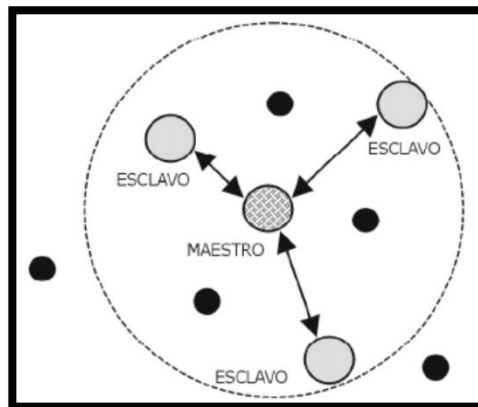
La estructura que maneja esta tecnología está compuesta, en su forma más básica, por lo que se denomina una Piconet y en una estructura un poco más compleja a la que se denomina una Scatternet.

##### **2.2.3.4.4.1 Piconet**

Dos o más dispositivos Bluetooth que comparten el mismo canal de conexión conforman una Piconet. Esta se establece a través de enlaces punto - multipunto, en donde uno de los dispositivos cumple el rol de maestro mientras los demás son esclavos. Una piconet puede tener un máximo de siete esclavos

activos. Si un equipo se encuentra dentro del radio de cobertura de otro, éstos pueden establecer conexión entre ellos.

Figura 36. Topología Piconet de una red Bluetooth.



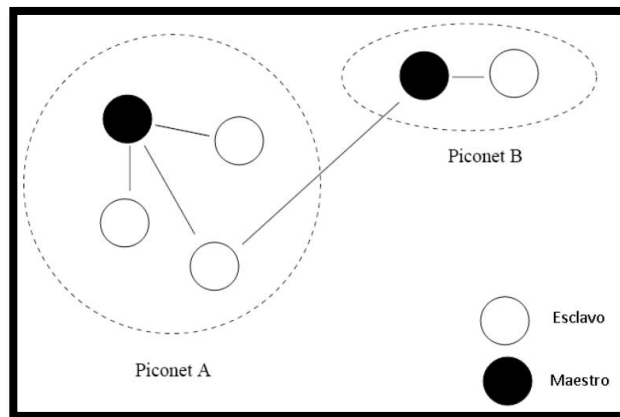
Fuente: [http://www.lgama.com.mx/images/publicaciones/sitioi09\\_id80\\_bluetooth.pdf](http://www.lgama.com.mx/images/publicaciones/sitioi09_id80_bluetooth.pdf)

Sin embargo, sólo aquellas unidades que realmente quieran intercambiar información comparten un mismo canal creando la piconet. Este hecho permite que se creen varias piconet en áreas de cobertura superpuestas.

#### 2.2.3.4.4.2 Scatternet

A un grupo de piconets se le llama scatternet. Se puede apreciar la formación de una scatternet en la que participan la piconet A y B, como se indica en la siguiente figura.

Figura 37. Topología Scatternet de una red Bluetooth



Fuente: [http://www.lgama.com.mx/images/publicaciones/sitioi09\\_id80\\_bluetooth.pdf](http://www.lgama.com.mx/images/publicaciones/sitioi09_id80_bluetooth.pdf)

Cuanto más piconets se añaden a la scatternet, el rendimiento del sistema de salto en frecuencia disminuye poco a poco, existiendo una reducción por término medio del 10%, por otra parte un dispositivo Bluetooth sólo puede actuar como maestro en una sola piconet, debido a que una piconet está determinada por la sincronización con el reloj Bluetooth del dispositivo maestro. En cambio, este dispositivo sí podrá hacer el papel de esclavo en diversas piconets.

#### **2.2.3.4.5 Ventajas y desventajas de Bluetooth**

Bluetooth al igual que todas las tecnologías de comunicación tiene su lado positivo y sus inconvenientes:

### *Ventajas.*

- Opera en la banda libre de ISM 2.4 GHz para conexiones inalámbricas.
- Bluetooth se usa para aplicaciones en teléfonos móviles y la informática casera.
- Permite la transmisión de datos, imágenes y videos entre dispositivos.

### *Desventajas.*

- Durabilidad de la batería muy corta.
- Velocidades de transmisión muy lentas y pérdida de datos en caso de desconectarse o desincronización.
- Tiene corto alcance por lo que no permite la movilización del usuario.
- Solo permite pocos dispositivos conectarse entre sí.

## **2.2.4 Selección de la tecnología inalámbrica adecuada.**

Como se mencionó anteriormente, se escogió una tecnología inalámbrica diferente a las que se utilizan hoy en día para las comunicaciones inalámbricas WPAN, en las cuales fueron analizadas sus principales características, mismas que son necesarias para este proyecto. Es importante destacar que una u otras tecnologías resulta un poco difícil de comparar debido a que se difiere en la utilización y servicios entregados. A continuación se comentan algunas observaciones, por lo tanto la comparativa realizada en la tabla siguiente sólo incluye campos medibles.

Tabla 6. Comparación de tecnologías inalámbricas.

TECNOLOGIA	Zigbee	Wi-Fi	Bluetooth
<b>Velocidad de transición</b>	250 Kbps	11 Mbps 54 Mbps	1 Mbps
<b>Frecuencia</b>	868 - 868,8 MHz 902 - 928 MHz 2.4 - 2.483 MHz 5 MHz	2.4 MHz 5 MHz	2.4 MHz
<b>Modulación</b>	BPSK - OQPSK	DBPSK - DQPSK	GFSK
<b>Técnicas de transmisión</b>	DSSS	DSSS y FHSS	FHSS
<b>Canales</b>	16	3	1
<b>Necesidad de línea de vista</b>	No	No	No
<b>Alcance máximo</b>	75 m	100m	10m
<b>Potencia</b>	1mW	100mW	100mW
<b>Duración de la batería</b>	Años	Energía continua	Meses
<b>Número de dispositivos</b>	65535	256	7
<b>Topología de red</b>	Varias	Varias	Varias
<b>Seguridad</b>	AES	AES, WEP, WAP, WAP2	-
<b>Tamaño</b>	Pequeño	Grande	Pequeño
<b>Costo</b>	40 \$	35 \$	25\$

Fuente: El autor.

Como se puede observar en la tabla anterior, las características de las tecnologías escogidas presentan pocas diferencias pero se decidió escoger la tecnología Zigbee por las siguientes razones:

- *Frecuencia:* Zigbee es una tecnología que permite trabajar a diferentes rangos de frecuencia, lo que es ideal para este proyecto debido a que en el lugar de trabajo existen equipos y personas que trabajan a diario con las otras tecnologías lo que puede generar problemas en un futuro.
- *Alcance:* Un equipo con tecnología Zigbee puede alcanzar una mayor cobertura de trabajo a la establecida en el estándar, ya que, hoy en día existen dispositivos capaces de llegar hasta 1Km de distancia en lugares abiertos y 100m en lugares cerrados, es por ello que, comparado con dispositivos con tecnología Bluetooth es sumamente ventajoso y si se habla de equipos Wi-Fi pueden ser semejantes en alcance, pero la diferencia está en que los equipos con tecnología Zigbee son manipulables para tener mayor alcance.
- *Duración de la batería:* Debido a que estos equipos utilizan aproximadamente una tensión relativamente baja es ideal para este proyecto porque es necesario que el sistema esté funcionando continuamente, y se puede realizar esto simplemente conectando baterías, sin la necesidad de estar conectado continuamente a la energía eléctrica. A diferencia de los equipos con tecnología Bluetooth, éstos pueden durar hasta años sin cargar las baterías, mientras que Bluetooth tiene una duración mucho más baja por la aplicación que éste representa y también

a diferencia de Wi-Fi, éste tiene que estar conectado continuamente a la energía, en caso de perder electricidad, se pierde la comunicación.

- *Número de dispositivos:* En Zigbee se pueden conectar un máximo de 65535 dispositivos y tener redundancia en la red sin la necesidad de configuración adicional, mientras que en Bluetooth sólo se puede conectar un máximo de 7 dispositivos. Con respecto a Wi-Fi, éste presenta una ventaja, ya que se puede ampliar más la red, pero en caso de fallar un equipo, la red pierde conectividad y para tener redundancia se debe realizar una mayor complejidad en la configuración de los equipos.
- *Tamaño:* El tamaño de los equipos con tecnología Zigbee es mucho más pequeño que el de los equipos con tecnología Wi-Fi, por lo que su implementación en un circuito es mucho más fácil de realizarlo. En lo que respecta a Bluetooth son casi semejantes, con un poco de diferencia en el tamaño, pero con pocas opciones de configuración física.



## **CAPÍTULO III**

### **DISEÑO DEL HARDWARE**

#### **3.1 Introducción**

En este capítulo se diseñan los diferentes circuitos que serán necesarios para integrar los dispositivos electrónicos analizados anteriormente tales como: el microcontrolador PIC18F4550, los módulos inalámbricos XBEE, etc. De igual forma, la topología de red en la que los módulos XBEE puedan trabajar para asegurar una convergencia fiable, y puedan transmitirse los datos sin ningún problema.

La programación del microcontrolador se realizará en base a los diagramas de flujo de las funciones a cumplir, tales como la transmisión y recepción de los datos; y finalmente para concluir con este capítulo se realizará físicamente los circuitos diseñados para sus posteriores pruebas.

#### **3.2 Diseño del sistema eléctrico para el control de acceso a las aulas**

La interfaz eléctrica para el control de acceso a las aulas de la Facultad de Ingeniería en Ciencias Aplicadas (FICA), se basa en el estudio de la estructura

lógica y física que ésta tendrá para el correcto funcionamiento; tanto en la parte de la comunicación inalámbrica, como en el consumo de energía de los diferentes elementos eléctricos que ésta contenga. En este punto, se tomará en cuenta cada uno de los parámetros de los elementos estudiados anteriormente, y se procederá a realizar los diferentes cálculos y diseños esquemáticos de cada parte del circuito, lo cual se detalla a continuación.

### **3.2.1 Descripción del sistema de control de acceso a las aulas**

Este sistema se desarrolla para facilitar el uso de las aulas de la facultad, mediante un control por parte de los docentes, manejado por una interfaz gráfica ubicada en la sala de profesores.

El sistema consta de tres partes fundamentales:

- La interfaz gráfica.
- El circuito transmisor.
- El circuito receptor.

✓ *Interfaz gráfica:* Es una aplicación que será utilizada por el docente para proceder a utilizar un aula determinada, esta aplicación se estudiará detalladamente en el capítulo siguiente.

✓ *Circuito transmisor:* Este permitirá realizar la comunicación entre la interfaz gráfica y el circuito receptor. En el circuito transmisor se tienen las siguientes funciones:

- Adquirir la información por medio de la interfaz gráfica diseñada.
- Procesar la información por medio del módulo inalámbrico XBEE.
- Transmitir y recibir la información procesada hacia y desde el circuito receptor respectivamente.

✓ *Circuito receptor:* Este permitirá realizar la comunicación entre el microcontrolador (implementado en este circuito) y el circuito transmisor. En el circuito receptor se tienen las siguientes funciones:

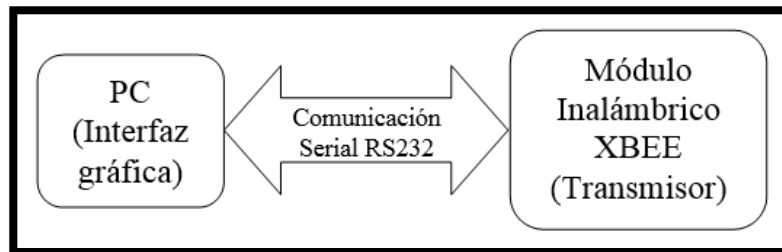
- Recibir la información enviada por parte del circuito transmisor
- Procesar la información en el microcontrolador.
- Realizar las funciones necesarias de apertura de puertas
- Transmitir los resultados obtenidos al circuito transmisor.

### **3.3 Diagrama de bloque**

Un diagrama de bloque es una representación gráfica general de un proceso o funcionamiento interno de un sistema. A continuación se muestra un

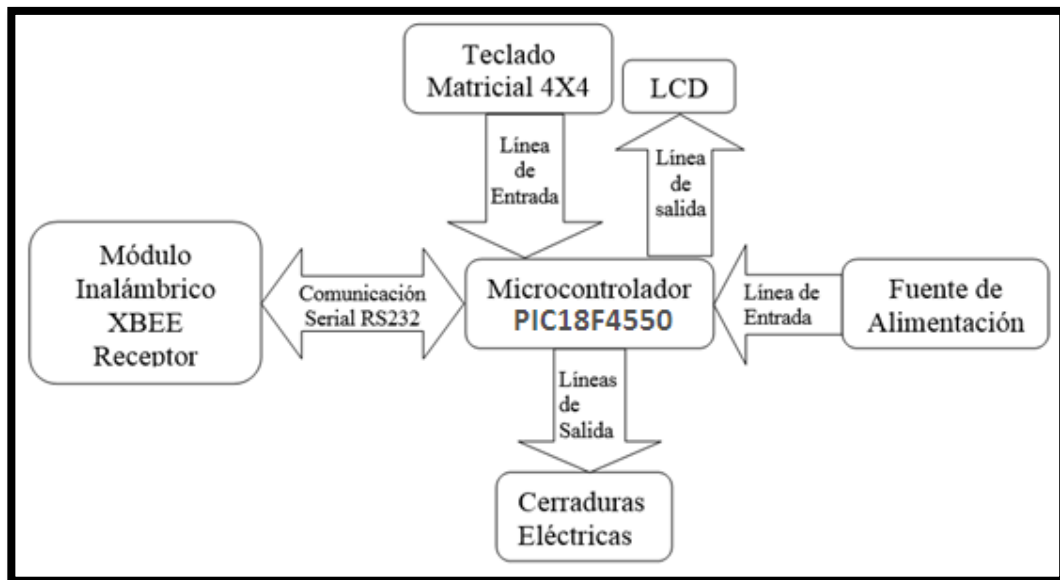
diagrama general de los componentes y función de cada uno de los circuitos que se emplean para este diseño.

Figura 38. Diagrama de bloque del circuito transmisor.



Fuente: El autor.

Figura 39. Diagrama de bloque del circuito receptor.

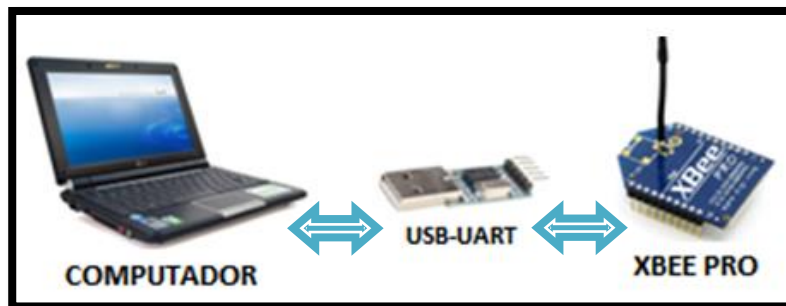


Fuente: El autor.

### 3.4 Diagrama de conexión física de los circuitos transmisor y receptor

Dado que el circuito transmisor se conecta directamente a una PC por medio de un puerto USB, se tomará el voltaje proporcionado por el mismo, el cual, es de 5VDC y se le conectará a un módulo USB-USART, este módulo, permitirá conectar el computador con el módulo XBEE Pro el mismo que recibirá y transmitirá los datos desde y hacia el circuito receptor.

Figura 40. Diagrama de conexión del circuito transmisor.



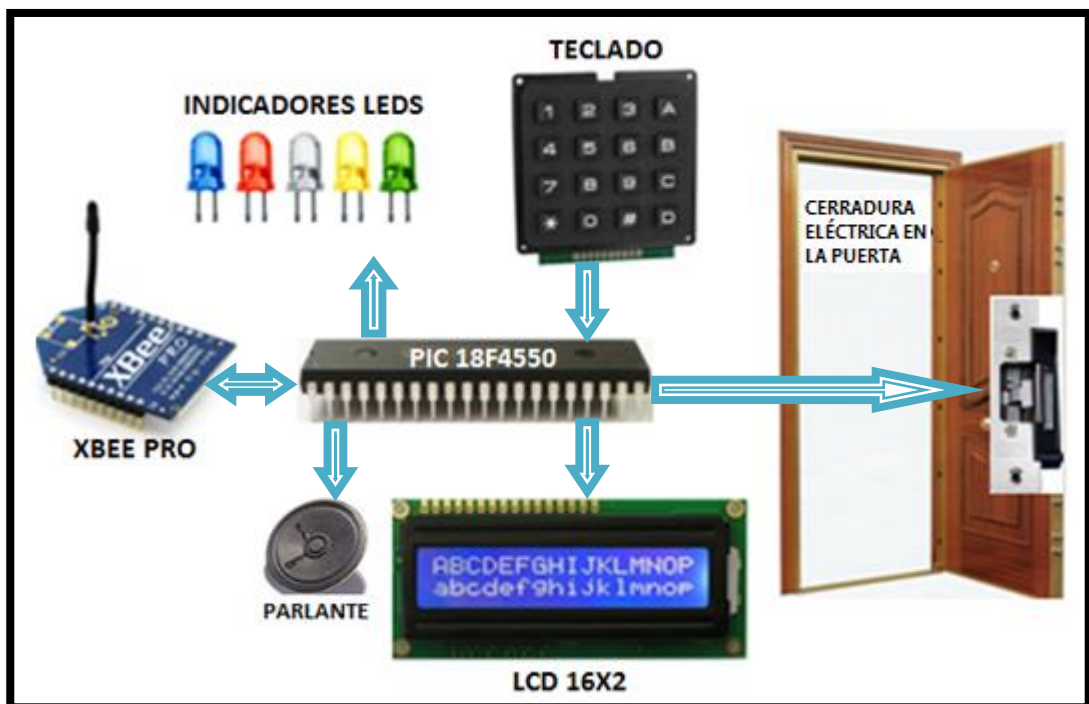
Fuente: El autor.

El circuito receptor es el que hace el mayor trabajo mecánico y eléctrico ya que este será el encargado de la apertura del aula a utilizar por el docente, visualizar el estado actual del aula, la autenticación de la misma, el envío y recepción de los datos.

En primera instancia el módulo XBEE Pro es el elemento más importante en este sistema, ya que los datos enviados y transmitidos serán la

responsabilidad de este elemento en el caso que se reciba los datos; estos serán enviados hacia el microcontrolador PIC18F4550, el cual por medio de la programación realizada, permitirá el ingreso de una clave por parte del docente, mediante la manipulación del teclado matricial 4X4, y, dependiendo de la clave la puerta será abierta o bloqueada. En caso de errores en la clave, estos parámetros serán visibilizados por medio de la LCD, para poder evidenciar el estado del aula, y en cual se encuentra, ya que existen varias en la facultad. Los indicadores leds permitirán evidenciar el estado del circuito, dependiendo de su color, se configurará para que estos puedan ayudar a evidenciar el estado de uno u otro elemento del circuito receptor, y, finalmente el parlante avisará el tiempo en el que el aula deba ser desocupada por el docente, por un tiempo determinado.

Figura 41. Diagrama de conexión del circuito receptor.



Fuente: El autor.

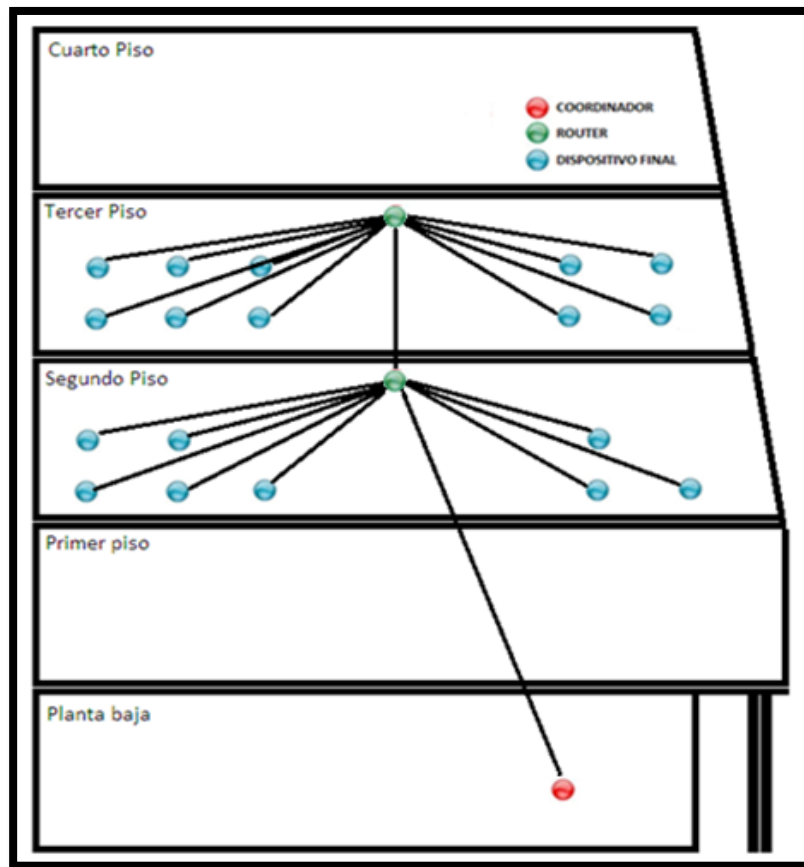
### **3.5 Topología de red**

Según el estándar Zigbee que permite realizar varias configuraciones en una topología de red, para el caso de este proyecto, se utiliza una topología tipo estrella por la situación geográfica de la facultad y debido a que los módulos utilizados con este tipo de tecnología no necesitan línea de vista, es decir, funcionan correctamente en lugares cerrados y tienen un alcance de aproximadamente treinta metros para poder comunicarse entre sí, el diseño de la red inalámbrica está realizado de la siguiente forma.

Cada puerta cuenta con un “dispositivo final”(módulo receptor), el cual se comunica con el “router” (módulo router) instalado en cada piso, para posteriormente enviar los datos transmitidos y recibidos hacia el “coordinador” (módulo transmisor) instalado en el computador, que cuenta con la aplicación del software diseñado para este proyecto.

Cabe recalcar que la posición física de los dispositivos “coordinador” y “routers” se ubican en un lugar estratégico para lograr la convergencia total de la red inalámbrica.

Figura 42. Topología de red utilizando dispositivos XBEE.



Fuente: El autor.

### 3.5.1 Ubicación de los diferentes dispositivos en la estructura física de las instalaciones de la facultad.

A continuación se verifica la ubicación de cada uno de los dispositivos inalámbricos conjuntamente con los planos de la estructura física de la facultad, donde se encuentra integrada la red inalámbrica tipo estrella.

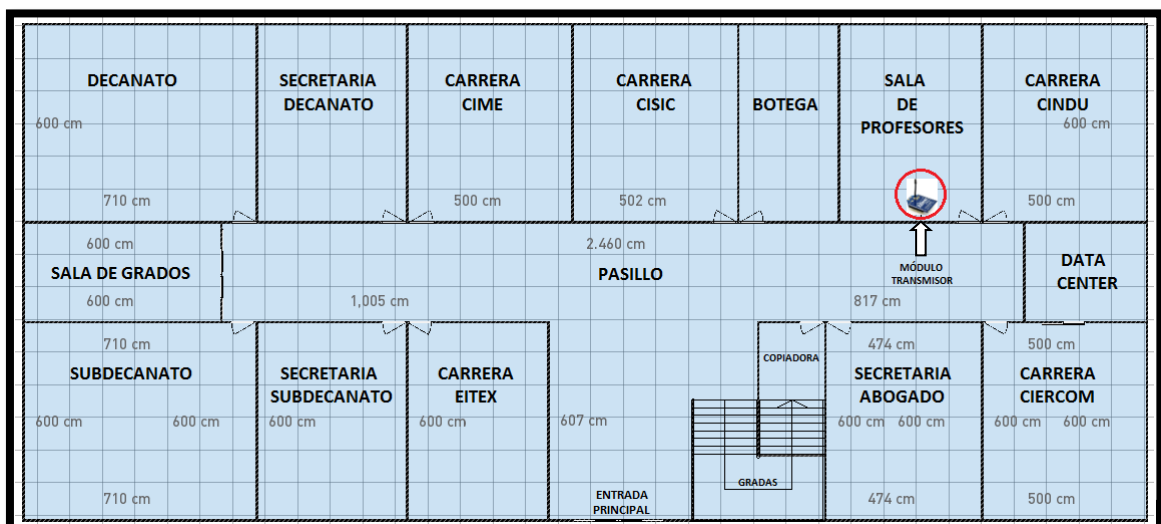


### 3.5.1.1 Plano de la planta baja

El módulo transmisor (dispositivo coordinador), es ubicado en la planta baja de la facultad, específicamente en la sala de profesores ya que es ahí donde los docentes registrarán su ingreso a las funciones laborales de cada uno y por ende utilizan el software de autenticación, ya que este dispositivo se conecta directamente con dicho software tal como se plantea para el diseño de este proyecto.

Cabe recalcar que la potencia de radiación de este dispositivo se aumentó al máximo en la configuración para alcanzar un mayor alcance. A continuación se puede apreciar lo dicho en la siguiente figura.

Figura 43. Planos de la planta baja de la facultad con el módulo transmisor.

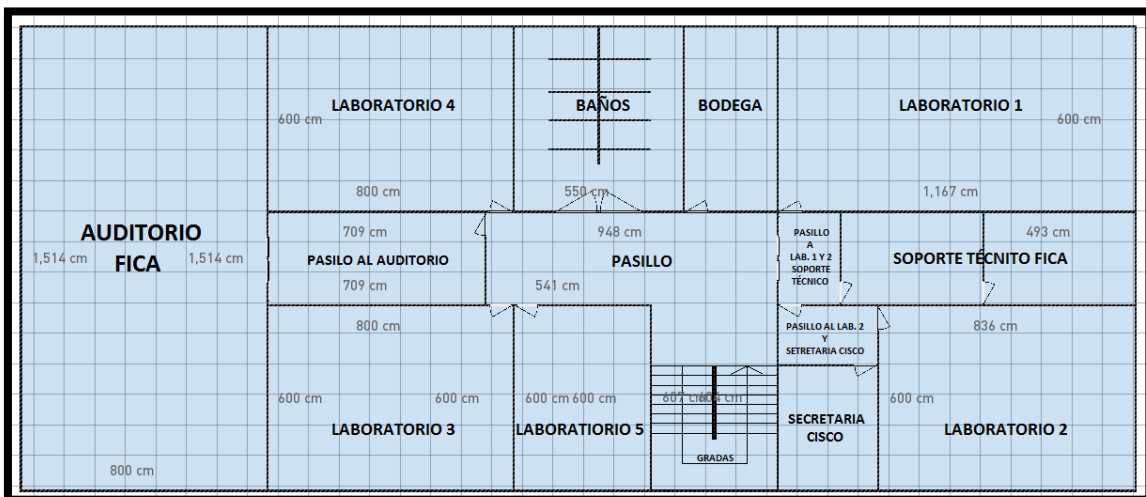


Fuente: El Autor.

### 3.5.1.2 Plano del primer piso

Dado que el primer piso de la facultad cuenta con laboratorios de informática, no necesitan de la ubicación de ningún dispositivo de red, por ende, este piso no es tomado en cuenta para la ubicación y distribución de ningún dispositivo electrónico, pero se considera la altura que se tiene para el análisis de la red inalámbrica. En la siguiente figura se puede apreciar las dimensiones físicas de este piso.

Figura 44. Planos del primer piso de la facultad.



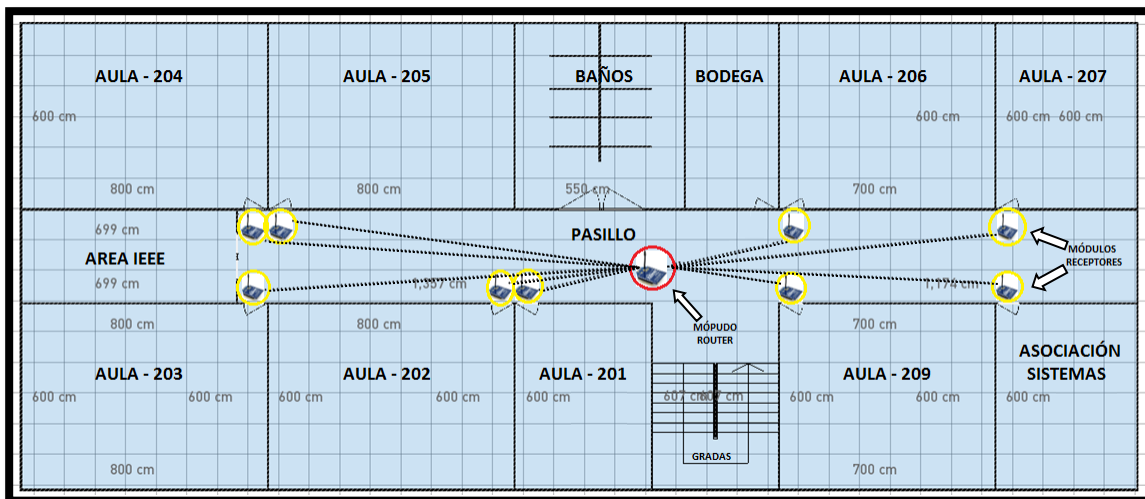
Fuente: El Autor.

### 3.5.1.3 Plano del segundo piso

En este piso, ya se cuenta con módulos receptores (dispositivo final) los mismos que se ubican en cada una de las puertas de las aulas y el primer módulo router (dispositivo router).

Los módulos receptores colocados en este piso se comunican únicamente con el módulo router ubicado en el techo de este piso y configurados con una potencia relativamente baja para evitar interferencia como el ruido provocado entre ellos, ya que se encuentran a una distancia muy corta. En la siguiente figura se puede apreciar las dimensiones físicas de este piso.

Figura 45. Planos del segundo piso de la facultad con los módulos receptores y router.

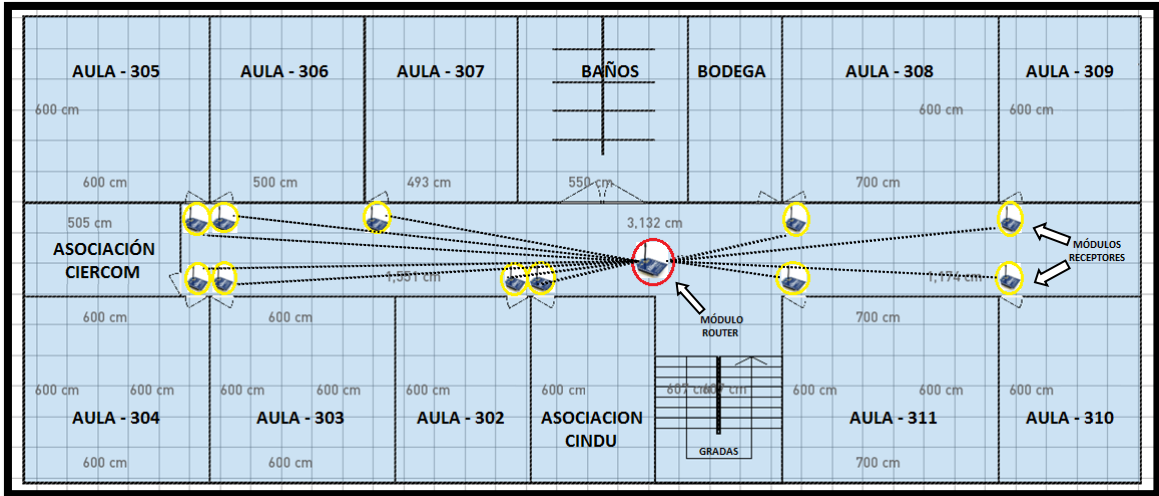


Fuente: El Autor.

### 3.5.1.4 Plano del tercer piso

Al igual que en el piso anterior, este también tendrá el mismo análisis, con la diferencia de la incorporación de dos módulos receptores adicionales. Tal como se aprecia en la siguiente figura.

Figura 46. Planos del tercer piso de la facultad con los módulos receptores y router.



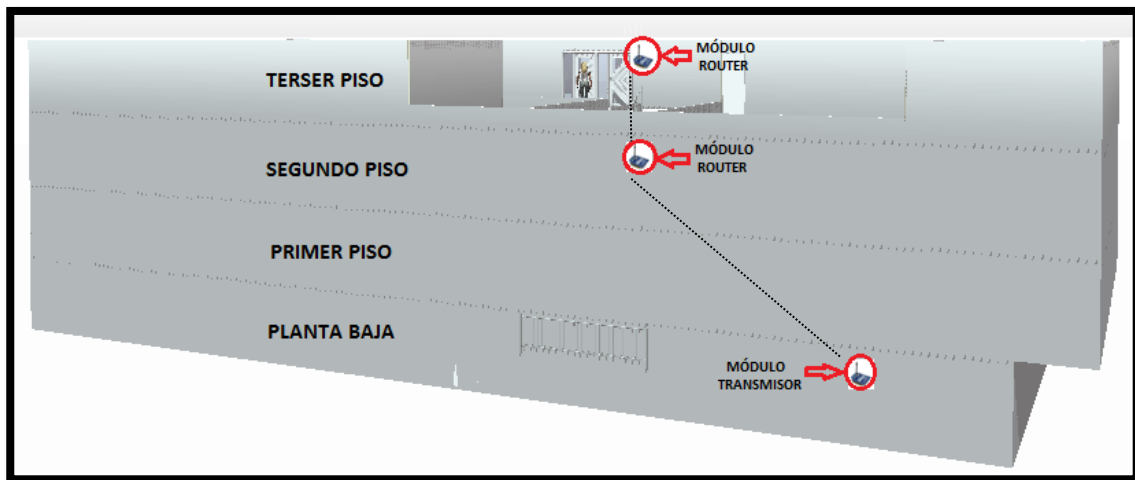
Fuente: El Autor.

### 3.5.1.5 Vista frontal de la facultad

Hasta el momento se ha hablado de la ubicación de los módulos receptores, los cuales se conectan con un solo módulo router ubicado en cada uno de los pisos de la facultad donde se encuentran las aulas, sin embargo, la comunicación entre los módulo router de cada piso se realiza verticalmente entre sí y con el módulo transmisor, el mismo que está ubicado en la planta baja.

A continuación se puede apreciar lo mencionado hasta el momento en la siguiente figura.

Figura 47. Comunicación entre módulos routers y transmisor.



Fuente: El Autor.

### 3.6 Alimentación de energía para la interfaz electrónica

Al realizar un diseño de cualquier tipo de circuitos se debe empezar con la fuente de alimentación con la cual este entrará en funcionamiento, para ello, se debe tomar en cuenta el consumo de energía desde la alimentación en el PIC18F4550 como en los módulos inalámbricos XBEE y cada uno de los diferentes dispositivos que se encuentren conectados al microcontrolador.

Estos datos se toman del datasheet (hojas de datos) que cada fabricante proporciona para una correcta manipulación de los elementos, este documento especifica todas las características que el dispositivo presenta, desde el voltaje de alimentación hasta las curvas características del dispositivo, si fuera este el caso.

En el anexo A del presente trabajo se adjunta los datasheets de los dispositivos electrónicos más importantes de este proyecto ya que según las especificaciones de estos se toman los requerimientos para el diseño de la fuente de alimentación.

### **3.6.1 Requisitos de alimentación del circuito transmisor**

Para este circuito no es necesario realizar una fuente de alimentación específica ya que como se conecta este circuito directamente a un computador se tomará las características eléctricas que éste proporcione para alimentar este circuito. Los parámetros eléctricos del circuito transmisor son:

- Voltaje de entrada: 5 VDC
- Voltaje de salida:
  - Alimentación del módulo inalámbrico: 3.3 VDC
- Corriente de salida: ~100mA

### **3.6.2 Requisitos de alimentación del circuito receptor**

Para este circuito es fundamental realizar la fuente de alimentación ya que este será independiente de otro elemento como lo fue en el circuito transmisor, para realizar esta fuente de alimentación son necesarios los siguientes parámetros eléctricos (datos tomados de los datasheets de cada dispositivo mencionado a continuación).

- Voltaje de entrada:
  - Alimentación del transformador: 110VAC.
- Voltaje de salida:
  - Alimentación del microcontrolador 5 VDC
  - Alimentación del módulo inalámbrico: 3.3 VDC
  - Alimentación máxima de los dispositivos eléctricos: 5VDC.
  - Alimentación de la cerradura eléctrica: 12VDC.
- Corriente de salida total: ~500mA

### **3.6.3 Desarrollo de la fuente de alimentación del circuito receptor**

En la entrada del circuito se contará con una alimentación de 110 VAC, la cual es proporcionada por las líneas de alimentación de la facultad, para que el circuito receptor funcione es necesario realizar una conversión de tensión, es decir, cambiar la tensión alterna proporcionada por la facultad a una tensión continúa.

En una fuente de alimentación se cuenta con cuatro bloques básicos:

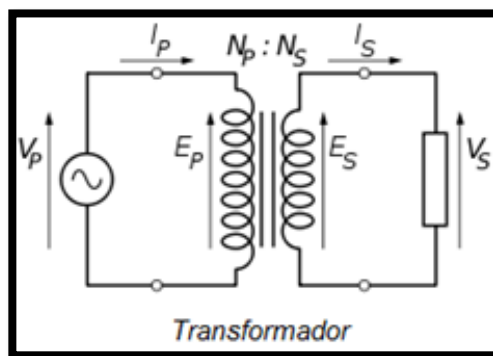
- Un transformador de red
- Un convertidor de alterna a continua o rectificador (AC/DC)
- Un convertidor de continua a continua o filtro (CD/CD)
- Un estabilizador o regulador de voltaje.

### 3.6.3.1 Transformador de red

El transformador permitirá reducir o aumentar la tensión que ingresa por medio de la red eléctrica altera, manteniendo la frecuencia de oscilación y proporcionando aislamiento galvánico.

(Toapanta & Jaramillo, 2009) “Al conectar el bobinado primario a la red eléctrica alterna ( $V_p$ ), circula una corriente alterna ( $I_p$ ) la cual inducirá un flujo magnético en el núcleo de hierro, este flujo alterno inducirá una fuerza electromotriz o tensión (fem) denominado ( $V_s$ ) en el bobinado secundario por la cual circulará una corriente alterna ( $I_s$ ).” Tal como se puede apreciar en la siguiente figura.

Figura 48. Transformador eléctrico en una fuente de alimentación.



Fuente: [https://es.wikipedia.org/wiki/Transformador#/media/File:Transformer\\_under\\_load.svg](https://es.wikipedia.org/wiki/Transformador#/media/File:Transformer_under_load.svg)



La relación entre la tensión inductora ( $E_p$ ) y la tensión inducida ( $E_s$ ), es directamente proporcional al número de espiras de los bobinados primario ( $N_p$ ) y secundario ( $N_s$ ), según la ecuación:

$$\frac{E_p}{E_s} = \frac{N_p}{N_s}$$

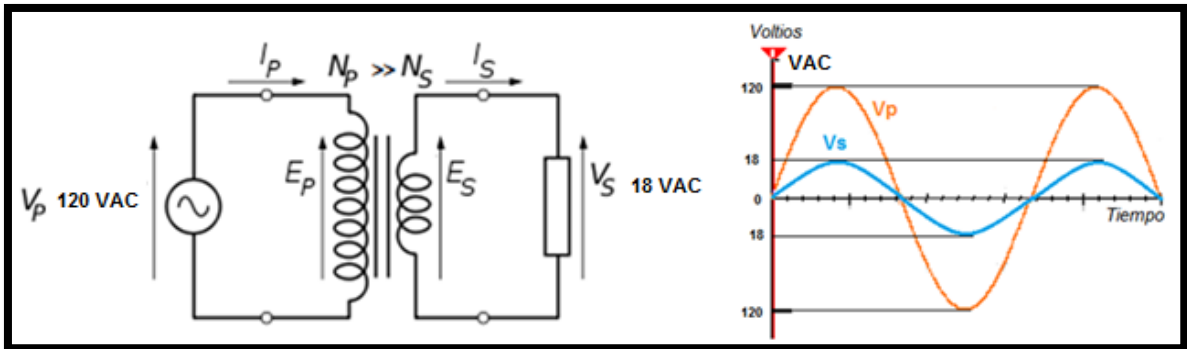
Por lo tanto, **La relación de transformación (m)** de la tensión entre el bobinado primario y el secundario depende de los números de vueltas que tenga cada uno.

$$\frac{N_p}{N_s} = \frac{V_p}{V_s} = \frac{I_s}{I_p} = m$$

Si el número de vueltas del secundario es el triple del primario, en el secundario habrá el triple de tensión (transformador elevador) o si el número de vueltas del primario es el triple del secundario, en el secundario habrá un tercio de tensión (transformador reductor).

Dado que en esta fuente de alimentación es necesario reducir la tensión de entrada, se utilizará un transformador reductor que será capaz de reducir dicha tensión a 18 voltios de corriente alterna (VAC) el mismo que se puede adquirir en el mercado.

Figura 49. Transformador reductor y curva característica.



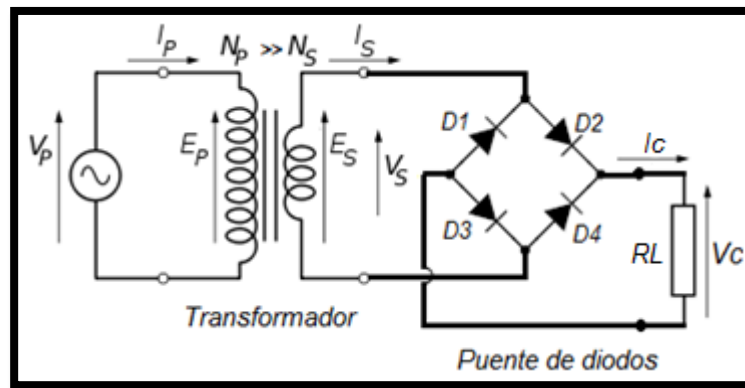
Fuente: El autor.

### 3.6.3.2 Convertidor de alterna a continua o rectificador (AC/DC)

El rectificador convierte la tensión alterna en una forma de onda continua pulsante, la cual está compuesta de una señal continua y un rizado, este puede ser compuesto por diodos.

Existen diversas formas de rectificar la tensión alterna, sin embargo para la realización de esta fuente de alimentación se procederá a diseñar mediante el rectificador de onda completa tipo puente de diodos.

Figura 50. Rectificador de onda completa tipo puente de diodos en una fuente de alimentación.



Fuente: El Autor.

Para calcular el valor del voltaje a la salida del puente de diodos y la corriente con su respectiva potencia se realizan los siguientes cálculos permitiendo obtener los valores a los cuales la fuente de alimentación proporcionará para alimentar el circuito respectivo.

$$V_{MAX} = (\sqrt{2})(V_{RMS})$$

$$V_{DC} = \frac{(2)(V_{MAX})}{\pi}$$

$$I_{DC} = \frac{V_{DC}}{R_L}$$

$$P_{DC} = (V_{DC})(I_{DC})$$

Donde:

- $V_{MAX}$ = Voltaje máximo.

- $V_{RMS}$ = Voltaje eficaz.
- $V_S$ = Voltaje en el devanado secundario del transformador.
- $V_{DC}=V_C$ = Voltaje en corriente directa y Voltaje en la carga.
- $I_{DC}= I_C$ = Corriente directa y Corriente en la carga.
- $RL$ = Carga útil.
- $P_{DC}$ = potencia en corriente directa.

$$V_{MAX} = (\sqrt{2})(V_{RMS}) = (\sqrt{2})(V_S) = (\sqrt{2})(18) = 25,45 V$$

$$V_{DC} = V_C = \frac{(2)(V_{MAX})}{\pi} = \frac{(2)(25,45)}{\pi} = 16,20 V$$

$$I_{DC} = I_C = \frac{V_{DC}}{RL}; \quad Si \quad RL = 100\Omega \rightarrow \frac{16,20}{100} = 162 \text{ mA}$$

$$P_{DC} = (V_{DC})(I_{DC}) = (16,20 V)(162 \text{ mA}) = 2,63 W$$

Este tipo de configuración de diodos permite eliminar la parte negativa de la onda senoidal proporcionada por el transformador. Este proceso se analiza según el semiciclo de la onda senoidal.

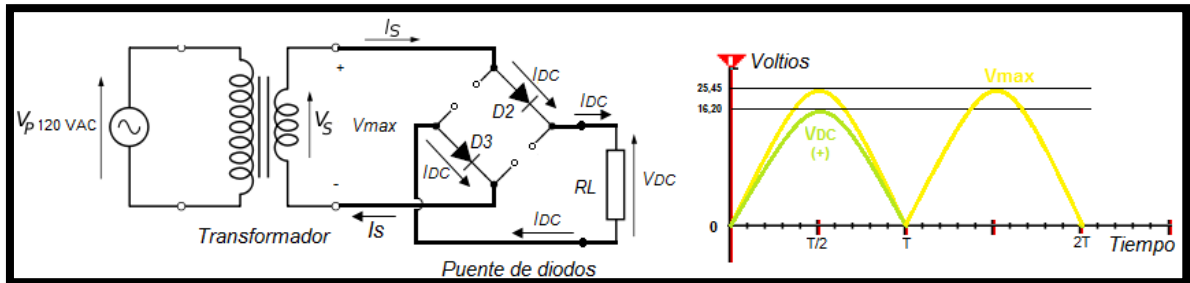
### **3.6.3.2.1 Análisis del semiciclo positivo.**

Tomando en cuenta las características de los diodos y el sentido de circulación de la corriente del transformador en el circuito de la figura siguiente, la corriente ( $I_s$ ) polariza al diodo diodos D1 inversamente por lo que este no permite

la circulación de la misma permaneciendo en un estado de “circuito abierto”, mientras que el diodo D4 se encuentra polarizado directamente dando paso a la circulación de la corriente hacia la carga (C) y para cerrar el circuito el diodo D3 se polariza directamente permitiendo cerrar el circuito en la parte inferior negativa del transformador mientras que en el diodo D4 está polarizado inversamente provocando el mismo estado que el diodo D1.

Lo dicho que puede apreciar en la siguiente figura dando como resultado el primer ciclo positivo de la onda sinusoidal proporcionado por el transformador.

Figura 51. Análisis del rectificador en el semiciclo positivo y curva característica.



Fuente: El Autor.

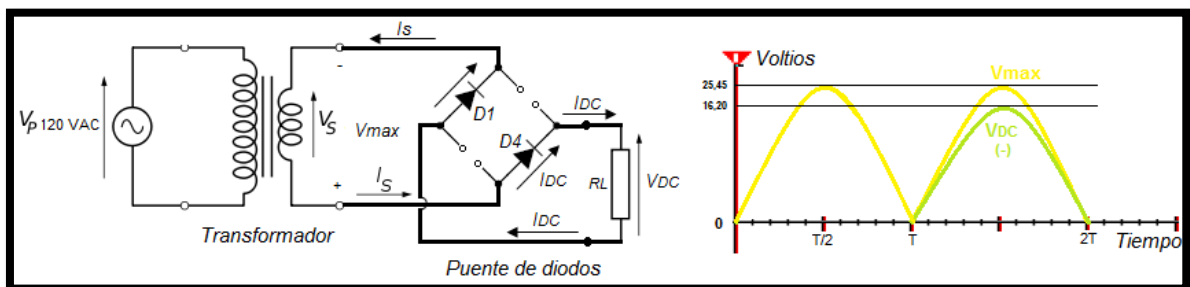
### 3.6.3.2 Análisis del semiciclo negativo.

De igual forma que el análisis anterior, en este los diodos D3 y D2 son polarizados inversamente permaneciendo estos en un estado de “circuito abierto” impidiendo la circulación de la corriente, mientras que los diodos D1 y D4 son polarizados directamente conduciendo la corriente hacia la parte negativa del

transformador cerrando el circuito y obteniendo la onda del semiciclo negativo continua al proceso anterior.

Lo dicho que puede apreciar en la siguiente figura dando como resultado el segundo ciclo positivo de la onda sinusoidal proporcionado por el transformador.

Figura 52. Análisis del rectificador en el semiciclo negativo y curva característica.



Fuente: El Autor.

La frecuencia de trabajo por ende será el doble ya que como se puede apreciar al cumplirse un periodo de tiempo la onda se duplica siendo esta frecuencia la siguiente.

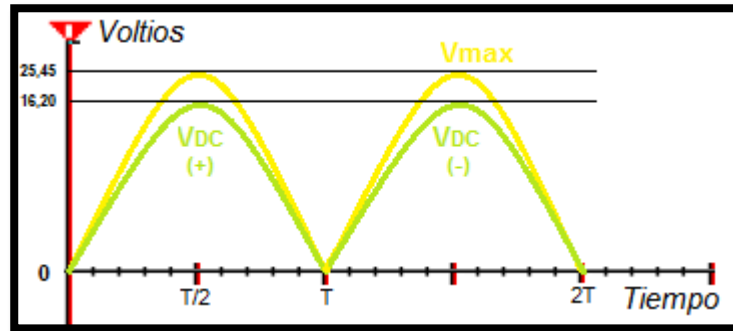
$$f_{out} = (2)(f_{in})$$

Donde:

- $f_{out}$  = Frecuencia de salida.
- $f_{in}$  = Frecuencia de entrada.

$$f_{out} = (2)(f_{in}) = (2)(60Hz) = 120 Hz$$

Figura 53. Rectificación de onda completa en los dos semiciclos.



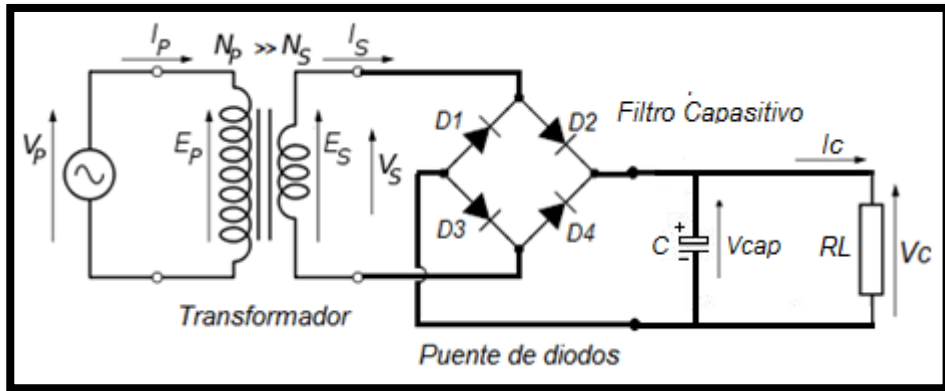
Fuente: El Autor.

De esta forma se obtiene la rectificación en continua de la tensión deseada para posteriormente pasar a un tercer proceso que es el filtro de esta onda.

### 3.6.3.3 Convertidor de continua a continua o filtro (CD/CD)

El filtro permite eliminar las componentes de frecuencias distintas de cero, es decir, reduce el rizado, aísla la componente alterna de la continua, asegura un comportamiento lineal de la tensión y permite que la componente continua llegue a la carga. Este filtro puede ser elementos como condensadores o bobinas.

Figura 54. Filtro capacitivo en una fuente de alimentación.



Fuente: El Autor.

Para hallar el valor del capacitor el cual realiza la función de filtrado se realizan los siguientes cálculos.

$$C = \frac{I_{DC}}{(V_{PP})(f_{out})}$$

$$V_{PP} = (m)(V_{DC})$$

$$P_r(\%) = \left( \frac{1}{(2)(f_{out})(C)(RL)} \right) (100)$$

Donde:

- $C$  = Valor del capacitor.
- $V_{PP}$  = Voltaje pico a pico.
- $P_r(\%)$  = Porcentaje de rizado.



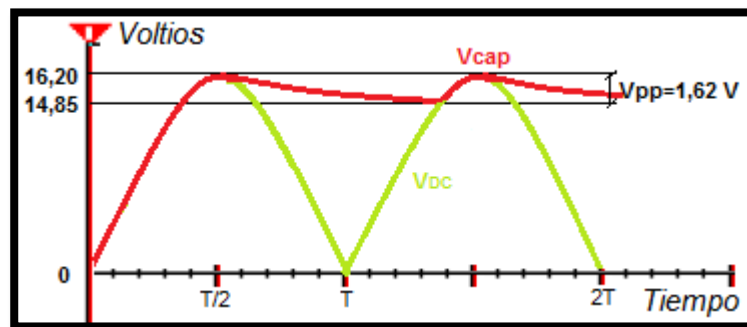
- $m$  = Porcentaje referencial para el cálculo del voltaje pico a pico.

$$\text{Si } m = 10\% \rightarrow V_{PP} = (10\%)(16,20) = 1,62 \text{ V}$$

$$C = \frac{I_{DC}}{(V_{PP})(f_{out})} = \frac{162mA}{(1,62V)(120Hz)} = 832,33\mu F \cong 1000\mu F$$

$$P_r(\%) = \left( \frac{1}{(2)(120Hz)(832,33\mu F)(100\Omega)} \right) (100) = 0,00005\%$$

Figura 55. Filtro de salida con el capacitor



Fuente: Al Autor.

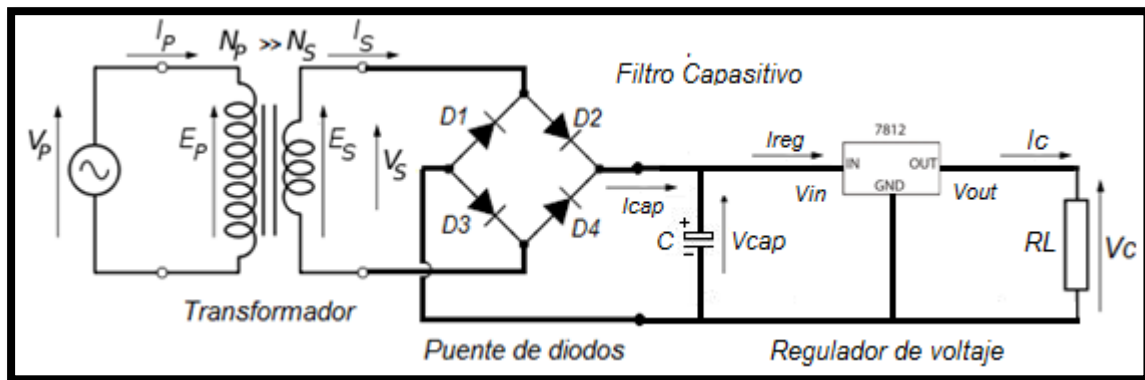
Para asegurar que una tensión no sea muy elevada en el rizado, conviene colocar un condensador más grande que el obtenido numéricamente para este caso se utilizara un capacitor electrolítico de 1000uF a 25 V.

### 3.6.3.4 Estabilizador o regulador de voltaje.

Por último, el regulador de voltaje permite estabilizar la tensión en un valor determinado, siendo este una tensión continua. Los reguladores pueden ser transistores, diodos zener, configuración de resistencias, etc.

Para el diseño de esta fuente de alimentación se utiliza transistores de regulación, uno de ellos puede ser el LM7812 el cual permite estabilizar un voltaje de entrada ( $V_{in}$ ) a un valor fijo de salida ( $V_{out}$ ) en 12 voltios. El proceso de estabilización se lo puede encontrar en su datasheet.

Figura 56. Regulador de voltaje en una fuente de alimentación.



Fuente: El Autor.

Uno de los requerimientos para poder estabilizar un voltaje con un regulador es que se debe cumplir lo siguiente.

$$V_{in} > V_{out}$$

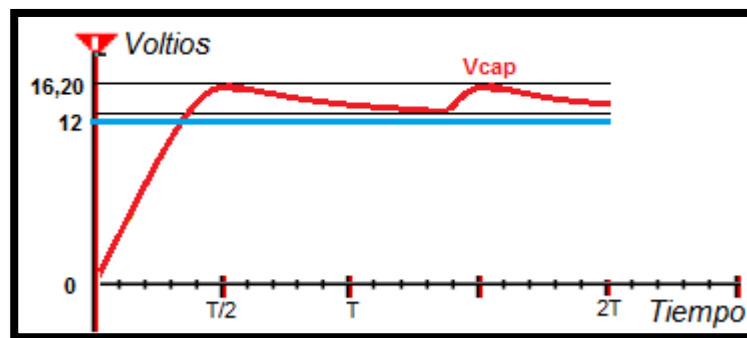
$$V_{dis} = V_{in} - V_{out}$$

Donde:

- $V_{in}$  = Voltaje de entrada del regulador.
- $V_{out}$  = Voltaje de salida del regulador.
- $V_{dis}$  = Voltaje disipado o reducido.

$$V_{dis} = V_{in} - V_{out} = 16,20 - 12 = 4,20 \text{ V}$$

Figura 57. Estabilización del voltaje en una fuente de alimentación.



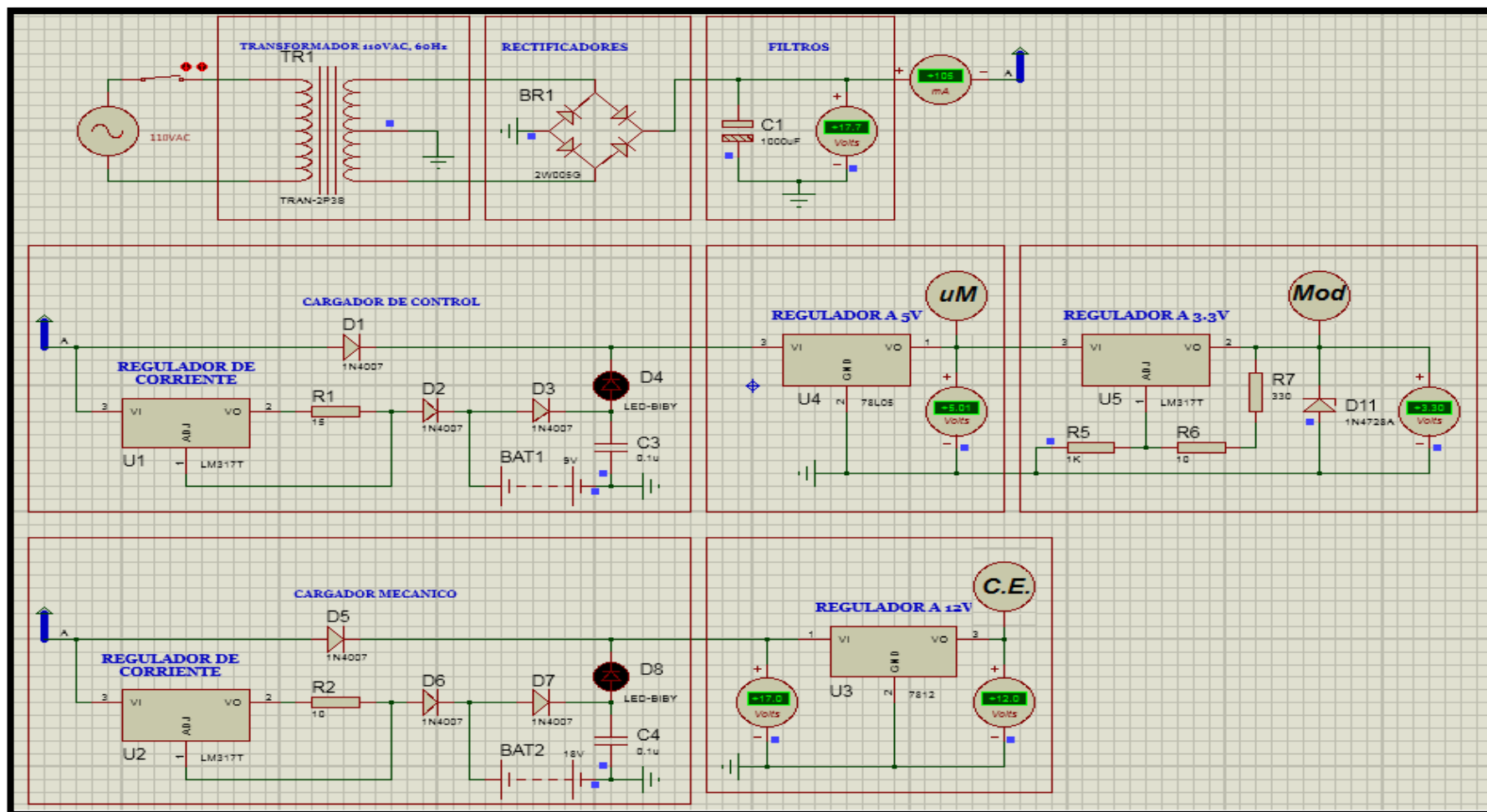
Fuente: El Autor.

Una fuente de alimentación es estabilizada o regulada si la variación relativa de la tensión de salida es menor que la variación relativa de tensión de la entrada.

En la fuente de alimentación que se diseña en este proyecto, el transformador reduce la tensión de 110VDC a 9, 18VAC y mediante rectificadores de onda con los reguladores de voltaje se obtendrá voltajes de 12, 5 y 3.3 VDC para alimentar los diferentes dispositivos eléctricos debido a las especificaciones eléctricas del datasheet de cada uno de ellos.

La fuente de alimentación del circuito receptor se indica en la siguiente figura y es realizada según los requerimientos explicados anteriormente. Esta fuente se realizó en el programa Proteus, para simularlo y poder utilizar los elementos necesarios.

Figura 58. Fuente de alimentación del circuito receptor.



Fuente: El autor (Diseño realizado en el programa Proteus).

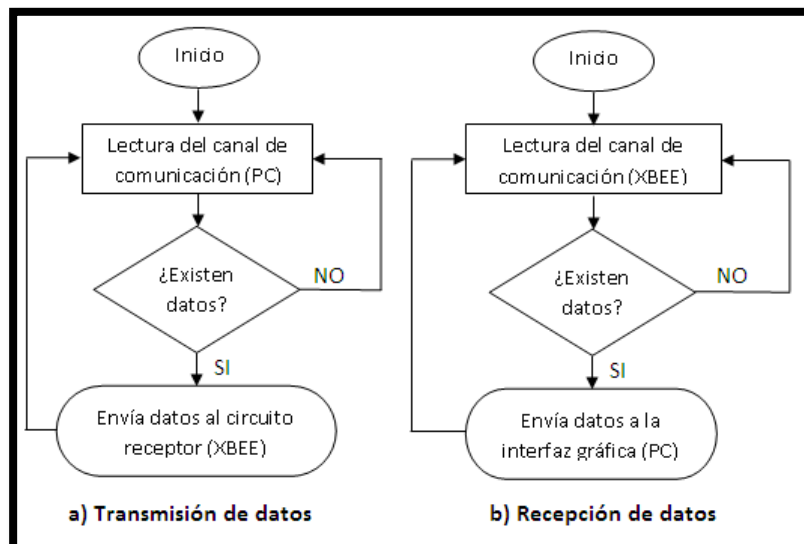


El proceso de transmisión y recepción de datos de los diferentes circuitos se lo representa a continuación en el siguiente diagrama de flujo, así como la lectura y escritura de los mismos ya sea en la interfaz gráfica como en el microcontrolador.

### 3.7.1 Diagrama de flujo del circuito transmisor

Dado que en el circuito transmisor solo tendrá la comunicación entre la PC y el módulo inalámbrico XBEE, en el diagrama de flujo de ese circuito se especifica la transmisión y recepción de la información proveniente de la interfaz gráfica y del circuito receptor respectivamente.

Figura 60. Diagrama de flujo de la transmisión y recepción de datos del circuito transmisor.



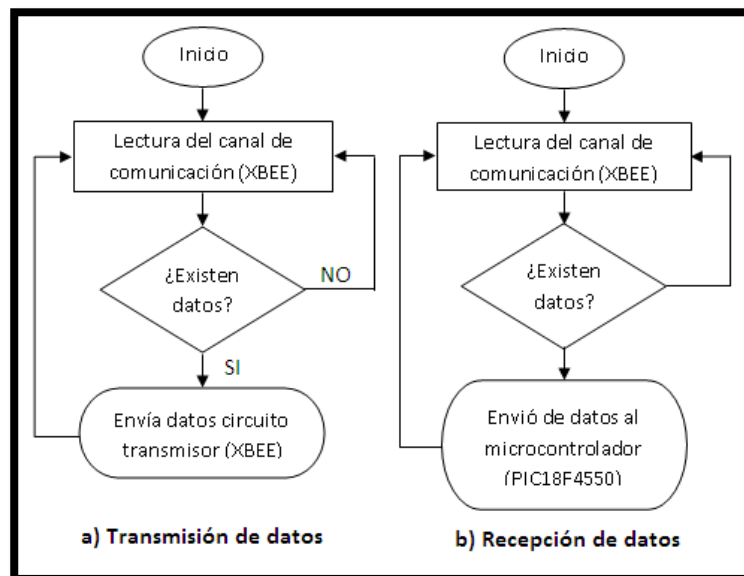
Fuente: El autor.

### 3.7.2 Diagrama de flujo del circuito receptor

En este punto el proceso es más complejo debido a que los datos recibidos por el módulo inalámbrico XBEE es transmitido hacia el microcontrolador, y por ende realiza otras funciones, es por ello que se crean subrutinas para el microcontrolador.

El diagrama de flujo de este circuito especifica la transmisión y recepción de la información proveniente del circuito receptor, del módulo inalámbrico XBEE y el microcontrolador respectivamente.

Figura 61. Diagrama de flujo de la transmisión y recepción de datos del circuito receptor.



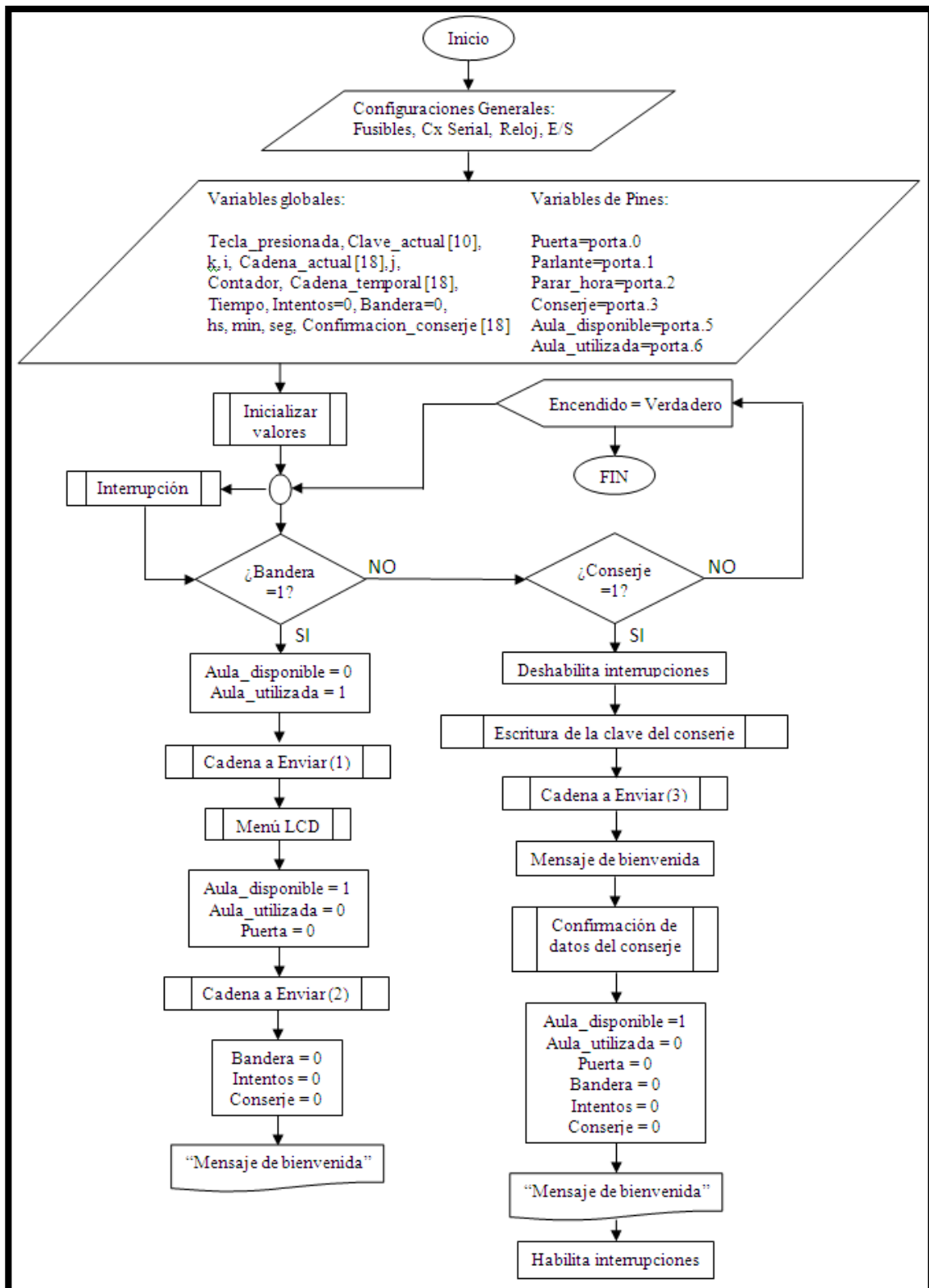
Fuente: El autor.



El proceso de análisis de los datos recibidos por parte del módulo inalámbrico XBEE en el circuito receptor es más complejo que el análisis de datos del circuito transmisor, debido a que el microcontrolador se encarga de realizar todo el proceso de comparación de variables, ejecución de datos, visualización y manipulación de los elementos electrónicos según los datos adquiridos.

Es por ello que a continuación en el siguiente diagrama de flujo se explica este proceso el mismo que está contenido por varias subrutinas. Las subrutinas se describen en diferentes diagramas ya que son diferentes métodos según se indiquen, cabe recalcar que una subrutina puede contener otra subrutina formando una cadena entre ellas.

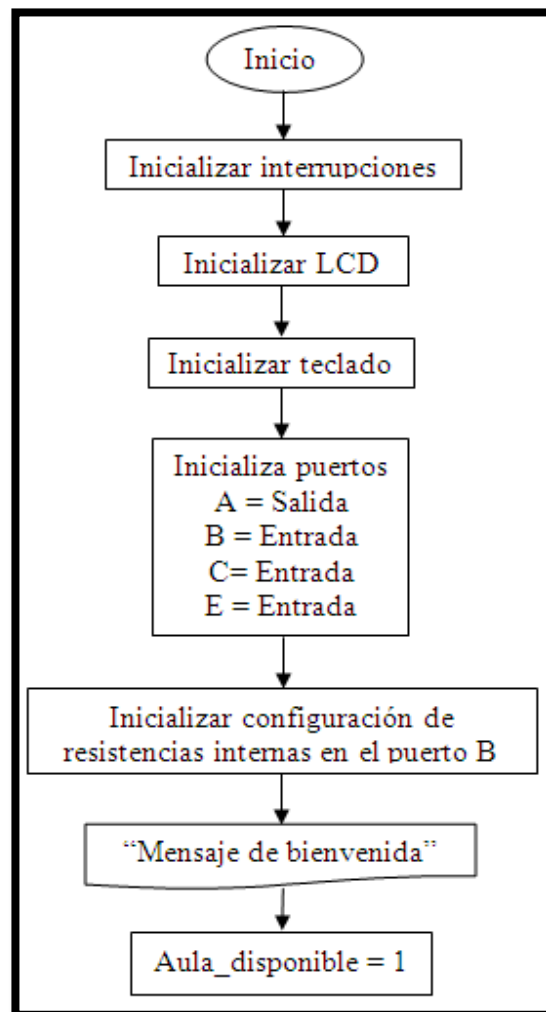
Figura 62. Diagrama de flujo del programa principal del microcontrolador (PIC18F4550)



Fuente: El autor.

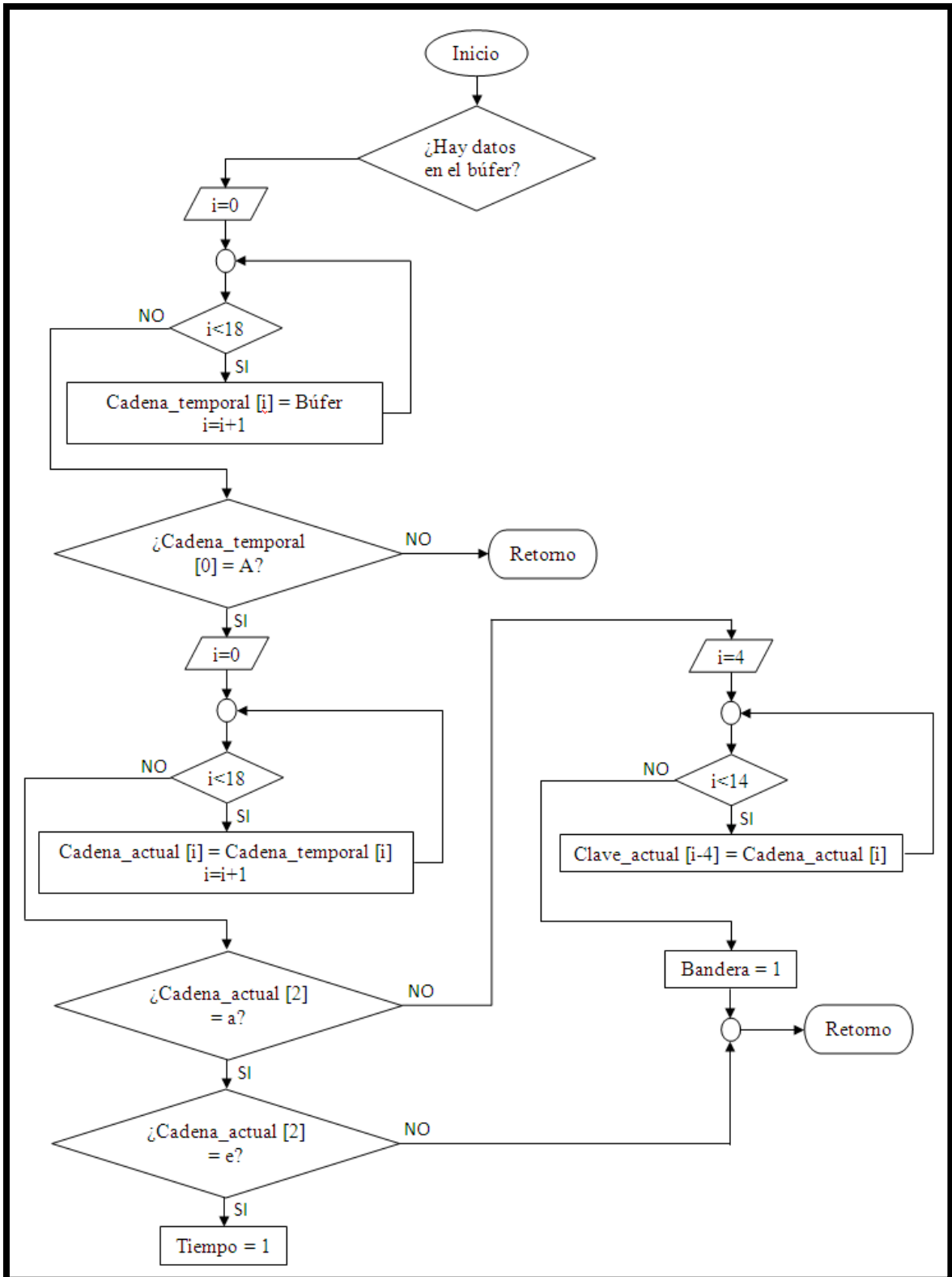
Como se puede apreciar en la figura anterior, en el programa principal del microcontrolador existen varias subrutinas o métodos que fueron creados para poder separar procesos, tal es el caso de los métodos: “Inicializar valores”, “Interrupción”, “Menú LCD”, “Escritura de la clave del conserje”, “Confirmación de datos” y “Cadena a enviar”. Este último tiene un parámetro de entrada el cual se expresa en paréntesis con un valor “()”, es decir, este método según el parámetro realizará una u otra función según se explique en el diagrama de flujo del mismo.

Figura 63. Diagrama de flujo de la subrutina para inicializar valores.



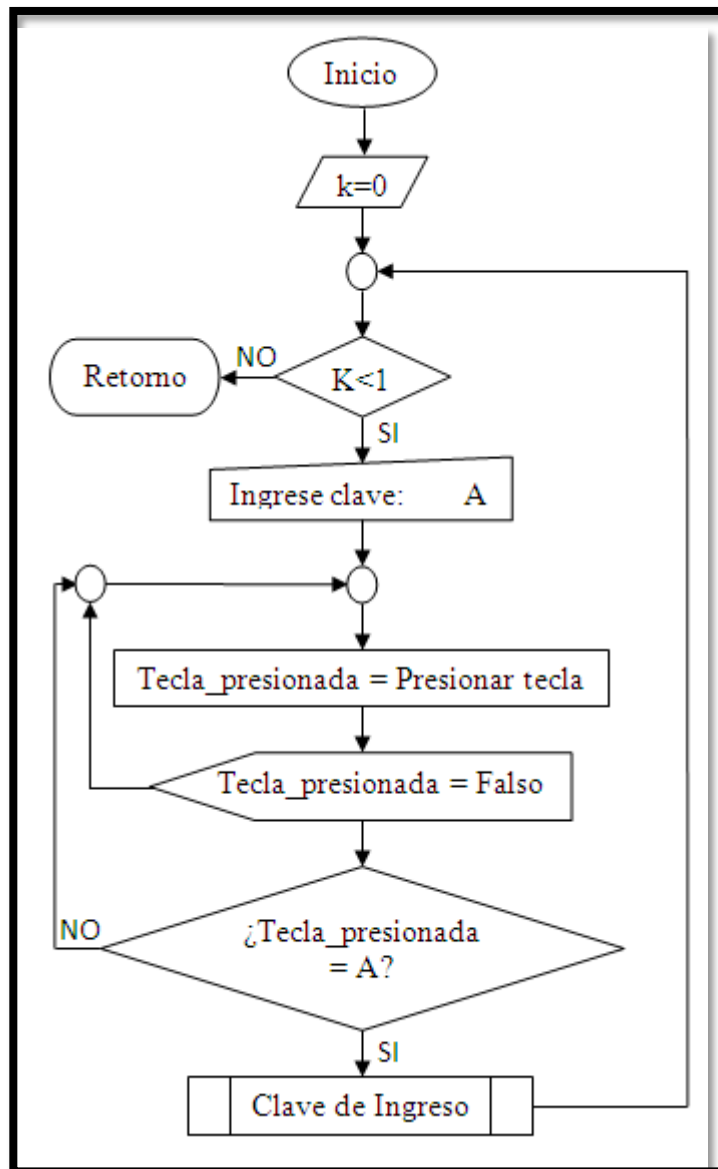
Fuente: El autor.

Figura 64. Diagrama de flujo de la subrutina interrupción.



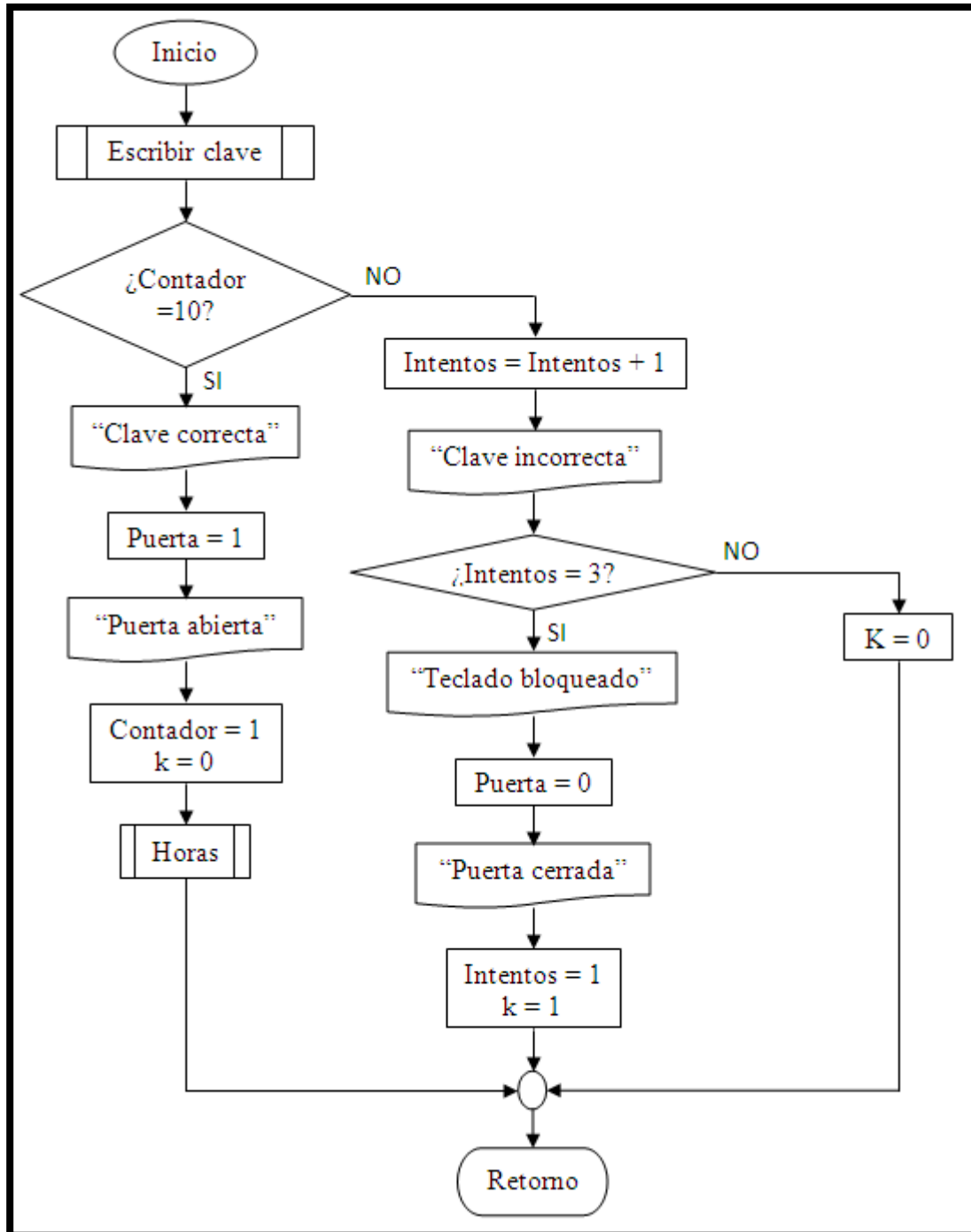
Fuente: El autor.

Figura 65. Diagrama de flujo de la subrutina para el menú de la LCD



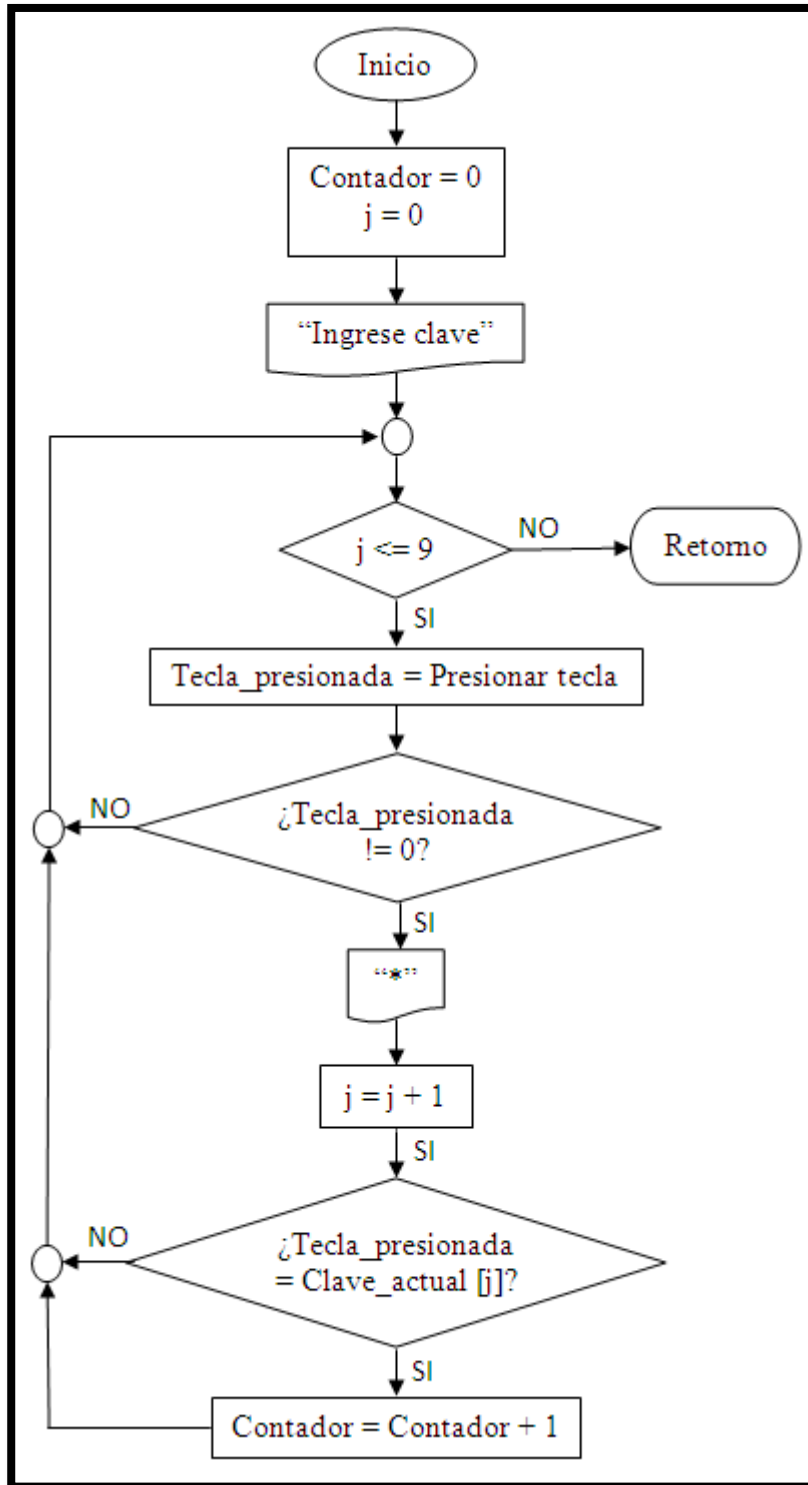
Fuente: El autor.

Figura 66. Diagrama de flujo de la subrutina clave de ingreso



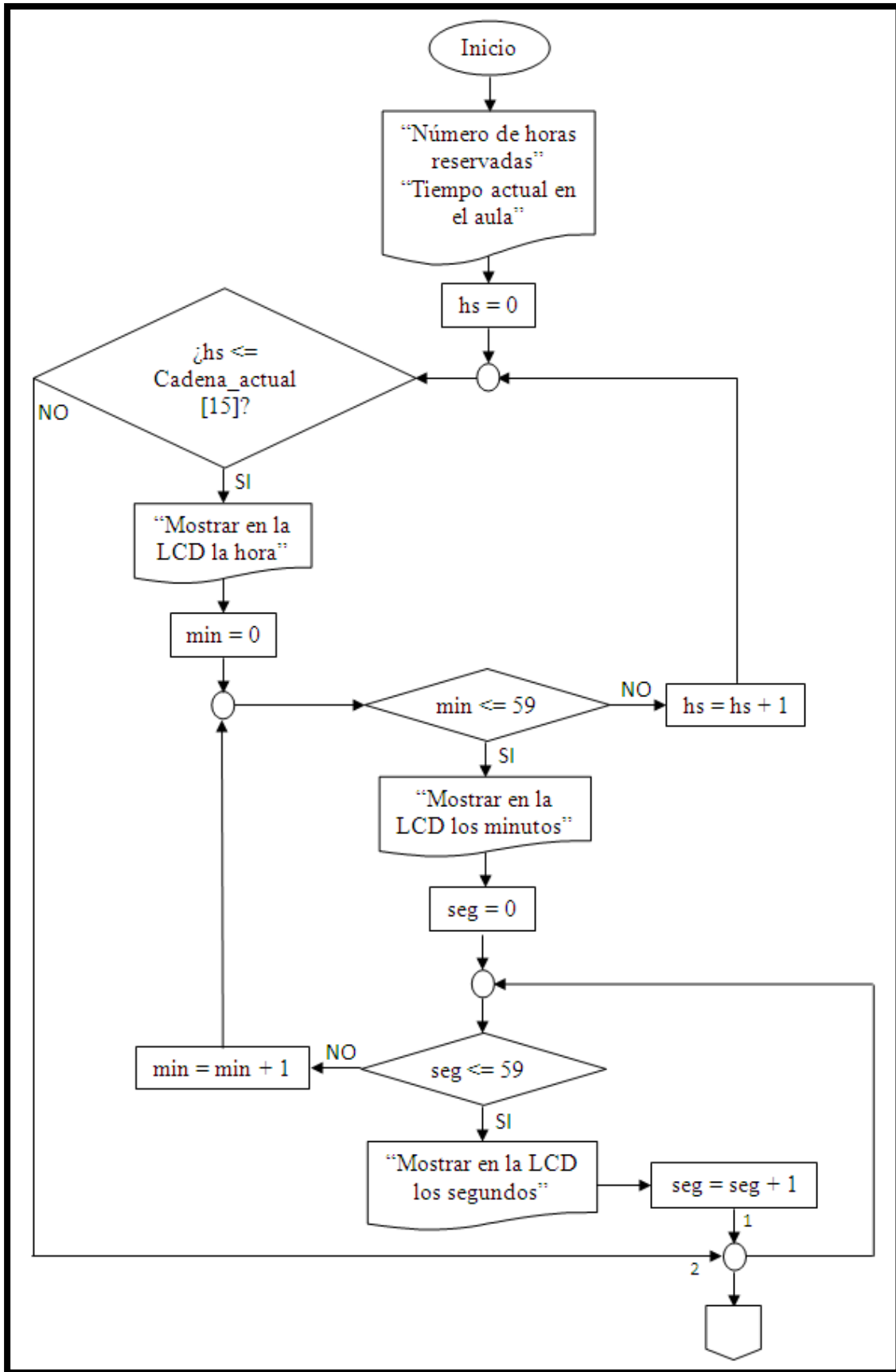
Fuente: El autor.

Figura 67. Diagrama de flujo de la subrutina escribir clave

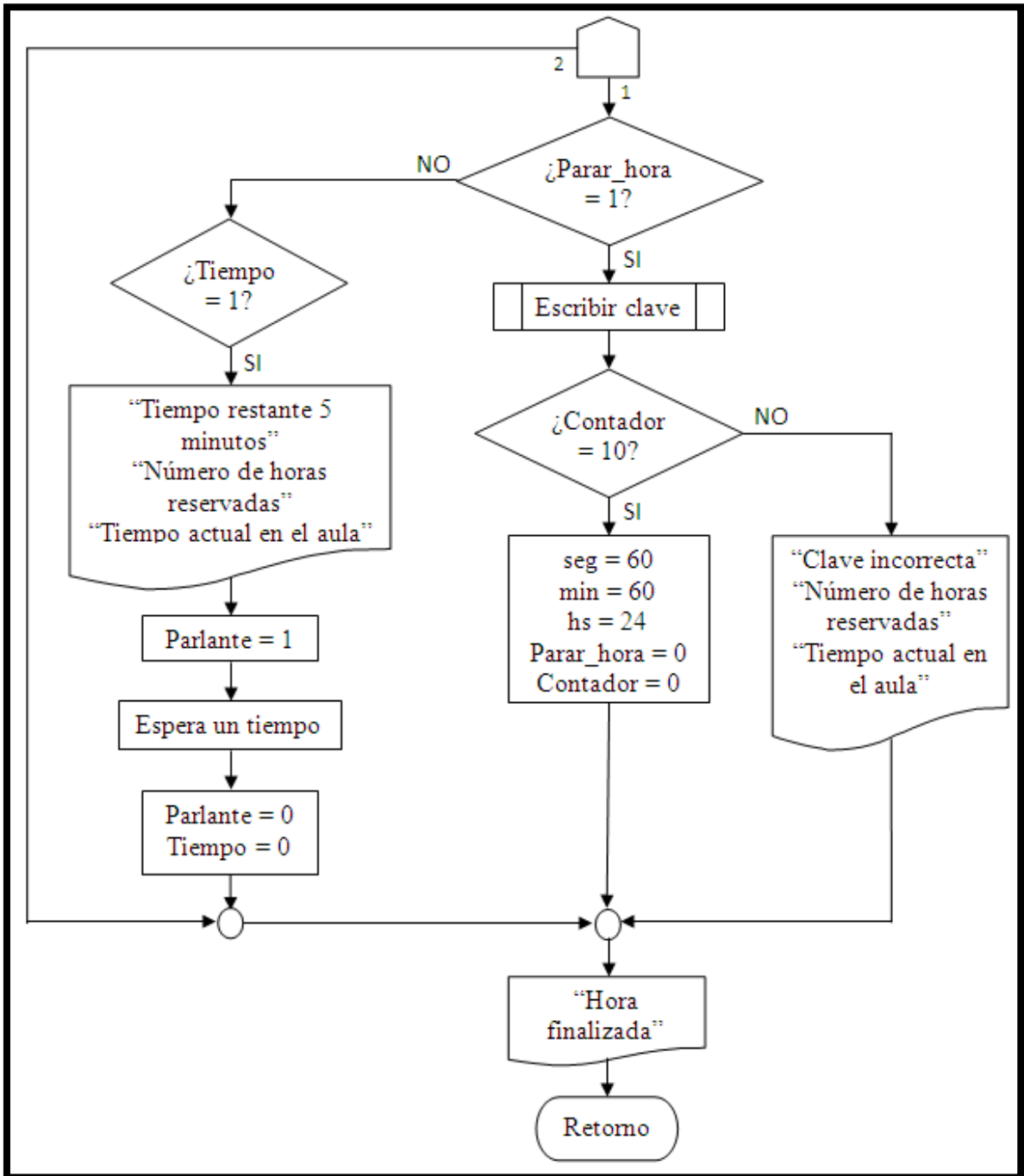


Fuente: El autor.

Figura 68. Diagrama de flujo de la subrutina horas

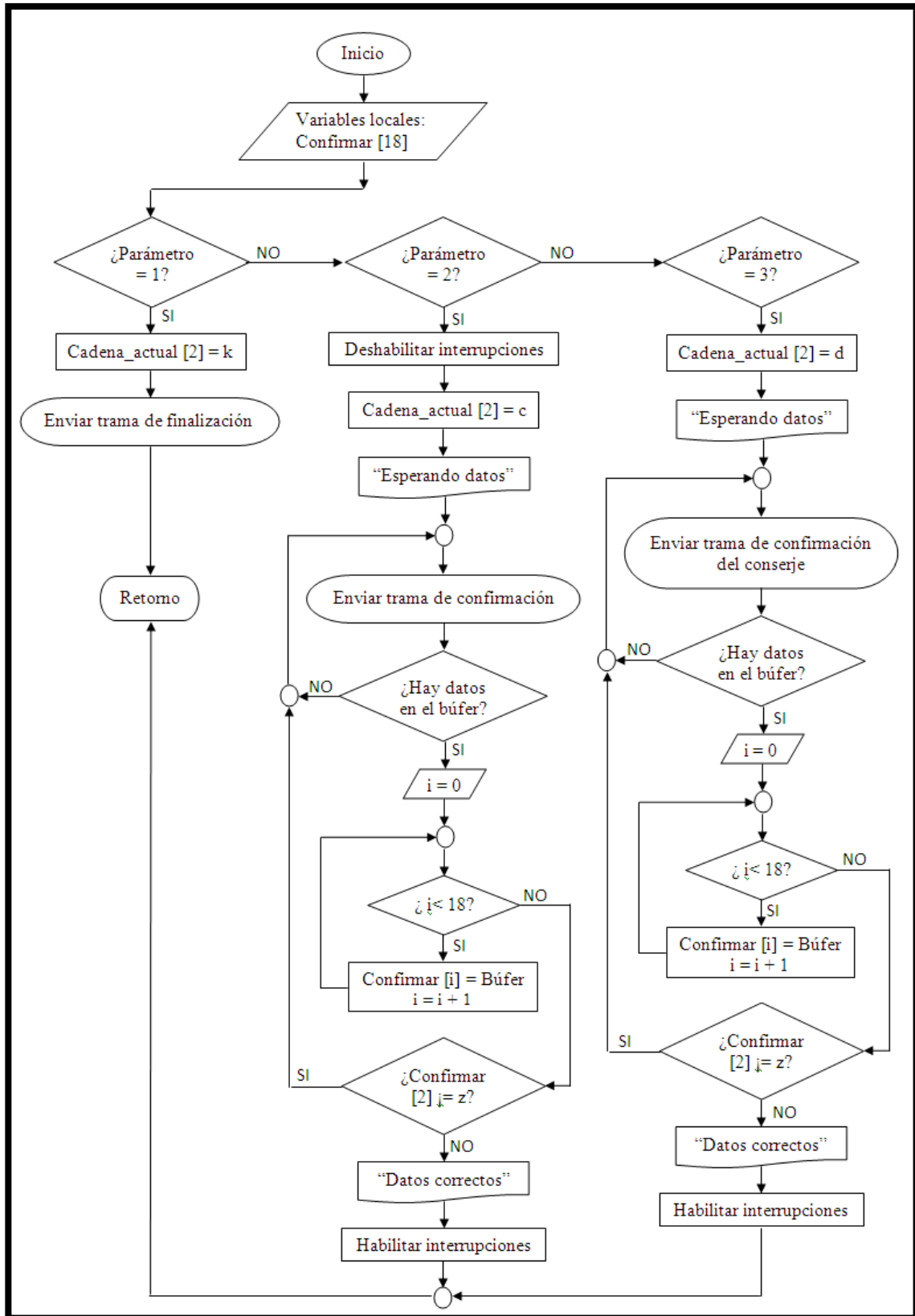






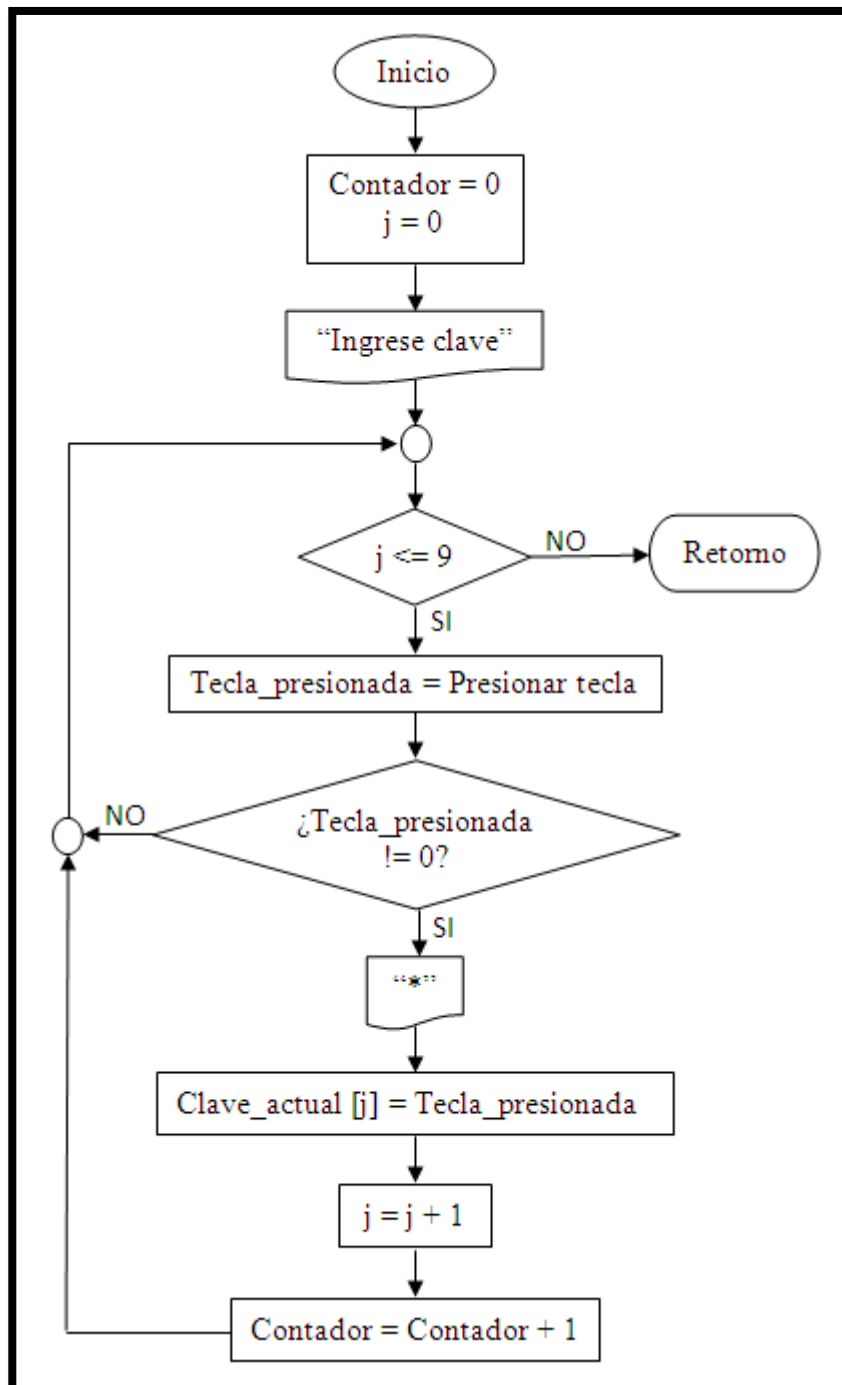
Fuente: El autor.

Figura 69. Diagrama de flujo de la subrutina cadena a enviar



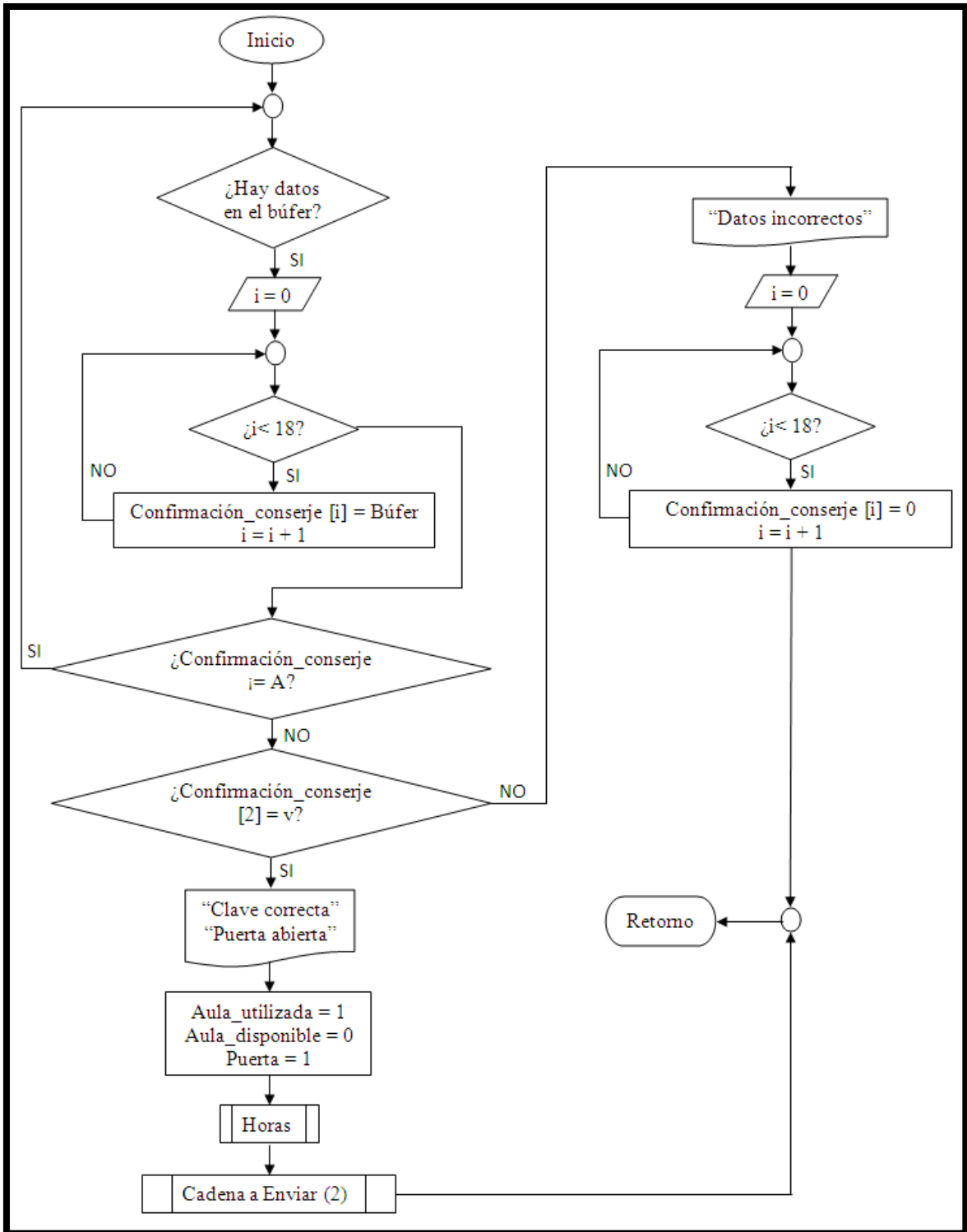
Fuente: El autor.

Figura 70. Diagrama de flujo de la subrutina escritura de la clave del conserje



Fuente: El autor.

Figura 71. Diagrama de flujo de la subrutina confirmación de datos del conserje

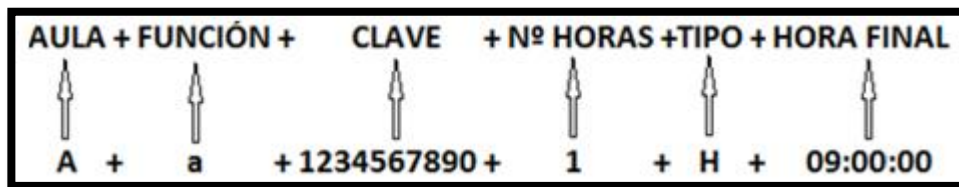


Fuente: El autor.

### 3.8 Tramas de envío y recepción.

El método para enviar y recibir tramas se menciona en el diagrama de flujo de la figura 53. Las lecturas de cada sistema electrónico son enviadas en texto plano y la información que contiene la trama que se transmite y recibe de cada sistema electrónico tiene el siguiente formato:

Figura 72. Formato general de las tramas



Fuente: El autor.

Donde:

- AULA: Pertenece al aula que corresponde la trama.
- FUNCIÓN: tiene los siguientes caracteres:
  - a: Indica que el aula será abierta por un determinado docente.
  - c: Indica la finalización de la hora de clases por parte del docente
  - d: Verifica al docente si está registrado para utilizar esa aula
  - e: Informa al docente que le quedan 5 minutos para que finalice la hora de clases en esa aula.
  - k: Identifica una confirmación del dato enviado.
  - v: Indica que la clave del docente es correcta
  - f: Indica que la clave del docente es incorrecta

- CLAVE: contiene 10 dígitos que serán ingresados por teclado, corresponde a los caracteres para ingresar al aula, en caso de ser digitados correctamente el aula será abierta, caso contrario el sistema se bloquea.
- HORAS: Indica el número de horas que el docente permanecerá en el aula.
- TIPO: Identifica si el aula será utilizada por:
  - H: El horario correspondiente a ese periodo académico en esa día.
  - R: Si el aula será utilizada por una reserva realizada con anticipación por parte del docente.
  - L: Si el aula será utilizada por un conserje para una limpieza al aula.
- HORA FINAL: Identifica el periodo de tiempo que el docente se encontrará en el aula.
- SIGNO (+): El signo “+” identifica una separación de cada uno de los datos a enviar y recibir.

### **3.9 Programación del microcontrolador**

La programación del microcontrolador se especifica en el anexo B del presente trabajo de titulación, en él se identifica cada una de las funciones que realiza las líneas de código, las mismas que fueron relacionadas conforme se explica en los diagramas de flujo anteriormente indicados, este código fue elaborado en el programa “PIC C Compiler”.

### 3.10 Funcionamiento del sistema eléctrico.

En este punto se analizan los parámetros eléctricos y las configuraciones físicas que son utilizadas en el circuito receptor, ya que en este es el que mayor trabajo se realiza, el mismo que será analizado por separado dependiendo el funcionamiento que realicen cada elemento electrónico.

#### 3.10.1 Recepción y transmisión de datos

Para poder recibir y transmitir los datos se debe analizar las características que el módulo XBEE Pro necesita, y la configuración con la que se conecte al microcontrolador, los mismos que se explican a continuación.

Al momento de transmitir los datos se debe tomar en cuenta que tanto el voltaje que proporciona el microcontrolador y el módulo XBEE no es el mismo, hay que tener cuidado al momento de integrar estos dos elementos, es por ello que se debe realizar una configuración extra en los pines de transmisión del microcontrolador (Tx), y recepción del módulo XBEE (Rx), para poder así regular el voltaje a su valor máximo según las especificaciones.

Datos:

$$Vm_{Tx} = 5V$$

$$Im_{Tx} = 0.17mA$$

$$V_0 = 3.3V$$

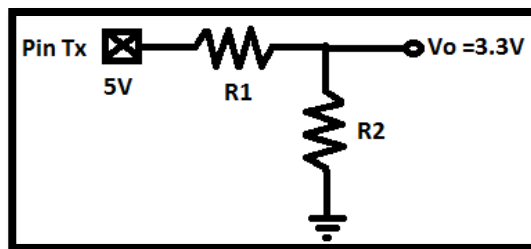
$$R_1, R_2 = ?$$

Donde:

- $V_{m_{Tx}}$  = Voltaje microcontrolador del pin de transmisor
- $I_{m_{Tx}}$  = Corriente microcontrolador del pin de transmisor
- $V_0$  = Voltaje de salida requerido para el módulo XBEE
- $R_1, R_2$  = Resistencias a calcular

A continuación se calcula los valores de las resistencias (R1 y R2). Para ello es necesario utilizar la ley de Ohm, para poder calcular dichas resistencias despejando los valores de voltaje y corriente, como se explica en los siguientes cálculos.

Figura 73. Diagrama de conexión para el cálculo de las resistencias R1 y R2.



Fuente: El autor.

$$V_{R1} = V_{m_{Tx}} - V_0$$

$$V_{R1} = 5V - 3.3V$$

$$V_{R1} = 1.7V$$

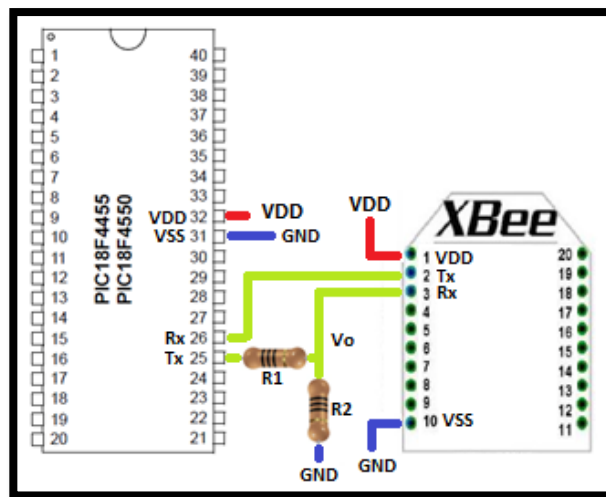
$$R_1 = \frac{V_{R1}}{I_{m_{Tx}}} = \frac{1.7V}{0.17mA} = 10K\Omega$$

$$R_2 = \frac{V_0}{I_{m_{Tx}}} = \frac{3.3V}{0.17mA} = 19,411K\Omega \cong 20K\Omega$$



En lo referente a la transmisión desde el módulo XBEE Pro hacia el microcontrolador, no es necesario regular el voltaje, ya que este permite trabajar con voltajes hasta 2.2V en sus pines, por lo que este puede funcionar sin ningún problema. Dado que en el mercado comercial no se puede encontrar resistencias de los valores que se indican en las ecuaciones anteriores, es necesario escoger o aproximar a un valor que si se los pueda encontrar, tal es el caso de la resistencia R2. En la siguiente figura se indica la conexión para la transmisión y recepción de los datos entre el microcontrolador y el módulo XBEE Pro.

Figura 74. Diagrama de conexión física entre el microcontrolador y el módulo XBEE Pro.



Fuente: El autor.

### 3.10.2 Visualización de los datos

Para poder visualizar datos del microcontrolador, se utiliza una LCD, la cual para que pueda funcionar es necesario disponer de una librería configurada



### 3.10.3 Mecanismo de apertura de la puerta

Para lograr abrir la puerta del aula se utiliza un mecanismo eléctrico con un relé y un transistor BJT en configuración emisor común, este transistor es el 2N3904 ya que permite trabajar con voltajes de hasta 40V y 200mA, para ello es necesario obtener los valores de la resistencia que se coloca entre el microcontrolador y el transistor, este cálculo se explica a continuación.

Datos:

$$I_{Bmax} = 50mA$$

$$h_{FE} = 30$$

$$Vm = 5V$$

$$V_{BE} = 0.7V$$

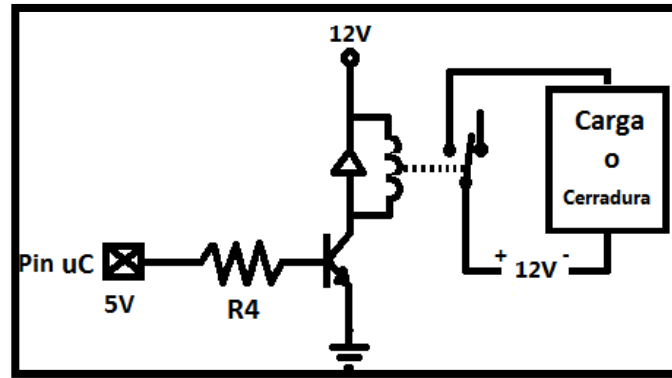
$$R_4 = ?$$

Donde:

- $I_{Bmax}$  = Corriente máxima del transmisor en la base
- $h_{FE}$  = Ganancia del transistor
- $Vm$  = Voltaje microcontrolador
- $V_{BE}$  = Voltaje base – emisor
- $R_4$  = Resistencia a calcular

A continuación se calcula los valores de la resistencia (R4). Para ello se realizará la configuración de emisor común de un transistor para que pueda ser accionado el relé en un estado positivo del pin del microcontrolador.

Figura 76. Diagrama de conexión para el cálculo de la resistencia R4.



Fuente: El autor.

$$I_B = \frac{I_{Bmax}}{h_{FE}} = \frac{50mA}{30} = 1.67mA$$

$$Vm = I_B * R_4 + V_{BE}$$

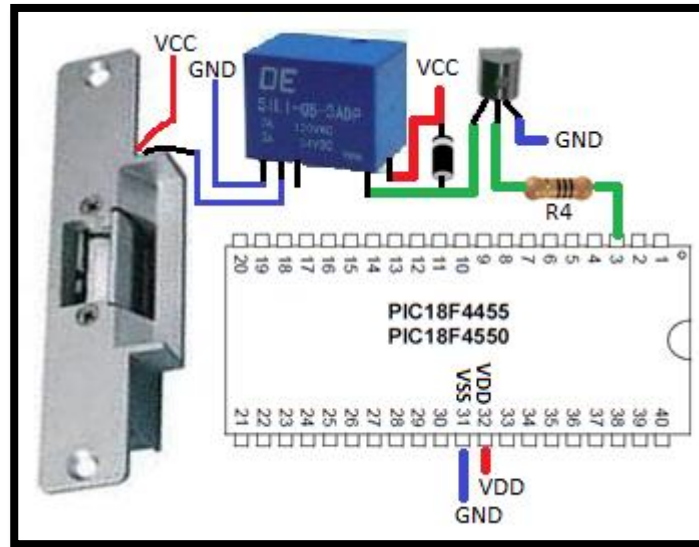
$$I_B * R_4 = Vm - V_{BE} = 5V - 0.7V$$

$$I_B * R_4 = 4.3V$$

$$R_4 = \frac{4.3V}{I_B} = \frac{4.3V}{1.67mA} = 2.57K\Omega \cong 2.7K\Omega$$

De igual forma que en el caso anterior, en el mercado comercial no se pueden encontrar resistencias del valor que se indica en la ecuación anterior, es por ello que se escoge una resistencia aproximada al valor calculado, tal es el caso de la resistencia R4. En la siguiente figura se indica la conexión para el mecanismo de apertura de la puerta.

Figura 77. Diagrama de conexión física entre el microcontrolador, el relé y la cerradura.



Fuente: El autor.

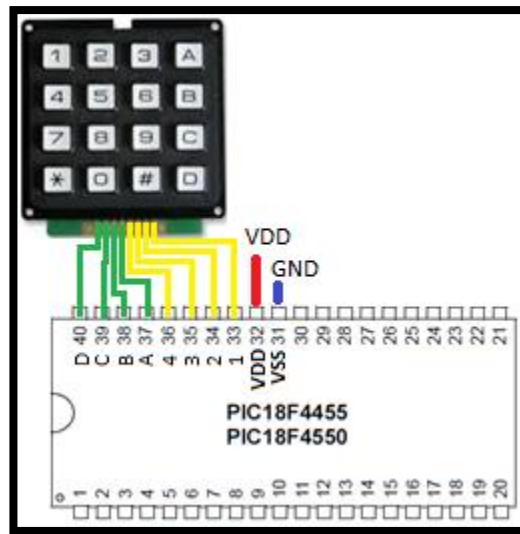
### 3.10.4 Seguridad del aula (Autenticación)

El utilizar el teclado matricial en el circuito receptor ayuda a dar seguridad al aula, ya que sólo la persona que tenga la clave asignada a una determinada hora podrá ingresar a la misma, de esta forma se puede asegurar que sólo la persona encargada de utilizar el aula sea quien permita el acceso o no, siendo esta la única responsable de los bienes que se encuentren en ella, el diagrama de conexión de este método es muy sencillo, ya que al configurar el microcontrolador de una forma especial, permite que solo sea necesario la conexión de las filas y las columnas del teclado, siendo estos los siguientes:

Las filas se encuentran numeradas del 1 al 4 para poder diferenciar de las columnas que se encuentran abreviadas con las letras A,B,C y D, esta es la forma

de conexión previa a la modificación de la librería utilizada por el microcontrolador, para que este dispositivo pueda funcionar correctamente. En la figura siguiente se indica la forma de conexión de estos dos dispositivos.

Figura 78. Diagrama de conexión física entre el microcontrolador y el teclado.

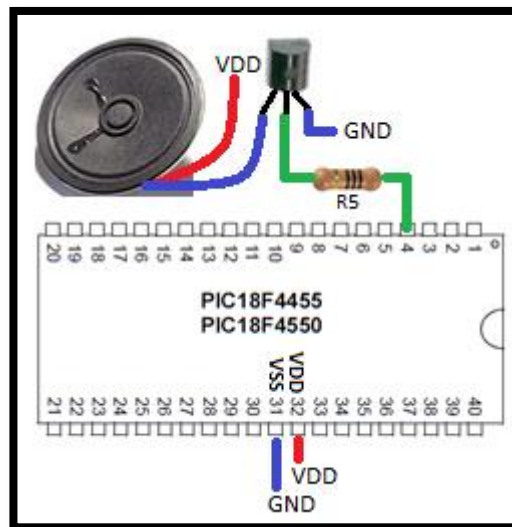


Fuente: El autor.

### 3.10.5 Información del tiempo restante en el aula (Alarma)

Este circuito se utiliza para poder informar, mediante un sonido, al docente que su tiempo está a punto de terminar. Los parámetros de este tiempo son configurados previamente en el microcontrolador, dando así una mejor forma de utilización de las aulas. La configuración física de este es similar al de la apertura de la puerta ya que se utilizan los mismos elementos y configuraciones por lo tanto no es necesario calcular nuevamente los valores de su resistencia. En la figura siguiente se indica la forma de conexión.

Figura 79. Diagrama de conexión física entre el microcontrolador y el parlante.



Fuente: El autor.

### 3.10.6 Interrupción de la hora de clase y utilización del aula por un conserje (Pulsadores)

Este módulo permite utilizar un aula sin necesidad de registrarse en el sistema (software descrito en el capítulo III), de ante mano este método será utilizado únicamente por los conserjes que realizan funciones diferentes a los docentes de la facultad, e interrumpir las horas de clases de un docente si él así lo amerite.

Datos:

$$V_{in} = 5 V$$

$$I_{mic} = 25 mA$$

$$R_6 = ?$$

Donde

- $V_{in}$  = Voltaje de entrada.
- $I_{mic}$  = Corriente máxima del microcontrolador.
- $R_6$  = Resistencia a calcular.

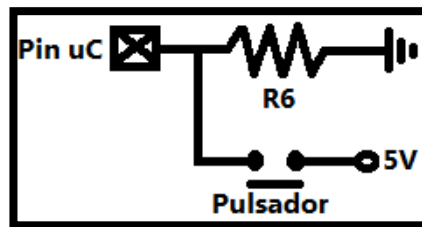
A continuación se indica el valor de la resistencia ( $R_6$ ). Para ello se utiliza una configuración denominada resistencia de polarización (pull-down), la cual permite mantener al pin del microcontrolador en un estado de 0 V (bajo consumo de energía) mientras el interruptor o pulsador no se encuentre presionado, caso contrario el pin pasa a estar en un estado de 5 V (alto consumo de energía) al momento de ser presionado dicho elemento.

Se realiza esta configuración en este tipo de módulos para evitar que el pin del microcontrolador esté censando constantemente el estado de dicho pin, es decir, debido a los niveles de voltaje que se manejan en un microcontrolador, al ser configurado previamente un pin del mismo como entrada, éste admitirá cualquier señal que sea posible y al estar en una posición liberada (sólo conectada con un pulsador) y tomando en cuenta que los niveles voltaje para un estado positivo (1 lógico) es de 2.8 V a 5V, y para un estado negativo (0 lógico), es de 0 V a 0.8 V; este pin va a estar constantemente cambiando de estado irregularmente.



Para evitar este tipo de eventos se coloca la resistencia de polarización (R6) permitiendo cerrar el circuito. Lo dicho, se indica a continuación en la siguiente figura con los cálculos realizados utilizando la ley de corrientes de Kirchoff, para ello se utilizará tres casos, para verificar cuál de los valores de resistencia pueden ser más aconsejable utilizar.

Figura 80. Diagrama de conexión para el cálculo de la resistencia R6.



Fuente: El autor.

Caso A:

$$Si, I_R = I_{mic}$$

$$I_R = 25 \text{ mA}$$

$$I_R = \frac{V_{in} - 0}{R_6} = \frac{5}{R_6}$$

$$R_6 = \frac{5}{I_R} = \frac{5}{25 \text{ mA}}$$

$$R_6 = 200 \Omega$$

Caso B:

$$Si, I_R < I_{mic}$$

$$I_R = \frac{I_{mic}}{5} = \frac{25 \text{ mA}}{5}$$

$$I_R = 5 \text{ mA}$$

$$I_R = \frac{V_{in} - 0}{R_6} = \frac{5}{R_6}$$

$$R_6 = \frac{5}{I_R} = \frac{5}{5 \text{ mA}}$$

$$R_6 = 1 \text{ K}\Omega$$

Caso C:

$$\text{Si } I_R \ll I_{mic}$$

$$I_R = \frac{I_{mic}}{50} = \frac{25 \text{ mA}}{50}$$

$$I_R = 0,5 \text{ mA}$$

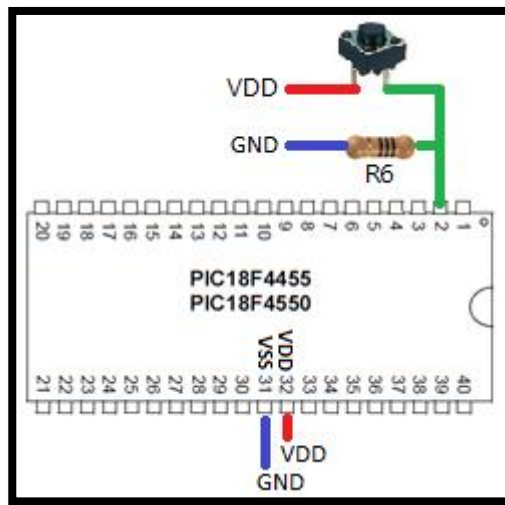
$$I_R = \frac{V_{in} - 0}{R_6} = \frac{5}{R_6}$$

$$R_6 = \frac{5}{I_R} = \frac{5}{0,5 \text{ mA}}$$

$$R_6 = 10 \text{ K}\Omega$$

Mientras mayor sea el valor de la resistencia R6, mayor será el flujo de corriente en el pin del microcontrolador, es decir, al utilizar un valor de resistencia mayor este impide el paso de corriente hacia el terminal común, permitiendo que fluya con mayor intensidad hacia el pin del microcontrolador, por ello se colocará un valor típico que es de 10 KΩ, el cual se encuentra entre los cálculos realizados con el caso C. En la siguiente figura se indica la conexión de este módulo en el circuito.

Figura 81. Diagrama de conexión física entre el microcontrolador y el pulsador



Fuente: El autor.

### 3.10.7 Indicadores de estado del circuito

Este módulo permite visualizar el estado actual del circuito, se utiliza para poder verificar si el aula está siendo utilizada o no físicamente por un docente en un aula determinada.

En la siguiente figura se indica la forma de conexión y se calcula el valor de la resistencia que será necesaria para poder encender un diodo led, tomando en cuenta que cada uno de los diodos led tiene diferentes características, ya que dependiendo del color que se utilice este soporta diferente voltaje y amperaje, por lo tanto se toma un rango general de estos elementos siendo este de 10 a 20mA y de 1.5 a 2.2 V según lo especifica en su datasheet.

Datos:

$$I_{d_{max}} = 20 \text{ mA}$$

$$V_{d_{max}} = 2.2 \text{ V}$$

$$V_m = 5 \text{ V}$$

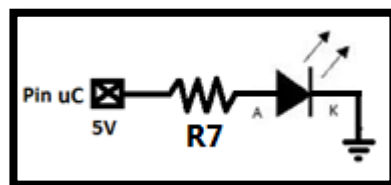
$$R_7 = ?$$

Donde

- $I_{d_{max}}$  = Corriente máxima que circula por el diodo led
- $V_{d_{max}}$  = Voltaje máximo que circula por el diodo led
- $V_m$  = Voltaje del microcontrolador
- $R_7$  = Resistencia a calcular

A continuación se calcula los valores de la resistencia (R7). Para ello se utiliza la ley de tensiones de Kirchhoff, la cual se basa en la igualdad de potenciales eléctricos, es decir, que la suma de las caídas de tensión en los elementos eléctricos de un circuito es igual a la energía suministrada por las fuentes. Lo dicho se indica a continuación con los cálculos realizados

Figura 82. Diagrama de conexión para el cálculo de la resistencia R7.



Fuente: El autor.

$$V_m = V_{d_{max}} + V_{R6}$$

$$V_m = V_{d_{max}} + R_7 * I_{d_{max}}$$

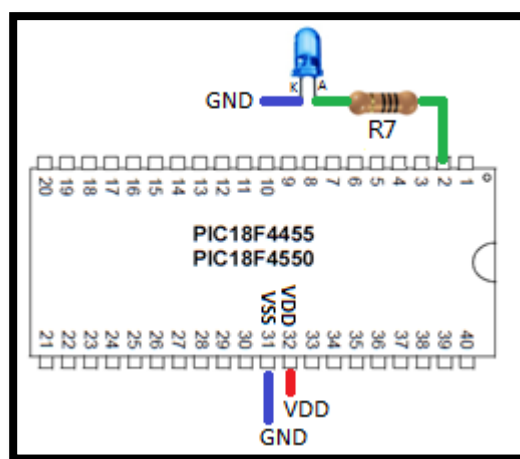
$$R_7 = \frac{V_m - V_{d_{max}}}{I_{d_{max}}}$$

$$R_7 = \frac{5V - 2.2V}{20mA}$$

$$R_7 = 0.14K\Omega = 140\Omega \cong 220\Omega$$

De igual forma que en los casos anteriores, en el mercado comercial no se puede encontrar resistencia del valor que se indica en la ecuación anterior, lo que se podría realizar es: colocar varias resistencias en serie (una a continuación de otra) para conseguir el valor calculado, pero esto llevaría una mayor cantidad de elementos en el circuito, es por ello que se escoge una resistencia aproximada al valor calculado, tal es el caso de la resistencia R7 para reducir en número y sólo colocar una resistencia. En la siguiente figura se indica la conexión para los indicadores del estado del circuito.

Figura 83. Diagrama de conexión física entre el microcontrolador y el indicador Led.



Fuente: El autor.

Cada una de estas configuraciones es necesaria para que el circuito receptor de cada una de las puertas funcione correctamente. La alimentación en ciertos módulos es diferente, para ello se toma en cuenta que las denominaciones: VDD, VCC y GND representan a esta alimentación siendo: VDD un voltaje de 5V, VCC un voltaje de 12V y GND las respectivas conexiones a tierra del circuito, ya que como se pudo apreciar en él se realizan varias funciones.

### **3.11 Dispositivos o elementos a utilizar en el circuito transmisor y receptor**

En este punto se realiza un listado de cada uno de los materiales que el proyecto contendrá, desde un diodo led hasta la cerradura eléctrica, para posteriormente proceder a realizarlo y armarlo para su funcionamiento final.

#### **3.11.1 Dispositivos del circuito transmisor**

En este circuito se utilizan pocos elementos electrónicos, ya que sólo se encargará de recibir o transmitir datos hacia el microcontrolador y desde la interfaz gráfica diseñada más adelante. Este circuito contendrá los siguientes materiales:

- Un módulo USB-USART 5 pines
- Un módulo XBEE Pro S1
- Una resistencia de 220 ohmios ( $\Omega$ )

- Un diodo led.

### 3.11.2 Dispositivos del circuito receptor

La diferencia de este circuito con el circuito transmisor es que: se necesita varios elementos eléctricos, ya que este permite realizar diferentes funciones: desde la visualización de los datos; hasta la apertura propia de la puerta del aula. Este circuito contendrá los siguientes materiales:

#### *Fuente de alimentación y backup:*

- Un transformador
  - Devanado primario a 110VAC
  - Devanado secundario a 3 salidas: 2 de 9VAC y 1 de 18VAC.
  - Corriente de 500mA.
- Un fusible
- Un rectificador de onda completa.
- Cuatro reguladores:
  - Un LM7805 (5 VDC a 1 A).
  - Dos LM317T (3,3VDC a 1 A).
  - Un LM7812 (12 VDC a 1 A)
- Un condensador electrolítico de 1000 microfaradios (uF) a 25 voltios.
- Un condensador electrolítico de 0.1 microfaradios (uF) a 50 voltios.
- Un diodo zener a 3.3 voltios (v)
- Cuatro resistencias:

- Una de 330 ohmios ( $\Omega$ )
- Dos de 10 ohmios ( $\Omega$ )
- Una de 1 kilo ohmio ( $K\Omega$ )
- Tres disipadores de calor.
- Tres diodos 1N4007
- Tres baterías de 9VDC recargables

*Elementos del circuito receptor*

- Un microcontrolador PIC18F4550
- Un módulo XBEE Pro S1
- Un teclado matricial de 4x4
- Una LCD de 16x2
- Un potenciómetro de 10 kilo ohmios ( $K\Omega$ )
- Diez resistencias:
  - Una de 100 ohmios ( $\Omega$ )
  - Dos de 220 ohmios ( $\Omega$ )
  - Cinco de 10 kilo ohmios ( $K\Omega$ )
  - Dos de 2,7 kilo ohmios ( $K\Omega$ )
- Un diodo 1N4007
- Dos transistor 2N3904
- Un Relé de 12 voltios
- Un parlante pequeño (Buzzer) a 5 voltios
- Una cerradura eléctrica de 12voltios.
- Cuatro pulsadores
- Dos diodos LED

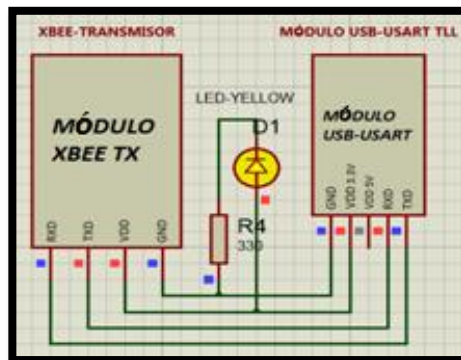


### 3.12 Diseño de los circuitos transmisor y receptor

El diseño de los diferentes circuitos se realizó en un programa que permite simularlo como si fuera físicamente, evitando cometer errores, el cual indica cómo sería el funcionamiento del mismo a la hora de crearlos o armarlos, este programa es Proteus y permite realizar cualquier tipo de proyectos electrónicos.

Para el circuito transmisor se conectan directamente entre los dos módulos y se incrementa un diodo y su respectiva resistencia, para verificar si el módulo está siendo alimentado correctamente o no, dependiendo del estado de dicho módulo, tal como se indica en la figura siguiente.

Figura 84. Diseño del circuito transmisor.

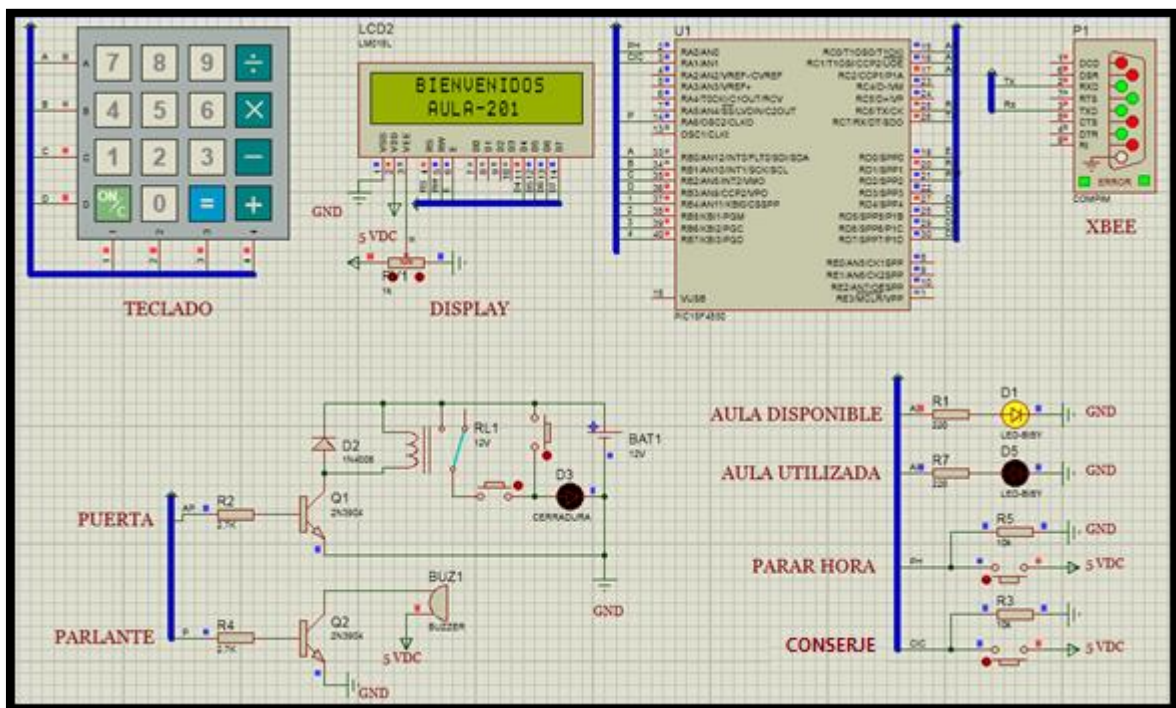


Fuente: El autor (Diseño realizado en el programa Proteus)

En el circuito receptor se conectan todos los elementos electrónicos estudiados anteriormente, donde el proceso de funcionamiento es recibir los datos por medio del módulo XBEE, los mismos que serán recibidos por el

microcontrolador, éste procesa los datos mediante diferentes métodos de programación, para posteriormente visualizarlos en la LCD ingresando datos por el teclado para poder abrir la puerta, adicionalmente, se incorporan pulsadores para poder tener acceso a las aulas, sin tener que usar la interfaz gráfica, este parámetro será utilizado sólo por los conserjes de la facultad, ya que ellos no necesitan registrarse previamente, pero se generan reportes de los ingresos.

Figura 85. Diseño del circuito receptor.

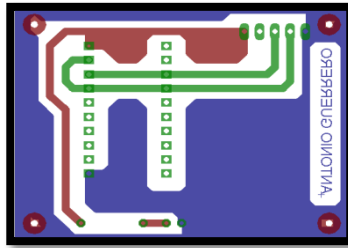


Fuente: El autor (Diseño realizado en el programa Proteus)

### 3.13 Diseño de la placa de los circuitos transmisor y receptor en Eagle

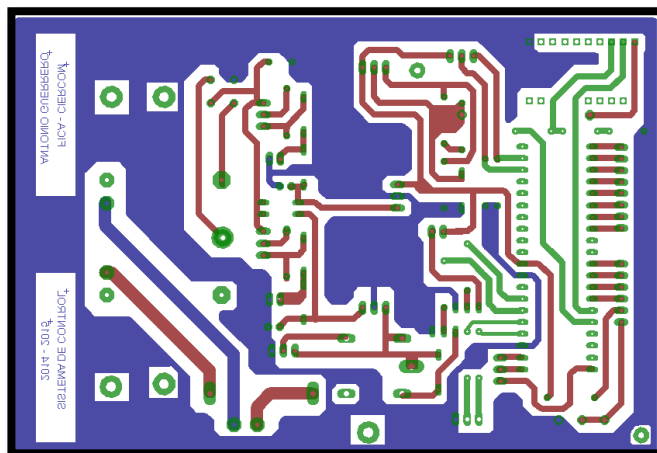
El diseño de las pistas de cada uno de los circuitos se realizó en el programa Eagle, el cual permite diseñar los diagramas electrónicos de un proyecto, para posteriormente montar en una baquelita los dispositivos que ésta contenga. Se decidió utilizar este software para este diseño debido a que tiene una licencia gratuita.

Figura 86. Pista del circuito transmisor.



Fuente: El autor (Diseño realizado en el programa Eagle)

Figura 87. Pista del circuito receptor.



Fuente: El autor (Diseño realizado en el programa Eagle)



El proceso de diseño de estos circuitos se especifica en el anexo C de este trabajo de titulación, desde la impresión de las pistas, así como la distribución de cada elemento en las placas respectivas.

## CAPÍTULO IV

### DISEÑO DEL SOFTWARE

#### 4.1 Introducción

En este capítulo se diseñará la aplicación o “interfaz gráfica” en el programa “Microsoft Visual C#” con su respectiva base de datos realizada en “SQL Server”, los cuales se comunicarán con el hardware (circuito transmisor), diseñado previamente en el capítulo anterior.

Esta interfaz gráfica consta de varios métodos que son necesarios construir en base a los requerimientos de la aplicación, entre ellos se tiene: la autenticación del docente, los diferentes modos de ingreso que se pueden crear; la creación, modificación y eliminación de un docente; permitir reservar un aula en una fecha determinada por el docente; modificar o crear datos de un aula, materia o carrera, ya que, debido al crecimiento de la facultad, el sistema debe permitir incluso dichos parámetros, etc.

También permitirá construir un registro de cada una de las aulas utilizadas, mediante un reporte que será guardado en la base de datos para constancia de la asistencia del docente al aula que le corresponda, y, adicionalmente se podrá

crear un campo que permita utilizar un aula que se encuentre libre, para así aprovechar de los recursos de la facultad y no mantener cerradas las aulas que, por cualquier razón, se encuentren disponibles. Estos requerimientos son tomados en cuenta en base a los problemas que se dan actualmente en la facultad.

Cada uno de estos modos de ingreso al sistema se valida mediante los datos previamente ingresados en la base de datos, es por ello que a continuación se realizará el diseño de la base de datos del sistema, tomando en cuenta estos parámetros.

#### **4.2 Diseño de la base de datos del sistema de ingreso a las aulas**

Para poder diseñar una base de datos se debe tomar en cuenta ciertos parámetros importantes como son:

- El diseño conceptual
- La elección del Sistema de Gestor de Base de Datos (SGBD)
- El diseño Lógico
- El diseño Físico

### 4.2.1 Diseño conceptual

Se trata de representar información de una manera simple y concreta mediante un diagrama denominado Entidad-Relación. Este diagrama tiene ciertas características que se mencionan a continuación:

- Entidad: No necesita de otros parámetros para ser descrito.
- Atributos: Son las características o parámetros propios de una entidad.
- Relación: Es representado mediante una figura geométrica (rombo), la cual identifica un vínculo entre entidades. Este vínculo se denomina cardinalidad de las que se puede menciona tres:
  - De uno a uno (1:1)
  - De uno a muchos (1:n)
  - De muchos a muchos (n:n)

En una base de datos, cada entidad debe contener un atributo clave, el cual es un identificador de la misma.

Tomando en cuenta esta información se procede a realizar el diseño conceptual de la base de datos del proyecto, basándose en los requerimientos que se indican a continuación.



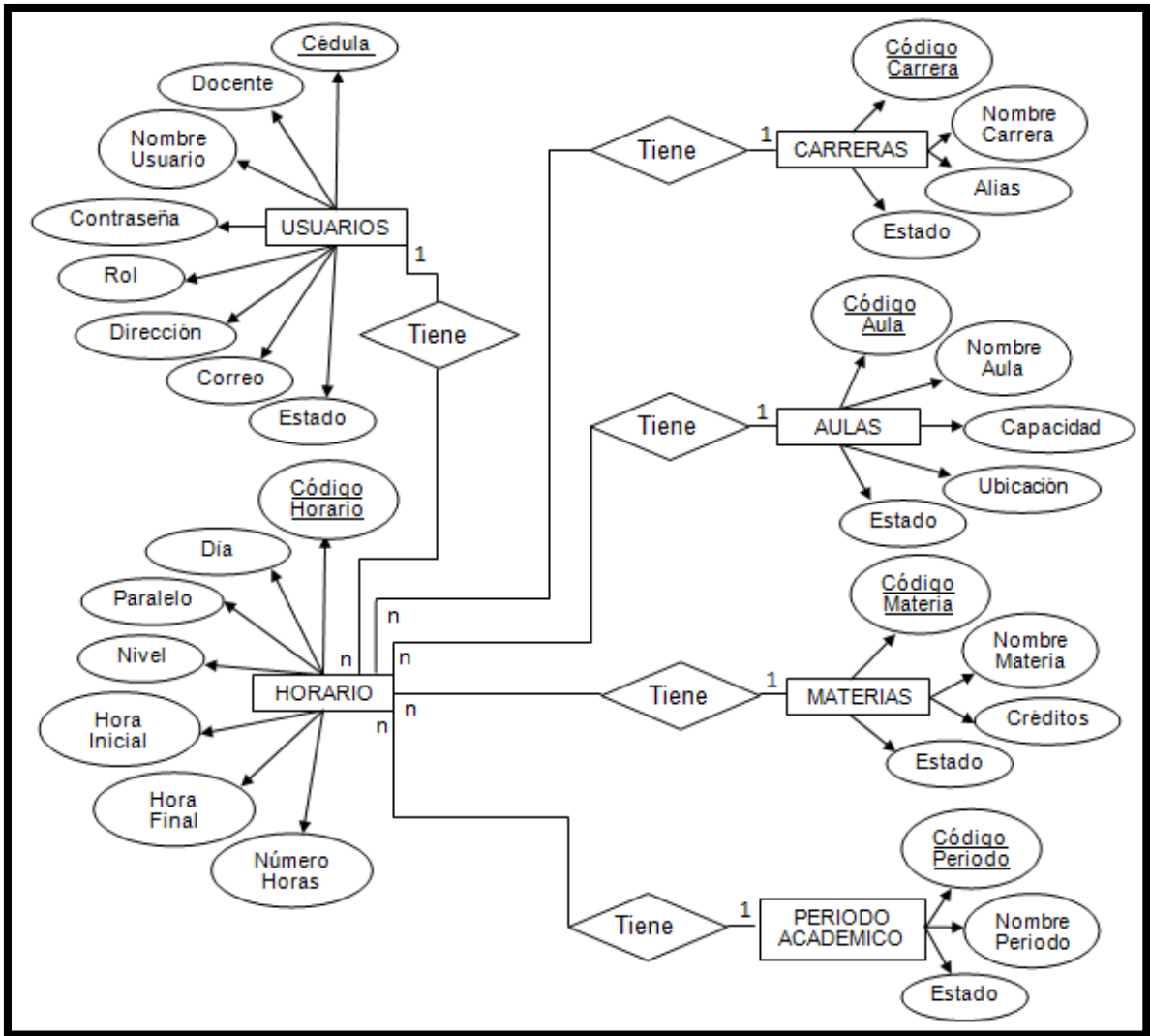
#### **4.2.1.1 Requerimientos para la base de datos del proyecto**

Se desea construir una base de datos que almacene el registro de los docentes o funcionarios que laboran en la facultad, los mismos que realizan diferentes trabajos en las aulas de la misma institución.

Para cada docente o funcionario se desea obtener los siguientes datos: el número de cédula, los nombres completos, la dirección, el teléfono, el rol que desempeña y su correo electrónico, así como los datos del aula al que este funcionario ingrese, la materia que imparta en una carrera determinada y el periodo académico que se encuentre, se debe tomar en cuenta que: debido al periodo académico, este dependerá de un horario al que se deben regir en caso de ser docentes y en caso de recuperación de horas se deberá realizar una reserva de un aula con anticipación.

Este registro debe contener la hora a la que el docente o funcionario ingresa y sale del aula, el número de horas, la fecha, el día y un detalle del trabajo que realizó.

Figura 90. Diseño del modelo conceptual de la base de datos.



Fuente: El autor.

#### 4.2.2 Elección del Sistema de Gestor de Base de Datos (SGBD)

Es un software que permite crear, mantener, utilizar e ingresar información en una base de datos, la elección del mismo depende del modelo en el que se desee trabajar. Estos modelos pueden ser: Relacional, Orientado a objetos,

Objeto relacional y Multidimensional. Siendo el más común y utilizado hoy en día el modelo relacional el cual se aplica en este proyecto.

#### **4.2.2.1 Modelos del Sistema de Gestor de Base de Datos (SGBD)**

Para el modelo relacional existendiferentessoftware, los más comunes de hoy en día son: SQL Server de Microsoft, Interbase de Borland, Oracle y DB2 de IBM.

Cada uno de estos software tienen muy buenas características, por lo que compararlos no sería conveniente, simplemente se decidió utilizar el software “SQL Server” de Microsoft, debido a la facilidad de configuración y amplio conocimiento adquirido. Tomando en cuenta que este software es licenciado, se toman las medidas necesarias, ya que, la licencia de este software fue obtenida por la institución educativa (Universidad Técnica del Norte), y así aprovechar de alguna manera este recurso.

Figura 91. Logo del software SQL Server 2008.



Fuente:<https://nugetgallery.blob.core.windows.net/icons/Microsoft.SqlServer.Types.10.50.1600.1.png>

“SQL Server” es un sistema creado por la empresa Microsoft, basado en un modelo relacional, el cual se basa en la construcción de tablas, las mismas que contienen campos. Los datos que se guardan en estos campos pueden ser recuperados y modificados mediante consultas con el lenguaje SQL (Lenguaje de Consulta Estructurado).

### **4.2.3 Diseño lógico**

En este punto se aplica el modelo al que se basará la base de datos, en este caso el modelo relacional, básicamente se utiliza el diseño conceptual para proceder a crear las diferentes tablas mediante “reglas”. Estas reglas son:

- Cada entidad del diagrama conceptual es una tabla del diseño lógico.
- Las relaciones con cardinalidad de (n: n): se destruyen y se crean como una tabla nueva.
- Las relaciones con cardinalidad de (1: n): se incluye la clave principal de la entidad que contenga la cardinalidad (n), siendo esta entidad parte de la entidad con cardinalidad (1).

Los atributos clave del diseño conceptual son las claves primarias en cada tabla creada en el diseño lógico. Tomando en cuenta esta información se procede a realizar el diseño lógico de la base de datos del proyecto, el cual se indica a continuación.

- Usuario (Cédula, Docente, Nombre Usuario, Contraseña, Rol, Dirección, Correo, Estado)
  - CP: Cédula
- Aulas (Código Aula, Nombre Aula, Capacidad, Ubicación, Estado)
  - CP: Código Aula
- Materias (Código Materia, Nombre Materia, Créditos, Estado)
  - CP: Código Materia
- Carreras (Código Carrera, Nombre Carrera, Alias, Estado)
  - CP: Código Carrera
- Periodo Académico (Código Periodo, Nombre Periodo, Estado)
  - CP: Código Periodo
- Horario (Código Horario, Día, Paralelo Nivel, Hora Inicial, Hora Final, Número Horas, Cédula, Código Materia, Código Periodo, Código Carrera, Código Aula)
  - CP: Código Horario
  - CAj: Cédula (Tabla Usuario); Código Materia (Tabla Materias); Código Periodo (Tabla Periodo Académico); Código Carrera (Tabla Carreras); Código Aula (Tabla Aulas)

Debido a que el sistema será utilizado por otros docentes que no pertenecen a la facultad, es decir, que no tienen un horario de clases con las materias del pénsum de estudio de las diferentes carreras de la facultad, es necesario crear una tabla extra para poder registrar los datos que dichos docentes reserven, ya que, se debe asegurar una fecha precisa para hacer uso

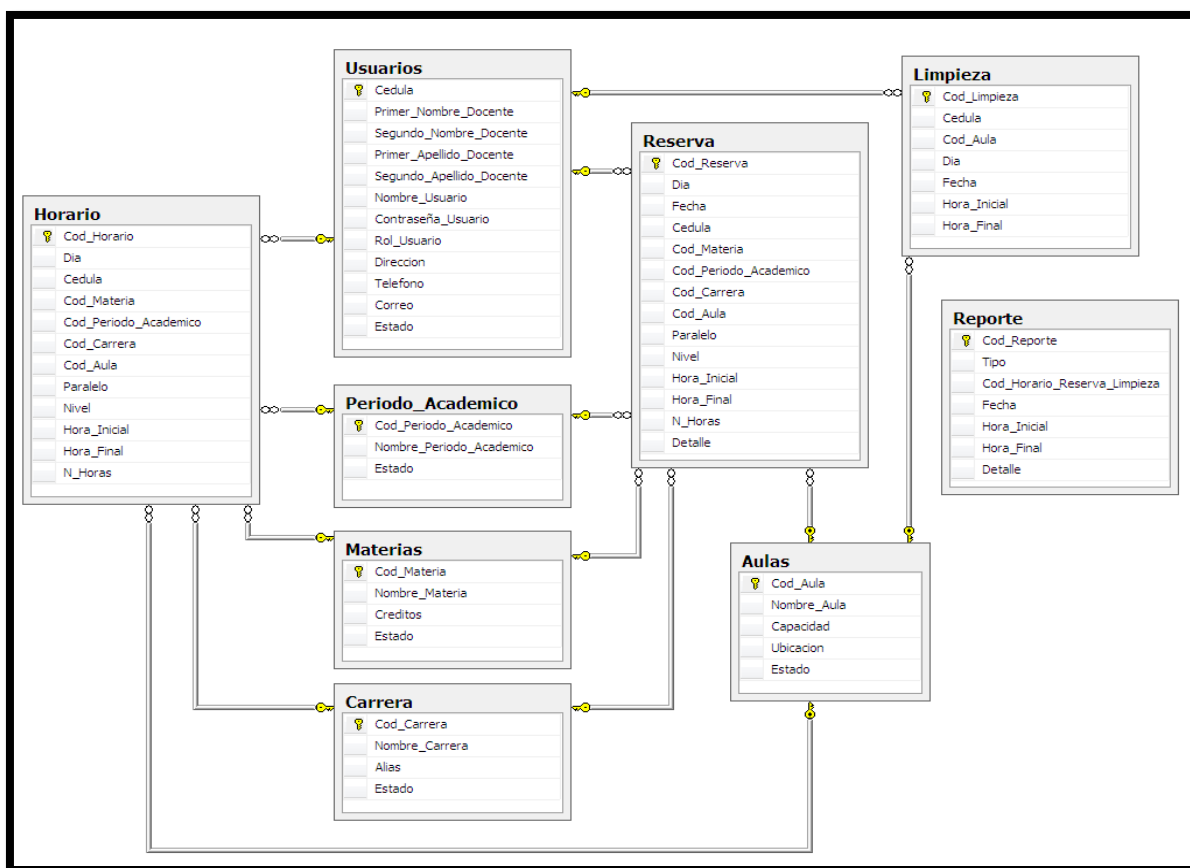
de las aulas y que estos datos sean guardados en la base de datos. Esta tabla se denominará (RESERVA) y de igual forma se aplica para los conserjes cuya tabla se denominará (LIMPIEZA).

- Reserva (Código Reserva, Día, Fecha, Paralelo, Nivel, Hora Inicial, Hora Final, Número Horas, Detalle, Cédula, Código Materia, Código Periodo, Código Carrera, Código Aula)
  - CP: Código Reserva
  - CAj: Cédula (Tabla Usuario); Código Materia (Tabla Materias); Código Periodo (Tabla Periodo Académico); Código Carrera (Tabla Carreras); Código Aula (Tabla Aulas)
- Limpieza (Código Limpieza, Día, Fecha, Hora Inicial, Hora Final, Cédula, Código Aula)
  - CP: Código Limpieza
  - CAj: Cédula (Tabla Usuario); Código Aula (Tabla Aulas)

#### **4.2.4 Diseño físico**

En este punto se procede a tomar la información obtenida en el diseño lógico y utilizarla en el software elegido, creando así, las tablas correspondientes con cada uno de sus atributos y relaciones entre las ellas. En la siguiente figura se indica el resultado final de la creación de las tablas de la base de datos finalizada, para proceder a utilizar en el software que se diseña más adelante.

Figura 92. Diseño físico de la base de datos



Fuente: El autor (Diseño realizado en el programa SQL Server)

### 4.3 Diseño de la interfaz gráfica del sistema de ingreso a las aulas

El diseño de la interfaz gráfica de este proyecto está basado dependiendo del uso que se le dé a las diferentes aulas, para ello se cuenta con diferentes modos de ingreso a los cuales se les denominan:

- Administrador
- Usuario
- Eventual

*Administrador:* Es el encargado de gestionar la aplicación y la base de datos según sea necesario, el cual permitirá realizar acciones como eliminar, crear, modificar, ya sea los datos de un docente como los datos de las aulas y materias, permitiéndole cambiar diferentes parámetros.

Figura 93. Interfaz gráfica del modo Administrador.

AUTENTICACIÓN

**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FICA - CIERCOM**  
**SISTEMA DE CONTROL DE ACCESO**

**Modo de ingreso**

- Administrador
- Usuario
- Eventual

**Datos del Sistema**

Fecha 01/06/2015

Día LUNES

Hora Actual 10:09:51

**Autenticación**

Ingreso Usuario

Ingreso Contraseña

Confirmar Contraseña

LIMPIAR INGRESAR

Fuente: El autor.

*Usuario:* Es aquel que tendrá ciertos permisos en la aplicación, es decir, sólo se le permite modificar y verificar los datos propios, así como ver las aulas que se encuentran disponibles en ese momento y utilizar una aula disponible en caso de necesitarla fuera de su horario normal.



Figura 94. Interfaz gráfica del modo Usuario.

The screenshot shows a window titled 'AUTENTICACIÓN' for the 'UNIVERSIDAD TÉCNICA DEL NORTE FICA - CIERCOM SISTEMA DE CONTROL DE ACCESO'. It features three radio buttons for login mode: 'Administrador', 'Usuario' (selected), and 'Eventual'. To the right, 'Datos del Sistema' shows 'Fecha' as 01/06/2015, 'Dia' as LUNES, and 'Hora Actual' as 10:10:28. The 'Autenticación' section has two input fields for 'Ingreso Usuario' and 'Ingreso Contraseña', with 'LIMPIAR' and 'INGRESAR' buttons below.

Fuente: El autor.

*Eventual:* Es aquel que puede tener las diferentes aulas que se encuentran disponibles u ocupadas en un periodo determinado.

Figura 95. Interfaz gráfica del modo Eventual.

The screenshot shows the same 'AUTENTICACIÓN' window, but with the 'Eventual' radio button selected. The 'Datos del Sistema' section now shows 'Hora Actual' as 10:10:59. The 'Autenticación' section has a single input field for 'Ingreso C.I.' and an 'INGRESAR' button below it.

Fuente: El autor.

### **4.3.1 Prioridades de los modos de ingreso al sistema de control**

Las prioridades de cada uno de los modos de ingreso se asignan dependiendo del trabajo que cada uno de los docentes realizan, es por ello que a continuación se describen brevemente cada una de estas.

#### **4.3.1.1 Prioridad del modo Administrador.**

- Ingresar a un nuevo docente o funcionario
- Buscar, modificar y eliminar los datos de un docente o funcionario
- Ingresar o modificar parámetros de las diferentes aulas y materias impartidas en las diferentes carreras de la facultad.
- Crear horarios de clases basándose a los parámetros de distribución de aulas impartidas en la facultad.
- Modificar los datos personales del docente según su registro.
- Utilizar un aula asignada correspondiente al horario designado o reservada previamente.
- Reservar un aula disponible en un determinado tiempo.
- Ver un mapa de aulas disponibles o utilizadas.
- Verificar los reportes de cada uno de los docentes que ingresaron a las diferentes aulas.

#### **4.3.1.2 Prioridad del modo Usuario.**

- Modificar los datos personales del docente según su registro.

- Utilizar un aula asignada correspondiente al horario designado o, un aula reservada previamente.
- Reservar un aula disponible en un determinado tiempo.
- Ver un mapa de aulas disponibles o utilizadas.

#### **4.3.1.3 Prioridad del modo Eventual.**

- Reservar un aula disponible en un determinado tiempo.
- Utilizar un aula reservada previamente.
- Ver un mapa de aulas disponibles o utilizadas.

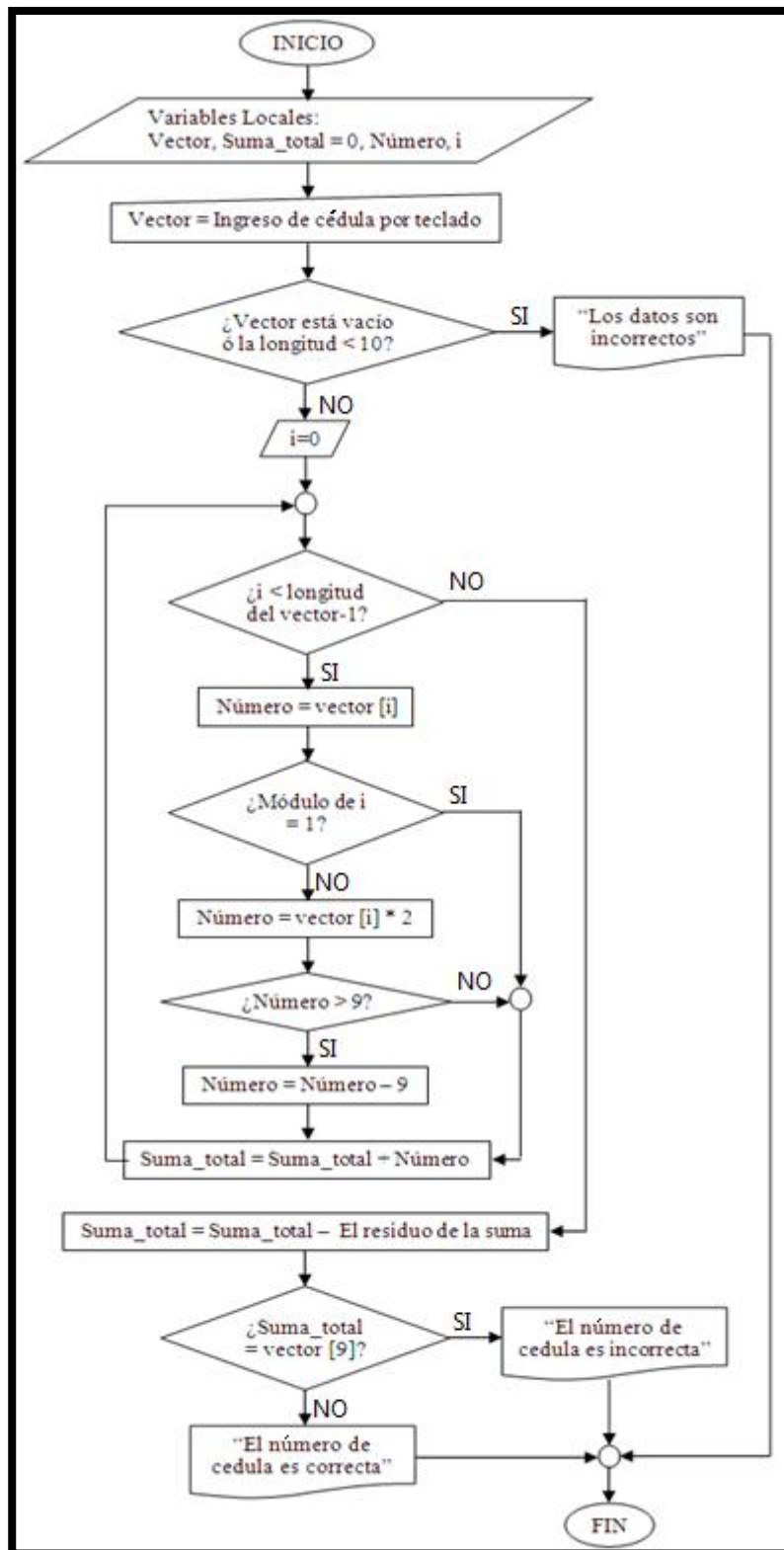
### **4.3.2 Proceso del diseño de las interfaces según su prioridad**

En este punto se analizan los requerimientos de las prioridades mencionadas anteriormente, para ello es necesario crear diagramas de flujo de los métodos que se creen en cada uno de ellos, si así lo amerite, para posteriormente pasarlo a código.

#### **4.3.2.1 Proceso para la creación de un nuevo docente.**

Para ingresar al sistema un nuevo docente o funcionario es necesario crear un método que permita la validación de una cédula, ya que no se puede ingresar datos que no correspondan a este parámetro, este método se explica en un diagrama de flujo, sobre el proceso que se realiza para verificar si una cédula es correcta o no.

Figura 96. Diagrama de flujo del método validación de cédula



Fuente: El autor.

Además del método anteriormente mencionado es necesario asegurarse que los campos contengan solo números o solo letras según lo amerite, como en el número de cédula y teléfono así como los nombres y apellidos respectivamente. Estos parámetros se pueden lograr gracias a las propiedades que facilita el uso del programa Microsoft Visual C# al crear clases para reutilizar el código escrito y ser llamadas cuando se lo requiera.

Figura 97. Interfaz gráfica de la creación de un nuevo docente

The screenshot shows a web application interface for managing teachers. At the top, there is a title bar 'ADMINISTRADOR' and a menu with 'SALIR' and 'REPORTES'. Below that is a navigation bar with 'Nuevo Docente' selected, and other options: 'Buscar Modificar y Eliminar Docentes', 'Crear, Modificar Materia Aula o Carrera', 'Crear Horario', 'Modificar Datos Personales', and 'Utilizar Al...'. The main content area is titled 'DATOS DEL DOCENTE' and is divided into several sections:

- Datos Personales:** Includes fields for 'C.I. Docente' (with a 'VALIDAR' button), 'Primer\_Nombre\_Docente', 'Segundo\_Nombre\_Docente', 'Primer\_Apellido\_Docente', 'Segundo\_Apellido\_Docente', 'Dirección', 'Teléfono\_Celular\_Principal', and 'Correo' (with a '@utn.edu.ec' suffix).
- Identificación de Registro:** Includes fields for 'Nombre\_Usuario' and 'Contraseña\_Usuario'.
- Rol Usuario:** Includes radio buttons for 'Administrador', 'Usuario', 'Eventual', and 'Conserje', along with the logo of 'UNIVERSIDAD TÉCNICA DEL NORTE'.
- Información:** A text box with the instruction: 'Asegúrese que los campos se encuentren ingresados correctamente para poder validar los datos'.
- Controles:** Includes 'GUARDAR' and 'LIMPIAR' buttons, and the logo for 'CIERCOM INGENIERÍA EN ELECTRONICA Y REDES DE COMUNICACIÓN'.

At the bottom of the interface, there is a banner image of a building with the text 'Facultad de Ingeniería en Ciencias Aplicadas'.

Fuente. Al autor.

### 4.3.2.2 Proceso para la búsqueda, modificación y eliminación de un docente.

En este parámetro se crean los métodos de: búsqueda, modificación y eliminación de un docente, dichos métodos son realizados mediante consultas entre el software y la base de datos. Es necesario mencionar que: para hacer una modificación o eliminación de un docente, en primera instancia, se debe buscar al docente para decidir cuál de las dos opciones (Modificar o Eliminar) realizar.

Figura 98. Interfaz gráfica de la búsqueda, modificación y eliminación de un docente

The screenshot shows a web application interface for managing teachers. The title bar reads "ADMINISTRADOR". The main menu includes "SALIR" and "REPORTES". The navigation bar contains "Nuevo Docente", "Buscar Modificar y Eliminar Docentes", "Crear, Modificar Materia Aula o Carrera", "Crear Horario", "Modificar Datos Personales", and "Utilizar Au...".

The interface is divided into several sections:

- Buscar Docente por:** Includes radio buttons for "C.I.\_Docente", "Nombre\_Docente", and "Apellido\_Docente", each followed by a text input field. A "LIMPIAR" button is located below these fields.
- Opciones:** Includes radio buttons for "Buscar", "Modificar", and "Eliminar". A large blue question mark icon is positioned to the right of these options.
- Datos Personales:** Includes text input fields for "C.I.\_Docente", "Primer\_Nombre\_Docente", "Segundo\_Nombre\_Docente", "Primer\_Apellido\_Docente", "Segundo\_Apellido\_Docente", "Dirección", "Teléfono\_Celular\_Principal", and "Correo".
- Identificación de Registro:** Includes text input fields for "Nombre\_Usuario", "Contraseña\_Usuario", and "Rol\_Usuario" (a dropdown menu), and a dropdown menu for "Estado".
- Resultados de la Búsqueda:** A large gray rectangular area representing the search results table.

At the bottom of the interface, there is a text prompt: "Seleccione un elemento de la tabla para cargar los datos del docente" and an "ACEPTAR" button. There are also "OK" and "CANCELAR" buttons located near the registration fields.

Fuente: El autor.

El proceso de eliminación no se hace completamente, es decir, en el caso de eliminar un docente no se lo realiza, sino que se pone al docente en un estado de “Inactivo”, ya que en caso de requerir los servicios del docente sólo sea necesario “Activarlo” para que este pueda cumplir con las tareas asignadas. Este punto se lo realiza de esta manera debido a que en ciertas ocasiones un docente puede regresar a impartir clases en la facultad nuevamente.

#### **4.3.2.3 Proceso para Ingresar o modificar parámetros de las diferentes aulas y materias.**

Este parámetro es necesario crearlo debido a que la facultad tiende a crecer en infraestructura, por lo que se pueden crear diferentes aulas como se puede constatar actualmente, ya que cuenta con un piso más en las edificaciones de la misma.

Respecto al parámetro de las materias, conforme pasa el tiempo, las mallas de cada una de las carreras que se imparte en la facultad cambian, es por ello que se decidió crear este parámetro, para poder aumentar materias que conforme pase el tiempo se creen y así poder modificar según sea el caso.

Figura 99. Interfaz gráfica de la creación y modificación de un aula materia o carrera

ADMINISTRADOR

SALIR REPORTE

Nuevo Docente | Buscar Modificar y Eliminar Docentes | **Crear, Modificar Materia Aula o Carrera** | Crear Horario | Modificar Datos Personales | Utilizar Au

### CREAR O MODIFICAR AULAS, MATERIAS O CARRERAS

**Datos del Aula**  
 Crear  Modificar  
Nombre del Aula  \_   
Ubicación   
Capacidad de Estudiantes

**Datos de la Materia**  
 Crear  Modificar  
Nombre de la Materia   
Número de Créditos

**Datos de la Carrera**  
 Crear  Modificar  
Nombre de la Carrera   
Alias de la Carrera

**CIERCOM**  
INGENIERÍA EN ELECTRÓNICA  
Y REDES DE COMUNICACIÓN

**Información**  
Asegúrese que los campos se encuentren ingresados correctamente para poder validar los datos

Fuente: El autor.

#### 4.3.2.4 Proceso para crear horarios de clases

En este punto se crea el horario conforme al distributivo de aulas proporcionado por el departamento de informática de la universidad en el periodo académico que corresponda. En éste, se llenan los campos con la información proporcionada por dicho departamento, cabe mencionar que los datos que son necesarios para utilizar este parámetro ya se encuentran en la base de datos o pueden ser llenados utilizando la casilla anterior del sistema, que es la creación



de una nueva aula, materia o carrera si fuese necesario, por lo que sólo se debe seleccionar los campos para crear el horario de un determinado docente.

Figura 100. Interfaz gráfica de la creación del horario de clases

ADMINISTRADOR

SALIR REPORTES

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | **Crear Horario** | Modificar Datos Personales | Utilizar Au...

### HORARIO ASIGNADO SEGÚN EL DEPARTAMENTO DE INFORMÁTICA DE LA UTN

**Opciones**

- Nuevo
- Modificar
- Duplicar

**Acción**

- [Dropdown] [OK]

- [Dropdown]

**Datos del Horario**

Día [Dropdown] Carrera [Dropdown] Aula [Dropdown]

Docente [Dropdown] Paralelo [Dropdown]

Materia [Dropdown] Nivel [Dropdown]

**Horas**

Inicio [Dropdown]

Final [Dropdown]

Nº Horas [Input]

**CREAR** **MODIFICAR** **LIMPIAR**

**Horarios del Periodo Académico**

[Empty Table Area]

Total de registros [Input] Seleccione un elemento de la tabla para cargar los datos **CARGAR**

Fuente: El autor.

Los procesos descritos hasta este momento son prioridad designada sólo para el administrador, ya que él será el único que tenga acceso a esta información. De aquí en adelante los siguientes procesos serán utilizados en los modos de acceso administrador y usuario.

#### 4.3.2.5 Proceso para modificar los datos personales

Este campo será necesario en caso de que los datos del docente que se autentique deban ser actualizados o cambien, es decir, si se cambia el teléfono, la dirección o si desea cambiar incluso su usuario y contraseña para el ingreso al sistema, etc., esto lo pueda hacer sin tener que requerir los servicios o la presencia del administrador que le pueda ayudar con las modificaciones.

Figura 101. Interfaz gráfica de la modificación de datos personales



ADMINISTRADOR

SALIR REPORTES

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar AL

### MODIFICAR DATOS PERSONALES Y DE REGISTRO DEL SISTEMA

<b>Datos Personales</b>		<b>Identificación de Registro</b>	
C.I._Docente	1003788401	Nombre_Usuario	tonny
Primer_Nombre_Docente	ANTONIO	Contraseña_Usuario	5
Segundo_Nombre_Docente	BOLÍVAR	Rol_Usuario	Administrador
Primer_Apellido_Docente	GUERRERO		
Segundo_Apellido_Docente	ASEICHA	<b>Control</b>	OK
Dirección	MIGUEL SANCHEZ Y TOBIÁS MENA		
Teléfono_Celular_Principal	0981162999		
Correo	abguerreroa@utn.edu.ec		

CIERCOM  
INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

UNIVERSIDAD TÉCNICA DEL NORTE  
IBARRA - ECUADOR

Facultad de Ingeniería en Ciencias Aplicadas

Fuente: Al autor.

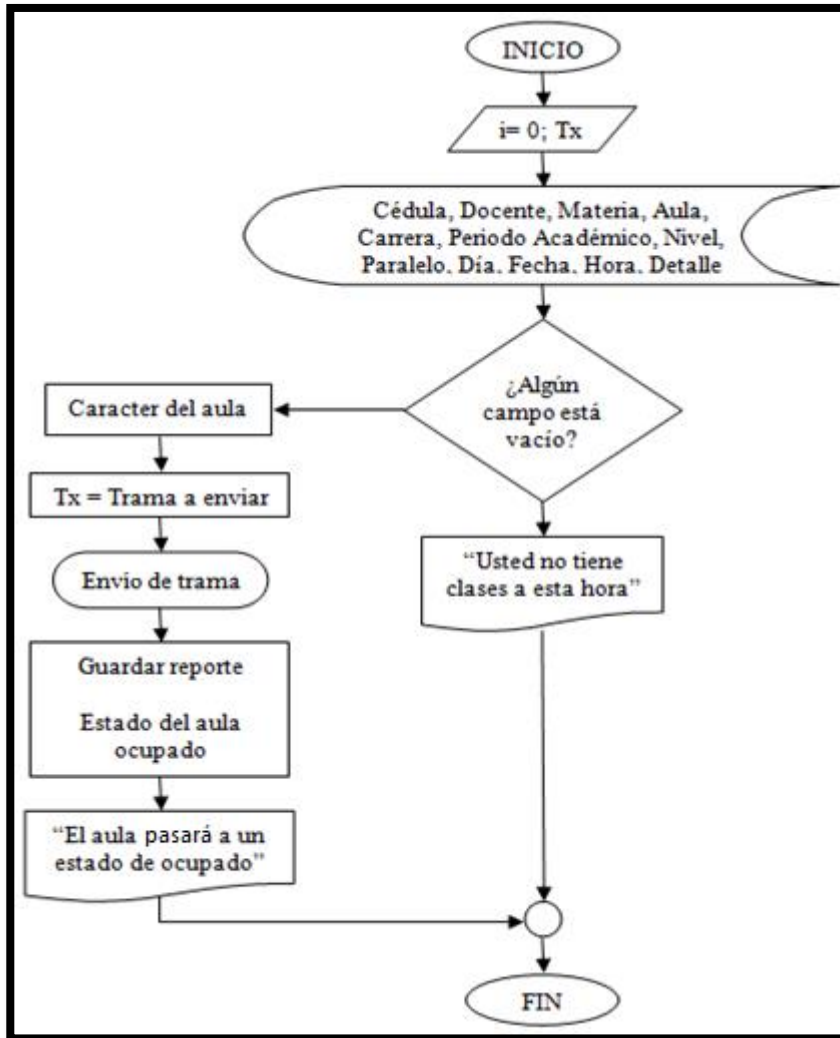
#### **4.3.2.6 Proceso para utilizar un aula asignada o reservada**

Según el docente que se autentique en el modo de ingreso, el sistema cargará los datos del mismo en caso de tener clases a una hora determinada, ya sea por el horario de clases o por una reserva del aula realizada con anterioridad en el sistema.

Inicialmente el sistema forma la trama a enviar con los datos cargados de la base de datos previamente y procede a enviar estos datos hacia el circuito receptor para poder utilizar el aula, caso contrario, si los campos de este proceso no son cargados con la información correspondiente, este campo no es válido para el docente, ya que si se autentica en una hora diferente al horario o la reserva, el sistema verificará que no tiene clases el docente e indica un mensaje con esta información, por lo que puede proceder a utilizar el campo siguiente si desea reservar una aula.

El proceso del envío de datos hacia el circuito receptor se especifica en el siguiente diagrama de flujo ya que en éste se forma la trama que será transmitida por el circuito transmisor hacia el respectivo circuito receptor al cual corresponda dicha trama.

Figura 102. Diagrama de flujo del envío de datos desde la interfaz gráfica



Fuente: El autor.

Para transmitir la trama hacia el circuito receptor, dependiendo del aula que se encuentre asignado o reservado por el docente, se agrupa los parámetros especificados en el capítulo anterior del formato de trama, formando así una cadena de caracteres y posteriormente, guardados los datos en un reporte previo, pasar el aula a un estado de ocupado.

Figura 103. Interfaz gráfica de la utilización del aula



Fuente: El autor.

#### 4.3.2.7 Proceso para utilizar un aula disponible

Este parámetro es importante ya que, en ocasiones un docente necesita utilizar un aula en un horario diferente al asignado por el distributivo, es por eso que puede utilizar este parámetro justificando el motivo por el cual realiza esta acción. También se utiliza para docentes que no sean parte de la facultad y deseen impartir clases en la misma, tomando en cuenta que dicho docente esté registrado previamente como un docente eventual.

Figura 104. Interfaz gráfica de la reserva del aula

ADMINISTRADOR

SALIR REPORTES

Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Aula Asignada o Reservada | Reservar Aula Disponible | Ver

### UTILIZAR AULA DISPONIBLE

**Datos del Aula**

Número de Aula

Ubicación

Capacidad de Estudiantes

**Datos Docente**

C.I. Docente 1003788401

Docente GUERRERO ASEICHA ANTONIO

Rol\_Usuario Administrador

Detalle:

MOTIVO DE LA RESERVA:

**Parámetros de Reserva**

Día

Fecha de Reserva 25/ 11/ 2015

Periodo Académico SEPTIEMBRE 2014 - FEBRERO 2015

Materia

Carrera  Nivel  Paralelo

**Tiempo**

Número de Horas

Hora Actual 20:05:52

Hora Inicial

Hora Final 00:00:00

**Controles**

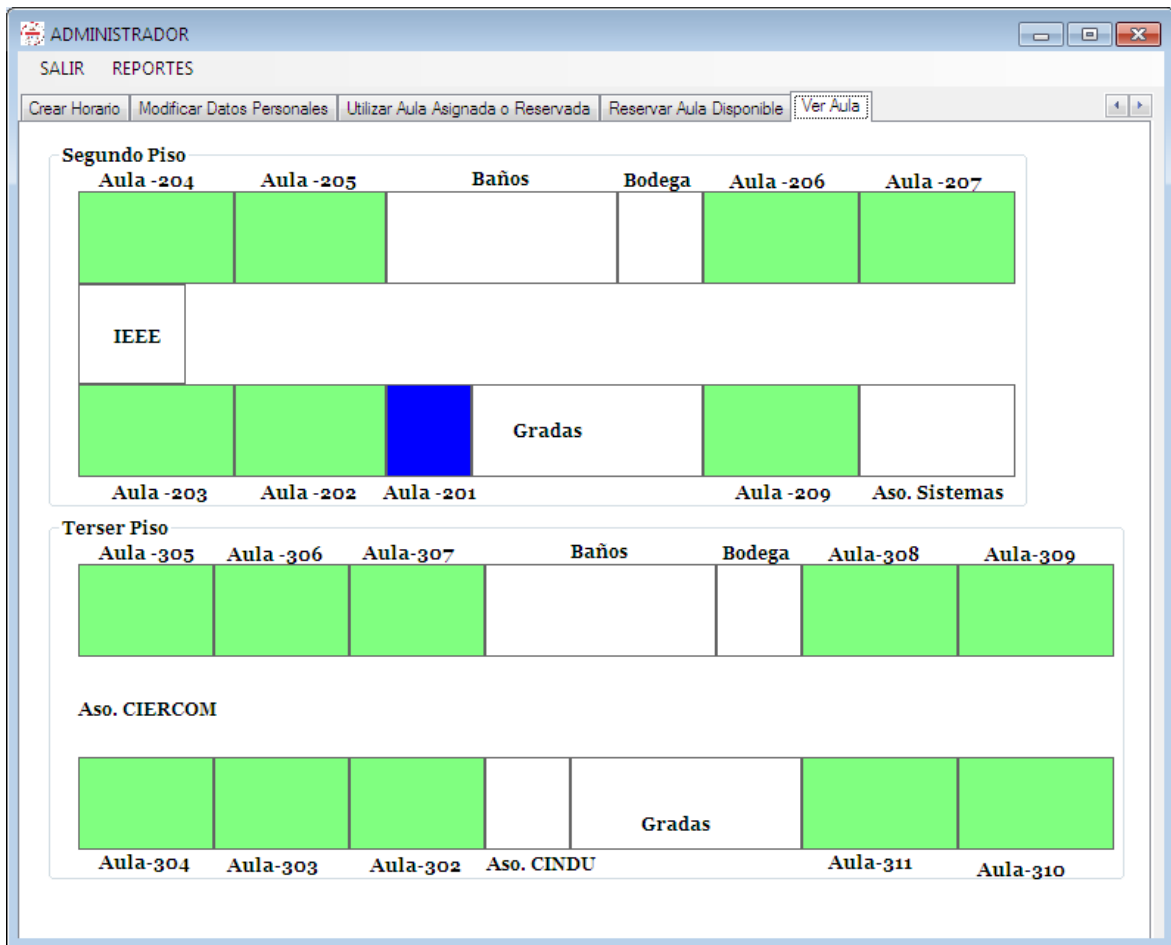
LIMPIAR OK

Fuente: El autor.

#### 4.3.2.8 Proceso para ver un mapa de aulas

Este campo permite identificar un aula ocupada o disponible gráficamente sin la necesidad de una consulta previa, en ésta se presentan las aulas de la facultad en un estado de “Disponible” en un color verde y en estado de “Ocupado” en un color azul.

Figura 105. Interfaz gráfica del estado de las aulas de la facultad



Fuente: El autor.

#### 4.4 Programación de la interfaz gráfica

La programación de la interfaz gráfica se especifica en el anexo D del presente trabajo de titulación, en él se aprecia: la creación de clases, métodos, subrutinas etc. para realizar las funciones descritas anteriormente, las mismas que fueron relacionadas conforme se explica en los diagramas de flujo anteriormente indicados y tomando en cuenta los requerimientos del proyecto, este código fue elaborado en el programa “Microsoft Visual Studio C#”.

#### **4.5 Manuales del sistema de control de acceso**

Este trabajo de titulación al ser una aplicación útil para diferentes personas que laboran en la facultad, cuenta con manuales de uso, ya sea del software (interfaz gráfica), como del hardware (circuitos electrónicos), los mismos que se encuentran en los anexos E y F respectivamente de dicho trabajo con cada uno de los pasos a seguir y recomendaciones, para el correcto funcionamiento de esta aplicación.



## **CAPÍTULO V**

### **PRUEBAS Y CORRECCIÓN DE ERRORES**

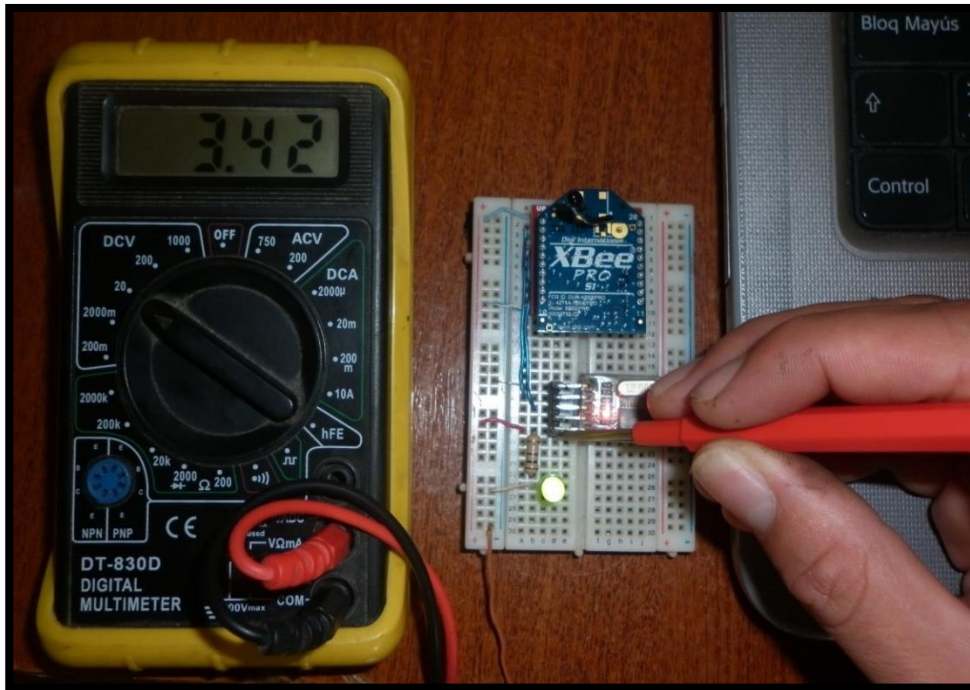
#### **5.1 Introducción**

En este capítulo se realizan las pruebas necesarias para garantizar el correcto funcionamiento, ya sea de conectividad entre módulos inalámbricos que especifica la topología de red así como también del sistema electrónico y verificar la información almacenada en la base de datos, haciendo uso de las diferentes aulas de la facultad mediante este sistema, con el objetivo de generar un reporte de las funciones que se han realizado en dichas aulas.

#### **5.2 Interfaz electrónica encargada de la adquisición y procesamiento de datos**

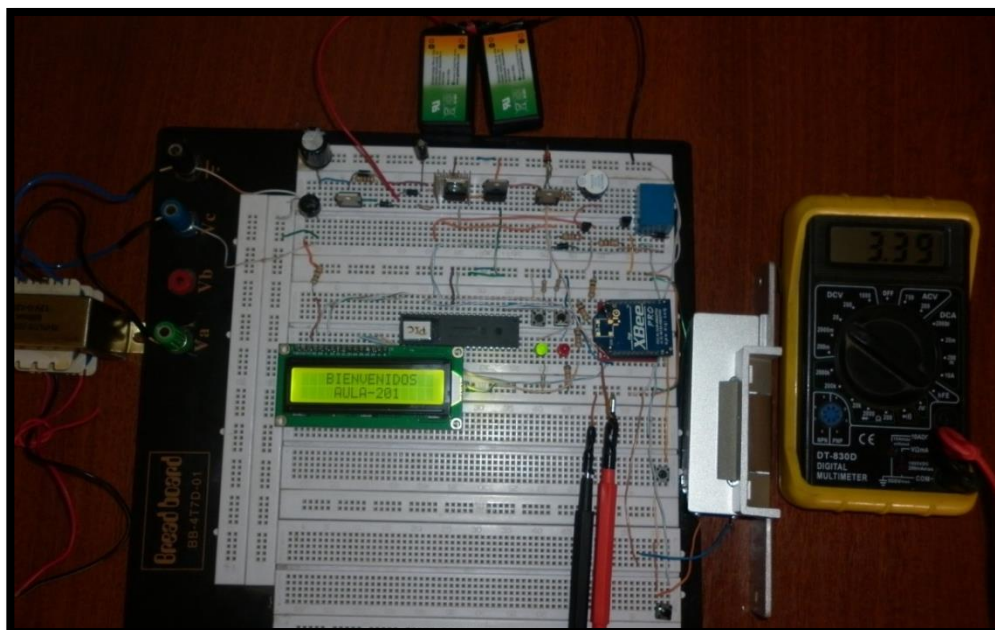
Para realizar las pruebas de la interfaz electrónica se desarrollaron primeramente los denominados circuito transmisor y circuito recepción, siguiendo las configuraciones mencionadas en el capítulo II, para posteriormente verificar la conectividad. En la figura siguiente se indica los circuitos anteriormente mencionados antes de ser ensamblados en una placa electrónica.

Figura 106. Pruebas eléctricas del circuito transmisor.



Fuente: El autor.

Figura 107. Pruebas eléctricas del circuito receptor.

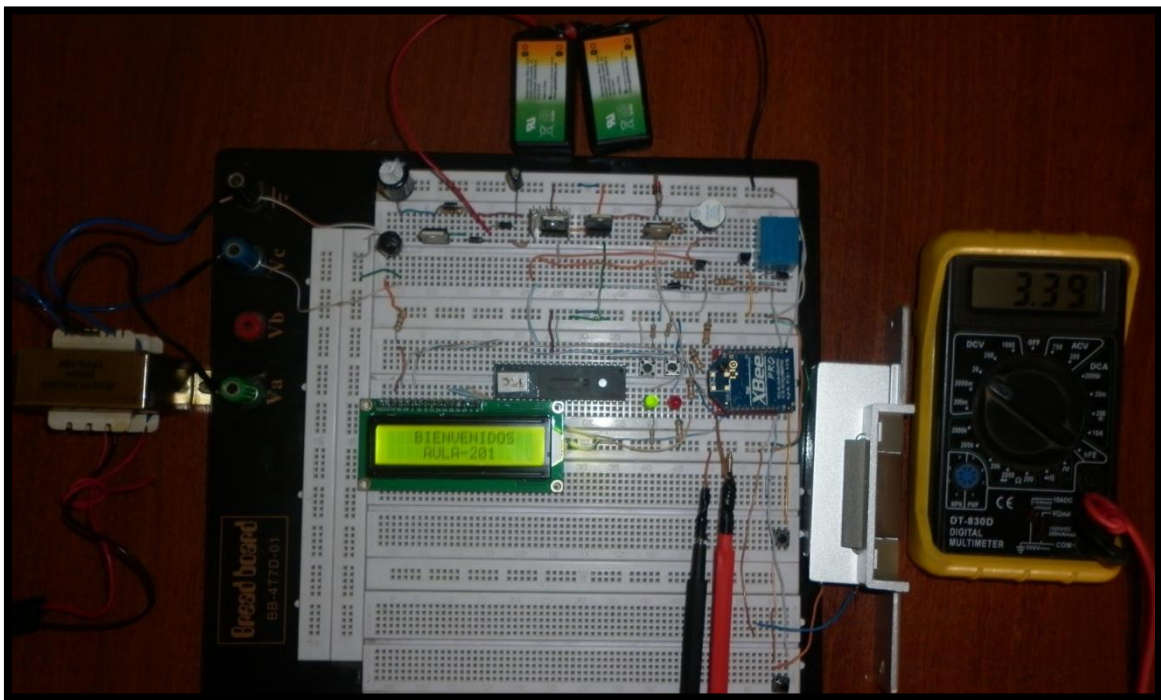


Fuente: El autor.

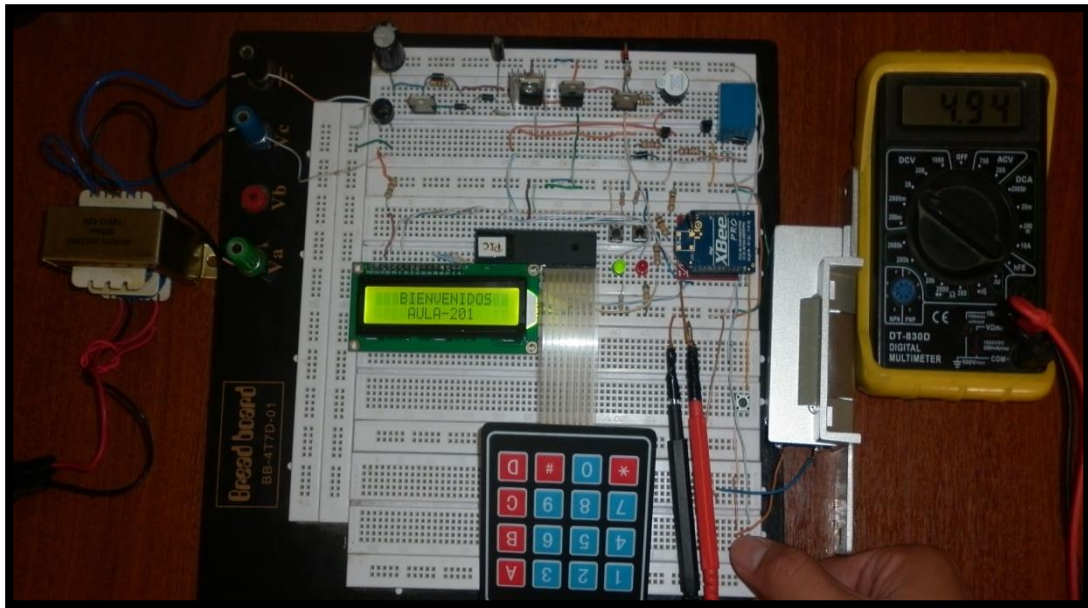
Resultados:

Normalmente el circuito receptor, conectado con las baterías y la fuente alterna, no presenta problemas tal como se puede apreciar en las figuras anteriores, sin embargo, al realizar las pruebas de apertura de la puerta (presionando el pulsador que abre la puerta desde adentro del aula) la cerradura eléctrica genera un pico de energía, por lo que el circuito presenta un reinicio involuntario debido a este evento. Este problema se puede apreciar en las figuras siguientes.

Figura 108. Errores eléctricos del circuito receptor.



a) Circuito trabajando normalmente.



b) Error provocado por la apertura de la puerta.

Fuente: El autor.

Debido a que las cerraduras eléctricas necesitan de un mayor consumo de energía lo cual provoca que el circuito receptor tenga caídas de tensión significativas, para el correcto funcionamiento, se procederá a realizar una configuración adicional en el circuito receptor.

### **5.2.1 Corrección de errores de la interfaz electrónica encargada de la adquisición y procesamiento de datos**

La opción que se adquiere para corregir este error es separar en dos partes al circuito receptor referente a la fuente de alimentación, es decir, se realizará dos fuentes de alimentación con un solo transformador, una para la parte de control y otra para la parte mecánica. Siendo:

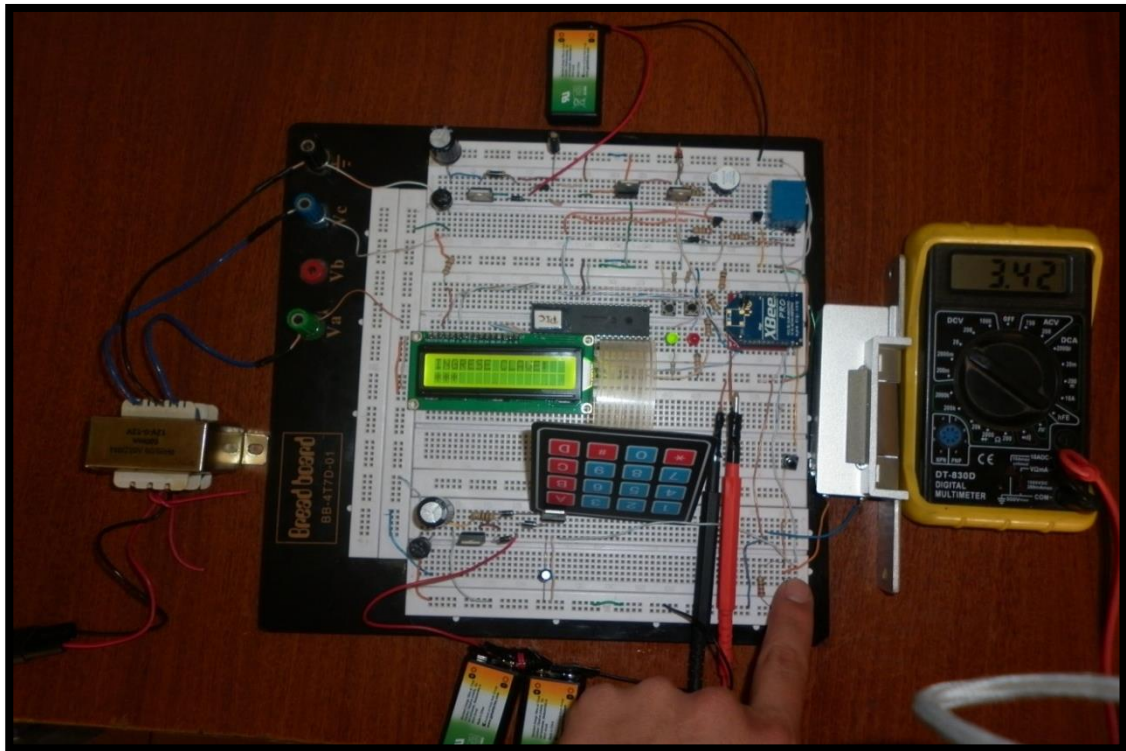
- La parte de control, los dispositivos electrónicos encargados de: La seguridad del sistema, la información del tiempo restante en el aula, los indicadores de estado del circuito, la visualización, el procesamiento, la recepción y transmisión de datos.
- La parte mecánica, los dispositivos electrónicos encargados de: El mecanismo de apertura de la puerta.

En la figura 51 se puede apreciar la fuente de alimentación del circuito receptor, en el cual, se observa los dos sistemas que backup de energía (cargador), de lo cual el segundo, que es el de control mecánico, es opcional para el sistema, ya que dicho control se refiere a la parte mecánica, y lo que se necesita controlar y tener respaldo de energía de mayor interés es la parte de control, para no perder conectividad con el software y permanecer comunicados continuamente. Sin embargo para la demostración de este proyecto se contará con los dos backup de energía para demostrar el correcto funcionamiento.

A continuación se procede a verificar el circuito receptor con dicha modificación y comprobar que el sistema funcione correctamente, en la figura siguiente se puede apreciar lo mencionado.



Figura 109. Pruebas eléctricas del circuito receptor modificada la fuente de alimentación.



Fuente: El autor.

#### Resultados:

Luego de realizar las pruebas respectivas tomando lecturas de niveles de voltaje y corriente, los circuitos transmisor y receptor ya no presentaron problemas, logrando ser totalmente operativos eléctricamente. Cabe recalcar que la toma de datos no se realiza hasta este punto, ya que primeramente deben ser configurados los módulos XBEE, y para que estos dispositivos funcionen correctamente deben contar con una buena alimentación.

### 5.3 Pruebas de conectividad en la red

Luego de haber obtenido resultados positivos en la configuración eléctrica y sin presentar problema alguno, se procedió a realizar las pruebas de conectividad en la red, para lo cual se tuvo que realizar pruebas en la edificación de la facultad, de tal forma que fue necesario utilizar un software que permita visualizar los datos que se envían y reciben desde un módulo XBEE hacia otro, dicho software es el denominado XCTU<sup>7</sup> el mismo que es proporcionado por el fabricante de dichos módulos.

#### 5.3.1 Configuración de los módulos XBEE.

En este punto se procede a utilizar el software XCTU, el cual permite cargar los parámetros de configuración de un módulo para posteriormente utilizar en la red WPAN. En la tabla siguiente se especifican los parámetros necesarios para la convergencia en la red.

Tabla 7. Parámetros de configuración de los módulos XBEE Pro de los diferentes circuitos.

PARÁMETROS	CIRCUITO TRANSMISOR	CIRCUITO RECEPTOR AULA 201	ROUTER SEGUNDO PISO (Rx/Tx)	CIRCUITO RECEPTOR AULA 301	ROUTER TERCER PISO (Rx/Tx)
ATCH	11	12	11/12	13	12/13
ATID	1111	2222	1111/2222	3333	2222/3333
ATSH	13A200	13A200	13A200	13A200	13A200
ATSL	402C89D0	402C89C4	402C89A1	404832C5	402C89B5
ATDH	13A200	13A200	13A200/13A200	13A200	13A200/13A200

<sup>7</sup>XCTU: Es una aplicación multiplataforma gratuita diseñada para permitir a los desarrolladores interactuar con módulos Digi RF a través de una interfaz gráfica fácil de usar. Incluye nuevas herramientas que hacen más fácil configurar y probar los módulos de RF XBee.

<b>ATDL</b>	402C89A1	402C89A1	402C89D0/402C89B5	402C89B5	402C89A1/404832C5
<b>ATMY</b>	1	20	2/20	30	3/30
<b>ATBD</b>	9600	9600	9600	9600	9600
<b>ATSM</b>	1	1	1	1	1

Fuente: El autor.

Las especificaciones de los parámetros de configuración mencionadas anteriormente tienen el siguiente significado:

- **ATCH:** Permite establecer un canal de comunicación en la red.
- **ATID:** Especifica un valor de identificación de la red.
- **ATSH /ATSL:** Es la dirección MAC del dispositivo, es decir, es la numeración única que identifica al dispositivo dentro de la red.
- **ATDH /ATDL:** Especifica la dirección de destino con el cual el dispositivo se va a conectar. Esta puede ser la dirección MAC de otro dispositivo o una dirección de broadcast.
- **ATMY:** Especifica un nombre o número para identificación del dispositivo dentro de la red.
- **ATBD:** Establece la velocidad, en baudios, a la cual el dispositivo va a trabajar.
- **ATSM:** Activa el modo de "sleep" en los dispositivos lo cual permite el ahorro de energía cuando se encuentren en reposo.



### 5.3.2 Verificación de envío y recepción de datos.

Esta prueba se realiza sólo entre los módulos XBEE conforme la topología de red lo especifica, para ello se “puentea” los pines de transmisión (TX) y recepción (RX) del módulo XBEE, para verificar que los datos que se envían, regresen de inmediato. Este “puenteo” se puede apreciar en la figura siguiente de cada circuito.

Figura 110. Puente de los pines TX y RX del módulo XBEE correspondiente al Aula 201.



Fuente: El autor.

Figura 111. Puente de los pines TX y RX del módulo XBEE correspondiente al Aula 301.



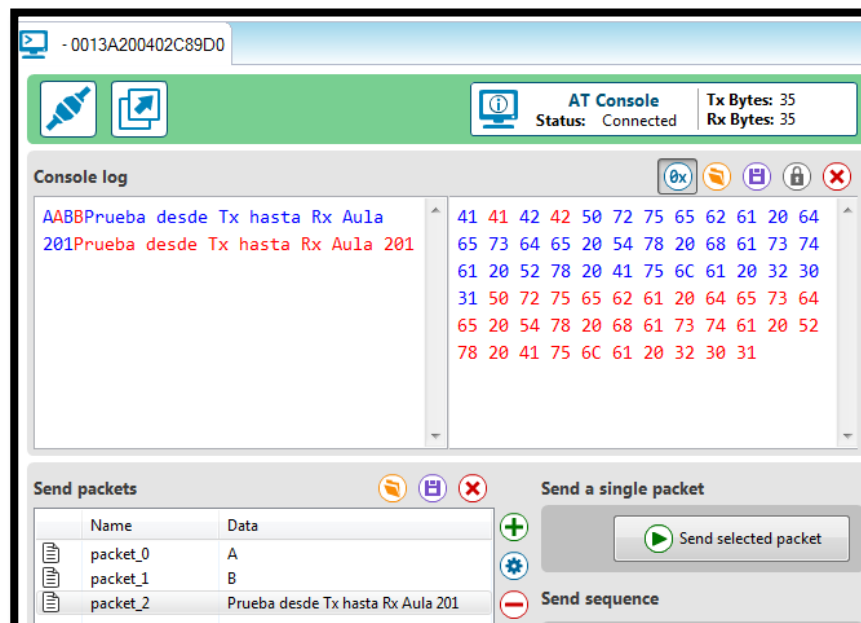
Fuente: El autor.

Una vez configurados los dos módulos XBEE Pro y realizada la modificación especificadas en las figuras 95 y 96 en los circuitos receptores, se realizan las pruebas de transmisión y recepción de datos. Cabe recalcar que los datos enviados no son los correspondientes a la trama de la aplicación, ya que se utiliza texto aleatorio para dicha prueba de verificación. En la figura siguiente se observa el resultado obtenido en el terminal del software XCTU.

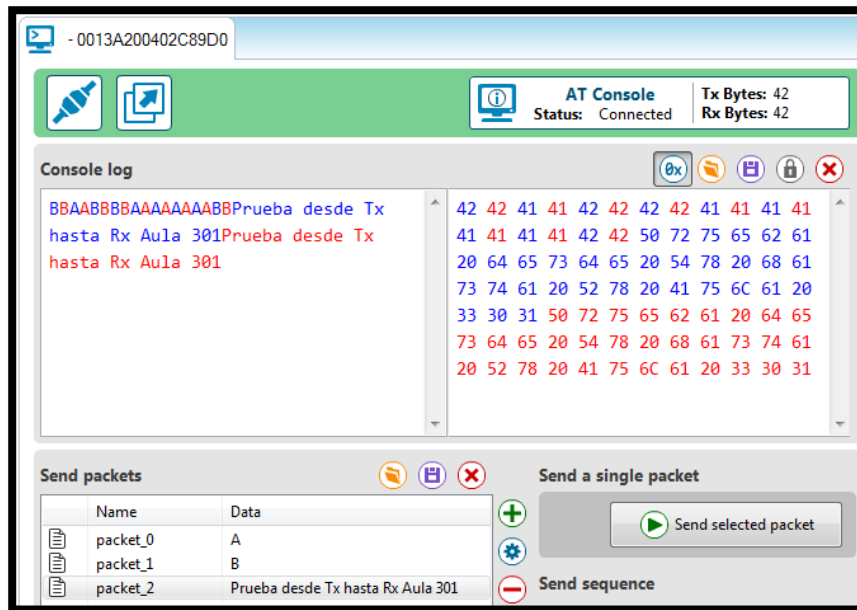
Donde:

- La trama de rojo, son los datos recibidos y
- La trama de color azul, son los datos enviados.

Figura 112. Pruebas de convergencia entre módulos XBEE en la red.



- a) Convergencia entre el circuito transmisor y el circuito receptor del Aula 201.



b) Convergencia entre el circuito transmisor y el circuito receptor del Aula 301.

Fuente: El autor.

Resultados:

De esta forma se comprueba el correcto funcionamiento de los módulos en la red, obteniendo en tiempo real la lectura de datos con una precisión exacta a la hora de transmitir y recibir las tramas, quedando listos para ser incorporados en los circuitos eléctricos. Cabe recalcar que las tramas enviadas y recibidas son sólo datos de pruebas.

## 5.4 Interfaz gráfica encargada del almacenamiento y procesamiento de datos

Tomando en cuenta que hasta este punto no existe ningún error en el diseño de hardware y cobertura de red, en la figura siguiente se visualiza la información guardada en la base de datos, haciendo uso del sistema, mediante la generación de un reporte, verificando de esta forma el almacenamiento y procesamiento de datos en dicho software y comprobando de esta manera el correcto funcionamiento del mismo.

Figura 113. Reporte generado por el sistema de control de acceso.

**SISTEMA DE CONTROL DE ACCESO  
INFORME DEL ADMINISTRADOR  
UTN - FICA - CIERCOM**

Cédula: 1003098215  
Docente: BENAVIDES CHAMORRO KATHERINE

Tipo	Día	Fecha	Carrera	Materia	Paralelo/Nivel	Aula	Horas	Hora_Inicio	Hora_Final	Detalle
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A 1º	AULA_201	3	08:23:47	08:25:28	DSFSDPSDF
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A 1º	AULA_201	3	08:41:03	08:42:01	DSFSDPSDF
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A 1º	AULA_201	3	08:45:27	08:46:45	DSFSDPSDF
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A 1º	AULA_201	3	08:51:33	08:52:26	DSFSDPSDF
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A 1º	AULA_201	3	08:54:18	08:56:41	DSFSDPSDF
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	B 1º	AULA_201	3	10:10:55	10:12:05	FB

Cédula: 1003765565  
Docente: ANDRADE VALENZUELA CARLOS

Tipo	Día	Fecha	Carrera	Materia	Paralelo/Nivel	Aula	Horas	Hora_Inicio	Hora_Final	Detalle
LIMPIEZA	SÁBADO	23/05/2015	-	-	-	AULA_201	0	11:10:41	11:26:55	LIMPIEZA DELAULA

Autor: Guerrero Aseicha Antonio

Página 1 de 1

Nº de página actual: 1      Nº total de páginas: 1      Factor de zoom: 75%

Fuente: El autor (Realizado en Microsoft Visual Studio 2010 en C#)

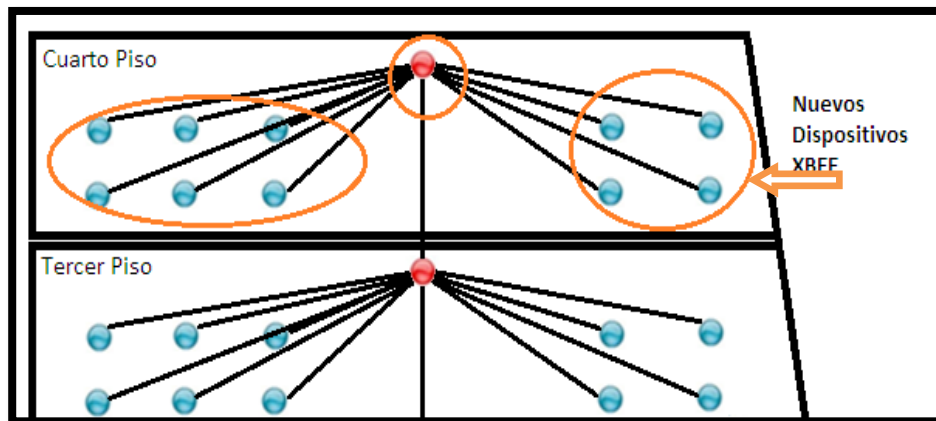
## **5.5 Ampliación de cobertura del sistema de control de acceso**

Dado que la facultad crece cada día más y como se puede apreciar en los últimos tiempos, es necesario de la dotación de nuevas aulas en la facultad, por lo que se puede decir que la cobertura de red para este sistema debe ser convergente, fiable y sobre todo seguro.

Para ello, se puede hacer uso de los mismos dispositivos inalámbricas instaladas en los diferentes pisos de la facultad para que funcionen como routers ya que caso contrario se presentarán problemas en la red dejando a los dispositivos más alejados fuera de cobertura. Es por ello que el aumento de dispositivos electrónicos se los puede realizar sin ningún problema, ya que se puede crear hasta 65635 nodos en una red, logrando así la ampliación de la misma

En la figura siguiente se puede apreciar el aumento de dispositivos en la topología de red que se puede realizar en caso de aumentar aulas en la facultad. Como ejemplo se tomó el cuarto piso de la edificación.

Figura 114. Ampliación de cobertura en la red.



Fuente: El autor.

En este capítulo se especificaron ciertas pruebas realizadas en el proceso de diseño, pruebas en las que fueron necesarios realizar ciertos cambios en el trabajo de titulación, logrando ser totalmente funcional eléctricamente y convergente en la transmisión y recepción de datos en la red WPAN.

## **CAPÍTULO VI**

### **ANÁLISIS ECONÓMICO Y FACTIBILIDAD**

#### **6.1 Introducción**

Este capítulo contendrá el análisis de factibilidad económica del sistema de control de acceso propuesto para la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte la cual cuenta con el presupuesto de los diferentes dispositivos electrónicos y de comunicación, analizando los parámetros del primer capítulo, en los cuales fueron indagados los elementos de hardware.

Es necesario indicar que el presente proyecto no generará ningún tipo de ingresos para la institución, por esta razón se realizará un análisis de indicadores de gestión - desempeño que permitirán determinar la factibilidad de la implementación de este proyecto.

#### **6.2 Costos**

Costo es el valor en efectivo o equivalente de un recurso necesario para manufacturar un bien o brindar un servicio. Al costo se lo clasifica de la siguiente manera:

- Materia Prima Directa
- Mano de Obra Directa
- Costos Indirectos de Fabricación

Debido a que este es un proyecto electrónico en lo referente a costos de Materia Prima se lo calcula dependiendo del precio de cada uno de ellos, tomando en cuenta también los materiales de construcción. A continuación en las tablas siguientes se menciona el costo de cada uno de los diferentes circuitos que contendrá el proyecto.

### 6.2.1 Materia Prima

Son los elementos sobre el cual se ejerce una labor con el objeto de transformarlo en el producto terminado analizado, la materia prima es cuantificable y medible en forma unitaria. Seguidamente se presenta cada material que se utilizará en la elaboración de los circuitos eléctricos.

Tabla 8. Costo del circuito transmisor.

<b>CIRCUITO TRANSMISOR</b>			
<b>DESCRIPCIÓN</b>	<b>CANT.</b>	<b>V. UNIT.</b>	<b>V. TOTAL</b>
Módulo XBEE Pro S1	1	35,20	35,20
USB a UART 5-Pin CP2102	1	17,60	17,60
Diodo Led	1	0,09	0,09
Resistencia	1	0,04	0,04
Caja plástica 5x5 cm	1	0,88	0,88
Construcción del circuito (Baquelita, Asido, etc.)	1	0,44	0,44
<b>SUB TOTAL</b>			<b>54,24</b>
<b>12 % IVA</b>			<b>6,51</b>
<b>TOTAL</b>			<b>60,75</b>



Tabla 9. Costo del circuito receptor.

<b>CIRCUITO RECEPTOR</b>			
<b>DESCRIPCIÓN</b>	<b>CANT.</b>	<b>V. UNIT.</b>	<b>V. TOTAL</b>
Transformador 9V a 500mA	1	3,52	3,52
Rectificador de onda completa	1	0,44	0,44
Diodo Led	2	0,09	0,18
Diodo electrolítico 1000uF/25 V	1	0,35	0,35
Diodos 1N4007	7	0,08	0,55
Diodo Zenner 3.3V	1	0,13	0,13
Resistencias	15	0,04	0,53
Baterías recargables	3	7,48	22,44
Regulador LM317T	3	0,53	1,58
Regulador LM7805	1	0,44	0,44
Regulador LM7812	1	0,53	0,53
Disipadores de calor	3	0,44	1,32
Módulo XBEE Pro S1	1	35,20	35,20
microcontrolador PIC18F4550	1	8,80	8,80
Teclado Matricial 4x4	1	4,40	4,40
LCD Liquido 2x16	1	7,92	7,92
Potenciómetro 10K	1	0,35	0,35
Fusible	1	0,44	0,44
Construcción del circuito (Baquelita, Asido, etc.)	1	0,44	0,44
Pulsadores pequeños	2	0,09	0,18
Zócalo de 20x2	1	0,35	0,35
Zócalo de 10x2	1	0,18	0,18
Relé 5V	1	0,79	0,79
Parlante (Buzer)	1	0,22	0,22
Transistor 3904	2	0,09	0,18
Caja plástica 10x15 cm	1	4,40	4,40
<b>SUB TOTAL</b>			<b>95,86</b>
<b>12 % IVA</b>			<b>11,50</b>
<b>TOTAL</b>			<b>107,36</b>

Tabla 10. Costo del circuito receptor parte mecánica.

<b>CIRCUITO RECEPTOR MECÁNICO</b>			
<b>DESCRIPCIÓN</b>	<b>CANT.</b>	<b>V. UNIT.</b>	<b>V. TOTAL</b>
Pulsadores pequeños	2	0,09	0,18
Cerradura electrónica	1	28,89	28,89
Cable UTP	3	0,35	1,05
Canaleta	2	1,50	3,00
<b>SUB TOTAL</b>			<b>33,12</b>
<b>12 % IVA</b>			<b>3,97</b>
<b>TOTAL</b>			<b>37,09</b>

Tabla 11. Costo del circuito router.

<b>CIRCUITO ROUTER</b>			
<b>DESCRIPCIÓN</b>	<b>CANT.</b>	<b>V. UNIT.</b>	<b>V. TOTAL</b>
Baterías recargables	1	9,73	9,73
Módulo XBEE Pro S1	1	35,20	35,20
Construcción del circuito (Baquelita, Asido, etc.)	1	0,44	0,44
Diodo Led	1	0,09	0,09
Resistencia	1	0,04	0,04
Caja plástica 5x5 cm	1	4,40	4,40
<b>SUB TOTAL</b>			<b>49,90</b>
<b>12 % IVA</b>			<b>5,99</b>
<b>TOTAL</b>			<b>55,89</b>

Fuente: El autor.

Como se puede apreciar en las tablas anteriores, el valor total de cada uno de ellos se relaciona a un solo circuito, por ende, tomando en consideración los requerimientos del proyecto y la topología de red, la cantidad total de circuitos en cuanto a materia prima se refiere, se aprecia en la siguiente tabla.

Tabla 12. Inversión de los equipos electrónicos.

<b>INVERSION DE EQUIPOS ELECTRONICOS</b>			
<b>DESCRIPCIÓN</b>	<b>CANT.</b>	<b>V. UNIT.</b>	<b>V. TOTAL</b>
Circuito transmisor	1	60,75	60,75
Circuito receptor	18	107,36	1932,48
Circuito receptor mecánico	18	37,09	667,62
Circuito router	2	55,89	111,78
<b>TOTAL MATERIA PRIMA</b>			<b>2772,63</b>

Fuente: El autor.

Es necesario recalcar que el presente proyecto de investigación cuenta con costos adicionales detallados a continuación:

### **6.2.2 Mano de Obra Directa**

Es la mano de obra consumida en las áreas que tienen una relación directa con la producción o la prestación de algún servicio.

En este caso es la generada por la instalación de cada uno de los circuitos electrónicos, tanto el transmisor con su respectivo software ubicado en una computadora o PC así como los respectivos circuitos receptores en cada una de las aulas con sus respectivas cerraduras eléctricas.

Tabla 13. Costo de la mano de obra directa.

DETALLE	CANT.	C. UNIT.	C. TOTAL	OBSERVACIONES
Instalación del circuito transmisor	1	\$ 10,00	\$10,00	La instalación se realizará únicamente con un operador
Instalación del circuito receptor	18	\$ 20,00	\$ 360,00	
Instalación del circuito receptor mecánico	18	\$ 40,00	\$ 720,00	
Instalación del circuito router	2	\$ 10,00	\$ 20,00	
<b>TOTAL MANO DE OBRA</b>			<b>\$ 1110,00</b>	

Fuente: El autor.

### 6.2.3 Costos Indirectos de Fabricación (CIF)

Son todos los costos que intervienen de forma indirecta en la fabricación del bien producto.

Tabla 14. Costos indirectos de fabricación.

DETALLE	Costo \$	OBSERVACIONES
Energía Eléctrica	\$360,00	Este costo está proyectado a un año.
Mantenimiento	\$500,00	El mantenimiento de los circuitos se realizará una vez al año
Software/ Microsoft Visual Studio 2010	\$0,00	El software es donado por la institución pues el presente proyecto es para su uso
Software/ Microsoft SQL Server 2008	\$0,00	El software es donado por la institución pues el presente proyecto es para su uso
<b>TOTAL CIF</b>	<b>\$860,00</b>	

Fuente: El autor.

## 6.2.4 Estado de costos de producción

A continuación se presenta el Estado de costos proyectado para un año.

Tabla 15. Estado de costos de producción.

	<b>Inventario Inicial Materia prima y materiales</b>	\$ -	
(+)	Compras Materia Prima y Materiales	\$ 2772,63	
(=)	<b>Disponible de Materia Prima y Materiales</b>	<b>\$ 2772,63</b>	
(-)	Inventario Final Materia Prima y Materiales	\$ -	
(=)	<b>Costo Real Materia Prima y Materiales</b>	<b>\$ 2772,63</b>	
	Mano de Obra	\$ 1110,00	
(=)	<b>Costo de Mano de Obra Directa</b>	<b>\$ 1110,00</b>	
	Costos Indirectos de Fabricación		
(+)	Energía Eléctrica	\$ 360,00	
(+)	Mantenimiento y Reparación	\$ 500,00	
(=)	<b>Costos Indirectos de Fabricación</b>	<b>\$ 860,00</b>	
(=)	<b>COSTO DE PRODUCCIÓN</b>		<b>\$ 4.742,63</b>

Fuente: El autor.

Como se puede observar el costo al que asciende el presente proyecto es de cuatro mil setecientos cuarenta y dos dólares americanos con sesenta y tres ctvs. (\$4.742,63).

### **6.3 Inversión**

(Torres & Aguirre, 2010)"La inversión es todo desembolso de recursos financieros para adquirir bienes concretos durables o instrumentos de producción, denominados bienes de equipo, y que la empresa utilizará durante varios años para cumplir su objeto social"

Es decir, la inversión es la cantidad de dinero requerido para la ejecución de un proyecto, en los cuales contiene todos los costos ya sea de los materiales que incluye el proyecto en sí, así como los valores de ingeniería.

Una vez realizado el estado de costos se llega a la conclusión de que la inversión para el desarrollo del presente proyecto es de cuatro mil setecientos cuarenta y dos dólares americanos con sesenta y tres ctvs. (\$4.742,63).

### **6.4 Factibilidad**

A continuación se presentan indicadores de desempeño para medir la factibilidad del presente proyecto, con la finalidad de que este sea ejecutado por la Facultad de Ingeniería de Ciencias Aplicadas, para mejorar y satisfacer las necesidades de sus docentes y alumnos.

## **6.5 Indicadores de Desempeño**

(Salguero, 2001) “Es necesario considerar que los índices de desempeño son herramientas de gestión que proveen un valor de referencia, a partir del cual se puede establecer una comparación entre las metas planteadas y el desempeño logrado.”

Para el desarrollo de los siguientes indicadores de gestión se toma como referencia a Salguero, Amado (2001) en su libro Indicadores de Gestión y Cuadro de Mando, por lo que dichos indicadores son ampliados y acoplados a la realidad y beneficio de la institución.

### **6.5.1 Indicador de Eficacia**

Este indicador mide lo planificado y lo obtenido, como también en qué medida la institución está cumpliendo con sus objetivos.

### **6.5.2 Indicador de Eficiencia**

Mide los costos respecto de beneficios recibidos como también el nivel de actividad y recursos utilizados.

### 6.5.3 Indicador de Economía

Es la capacidad de un proyecto o programa para generar y movilizar adecuadamente los recursos financieros. A continuación se presenta de manera detallada los siguientes indicadores:

Tabla 16. Indicadores de gestión y desempeño.

PRODUCTO ESTRATÉGICO AL QUE SE VINCULA	Indicador	Fórmula de Cálculo	Fórmula	%
Servicios Asistenciales	Eficacia / Resultado	N° de docentes, alumnos y empleados de la FICA que sienten comodidad y seguridad con el nuevo circuito eléctrico / N° de docentes, alumnos y empleados de la FICA	$= (1500/1560) * 100$	96,15%
Implementación de un circuito eléctrico en las puertas de la FICA				
Servicios Asistenciales	Efectividad/Resultado	N° de personal encargado del manejo del sistema a implementarse / N° de personal encargado del manejo del Sistema actual	$= (2/5) * 100$	40,00%
Implementación de un circuito eléctrico en las puertas de la FICA				
Servicios Asistenciales	Efectividad/Resultado	N° de personas beneficiadas / Población objetivo	$= (1560/1560) * 100$	100%
Implementación de un circuito eléctrico en las puertas de la FICA				
Servicios Asistenciales	Eficiencia/Resultados	N° de instalaciones de circuitos eléctricos/Horas hombre trabajadas	$= (18/32) * 100$	56%
Implementación de un circuito eléctrico en las puertas de la FICA				
Servicios	Eficiencia/Resultados	Minutos de espera	$= (2/10) * 100$	20%



Asistenciales		para ingresar al aula de clases con el circuito eléctrico/ Minutos de espera para ingresar al aula de clases sistema actual		
Implementación de un circuito eléctrico en las puertas de la FICA				
Servicios Asistenciales	Economía/Resultados	Costos de Materiales / Costo Total	=(2772,63 /4.742,63) *100	58,46%
Implementación de un circuito eléctrico en las puertas de la FICA				

Fuente: El autor.

Con lo anteriormente expuesto se puede observar que los porcentajes en los índices de eficacia, eficiencia, efectividad y economía son bastante favorables para poder ejecutar el presente trabajo, pues la implementación del sistema de control de acceso en cada aula es muy beneficioso para la Facultad de Ingeniería de Ciencias Aplicadas de la Universidad Técnica del Norte; pues de esta manera facilitará la organización interna de la facultad y se minimizarán recursos como: el tiempo perdido al momento de ingresar a las aulas, la utilización de hojas mediante el control de asistencia de los docentes, pérdida de bienes en la facultad.

Cabe recalcar que este sistema de control de acceso fue diseñado tomando en cuenta a la Facultad de Ingeniería en Ciencias Aplicadas pero puede ser implementado en cualquier otra edificación ya que es un sistema universal ya sea de la Universidad Técnica del Norte como en cualquier institución o lugar que se desee implementar.

## CAPÍTULO VII

### CONCLUSIONES Y RECOMENDACIONES

#### 7.1 Introducción

Finalmente, se procede a realizar las conclusiones y las recomendaciones que se darán para que cuando el proyecto sea implementado, éste sea más fácil de realizar por una u otra persona interesada en el tema.

#### 7.2 Conclusiones

- En el presente proyecto se logró diseñar un sistema de control de acceso para las aulas de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica del Norte, el cual es capaz de automatizar el ingreso a las aulas por parte de los docentes, tomando datos de la transmisión y recepción en una topología de red WPAN, dicho sistema funciona mediante una interfaz gráfica diseñada también en el software Microsoft Visual Studio 2010 C#.
- La implementación de un nuevo sistema de control para las aulas en la facultad es de suma importancia ya que, como se demuestra en este proyecto, el sistema es capaz de optimizar el proceso de ingreso a las

mismas reduciendo los tiempos de ingreso, es más seguro que la utilización de las cerraduras tradicionales, permitiendo eliminar las rejas que se encuentran en la edificación y así poder llevar un registro exacto de las personas que hacen uso de las aulas.

- El diseño de la red inalámbrica es estructurado de tal manera que, permita una futura expansión mediante la inclusión de nuevos equipos en la red, logrando de esta manera ampliar el área de cobertura y la cantidad de circuitos electrónicos que realizan el trabajo de apertura en las puertas.
- Los circuitos electrónicos fueron diseñados para brindar una interfaz confiable y de fácil manipulación tanto para estudiantes, docentes y personal que labora en la facultad.
- La interfaz gráfica cuenta con los manuales correspondientes a cada modo de ingreso, lo cual facilita el uso del sistema, siendo más fácil de manejar. Este sistema fue diseñado tomando en cuenta los requerimientos necesarios estudiados en capítulos anteriores.
- En la actualidad los sistemas electrónicos de control de acceso están sustituyendo a las tradicionales cerraduras manipuladas manualmente por llaves, es por esta razón que el proyecto fue desarrollado para este tipo de sistemas, además de poseer características óptimas para la implementación en cualquier lugar en el que se desee controlar el acceso a una área determinada.

### 7.3 Recomendaciones

- Hay que tener cuidado en la construcción de la caja que alojará a la placa del circuitos electrónico, ésta debe poseer características importantes en caso de requerir que dicha placa sea ubicada en exteriores (lugar abierto), como son seguridad, y protección hermética contra la lluvia.
- Para añadir características a los circuitos electrónicos, se puede realizar retirando el microcontrolador de la placa; ya que éste es el encargado de controlar el proceso de adquisición y transmisión de datos, así como la apertura de la puerta, a éste se puede modificar el programa fácilmente para lograr esta característica.
- Se debe tener presente que los circuitos receptores se encuentren en lugares que no presenten obstáculos en la red inalámbrica, si se detecta que la señal de la red es muy débil, se recomienda ubicar un circuito “puente” en un lugar estratégico para ampliar la red y tener una mayor cobertura, de esta manera la red puede ser totalmente operativa.
- El sistema de control de acceso posee seguridades basadas en claves, por tal motivo se recomienda que el proceso de administración de este sistema esté a cargo de una persona que labora en la facultad.
- Al diseñar un circuito electrónico se recomienda tomar medidas de voltaje y amperaje conforme el diseño se esté realizando, para verificar el

funcionamiento de cada dispositivo electrónico que se conecte, de esta forma corregir cualquier error que se presente con anticipación.

#### 7.4 Bibliografía:

Anibal, B. S. (17 de Noviembre de 2012). Estructura y Elementos de los Microcontroladores.

Collaguazo, G. (2010). *Scribd*. Obtenido de Scribd: <http://es.scribd.com/doc/244728217/MICROCONTROLADORES-pdf#scribd>

Jara Werchau, P., & Nazar, P. (2009). *Departamento de Ingeniería en Sistemas de Información*. Obtenido de Estándar IEEE 802.11 X de las WLAN: [http://www.edutecne.utn.edu.ar/monografias/standard\\_802\\_11.pdf](http://www.edutecne.utn.edu.ar/monografias/standard_802_11.pdf)

kioskea. (Junio de 2014). *kioskea.net*. Obtenido de <http://es.kioskea.net/contents/70-bluetooth>

Kioskea. (Junio de 2014). *Kioskea.net*. Obtenido de Introducción a Wi-Fi (802.11 o WiFi): <http://es.kioskea.net/contents/789-introduccion-a-wi-fi-802-11-o-wifi>

Maroné, J. (s.f.). *EXA UNICEN*. Obtenido de EXA UNICEN: [http://www.exa.unicen.edu.ar/catedras/tmicrocon/Material/3\\_Overview\\_Microcontroladores\\_ATMEL.pdf](http://www.exa.unicen.edu.ar/catedras/tmicrocon/Material/3_Overview_Microcontroladores_ATMEL.pdf)

Pascual, A. E. (Octubre de 2007). *Estándares en Tecnologías Inalámbricas*. Obtenido de [http://www.itrainonline.org/itrainonline/mmtk/wireless\\_es/files/02\\_es\\_estandares-inalambricos\\_guia\\_v02.pdf](http://www.itrainonline.org/itrainonline/mmtk/wireless_es/files/02_es_estandares-inalambricos_guia_v02.pdf)

Revista de la Asociación de Ingenieros del ICAI. (2012). Gestión de redes inteligentes domésticas mediante ZigBee Smart Energy. *ANALES DE MECÁNICA Y ELECTRICIDAD*, 28-33.

Salguero, A. (2001). *Indicadores de Gestión y Cuadro de Mando*. Juan Bravo, Madrid - España: Diaz de Santos S.A.

Sánchez, S. (2013). *Microcontroladores y sus Aplicaciones*. Obtenido de Microcontroladores y sus Aplicaciones: <https://microcontroladoresesv.wordpress.com/microcontroladores-pic-y-sus-variedades/>

Saucedo, C. I. (29 de Febrero de 2012). *Universidad Nacional Autónoma de México*. Obtenido de Eficiencia y seguridad en Bluetooth y Zigbee: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/229/A6.pdf?sequence=6>

Toapanta, D., & Jaramillo, A. (2009). *Construcción de una fuente digital variable regulada entre 2V a 20VDC*. Quito.

Torres, C., & Aguirre, P. (Diciembre de 2010). *Repositorio Digital UTN*. Obtenido de Repositorio Digital UTN: <http://repositorio.utn.edu.ec/bitstream/123456789/752/13/04%20RED%20002%20CAPITULO%20II.pdf>

# **ANEXOS**



## **ANEXO A**

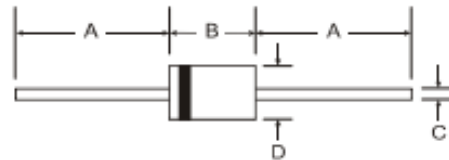
### **7.5 DATASHEETS DE LOS DISPOSITIVOS ELECTRÓNICOS MÁS RELEVANTES**

### Features

- Diffused Junction
- High Current Capability and Low Forward Voltage Drop
- Surge Overload Rating to 30A Peak
- Low Reverse Leakage Current
- Lead Free Finish, RoHS Compliant (Note 3)

### Mechanical Data

- Case: DO-41
- Case Material: Molded Plastic. UL Flammability Classification Rating 94V-0
- Moisture Sensitivity: Level 1 per J-STD-020D
- Terminals: Finish - Bright Tin. Plated Leads Solderable per MIL-STD-202, Method 208
- Polarity: Cathode Band
- Mounting Position: Any
- Ordering Information: See Page 2
- Marking: Type Number
- Weight: 0.30 grams (approximate)



Dim	DO-41 Plastic	
	Min	Max
A	25.40	—
B	4.06	5.21
C	0.71	0.864
D	2.00	2.72

All Dimensions in mm

### Maximum Ratings and Electrical Characteristics @ $T_A = 25^\circ\text{C}$ unless otherwise specified

Single phase, half wave, 60Hz, resistive or inductive load.  
For capacitive load, derate current by 20%.

Characteristic	Symbol	1N4001	1N4002	1N4003	1N4004	1N4005	1N4006	1N4007	Unit	
Peak Repetitive Reverse Voltage	$V_{RRM}$	50	100	200	400	600	800	1000	V	
Working Peak Reverse Voltage	$V_{RWM}$									
DC Blocking Voltage	$V_R$									
RMS Reverse Voltage	$V_{R(RMS)}$	35	70	140	280	420	560	700	V	
Average Rectified Output Current (Note 1) @ $T_A = 75^\circ\text{C}$	$I_O$	1.0								A
Non-Repetitive Peak Forward Surge Current 8.3ms single half sine-wave superimposed on rated load	$I_{FSM}$	30								A
Forward Voltage @ $I_F = 1.0\text{A}$	$V_{FM}$	1.0								V
Peak Reverse Current @ $T_A = 25^\circ\text{C}$	$I_{RM}$	5.0								$\mu\text{A}$
at Rated DC Blocking Voltage @ $T_A = 100^\circ\text{C}$		50								
Typical Junction Capacitance (Note 2)	$C_j$	15				8				pF
Typical Thermal Resistance Junction to Ambient	$R_{\theta JA}$	100								K/W
Maximum DC Blocking Voltage Temperature	$T_A$	+150								$^\circ\text{C}$
Operating and Storage Temperature Range	$T_A, T_{STG}$	-65 to +150								$^\circ\text{C}$

- Notes:
1. Leads maintained at ambient temperature at a distance of 9.5mm from the case.
  2. Measured at 1.0 MHz and applied reverse voltage of 4.0V DC.
  3. EU Directive 2002/95/EC (RoHS). All applicable RoHS exemptions applied, see EU Directive 2002/95/EC Annex Notes.

#### IMPORTANT NOTICE

Diodes Incorporated and its subsidiaries reserve the right to make modifications, enhancements, improvements, corrections or other changes without further notice to any product herein. Diodes Incorporated does not assume any liability arising out of the application or use of any product described herein; neither does it convey any license under its patent rights, nor the rights of others. The user of products in such applications shall assume all risks of such use and will agree to hold Diodes Incorporated and all the companies whose products are represented on our website, harmless against all damages.

#### LIFE SUPPORT

Diodes Incorporated products are not authorized for use as critical components in life support devices or systems without the expressed written approval of the President of Diodes Incorporated.

## SMALL SIGNAL NPN TRANSISTOR

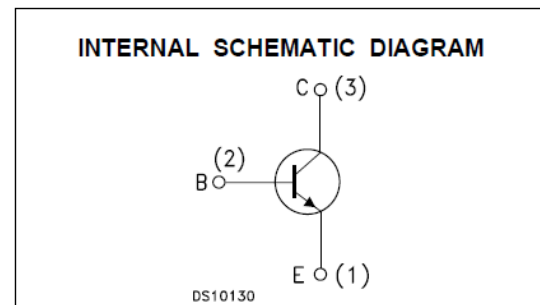
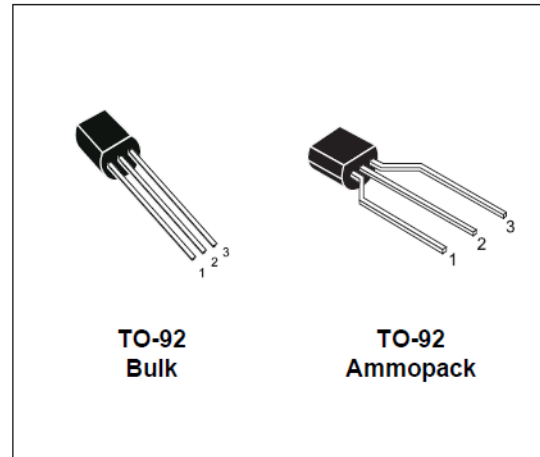
PRELIMINARY DATA

Ordering Code	Marking	Package / Shipment
2N3904	2N3904	TO-92 / Bulk
2N3904-AP	2N3904	TO-92 / Ammopack

- SILICON EPITAXIAL PLANAR NPN TRANSISTOR
- TO-92 PACKAGE SUITABLE FOR THROUGH-HOLE PCB ASSEMBLY
- THE PNP COMPLEMENTARY TYPE IS 2N3906

### APPLICATIONS

- WELL SUITABLE FOR TV AND HOME APPLIANCE EQUIPMENT
- SMALL LOAD SWITCH TRANSISTOR WITH HIGH GAIN AND LOW SATURATION VOLTAGE



### ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_{CB0}$	Collector-Base Voltage ( $I_E = 0$ )	60	V
$V_{CE0}$	Collector-Emitter Voltage ( $I_B = 0$ )	40	V
$V_{EB0}$	Emitter-Base Voltage ( $I_C = 0$ )	6	V
$I_C$	Collector Current	200	mA
$P_{tot}$	Total Dissipation at $T_C = 25\text{ }^\circ\text{C}$	625	mW
$T_{stg}$	Storage Temperature	-65 to 150	$^\circ\text{C}$
$T_j$	Max. Operating Junction Temperature	150	$^\circ\text{C}$

## 2N3904

### THERMAL DATA

$R_{thj-amb}$ •	Thermal Resistance Junction-Ambient	Max	200	$^{\circ}\text{C}/\text{W}$
$R_{thj-case}$ •	Thermal Resistance Junction-Case	Max	83.3	$^{\circ}\text{C}/\text{W}$

### ELECTRICAL CHARACTERISTICS ( $T_{case} = 25^{\circ}\text{C}$ unless otherwise specified)

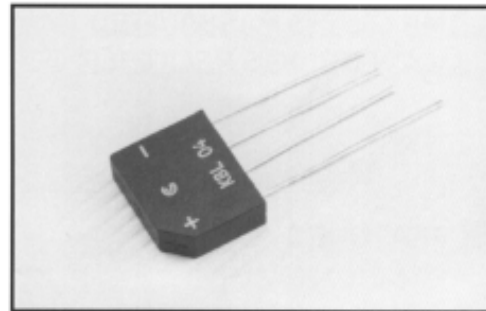
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$I_{CEX}$	Collector Cut-off Current ( $V_{BE} = -3\text{ V}$ )	$V_{CE} = 30\text{ V}$			50	nA
$I_{BEX}$	Base Cut-off Current ( $V_{BE} = -3\text{ V}$ )	$V_{CE} = 30\text{ V}$			50	nA
$V_{(BR)CEO}$ *	Collector-Emitter Breakdown Voltage ( $I_B = 0$ )	$I_C = 1\text{ mA}$	40			V
$V_{(BR)CBO}$	Collector-Base Breakdown Voltage ( $I_E = 0$ )	$I_C = 10\text{ }\mu\text{A}$	60			V
$V_{(BR)EBO}$	Emitter-Base Breakdown Voltage ( $I_C = 0$ )	$I_E = 10\text{ }\mu\text{A}$	6			V
$V_{CE(sat)}$ *	Collector-Emitter Saturation Voltage	$I_C = 10\text{ mA}$ $I_B = 1\text{ mA}$ $I_C = 50\text{ mA}$ $I_B = 5\text{ mA}$			0.2 0.2	V V
$V_{BE(sat)}$ *	Base-Emitter Saturation Voltage	$I_C = 10\text{ mA}$ $I_B = 1\text{ mA}$ $I_C = 50\text{ mA}$ $I_B = 5\text{ mA}$	0.65		0.85 0.95	V V
$h_{FE}$ *	DC Current Gain	$I_C = 0.1\text{ mA}$ $V_{CE} = 1\text{ V}$ $I_C = 1\text{ mA}$ $V_{CE} = 1\text{ V}$ $I_C = 10\text{ mA}$ $V_{CE} = 1\text{ V}$ $I_C = 50\text{ mA}$ $V_{CE} = 1\text{ V}$ $I_C = 100\text{ mA}$ $V_{CE} = 1\text{ V}$	60 80 100 60 30		300	
$f_T$	Transition Frequency	$I_C = 10\text{ mA}$ $V_{CE} = 20\text{ V}$ $f = 100\text{ MHz}$	250	270		MHz
$C_{CBO}$	Collector-Base Capacitance	$I_E = 0$ $V_{CB} = 10\text{ V}$ $f = 1\text{ MHz}$		4		pF
$C_{EBO}$	Emitter-Base Capacitance	$I_C = 0$ $V_{EB} = 0.5\text{ V}$ $f = 1\text{ MHz}$		18		pF
NF	Noise Figure	$V_{CE} = 5\text{ V}$ $I_C = 0.1\text{ mA}$ $f = 10\text{ Hz}$ to 15.7 KHz $R_G = 1\text{ K}\Omega$		5		dB
$t_d$	Delay Time	$I_C = 10\text{ mA}$ $I_B = 1\text{ mA}$			35	ns
$t_r$	Rise Time	$V_{CC} = 30\text{ V}$			35	ns
$t_s$	Storage Time	$I_C = 10\text{ mA}$ $I_{B1} = -I_{B2} = 1\text{ mA}$			200	ns
$t_f$	Fall Time	$V_{CC} = 30\text{ V}$			50	ns

\* Pulsed: Pulse duration = 300  $\mu\text{s}$ , duty cycle  $\leq 2\%$



# KBL005 Thru KBL10

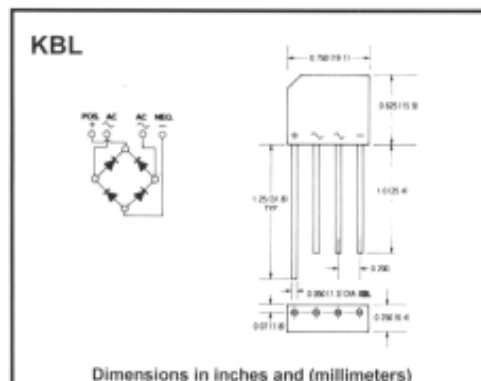
## 4 AMP SILICON BRIDGE RECTIFIER



### FEATURES

- Rating to 1000V PRV
- Surge overload rating to 200 Amperes peak
- Ideal for printed circuit board
- Reliable low cost construction utilizing molded plastic technique results in inexpensive product
- UL recognized: File #E106441
- UL recognized 94V-O plastic material

### Outline Drawing



### Mechanical Data

- Case: Molded plastic
- Leads: Silver plated copper
- Leads solderable per MIL-STD-202, Method 208
- Weight: 0.2 ounce, 5.6 grams

### Maximum Ratings & Characteristics

- Ratings at 25° C ambient temperature unless otherwise specified
- Single phase, half wave, 60Hz, resistive or inductive load
- For capacitive load, derate current by 20%

		KBL005	KBL01	KBL02	KBL04	KBL06	KBL08	KBL10	Units
Maximum Recurrent Peak Reverse Voltage	$V_{RRM}$	50	100	200	400	600	800	1000	V
Maximum RMS Voltage	$V_{RMS}$	35	70	140	280	420	560	700	V
Maximum DC Blocking Voltage	$V_{DC}$	60	100	200	400	600	800	1000	V
Maximum Average Forward Output Current @ $T_A = 50^\circ\text{C}$	$I_{(AV)}$	4.0							A
Peak Forward Surge Current 8.3 ms Single Half-Sine-Wave Superimposed On Rated Load	$I_{FSM}$	200							A
Maximum DC Forward Voltage Drop per Element At 3.0A DC	$V_F$	1.1							V
Maximum DC Reverse Current At Rated DC Blocking Voltage per Element @ $T_C = 100^\circ\text{C}$	$I_R$	10							$\mu\text{A}$
$I^2 t$ Rating for Fusing ( $t < 8.3\text{ms}$ )	$I^2 t$	166							$\text{A}^2 \text{S}$
Operating Temperature Range	$T_J$	-55 to +125							$^\circ\text{C}$
Storage Temperature Range	$T_{STG}$	-55 to +150							$^\circ\text{C}$

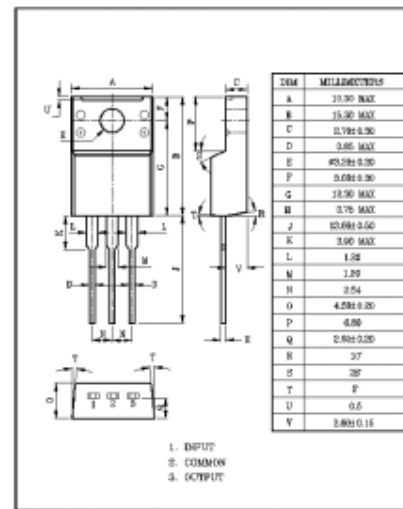
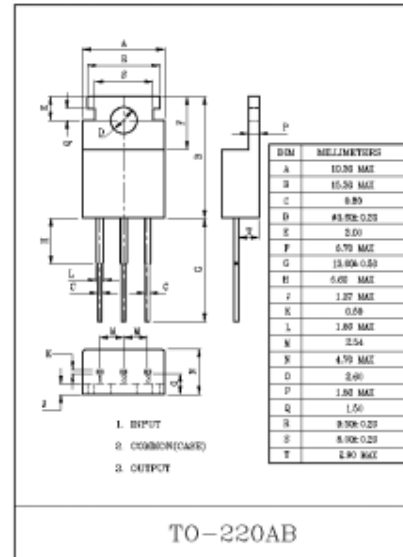
THREE TERMINAL POSITIVE VOLTAGE REGULATORS  
5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V, 20V, 24V.

**FEATURES**

- Suitable for C-MOS, TTL, the Other Digital IC's Power Supply.
- Internal Thermal Overload Protection.
- Internal Short Circuit Current Limiting.
- Output Current in Excess of 1A.
- Satisfies IEC-65 Specification. (International Electrotechnical Commission).

**MAXIMUM RATINGS (Ta=25°C)**

CHARACTERISTIC		SYMBOL	RATING	UNIT
Input Voltage	KIA7805AP/API~ KIA7815AP/API	V <sub>IN</sub>	35	V
	KIA7818AP/API~ KIA7824AP/API		40	
Power Dissipation (Tc=25°C)		P <sub>D</sub>	20.8	W
Power Dissipation (Without Heatsink)		P <sub>D</sub>	2.0	W
Operating Junction Temperature		T <sub>J</sub>	-30~150	°C
Storage Temperature		T <sub>stg</sub>	-55~150	°C



## KIA7805AP/API ~ KIA7824AP/API

KIA7805AP/API

ELECTRICAL CHARACTERISTICS ( $V_{IN}=10V$ ,  $I_{OUT}=500mA$ ,  $0^{\circ}C \leq T_j \leq 125^{\circ}C$ )

CHARACTERISTIC	SYMBOL	TEST CIRCUIT	TEST CONDITION	MIN.	TYP.	MAX.	UNIT	
Output Voltage	$V_{OUT}$	1	$T_j=25^{\circ}C$ , $I_{OUT}=100mA$	4.8	5.0	5.2	V	
Input Regulation	Reg line	1	$T_j=25^{\circ}C$	$7.0V \leq V_{IN} \leq 25V$	-	3	100	mV
				$8.0V \leq V_{IN} \leq 12V$	-	1	50	
Load Regulation	Reg load	1	$T_j=25^{\circ}C$	$5mA \leq I_{OUT} \leq 1.4A$	-	15	100	mV
				$250mA \leq I_{OUT} \leq 750mA$	-	5	50	
Output Voltage	$V_{OUT}$	1	$7.0V \leq V_{IN} \leq 20V$ $5.0mA \leq I_{OUT} \leq 1.0A$ , $P_o \leq 15W$	4.75	-	5.25	V	
Quiescent Current	$I_B$	1	$T_j=25^{\circ}C$ , $I_{OUT}=5mA$	-	4.2	8.0	mA	
Quiescent Current Change	$\Delta I_B$	1	$7.0V \leq V_{IN} \leq 25V$	-	-	1.3	mA	
Output Noise Voltage	$V_{NO}$	1	$T_a=25^{\circ}C$ , $10Hz \leq f \leq 100kHz$ $I_{OUT}=50mA$	-	50	-	$\mu V_{rms}$	
Ripple Rejection Ratio	RR	1	$f=120Hz$ , $8.0V \leq V_{IN} \leq 18V$ , $I_{OUT}=50mA$ , $T_j=25^{\circ}C$	62	78	-	dB	
Dropout Voltage	$V_D$	1	$I_{OUT}=1.0A$ , $T_j=25^{\circ}C$	-	2.0	-	V	
Short Circuit Current Limit	$I_{SC}$	1	$T_j=25^{\circ}C$	-	1.6	-	A	
Average Temperature Coefficient of Output Voltage	$TC_{VO}$	1	$I_{OUT}=5mA$ , $0^{\circ}C \leq T_j \leq 125^{\circ}C$	-	-0.6	-	mV/ $^{\circ}C$	

## KIA7805AP/API ~ KIA7824AP/API

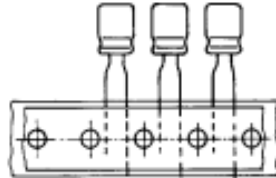
KIA7812AP/API

ELECTRICAL CHARACTERISTICS ( $V_{IN}=19V$ ,  $I_{OUT}=500mA$ ,  $0^{\circ}C \leq T_j \leq 125^{\circ}C$ )

CHARACTERISTIC	SYMBOL	TEST CIRCUIT	TEST CONDITION	MIN.	TYP.	MAX.	UNIT	
Output Voltage	$V_{OUT}$	1	$T_j=25^{\circ}C$ , $I_{OUT}=100mA$	11.5	12.0	12.5	V	
Input Regulation	Reg line	1	$T_j=25^{\circ}C$	$14.5V \leq V_{IN} \leq 30V$	-	10	240	mV
				$16V \leq V_{IN} \leq 22V$	-	3	120	
Load Regulation	Reg load	1	$T_j=25^{\circ}C$	$5mA \leq I_{OUT} \leq 1.4A$	-	12	240	mV
				$250mA \leq I_{OUT} \leq 750mA$	-	4	120	
Output Voltage	$V_{OUT}$	1	$14.5V \leq V_{IN} \leq 27V$ $5.0mA \leq I_{OUT} \leq 1.0A$ , $P_o \leq 15W$	11.4	-	12.6	V	
Quiescent Current	$I_B$	1	$T_j=25^{\circ}C$ , $I_{OUT}=5mA$	-	4.3	8.0	mA	
Quiescent Current Change	$\Delta I_B$	1	$14.5V \leq V_{IN} \leq 30V$	-	-	1.0	mA	
Output Noise Voltage	$V_{ND}$	1	$T_a=25^{\circ}C$ , $10Hz \leq f \leq 100kHz$ $I_{OUT}=50mA$	-	90	-	$\mu V_{rms}$	
Ripple Rejection Ratio	RR	1	$f=120Hz$ , $15V \leq V_{IN} \leq 25V$ $I_{OUT}=50mA$ , $T_j=25^{\circ}C$	55	71	-	dB	
Dropout Voltage	$V_D$	1	$I_{OUT}=1.0A$ , $T_j=25^{\circ}C$	-	2.0	-	V	
Short Circuit Current Limit	$I_{SC}$	1	$T_j=25^{\circ}C$	-	0.7	-	A	
Average Temperature Coefficient of Output Voltage	$TC_{VO}$	1	$I_{OUT}=5mA$ , $0^{\circ}C \leq T_j \leq 125^{\circ}C$	-	-1.6	-	mV/ $^{\circ}C$	



## Aluminum Electrolytic Capacitors, Radial Style



### FEATURES

- Polarized Al electrolytic capacitor
- High C-U product
- Very small dimensions
- Long lifetime
- Extended temperature range: 105°C

### APPLICATIONS

- General uses, industrial electronics, automotive electronics, audio / video systems
- Smoothing, filtering, coupling, decoupling, timing elements
- Little space requirement
- Portable and mobile units

MAIN SPECIFICATIONS				
Nominal Case Size (D x L)	[mm]	5 x 11 to 18 x 11.5		10 x 12.5 to 18 x 40
Rated Capacitance Range	[ $\mu$ F]	2.2 to 22000		
Capacitance Tolerance	[%]	$\pm 20$		
Rated Voltage Range	[V]	6.3 to 100	160 to 350	400, 450
Category Temperature Range	[°C]	-55 to 105	-40 to 105	-25 to 105
Endurance Test at UCT	[h]	1000		2000
Lifetime at 105°C and $I_R$	[h]	1500		2500
Lifetime at 85°C and $I_R$	[h]	6000		10000
Lifetime at 40°C and $I_R$	[h]	140000		230000
Sectional Specifications		IEC 384-4, CECC 30300, GP/LL grade		
Detail Specifications		CECC 30301-037, similar to DIN 45 910 Part 124 without quality assessment		
Climatic Category IEC 68 DIN 40040		55 / 105 / 56 FMF	40 / 105 / 56 GMF	25 / 105 / 56 HMF
Failure Rate	[10 <sup>-6</sup> / h]	$\leq 45$		



**TECHNICAL AND ORDERING INFORMATION**

If not indicated otherwise the following test conditions apply to all electrical parameters:

$T_a = 20^{\circ}\text{C}$ ,  $p = 80\text{-}120\text{ kPa}$ ,  $\text{RH} = 45\text{-}75\%$

- $C_R$  Rated Capacitance at 120Hz
- $U_R$  Rated Voltage
- $\tan \delta$  Max. Dissipation Factor at 120Hz
- $R_{ESR}$  Max. Equivalent Series Resistance at 120Hz
- $I_R$  Rated Alternating Current at 120Hz and Upper Category Temperature

Ordering example:

EKB 100 $\mu\text{F}$  / 35V,  $\pm 20\%$ , size: 6.3mm x 11mm

Leads: Long

Ordering code: EKB00BA310F00

Leads: Short

Ordering code: EKB 05...

Leads: Bent open, shortened

Ordering code: EKB 09...

Leads: Bent open, shortened and formed

Ordering code: EKB 06...

ELECTRICAL CHARACTERISTICS, WEIGHT AND ORDERING CODE							
CAPACITANCE 120Hz CR [ $\mu\text{F}$ ]	RATED VOLTAGE UR [V]	DIMENSIONS D x L [mm]	DISSIPATION FACTOR 120Hz	EQUIVALENT SERIES RESISTANCE 120Hz [ $\Omega$ ]	RATED CURRENT IR 120Hz, 85 $^{\circ}\text{C}$ [mA]	WEIGHT [g]	ORDERING CODE
680	6.3	8 x 11.5	0.28	0.55	338	1.1	EKB00PB368B00
1000	6.3	8 x 11.5	0.25	0.37	410	1.1	EKB00PB410B00
1500	6.3	10 x 16	0.29	0.26	575	2.0	EKB00DD415B00
2200	6.3	10 x 20	0.30	0.18	726	2.5	EKB00DE422B00
3300	6.3	13 x 16	0.33	0.13	922	3.0	EKB00GD433B00
4700	6.3	13 x 20	0.55	0.10	1149	3.8	EKB00GE447B00
6800	6.3	13 x 25	0.40	0.08	1406	4.5	EKB00GG468B00
10000	6.3	16 x 25	0.46	0.06	1732	7.0	EKB00JG510B00
15000	6.3	16 x 35.5	0.56	0.05	2161	11.0	EKB00JL515B00
22000	6.3	18 x 40	0.70	0.04	2586	16.0	EKB00KK522B00
150	10	5 x 11	0.24	2.12	129	0.5	EKB00AA315C00
220	10	5 x 11	0.24	1.45	156	0.5	EKB00AA322C00
330	10	6.3 x 11	0.24	1.00	220	0.8	EKB00BA333C00
470	10	6.3 x 11	0.24	0.68	262	0.8	EKB00BA347C00
680	10	10 x 12.5	0.24	0.47	433	1.5	EKB00CC368C00
1000	10	10 x 12.5	0.24	0.32	525	1.5	EKB00CC410C00
1500	10	10 x 16	0.25	0.22	604	2.0	EKB00DD415C00
2200	10	10 x 20	0.26	0.16	759	2.5	EKB00DE422C00
3300	10	13 x 20	0.29	0.12	1043	3.8	EKB00GE433C00
4700	10	13 x 25	0.31	0.09	1302	4.5	EKB00GG447C00
6800	10	16 x 25	0.36	0.07	1613	7.0	EKB00JG468C00
10000	10	16 x 35.5	0.42	0.06	2044	11.0	EKB00JL510C00
15000	10	18 x 35.5	0.52	0.05	2369	13.0	EKB00KL515C00
100	16	5 x 11	0.20	2.65	114	0.5	EKB00AA310D00
220	16	6.3 x 11	0.20	1.21	194	0.8	EKB00BA322D00
330	16	8 x 11.5	0.20	0.80	280	1.1	EKB00PB333D00
470	16	8 x 11.5	0.20	0.56	334	1.1	EKB00PB347D00
680	16	10 x 12.5	0.20	0.39	467	1.5	EKB00CC368D00
1000	16	10 x 16	0.20	0.27	621	2.0	EKB00DD410D00
1500	16	13 x 16	0.21	0.19	777	3.0	EKB00GD415D00
2200	16	13 x 20	0.22	0.13	963	3.8	EKB00GE422D00
3300	16	13 x 25	0.24	0.10	1220	4.5	EKB00GG433D00
4700	16	16 x 25	0.27	0.08	1540	7.0	EKB00JG447D00
6800	16	16 x 31.5	0.32	0.06	1864	9.0	EKB00JS468D00
10000	16	18 x 35.5	0.38	0.05	2294	13.0	EKB00KL510D00



**MICROCHIP**

**PIC18F2455/2550/4455/4550**

**28/40/44-Pin, High-Performance, Enhanced Flash,  
USB Microcontrollers with nanoWatt Technology**

**Universal Serial Bus Features:**

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

**Power-Managed Modes:**

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8  $\mu$ A typical
- Sleep mode currents down to 0.1  $\mu$ A typical
- Timer1 Oscillator: 1.1  $\mu$ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1  $\mu$ A typical
- Two-Speed Oscillator Start-up

**Flexible Oscillator Structure:**

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

**Peripheral Highlights:**

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 5.2 ns (TCY/16)
  - Compare is 16-bit, max. resolution 83.3 ns (TCY)
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I<sup>2</sup>C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

**Special Microcontroller Features:**

- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		IO	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EUSART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I <sup>2</sup> C™			
PIC18F2455	32K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	34K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

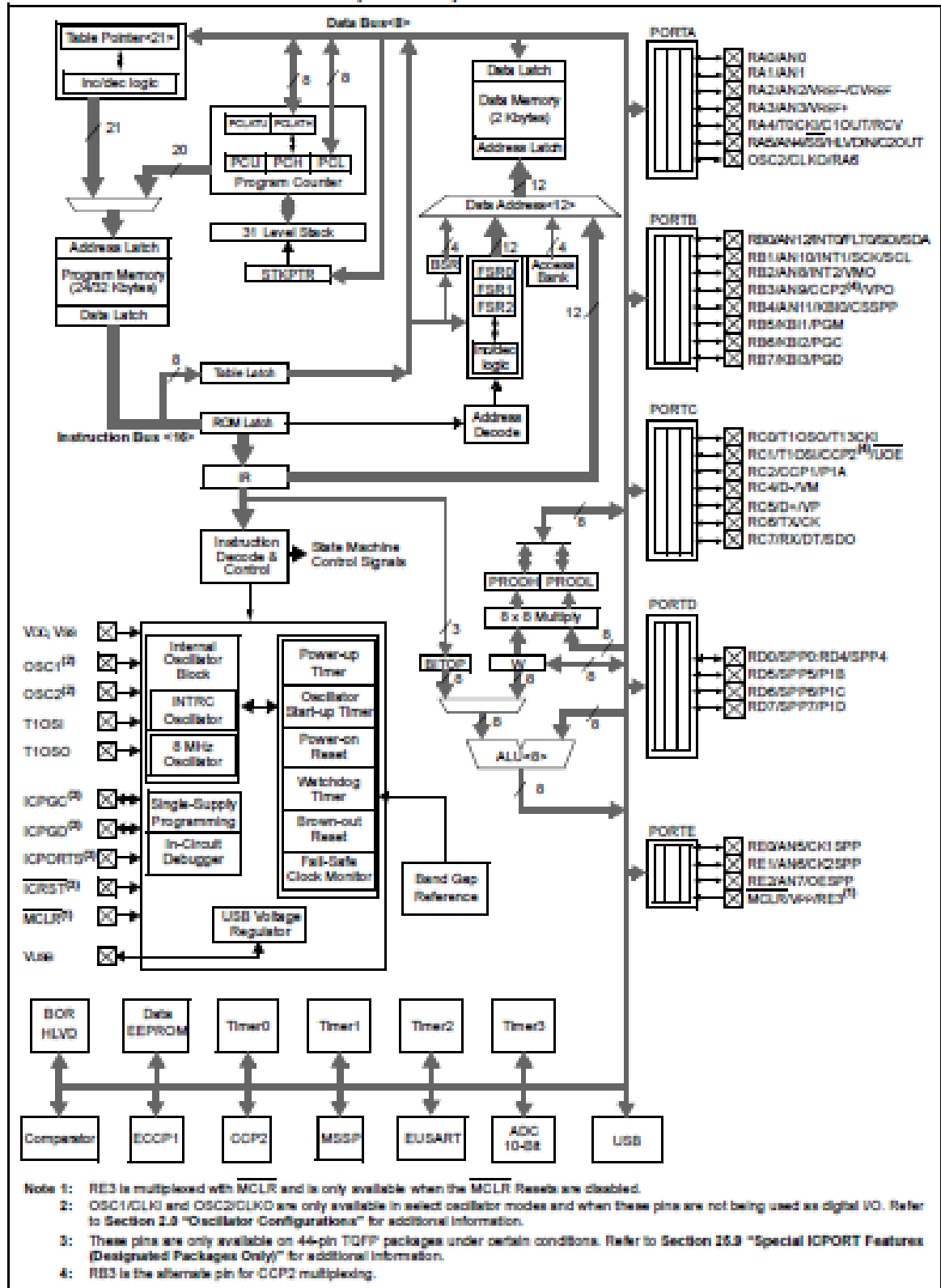
# PIC18F2455/2550/4455/4550

**TABLE 1-1: DEVICE FEATURES**

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/ Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

# PIC18F2455/2550/4455/4550

FIGURE 1-2: PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM



## **ANEXO B**

### **7.6 CÓDIGO DE PROGRAMA DEL MICROCONTROLADOR PIC 18F4550 PARA UN CIRCUITO RECEPTOR DE UNA AULA DETERMINADA DE LA FACULTAD.**

```

//----- SELECCIÓN DEL MICROCONTROLADOR A UTILIZAR -----
#include <18F4550.h>

//----- - CONFIGURACIÓN DE LOS FUSIBLES DEL MICROCONTROLADOR -----

#FUSES NOWDT, INTRC_IO, NOPROTECT, BROWNOUT, NOMCLR

//----- SELECCIÓN DE LA FRECUENCIA DE TRABAJO -----

#USE DELAY (CLOCK=4000000)//,RESTART_WDT)

//----- ACTIVACIÓN DE LA COMUNICACIÓN SERIAL -----

#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)

//----- LLAMADA DE LIBRERÍAS A UTILIZAR -----

#include <C:\Users\ROSITA\Desktop\Final\Antonio Bolívar Guerrero Aseicha\Programacion de Micros\Rx-Teclado-
LCDM_LCD.c>

#include <C:\Users\ROSITA\Desktop\Final\Antonio Bolívar Guerrero Aseicha\Programacion de Micros\Rx-Teclado-
LCD\KBD4por4.c>

//----- CONFIGURACIÓN DE PUERTOS -----

#BYTE PORTA=0xf80
#BYTE PORTB=0xf81
#BYTE PORTC=0xf82
#BYTE PORTE=0xf84

//----- VARIABLES DE PINES DE LOS PUERTOS -----

#bit aula_utilizada=portc.1
#bit aula_disponible=portc.2
#bit puerta=portc.0
#bit parlante=porta.6
#bit parar_hora=porta.0
#bit conserje=porta.1

//----- MÉTODOS -----

void clave_de_ingreso(void);
void bloquear_puerta(void);
void inicio(void);
void horas(void);
void escritura_clave(void);
void escritura_clave_conserje(void);
void confirmacion_datos_conserje(void);
void horas_conserje(void);
void cadena(void);

//----- VARIABLES GLOBALES -----

char tecla_presionada;

```

```

char clave_actual[10];
char cadena_actual[27];
char cadena_temporal[27];
char cadena_enviar[27];
char hora_final[8];
char tiempo;
int intentos=0,bandera=0,verificador_hora_final;
int hs,min,seg,k,i,j,contador,salir;
char confirmacion_conserje[27];

//----- INTERRUPCIÓN -----

#int_rda
void rda_isr(void)
{
    if(kbhit()) //Si ha recibido un caracter por la serial
    {
        for(i=0;i<27;i++)
        {
            cadena_temporal[i]=getc(); //Llena la cadena temporal con los caracteres recibidos
        }
    }
    if(cadena_temporal[0]=='I') //Compara el caracter de la cadena temporal de la posición 0 con 'A'
    {
        for(i=0;i<27;i++)
        {
            cadena_actual[i]=cadena_temporal[i]; //Llena la cadena actual con los datos de la cadena temporal
            cadena_enviar[i]=cadena_actual[i]; //Llena la cadena enviar con los datos de la cadena actual
        }
        if(cadena_actual[2]=='a') //Compara el caracter de la cadena actual de la posición 2 con 'a'
        {
            for(i=4;i<14;i++)
            {
                clave_actual[i-4] = cadena_actual[i]; //Llena la cadena clave actual con los datos de la cadena actual
            }
            for(i=19;i<27;i++)
            {
                hora_final[i-19] = cadena_actual[i]; //Llena la cadena hora final con los datos de la cadena actual
            }
        }
    }
}

```



```

    }

    bandera=1; //Permite ingresar al menú principal del programa principal
}
}
else if(cadena_temporal[0]=='Z') //Compara el caracter de la cadena temporal de la posición 0 con 'Z'
{
    verificador_hora_final=0; //Inicializa variable global verificador hora final en 0
    for(i=19;i<27;i++) //Llena la cadena hora final con los datos de la cadena temporal
    {
        if(hora_final[i-19] == cadena_temporal[i])
        {
            verificador_hora_final++; //Incremento de la variable global verificador hora final en +1
        }
    }
    if(verificador_hora_final==8) // Compara el tamaño de la variable global verificador hora final si es
    // igual a 8 caracteres
    {
        tiempo=1; // Establece la variable global tiempo en 1
    }
}
}
}

//----- MÉTODO DE ENVIO DE LA TRAMA -----

void Cadena(int parametro)
{
    char confirmar[27]; //Inicializa variable local como vector
    if(parametro==1) //Compara si el parámetro recibido es igual a 1
    {
        cadena_enviar[2]='k'; //Modifica el dato de la cadena enviar de la posición 2 en 'k'
        for(i=0; i<27; i++)
        {
            putc(cadena_enviar[i]); //Envía la trama cadena enviar por el puerto
        }
    }
    else if(parametro==2) //Compara si el parámetro recibido es igual a 2
    {
        disable_interrupts(int_rda); //Deshabilita la interrupción serial
        disable_interrupts(global); //Deshabilita interrupciones globales
    }
}

```

```

cadena_enviar[2]='c'; //Modifica el dato de la cadena enviar de la posición 2 en 'c'
lcd_putc("\fESPERANDO DATOS \nPOR FAVOR ESPERE"); //Mensaje para la LCD

for(i=0; i<27; i++)
{
    putc(cadena_enviar[i]); //Envía la trama cadena enviar por el puerto
}

do
{
    if(kbhit() //Si ha recibido un caracter por la serial
    {
        for(i=0;i<27;i++)
        {
            confirmar[i]=getc(); //Llena la cadena confirmar con los datos recibidos del puerto
        }
    }
}

while((confirmar[0]!='\n')); //Mantiene en el ciclo si el dato de la variable confirmar en la posición 0
                             //es igual a '\n'

if(confirmar[2]=='v') //Compara si el dato de la variable confirmar en la posición 2 es igual a 'v'
{
    lcd_putc("\n CORRECTO"); //Mensaje para la LCD
    delay_ms(1000); //Tiempo de espera
}

else
{
    lcd_putc("\f LOS DATOS SON :\n FALSOS"); //Mensaje para la LCD
    delay_ms(1000); //Tiempo de espera
}

enable_interrupts(int_rda); //Deshabilita la interrupción serial
enable_interrupts(global); //Deshabilita interrupciones globales
}

else if(parametro==3) //Compara si el parámetro recibido es igual a 3
{
    //--Establece un valor en la variable cadena enviar en una determinada posición--

```

```

cadena_enviar[0]='l';
cadena_enviar[1]='+';
cadena_enviar[2]='d';
cadena_enviar[3]='+';
for(i=0;i<14;i++)
{
    cadena_enviar[i+4]=clave_actual[i];           //Llena la cadena enviar con los datos de la clave actual
}
cadena_enviar[14]='+';
cadena_enviar[15]='1';
cadena_enviar[16]='+';
cadena_enviar[17]='L';
cadena_enviar[18]='+';
cadena_enviar[19]='0';
cadena_enviar[20]='0';
cadena_enviar[21]=':.';
cadena_enviar[22]='0';
cadena_enviar[23]='0';
cadena_enviar[24]=':.';
cadena_enviar[25]='0';
cadena_enviar[26]='0';
for(i=0; i<27; i++)
{
    putc(cadena_enviar[i]);                       //Envía la trama cadena enviar por el puerto
}
}
else if(parametro==4)                            //Compara si el parámetro recibido es igual a 4
{
    for(i=19;i<27;i++)
    {
        cadena_enviar[i]=confirmacion_conserje[i]; //Llena la cadena enviar con los datos de la confirmación conserje
    }
    cadena_enviar[2]='c';                          //Modifica el dato de la cadena enviar de la posición 2 en 'c'
    lcd_putc("\fESPERANDO DATOS \nPOR FAVOR ESPERE"); //Mensaje para la LCD
    for(i=0; i<27; i++)
    {

```

```

    putc(cadena_enviar[i]);          //Envía la trama cadena enviar por el puerto
}
do
{
    if(kbhit())                    //Si ha recibido un caracter por la serial
    {
        for(i=0;i<27;i++)
        {
            confirmar[i]=getc();    //Llena la cadena confirmar con los datos recibidos del puerto
        }
    }
}
while((confirmar[0]!='\n'));       //Mantiene en el ciclo si el dato de la variable confirmar en la posición 0
                                  //es igual a '\n'

if(confirmar[2]=='v')              //Compara si el dato de la variable confirmar en la posición 2 es igual a 'v'
{
    lcd_putc("\n  CORRECTO");      //Mensaje para la LCD
    delay_ms(1000);                //Tiempo de espera
}
else
{
    lcd_putc("\n  LOS DATOS SON : \n  FALSOS"); //Mensaje para la LCD
    delay_ms(1000);                //Tiempo de espera
}
for(i=0;i<27;i++)
{
    confirmar[i]="";                //Borra los datos de la cadena confirmar
}
}

//----- PROGRAMA PRINCIPAL -----

void main(void)
{
    enable_interrupts(int_rda);     //Habilita la interrupción serial
    enable_interrupts(global);     //Habilita interrupciones globales

    lcd_init();                    //Inicializa lcd
    kbd_init();                    //Inicializa teclado
}

```

```

set_tris_a(0b00000000); //Establece el estado del puerto 0 salida y 1 entradas
set_tris_b(0b11111111);
set_tris_c(0b10000000);
set_tris_e(0b00001000);

porta=0; //Inicializa los puertos A,B,C,E en 0
portb=0;
portc=0;
porte=0;

port_b_pullups(TRUE); //Activación de la configuración Pull-Up (resistencias) del puerto B
lcd_putc("\f BIENVENIDOS\n AULA-302"); //Mensaje para la LCD
aula_disponible=1;

do //Bucle infinito
{
if(bandera==1) //Si los caracteres de la cadena actual fueron A y 1 la bandera se
establece en 1
{
aula_disponible=0; //Apaga un led
aula_utilizada=1; //Prende un led
Cadena(1); //Inicializa el método de envío de tramas con el parámetro igual a 1
inicio(); //Ir al método Inicio
Cadena(2); //Inicializa el método de envío de tramas con el parámetro igual a 2
aula_utilizada=0; //Apaga un led
aula_disponible=1; //Prende un led
puerta=0; //Cierra la puerta
bandera=0; //Reinicia la bandera y se pone en 0
intentos=0; //Los intentos al digitar la clave se reinician en 0
conserje=0; //Establece la variable global conserje en 0
lcd_putc("\fHORA FINALIZADA \n GRACIAS..!!!"); //Mensaje para la LCD
delay_ms(1000);
lcd_putc("\f BIENVENIDOS\n AULA-302"); //Mensaje para la LCD
for(i=0;i<27;i++)
{
cadena_enviar[i]=""; //Borra los datos de la cadena enviar
cadena_actual[i]=""; //Borra los datos de la cadena actual
}
for(i=0;i<=9;i++)
{

```

```

    clave_actual[i] = "";           //Borra los datos de la clave actual
}
for(i=0;i<=7;i++)
{
    hora_final[i] = "";           //Borra los datos de la hora final
}
}
if(conserje==1)                   //Compara si la variable global conserje es igual a 1
{
    disable_interrupts(int_rda);   //Deshabilita la interrupción serial
    disable_interrupts(global);    //Deshabilita interrupciones globales
    escritura_clave_conserje();    //Inicializa el método escritura de la clave del conserje
    if(salir==0)                   //Compara si la variable global salir es igual a 0
    {
        Cadena(3);                 //Inicializa el método de envío de tramas con el parámetro igual a 3
        confirmacion_datos_conserje(); //Inicializa el método de confirmación datos conserje
    }
    aula_disponible=1;              //Prende un led
    aula_utilizada=0;               //Apaga un led
    puerta=0;                       //Cierra la puerta
    bandera=0;                      //Reinicia la bandera y se pone en 0
    intentos=0;                     //Los intentos al digitar la clave se reinician en 0
    conserje=0;                     //Establece la variable global conserje en 0
    lcd_putc("\fHORA FINALIZADA \n GRACIAS..!!!"); //Mensaje para la LCD
    delay_ms(1000);
    lcd_putc("\f BIENVENIDOS\n AULA-302"); //Mensaje para la LCD
    enable_interrupts(int_rda);     //Habilita la interrupción serial
    enable_interrupts(global);     //Habilita interrupciones globales
    for(i=0;i<27;i++)
    {
        cadena_enviar[i]="";       //Borra los datos de la cadena enviar
    }
    for(i=0;i<=9;i++)
    {
        clave_actual[i] = "";      //Borra los datos de la clave actual
    }
}

```

```

    }
}
while(true);
}

//----- MÉTODO DE INICIO -----

void inicio(void)
{
    for(k=0;k<1;) //Regresa si aun no está la clave correcta
    {
        lcd_putc("\fA->INGRESE CLAVE\nB->BLOQUEAR PUERTA"); //Mensaje para la LCD
        do
        {
            tecla_presionada=kbd_getc(); //Obtiene la tecla presionada y la guarda
        }
        while(tecla_presionada==0); //Realiza el bucle mientras esté pulsada una tecla
        if(tecla_presionada=='A') //Si la tecla guardada es A realiza el método clave de ingreso
        {
            clave_de_ingreso();
        }
        if(tecla_presionada=='B') //Si la tecla guardada es B realiza el método bloquear puerta
        {
            bloquear_puerta();
        }
    }
}

//----- MÉTODO ESCRITURA DE LA CLAVES -----

void escritura_clave(void)
{
    j=0,contador=0, //Inicializa variables en 0
    lcd_putc("\fINGRESE CLAVE:\n"); //Mensaje para la LCD
    while(j<=9)
    {
        tecla_presionada=kbd_getc(); //Guarda el caracter de la tecla presionada
        if(tecla_presionada!=0) //Si se presionó una tecla visualiza y compara con la de la
            clave
        {
            lcd_putc('*'); //Visualiza el caracter *
        }
    }
}

```

```

if(tecla_presionada==clave_actual[j]) //Si la tecla presionada es igual al de la clave incrementa la
                                     //variable contador

{
    contador++; //Incremento de la variable contador
}

if(tecla_presionada=='D') //Si la tecla presionada es igual al de la clave incrementa la
                           //variable contador

{
    lcd_putc("\n CLAVE BORRADA"); //Incremento de la variable contador
    delay_ms(1000); //Tiempo de espera
    lcd_putc("\fINGRESE CLAVE:\n");
    j=-1; //Decremento la variable j y obtiene el valor de -1
    contador=0;
}

j=j+1; //Incremento de la variable j
}
}

//----- MÉTODO ESCRITURA DE LA CLAVES CONSERJE-----

void escritura_clave_conserje(void)
{
    j=0,contador=0,salir=0; //Inicializa variables en 0
    lcd_putc("\fINGRESE CLAVE:\n"); //Mensaje para la LCD
    while(j<=9)
    {
        tecla_presionada=kbd_getc(); //Guarda una tecla presionada
        if(tecla_presionada!=0) //Si se presionó una tecla visualiza y compara con la de la clave
        {
            lcd_putc('*'); //Visualiza el caracter *
            if(tecla_presionada=='D') //Si la tecla presionada es igual al de la clave incrementa la variable
                                     //contador
            {
                lcd_putc("\n CLAVE BORRADA"); //Mensaje para la LCD
                delay_ms(1000); //Tiempo de espera
                lcd_putc("\fINGRESE CLAVE:\n"); //Mensaje para la LCD
                j=-1; //Decremento la variable j y obtiene el valor de -1
                contador=0; //Estabiliza la variable contador en 0
            }
        }
    }
}

```



```

if(tecla_presionada=='C') //Si la tecla presionada es igual al de la clave incrementa la variable
                           contador

{
  lcd_putc("\nPROCESO CANCELADO"); //Mensaje para la LCD
  delay_ms(1000); //Tiempo de espera
  lcd_putc("\fINGRESE CLAVE:\n"); //Mensaje para la LCD
  j=9; //Incremento de la variable j y obtiene el valor de 9
  contador=0; //Estabiliza la variable contador en 0
  salir=1; //Estabiliza la variable salir en 0
}
else
{
  clave_actual[j]=tecla_presionada; //Ingresa el caracter presionado en la variable clave actual en la posición j
  contador++; //Incremento de la variable contador
}
j=j+1; //Incremento de la variable j
}
}
}

//----- MÉTODO CONFIRMACIÓN DATOS CONSERJE -----

void confirmacion_datos_conserje(void)
{
  lcd_putc("\fESPERANDO DATOS \nPOR FAVOR ESPERE"); //Mensaje para la LCD
  do
  {
    if(kbhit()) //Si ha recibido un caracter por la serial
    {
      for(i=0;i<27;i++)
      {
        confirmacion_conserje[i]=getc(); //Llena la cadena temporal con los caracteres recibidos
      }
    }
  }
  while((confirmacion_conserje[0]!='!')); //Mantiene en el ciclo si el dato de la variable confirmación
  if(confirmacion_conserje[2]=='v') //Compara si el dato de la variable confirmación conserje en la
  {
    //Compara si el dato de la variable confirmación conserje en la
    posición 2 es igual a 'v'
  }
}

```

```

    lcd_putc("\f CLAVE CORRECTA \n PUERTA ABIERTA");           //Mensaje para la LCD
    delay_ms(1000);                                           //Tiempo de espera
    aula_utilizada=1;                                         //Prende un led
    aula_disponible=0;                                        //Apaga un led
    puerta=1;                                                //Abre la puerta
    delay_ms(1000);                                           //Tiempo de espera
    horas_conserje();                                         //Inicializa el método horas conserje
    Cadena(4);                                               //inicializa el método de envío de tramas con el parámetro igual a 4
}
else
{
    lcd_putc("\f LOS DATOS SON :\n FALSOS");                 //Mensaje para la LCD
    delay_ms(1000);                                         //Tiempo de espera
}
for(i=0;i<27;i++)
{
    confirmacion_conserje[i]="";                             //Borra los datos de la confirmación conserje actual
}
}

//----- MÉTODO DEL INGRESO DE LA CLAVE -----

void clave_de_ingreso(void)
{
    escritura_clave();                                       //Realiza el método de escritura de la clave
    if(contador==10)                                        //Si el contador es igual a 10 visualiza mensajes, abre la puerta
    y cuenta el tiempo en el aula
    {
        lcd_putc("\n CLAVE CORRECTA");                       //Mensaje para la LCD
        delay_ms(1000);                                       //Tiempo de espera
        puerta=1;                                             //Abre la puerta
        lcd_putc("\n PUERTA ABIERTA");                       //Mensaje para la LCD
        delay_ms(1000);                                       //Tiempo de espera
        contador=0;                                           //Se reinicia el contador
        horas();                                              //Realiza el método de las horas en el aula
        k=1;                                                  //Establece la variable k en 1 para salir del método de inicio
    }
    else
        //En caso de no cumplirse la condición del contador igual a 10 se cuenta
        los intentos realizados

```

```

{
    intentos++;                                //Incrementa la variable intentos en 1
    printf(lcd_putc, "\fCLAVE INCORRECTA\nINTENTOS= %U", intentos);                //Mensaje para la LCD con el
                                                                                   número de intentos realizados

    delay_ms(1000);                            //Tiempo de espera

    if(intentos==3)                            //Si los intentos realizados son igual a 3 se envían mensajes a la LCD, se
                                                                                   bloquea el teclado, la puerta se cierra y envía ACK de información

    {
        lcd_putc("\fTECLADO\nBLOQUEADO...!!!");    //Mensaje para la LCD
        puerta=0;                                //Cierra la puerta
        delay_ms(2000);                          //Tiempo de espera
        lcd_putc("\f AULA 302\n CERRADA");        //Mensaje para la LCD
        delay_ms(1000);                          //Tiempo de espera
        k=1;                                     //Establece la variable k en 1 para salir del método de inicio
        intentos=0;                              //Se reinicia la variable intentos en 0
    }

    else                                        //Si no se cumple la condición de los intentos igual a 3 regresa
                                                                                   al método de inicio

    {
        k=0;                                    //Establece la variable k en 0
    }
}
}

//----- MÉTODO PARA BLOQUEAR LA PUERTA -----

void bloquear_puerta()
{
    escritura_clave();                          //Realiza el método de escritura de la clave

    if(contador==10)                            //Si el contador es igual a 10 visualiza mensajes, abre la puerta y cuenta
                                                                                   el tiempo en el aula

    {
        lcd_putc("\f AULA-302\n CLAVE CORRECTA");    //Mensaje para la LCD
        delay_ms(1000);                            //Tiempo de espera
        puerta=0;                                //Cierra la puerta
        lcd_putc("\f AULA-302\n PUERTA CERRADA");    //Mensaje para la LCD
        contador=0;                              //Se reinicia el contador
        k=1;                                     //Establece la variable k en 1 para salir del método de inicio
    }

    else                                        //En caso de no cumplirse la condición del contador igual a 10 se cuenta
                                                                                   los intentos realizados
}

```

```

{
    intentos++;                                //Incrementa la variable intentos en 1

    printf(lcd_putc, "\fCLAVE INCORRECTA\nINTENTOS= %U", intentos);    //Mensaje para la LCD con el número de
                                                                           intentos realizados

    delay_ms(1000);                            //Tiempo de espera

    if(intentos==3)                            //Si los intentos realizados son igual a 3 se envían mensajes a la LCD,
                                                                           se bloquea el teclado, la puerta se cierra y envía ACK de información

    {
        lcd_putc("\fCLAVE INCORRECTA");        //Mensaje para la LCD

        delay_ms(1000);                        //Tiempo de espera

        k=0;                                   //Establece la variable k en 0 y regresa al menú de
inicio
        intentos=0;                            //Se reinicia la variable intentos en 0

    }

}

}

//----- MÉTODO DE HORAS EN EL AULA -----

void horas()
{
    int numero;

    char b;

    lcd_putc("\fHORAS RESERV: \nTIEMPO: ");    //Mensaje para la LCD

    lcd_gotoxy(15,1);                          //Posición del siguiente mensaje en la LCD

    lcd_putc(cadena_actual[15]);               //Ubicación del número de horas en la LCD

    lcd_gotoxy(11,2);                          //Posición del siguiente mensaje

    lcd_putc(":");                              //Muestra la separación de las horas con los minutos en la LCD

    lcd_gotoxy(14,2);                          //Posición del siguiente mensaje en la LCD

    lcd_putc(":");                              //Muestra la separación de los minutos con los segundos en la
LCD

    b = cadena_actual[15];                     //b" obtiene el valor de la variable cadena actual de la posición
15

    numero = b-48;                             //Convierte a decimal el valor de "b" y lo ingresa en la variable
número

    for(hs=0;hs<numero;hs++)                  //Ciclo repetitivo para contar el número de horas en el aula

    {
        lcd_gotoxy(9,2);                      //Posición del siguiente mensaje en la LCD

        printf(lcd_putc, "%02u", hs);         //Muestra las horas en la LCD

        for(min=0;min<=59;min++)              //Ciclo repetitivo para contar el número de minutos en el aula

        {

```

```

lcd_gotoxy(12,2); //Posición del siguiente mensaje en la LCD
printf(lcd_putc,"%02u"min); //Muestra la los minutos en la LCD
for(seg=0;seg<=59;seg++) //Ciclo repetitivo para contar el número de segundos en el aula
{
    lcd_gotoxy(15,2); //Posición del siguiente mensaje en la LCD
    printf(lcd_putc,"%02u"seg); //Muestra la los segundos en la LCD
    delay_ms(970); //Periodo de tiempo para contar los segundos a transcurrir
    if(parar_hora==1) //Si se presiona el botón de parar las horas de clase pedirá la
                    //clave de confirmación y verifica si es correcta
    {
        escritura_clave(); //Realiza el método de escritura de la clave
        if(contador==10) //Si el contador es igual a 10 visualiza mensajes, abre la puerta
                        //y cuenta el tiempo en el aula
        {
            seg=60; //Se desborda la variable seg en 60 para salir del ciclo
            min=60; //Se desborda la variable min en 60 para salir del ciclo
            hs=numero; //Se desborda la variable hs en 60 para salir del ciclo
            parar_hora=0; //Se establece el pin A3 del puerto A en 0 para esperar un
                          //futuro reinicio del sistema
            contador=0; //Se reinicia el contador
        }
        else //En caso de no cumplirse la condición del contador igual a 10
            se visualiza un mensaje
        {
            lcd_putc("\fLA CLAVE NO ES \n CORRECTA"); //Mensaje para la LCD
            delay_ms(1000); //Tiempo de espera
            lcd_putc("\fHORAS RESERV: \nTIEMPO: "); //Mensaje para la LCD
            lcd_gotoxy(15,1); //Posición del siguiente mensaje en la LCD
            lcd_putc(cadena_actual[15]); //Ubicación del número de horas en la LCD
            lcd_gotoxy(11,2); //Posición del siguiente mensaje
            lcd_putc(":"); //Muestra la separación de las horas con los minutos
                          //en la LCD
            lcd_gotoxy(14,2); //Posición del siguiente mensaje en la LCD
            lcd_putc(":");
            lcd_gotoxy(9,2); //Posición del siguiente mensaje en la LCD
            printf(lcd_putc,"%02u"hs); //Muestra las horas en la LCD
            lcd_gotoxy(12,2);
            printf(lcd_putc,"%02u"min);
        }
    }
}

```

```

    }
    if(tiempo==1) //Si la variable tiempo es igual a 1 visualiza el tiempo
                    //restante en el aula utilizada
    {
        lcd_putc("\fTIEMPO RESTANTE\n 5 MINUTOS"); //Mensaje para la LCD
        parlante=1; //Prende el parlante
        delay_ms(1000); //Tiempo de espera
        lcd_putc("\fHORAS RESERV: \nTIEMPO: "); //Mensaje para la LCD
        lcd_gotoxy(15,1); //Posición del siguiente mensaje en la LCD
        lcd_putc(cadena_actual[15]); //Ubicación del número de horas en la LCD
        lcd_gotoxy(11,2); //Posición del siguiente mensaje
        lcd_putc("."); //Muestra la separación de las horas con los minutos
                        //en la LCD
        lcd_gotoxy(14,2); //Posición del siguiente mensaje en la LCD
        lcd_putc(".");
        lcd_gotoxy(9,2); //Posición del siguiente mensaje en la LCD
        printf(lcd_putc,"%02u"hs); //Muestra las horas en la LCD
        lcd_gotoxy(12,2);
        printf(lcd_putc,"%02u"min);
        parlante=0; //Apaga el parlante
        tiempo=0; //Se restablece la variable tiempo en 0
    }
}
}
}

//lcd_putc("\fHORA FINALIZADA \n GRACIAS..!!!"); //Mensaje para la LCD
//delay_ms(1000); //Tiempo de espera
}

//----- MÉTODO DE HORAS DEL CONSERJE -----

void horas_conserje()
{
    int numero;
    char b;
    lcd_putc("\fLIMPIEZA: \nTIEMPO: "); //Mensaje para la LCD
    lcd_gotoxy(15,1); //Posición del siguiente mensaje en la LCD
    lcd_putc(confirmacion_conserje[15]); //Ubicación del número de horas en la LCD
    lcd_gotoxy(11,2); //Posición del siguiente mensaje
}

```

```

lcd_putc(":"); //Muestra la separación de las horas con los minutos en la LCD
lcd_gotoxy(14,2); //Posición del siguiente mensaje en la LCD
lcd_putc(":"); //Muestra la separación de los minutos con los segundos en la LCD
b = confirmacion_conserje[15]; //b" obtiene el valor de la variable cadena actual de la posición 15
numero = b-48; //Convierte a decimal el valor de b y lo ingresa en la variable número
for(hs=0;hs<numero;hs++) //Ciclo repetitivo para contar el número de horas en el aula
{
    lcd_gotoxy(9,2); //Posición del siguiente mensaje en la LCD
    printf(lcd_putc,"%02u"hs); //Muestra las horas en la LCD
    for(min=0;min<=59;min++) //Ciclo repetitivo para contar el número de minutos en el aula
    {
        lcd_gotoxy(12,2); //Posición del siguiente mensaje en la LCD
        printf(lcd_putc,"%02u"min); //Muestra la los minutos en la LCD
        for(seg=0;seg<=59;seg++) //Ciclo repetitivo para contar el número de segundos en el aula
        {
            lcd_gotoxy(15,2); //Posición del siguiente mensaje en la LCD
            printf(lcd_putc,"%02u"seg); //Muestra la los segundos en la LCD
            delay_ms(970); //delay_ms(970); //Periodo de tiempo para contar los segundos a transcurrir
            if(parar_hora==1) //Si se presiona el botón de parar las horas de clase pedirá la clave de confirmación y verifica si es correcta
            {
                j=0,contador=0; //Inicializa variables en 0
                lcd_putc("\fINGRESE CLAVE:\n"); //Mensaje para la LCD
                while(j<=9)
                {
                    tecla_presionada=kbd_getc(); //Guarda un tecla presionada
                    if(tecla_presionada!=0) //Si se presionó una tecla visualiza y compara con la de la clave
                    {
                        lcd_putc("*"); //Visualiza el caracter *

                        if(tecla_presionada==clave_actual[j]) //Si la tecla presionada es igual al de la clave incrementa la variable contador
                        {
                            contador++; //Incremento de la variable contador
                        }
                    }
                    else if(tecla_presionada=='D') //Si la tecla presionada es igual al de la clave incrementa la variable contador
                    {

```

```

        lcd_putc("\n CLAVE BORRADA");           //Incremento de la variable contador
        delay_ms(1000);
        lcd_putc("\fINGRESE CLAVE:\n");
        j=-1;
        contador=0;
    }
    j=j+1;           //Incremento de la variable j
}
}
if(contador==10)           //Si el contador es igual a 10 visualiza mensajes,
                           abre la puerta y cuenta el tiempo en el aula
{
    seg=60;           //Se desborda la variable seg en 60 para salir del ciclo
    min=60;           //Se desborda la variable min en 60 para salir del ciclo
    hs=numero;           //Se desborda la variable hs en 60 para salir del ciclo
    parar_hora=0;     //Se establece el pin A3 del puerto A en 0 para esperar un futuro reinicio del sistema
    contador=0;       //Se reinicia el contador
}
else           //En caso de no cumplirse la condición del contador igual a 10 se visualiza un mensaje
{
    lcd_putc("\fLA CLAVE NO ES \n  CORRECTA");           //Mensaje para la LCD
    delay_ms(1000);           //Tiempo de espera
    lcd_putc("\fLIMPIEZA: \nTIEMPO: ");           //Mensaje para la LCD
    lcd_gotoxy(15,1);           //Posición del siguiente mensaje en la LCD
    lcd_putc(confirmacion_conserje[15]);           //Ubicación del número de horas en la LCD
    lcd_gotoxy(11,2);           //Posición del siguiente mensaje
    lcd_putc(":");           //Muestra la separación de las horas con los minutos en la LCD
    lcd_gotoxy(14,2);           //Posición del siguiente mensaje en la LCD
    lcd_putc(":");
    lcd_gotoxy(9,2);           //Posición del siguiente mensaje en la LCD
    printf(lcd_putc,"%02u"hs);           //Muestra las horas en la LCD
    lcd_gotoxy(12,2);           //Posición del siguiente mensaje en la LCD
    printf(lcd_putc,"%02u"min);
}
}
}
}
}

```



```
}  
//lcd_putc("\fHORA FINALIZADA \n GRACIAS..!!!");           //Mensaje para la LCD  
//delay_ms(1000);  
}
```

## **ANEXO C**

### **7.7 PROCESO DE ELABORACIÓN DE LOS CIRCUITOS ELECTRÓNICOS DEL SISTEMA DE CONTROL DE ACCESO PARA LAS AULAS DE LA FACULTAD**

## 7.7.1 Circuitos electrónicos

La elaboración de un circuito electrónico se realiza de la siguiente manera, tomando en cuenta los siguientes pasos, tal como se los realizó para el circuito transmisor de este trabajo de titulación.

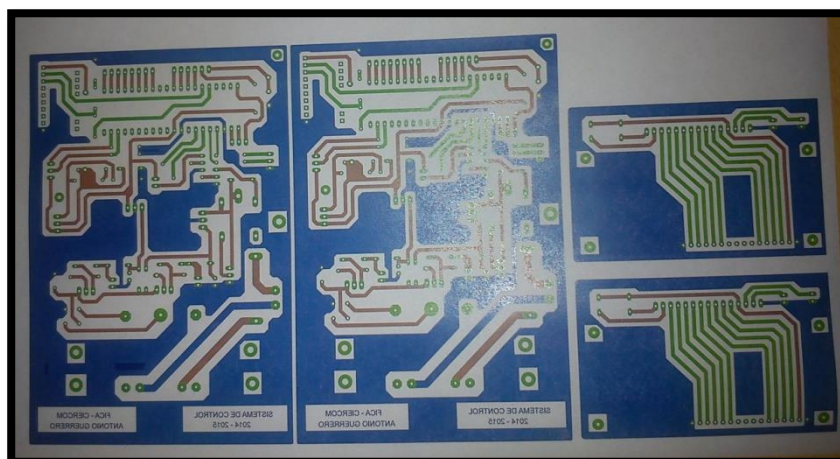
### 7.7.1.1 Impresión del circuito

Luego de diseñar la pista del circuito transmisor en el programa de Eagle se procede a imprimir dicho circuito, esta impresión debe ser realizada en una hoja especial llamada asetato, y en una impresora que sea a láser, ya que luego pasará por otro proceso de impregnación, el cual debe soportar altas temperaturas. En la siguiente figura se puede apreciar este proceso.

Figura 115. Impresión del circuito transmisor.



Figura 116. Impresión del circuito receptor.



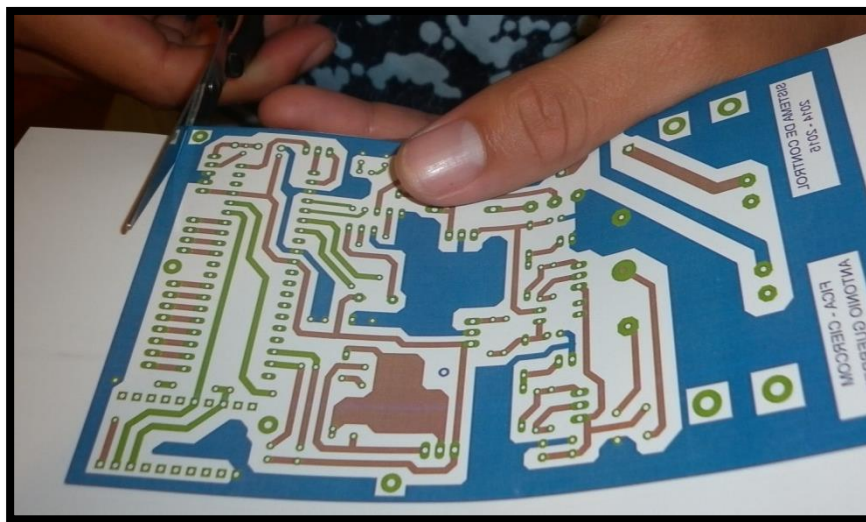
### 7.7.1.2 Recortar el circuito impreso

Una vez impreso el circuito se recorta el mismo al margen de la pista para proceder a colocarlo en la baquelita, tal como se puede apreciar en la figura siguiente.

Figura 117. Recorte del circuito transmisor.



Figura 118. Recorte del circuito receptor.



### 7.7.1.3 Medición de la baquelita con el circuito impreso

A continuación se coloca el circuito impreso en la parte frontal de la baquelita (en la parte del cobre) y se traza un margen con un marcador o se mide las

dimensiones del circuito para posteriormente recortar la baquelita. Este proceso se indica en la figura siguiente.

Figura 119. Medición del circuito transmisor.

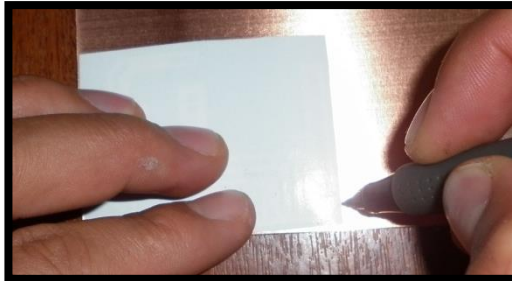


Figura 120. Medición del circuito receptor.



#### **7.7.1.4 Recortar la baquelita**

Una vez realizado el margen del circuito en la baquelita, se procede a recotar con una sierra, hay que asegurarse que las cuchillas estén correctamente alineadas, para que el cobre de la baquelita no se levante de la placa, ya que esto

puede presentar problemas. En la siguiente figura se puede apreciar este proceso de corte.

Figura 121. Recorte de la baquelita del circuito transmisor.



Figura 122. Recorte de la baquelita del circuito receptor.



#### **7.7.1.5 Pulir la baquelita**

Una vez recortada la baquelita se procede a pulir con una lija muy delgada o con un lustre fino, para fijar el cobre con la impresión del circuito o retirar óxido de la baquelita si así fuera el caso, tal como se indica en la figura siguiente.

Figura 123. Lijado de la baquelita del circuito transmisor.



Figura 124. Lijado de la baquelita del circuito receptor.



#### **7.7.1.6 Ubicar y fijar el circuito en la baquelita**

Este proceso es muy importante ya que, en caso de no fijar bien la impresión del circuito con la baquelita, el circuito puede tener complicaciones de funcionamiento debido a que la pista no estaría completa. Para evitar este problema se decide fijar al circuito con cinta adhesiva para que la impresión no se mueva al momento de realizar el proceso de impregnación de la pista.



Figura 125. Ubicación de la pista impresa en la baquelita del circuito transmisor.

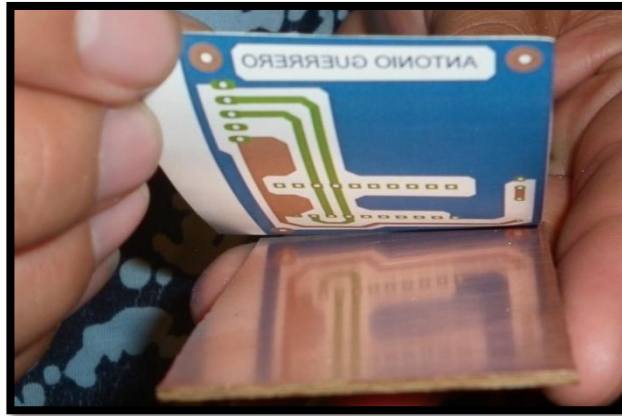


Figura 126. Fijación de la pista impresa a la baquelita del circuito transmisor.



Figura 127. Ubicación de la pista impresa en la baquelita del circuito receptor.

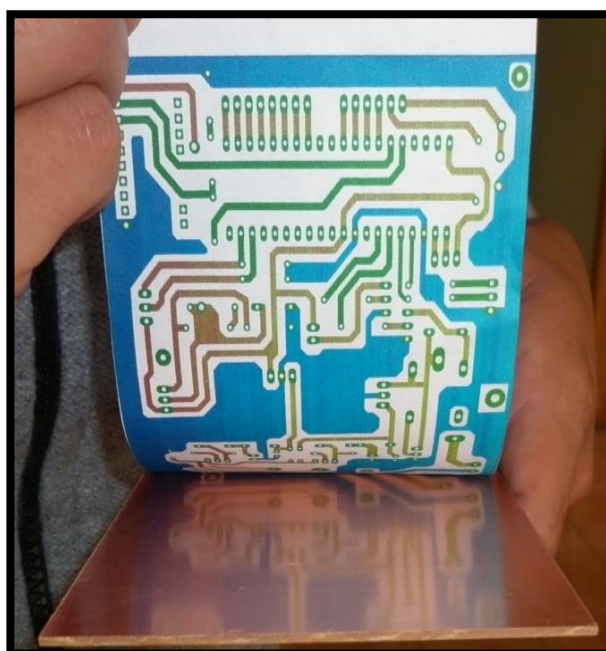
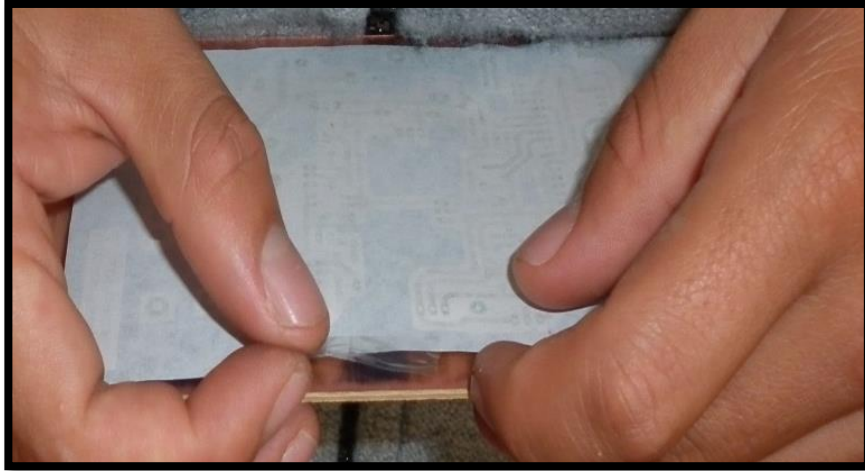




Figura 128. Fijación de la pista impresa a la baquelita del circuito receptor.



#### 7.7.1.7 Impregnar la pista del circuito en la baquelita

Para que la pista quede impresa totalmente en la baquelita, se debe poner a temperaturas altas de calor con una presión determinada con una plancha casera para que la pista del circuito quede totalmente pegada en la baquelita. Lo dicho, se puede apreciar en la siguiente figura.

Figura 129. Impregnación del circuito transmisor en la baquelita.



Figura 130. Impregnación del circuito receptor en la baquelita.



#### 7.7.1.8 Desmontar el papel de acetato

Una vez planchada la pista en la baquelita se procede a retirar el papel restante de la baquelita, para ello se debe colocar la baquelita en agua fría y esperar a que el papel se retire completamente sin forzar el proceso ya que al tratar de retirar manualmente tiende a levantarse la pista de la placa, por lo que el proceso será interrumpido.

Figura 131. Desmontaje del papel de las placas en los diferentes circuitos.



### 7.7.1.9 Separar el cobre adicional del circuito

Una vez terminado el proceso de impregnación del circuito, se coloca al mismo en agua caliente con ácido para retirar el cobre restante (lo que no forma parte del circuito), esto demora un poco, para apresurar el tiempo de espera se recomienda realizar movimientos leves, permitiendo mover la placa dentro del ácido. Lo dicho, se puede observar en la figura siguiente.

Figura 132. Separación del cobre adicional de la pista en los circuitos transmisor.

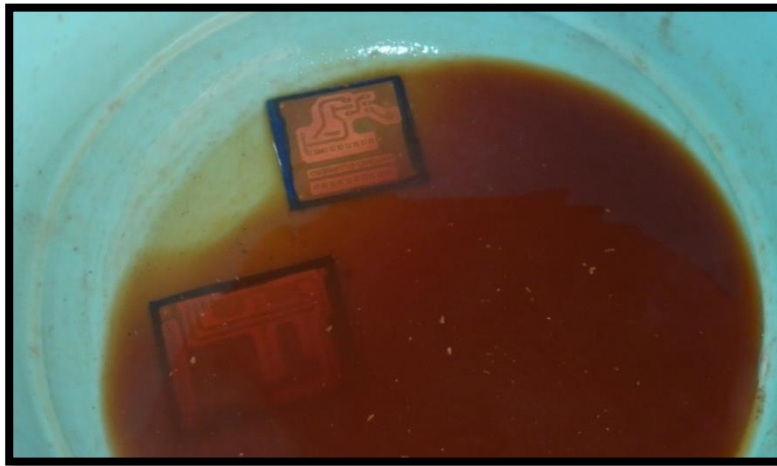
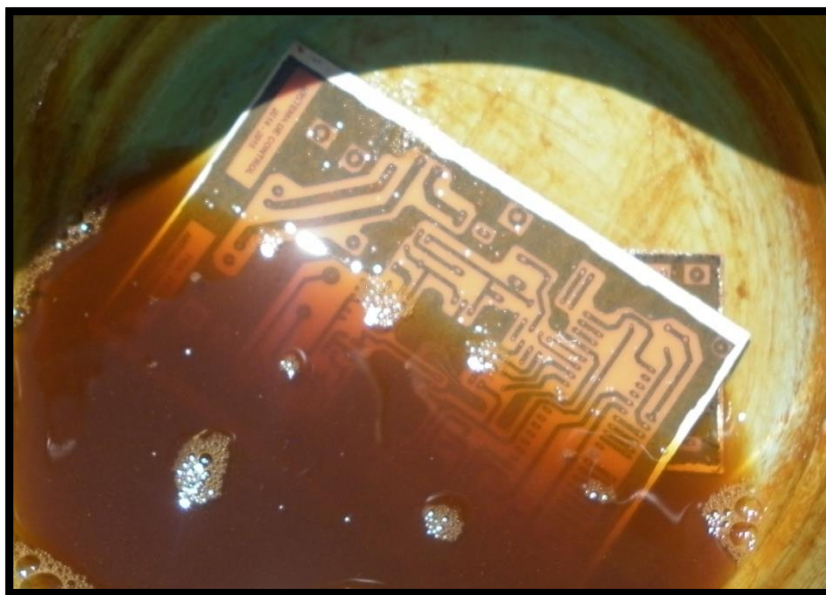


Figura 133. Separación del cobre adicional de la pista en los circuitos receptor.



### 7.7.1.10 Secary Lijar el circuito

Terminado el proceso de retiro del cobre, se debe secar el circuito y pulir la pista con la misma lija para retirar la capa de la tinta del circuito, tal como se indica en la figura siguiente.

Figura 134. Secado del circuito transmisor.



Figura 135. Lijado del circuito transmisor.

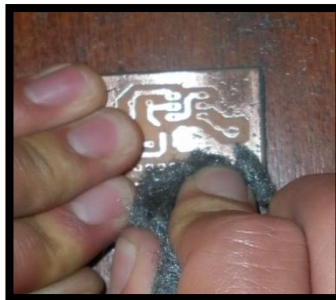
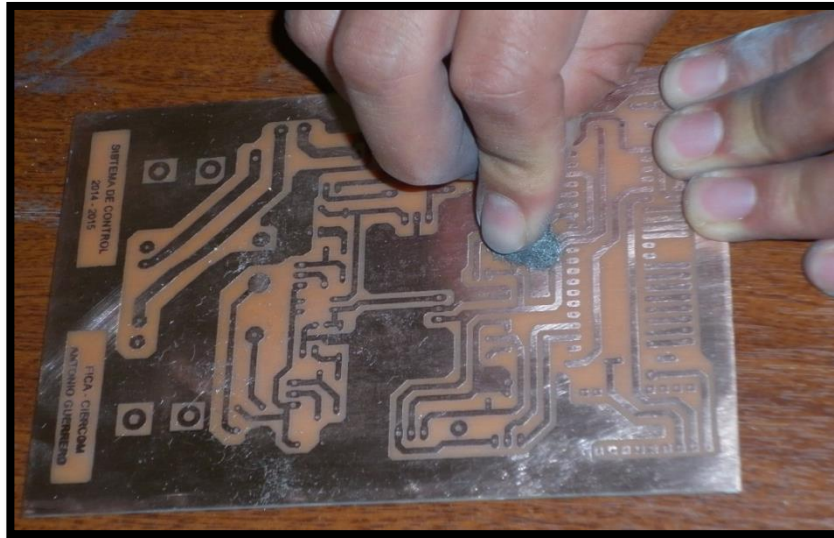


Figura 136. Secado del circuito receptor.





Figura 137. Lijado del circuito receptor.



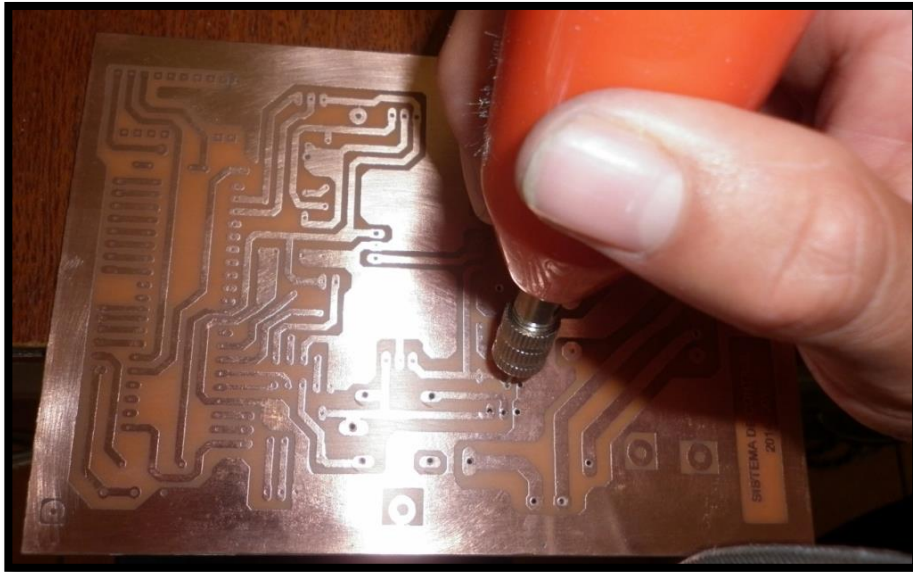
#### 7.7.1.11 Perforar el circuito

Una vez completado el proceso de secado y lijado del circuito, se procede a realizar las perforaciones del circuito para montar los elementos electrónicos. Estas perforaciones se las realiza con un taladro, con las brocas respectivas a cada uno de los elementos, no todos los orificios del circuito tienen el mismo tamaño, por lo tanto se debe contar con cada una de estas brocas para no aislar los extremos de un determinado dispositivo electrónico con la pista del circuito. Este proceso se indica a continuación en la siguiente figura.

Figura 138. Perforación del circuito transmisor.



Figura 139. Perforación del circuito receptor.



#### 7.7.1.12 Montar los elementos electrónicos en el circuito

Completado el proceso de perforación del circuito, se procede a colocar y soldar los elementos electrónicos en el circuito, de igual forma la soldadura de estos elementos deben ser realizados cuidadosamente para evitar hacer contacto con otro terminal, ya que la separación entre ellos es muy diminuta, lo que podría ocasionar hacer “puente” entre ellos. Este proceso se indica en la siguiente figura.

Figura 140. Ubicación de los elementos electrónicos en la placa del circuito transmisor.

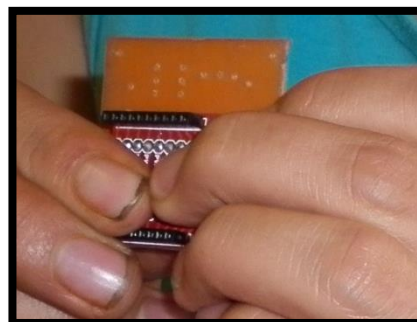
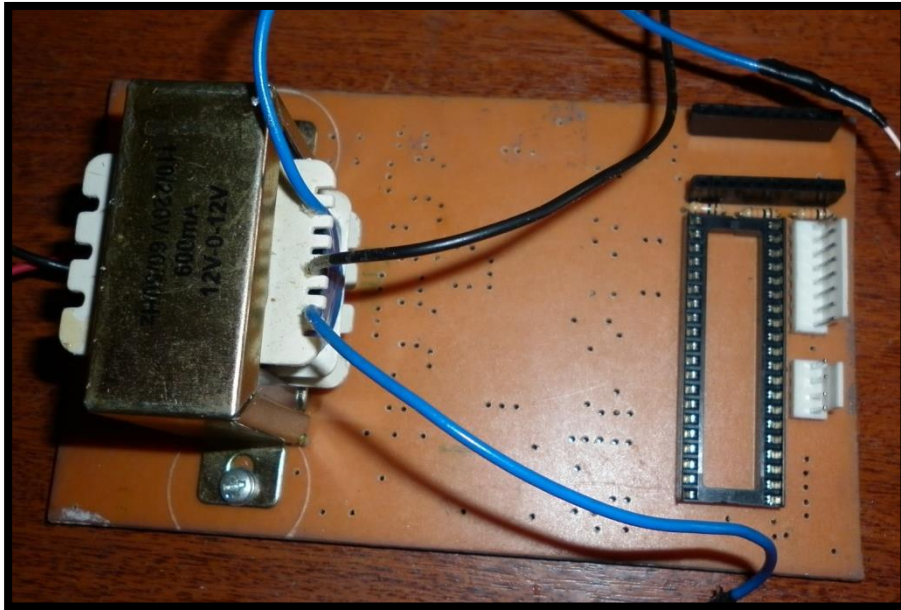


Figura 141. Ubicación de los elementos electrónicos en la placa del circuito receptor.

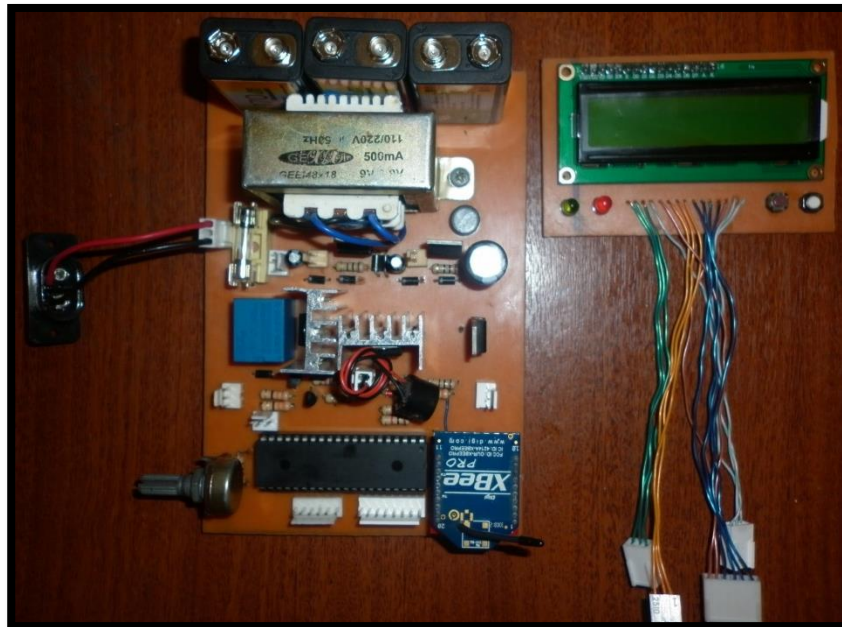


Una vez terminado el proceso de ubicación y soldado de los elementos electrónicos en cada una de las placas, los circuitos quedan finalmente completos y listos para ser puestos en práctica. Estos circuitos se pueden apreciar en las figuras siguientes.

Figura 142. Circuito transmisor terminado.



Figura 143. Circuito receptor terminado.



#### 7.7.1.13 Montar el circuito en la carcasa

Finalmente, realizado correctamente estos pasos se coloca en un estuche o carcasa, la cual debe ser diseñada según las dimensiones de la placa, y asegurada, para que el circuito no sea manipulado internamente con facilidad, lo dicho se indica en la figura siguiente.

Figura 144. Ubicación del circuito transmisor en su respectiva carcasa.





Figura 145. Aseguramiento del circuito transmisor.



Figura 146. Ubicación del circuito receptor en su respectiva carcasa.

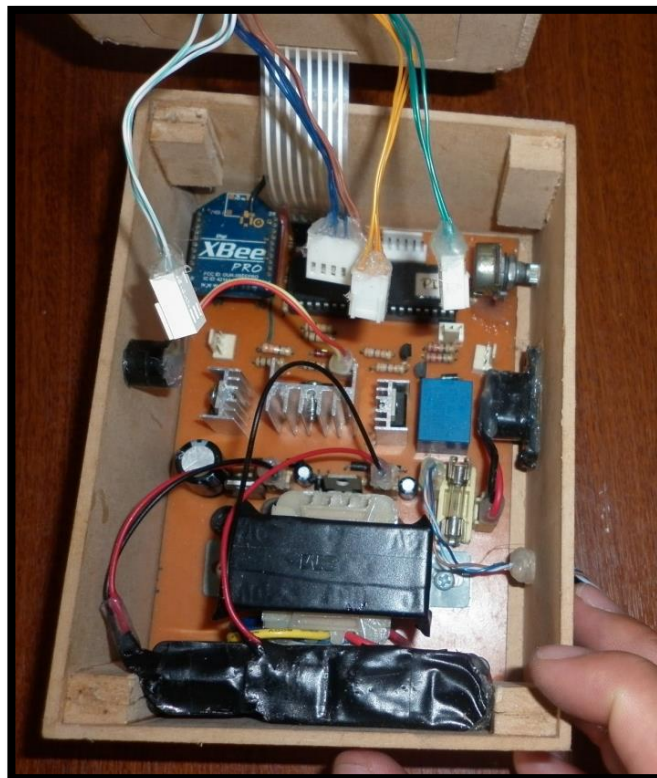


Figura 147. Aseguramiento del circuito receptor.



Con ello, se completa el proceso de diseño de los diferentes circuitos electrónicos con los que cuenta este trabajo de titulación.

## **ANEXO D**

### **7.8 CÓDIGO DE PROGRAMA LA INTERFAZ GRÁFICA CREADA EN MICROSOFT VISUAL STUDIO C#**

```

//-----MÉTODOS-----
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.Sql;
using System.Data.SqlClient;
using System.Windows.Forms;
using System.ComponentModel;
using System.Drawing;
using System.Threading;
using System.IO;
using System.Drawing.Imaging;

namespace Interfaz_Grafica_Reporte
{
    public class Metodos
    {
        public static void Solo_Numeros_CI_Docente(KeyPressEventArgs e)
        {
            if (char.IsNumber(e.KeyChar) || char.IsControl(e.KeyChar))
            {
                e.Handled = false;
            }
            else
            {
                e.Handled = true;
                MessageBox.Show("No se aceptan caracteres especiales, letras ni espacios en la cédula del docente. Ingrese solo números con un máximo de 10 dígitos", "ERROR DE SINTAXIS", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }
        public static void Solo_Numeros_Telefono(KeyPressEventArgs e)
        {
            if (char.IsNumber(e.KeyChar) || char.IsControl(e.KeyChar))
            {
                e.Handled = false;
            }
            else
            {
                e.Handled = true;
                MessageBox.Show("No se aceptan caracteres especiales, letras ni espacios en el nombre del docente.", "ERROR DE SINTAXIS", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }
        public static void Solo_Letras_Nombre_Docente(KeyPressEventArgs e)
        {
            if (char.IsLetter(e.KeyChar) || char.IsControl(e.KeyChar))
            {
                e.Handled = false;
            }
            else if (char.IsLower(e.KeyChar))
            {
                e.Handled = false;
            }
            else if (char.IsUpper(e.KeyChar))
            {
                e.Handled = false;
            }
            else
            {
                e.Handled = true;
                MessageBox.Show("No se aceptan caracteres especiales, números ni espacios en el nombre del docente", "ERROR DE SINTAXIS", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }
        public static void Solo_Letras_PApellido_Docente(KeyPressEventArgs e)
        {
            if (char.IsLetter(e.KeyChar) || char.IsControl(e.KeyChar))
            {
                e.Handled = false;
            }
            else if (char.IsLower(e.KeyChar))
            {
                e.Handled = false;
            }
        }
    }
}

```

```

        e.Handled = false;
    }
    else if (char.IsUpper(e.KeyChar))
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
        MessageBox.Show("No se aceptan caracteres especiales, números ni espacios en el primer apellido del docente", "ERROR DE SINTAXIS", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
public static void Solo_Letras_SAellido_Docente(KeyPressEventArgs e)
{
    if (char.IsLetter(e.KeyChar) || char.IsControl(e.KeyChar))
    {
        e.Handled = false;
    }
    else if (char.IsLower(e.KeyChar))
    {
        e.Handled = false;
    }
    else if (char.IsUpper(e.KeyChar))
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
        MessageBox.Show("No se aceptan caracteres especiales, números ni espacios en el segundo apellido del docente", "ERROR DE SINTAXIS", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
}
public static int Crear_Reserva(string Dia, string Fecha, string CI_Docente, string Materia,
string Periodo_Academico, string Carrera, string Aula, string Paralelo, string Nivel, string Hora_Inicial,
string Hora_Final, string Numero_Horas, string Detalle)
{
    int retornar = 0;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Insert Into Reserva (Dia, Fecha, Cedula,
Cod_Materia, Cod_Periodo_Academico, Cod_Carrera, Cod_Aula, Paralelo, Nivel, Hora_Inicial,
Hora_Final, N_Horas, Detalle) values ('{0}','{1}', (select Cedula from Usuarios where Cedula='{2}'), (
select Cod_Materia from Materias where Nombre_Materia='{3}'), (select Cod_Periodo_Academico from
Periodo_Academico where Nombre_Periodo_Academico='{4}'), (select Cod_Carrera from Carrera
where Alias='{5}'), (select Cod_Aula from Aulas where Nombre_Aula='{6}'), '{7}', '{8}', '{9}', '{10}', '{11}', '{12}'),
Dia, Fecha, CI_Docente, Materia, Periodo_Academico, Carrera, Aula, Paralelo, Nivel, Hora_Inicial,
Hora_Final, Numero_Horas, Detalle), Abrir_Conexion);
    retornar = cmd.ExecuteNonQuery();
    Abrir_Conexion.Close();
    return retornar;
}
}
public static int Comprobar_Reserva(string Dia, string Fecha, string Aula, string Hora_Inicio, string Hora_Final,
string Periodo)
{
    int retornar = 0;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("select * from Reserva where (Dia='{0}') and (Fecha='{1}')
and (Cod_Aula=(select Cod_Aula from Aulas where Nombre_Aula='{2}')) and (((select convert(time,
(select DATEADD(SECOND,+1,'{3}') ))) between Hora_Inicial and Hora_Final) or ((select convert
(time, (select DATEADD(SECOND,-1,'{4}') ))) between Hora_Inicial and Hora_Final)) and
(Cod_Periodo_Academico=(select Cod_Periodo_Academico from Periodo_Academico
where Nombre_Periodo_Academico='{5}'))", Dia, Fecha, Aula, Hora_Inicio, Hora_Final, Periodo), Abrir_Conexion);
    SqlDataReader retorna = cmd.ExecuteReader();
    while (retorna.Read())
    {
        retornar++;
    }
    Abrir_Conexion.Close();
    return retornar;
}
}
public static int Crear_Reporte(string Tipo, int Cod_Horario_Reserva_Limpieza, string Fecha, string Hora_Inicial,
string Hora_Final, string Detalle)
{
    int retornar = 0;

```

```

        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Insert Into Reporte (
Tipo, Cod_Horario_Reserva_Limpieza,Fecha,Hora_Inicial,Hora_Final,Detalle) values ('{0}','{1}','{2}','{3}','{4}','{5}'))",
Tipo, Cod_Horario_Reserva_Limpieza,Fecha, Hora_Inicial, Hora_Final, Detalle), Abrir_Conexion);
returnar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return returnar;
    }
    public static int Modificar_Reporte_Horario(string Hora_Final_Real, string Tipo, string Aula, string Fecha,
string Cedula)
    {
        int returnar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Update Reporte set Hora_Final='{0}' where Tipo='{1}'
AND Cod_Horario_Reserva_Limpieza=(select max(cod_Horario) from Horario where cod_Aula=(select
COD_AULA FROM AULAS WHERE NOMBRE_AULA='{2}') AND Fecha='{3}'AND Cedula='{4}') AND
Fecha='{3}'", Hora_Final_Real, Tipo, Aula, Fecha, Cedula), Abrir_Conexion);
returnar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return returnar;
    }
    public static int Modificar_Reporte_Reserva(string Hora_Final_Real, string Tipo, string Aula, string Fecha,
string Cedula)
    {
        int returnar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Update Reporte set Hora_Final='{0}' where Tipo='{1}'
AND Cod_Horario_Reserva_Limpieza=(select max(cod_Reserva) from Reserva where cod_Aula=(
select COD_AULA FROM AULAS WHERE NOMBRE_AULA='{2}') AND Fecha='{3}'AND Cedula='{4}')
AND Fecha='{3}'", Hora_Final_Real, Tipo, Aula, Fecha,Cedula), Abrir_Conexion);
returnar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return returnar;
    }
    public static int Modificar_Reporte_Limpieza(string Hora_Final_Real, string Tipo, string Aula,string Dia, string
Fecha, string Cedula)
    {
        int returnar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Update Reporte set Hora_Final='{0}' where Tipo='{1}'
AND Cod_Horario_Reserva_Limpieza=( select max(cod_limpieza) from limpieza where cod_Aula=(
select COD_AULA FROM AULAS WHERE NOMBRE_AULA='{2}') AND Dia='{3}' AND Fecha='{4}'
AND Cedula='{5}') AND Fecha='{4}'", Hora_Final_Real, Tipo, Aula, Dia, Fecha, Cedula), Abrir_Conexion);
returnar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return returnar;
    }
    public static int Crear_Limpieza(string Cedula, string Aula, string Dia, string Fecha, string Hora_Inicial,
string Hora_Final)
    {
        int returnar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Insert Into Limpieza (Cedula,Cod_Aula,Dia,
Fecha,Hora_Inicial,Hora_Final) values ('{0}',(select Cod_Aula from Aulas whereNombre_Aula='{1}','{2}','{3}','{4}',
'{5}'))",Cedula, Aula, Dia, Fecha, Hora_Inicial, Hora_Final), Abrir_Conexion);
returnar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return returnar;
    }
    public static int Comprobar_CI(string Cedula)
    {
        int returnar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("select Cedula from Usuarios where Cedula='{0}'", Cedula),
Abrir_Conexion);
        SqlDataReader retorna = cmd.ExecuteReader();
        while (retorna.Read())
        {
            returnar++;
        }
        Abrir_Conexion.Close();
        return returnar;
    }
    public static int Comprobar_Rol(string Cedula,string Rol)
    {

```

```

        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("select Rol_Usuario from Usuarios where Cedula='{0}' and
Rol_Usuario='{1}'", Cedula,Rol), Abrir_Conexion);
        SqlDataReader retorna = cmd.ExecuteReader();
        while (retorna.Read())
        {
            retornar++;
        }
        Abrir_Conexion.Close();
        return retornar;
    }

    public static int Insertar_Imagen(string Cedula, PictureBox Foto)
    {
        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand("Insert Into Imagenes (Imagen,Cedula) values (@Imagen,@Cedula)",
Abrir_Conexion);
        cmd.Parameters.Add("@Imagen", SqlDbType.Image);
        cmd.Parameters.Add("@Cedula", SqlDbType.VarChar);
        MemoryStream ms = new MemoryStream();
        Foto.Image.Save(ms, ImageFormat.Jpeg);
        cmd.Parameters["@Imagen"].Value = ms.GetBuffer();
        cmd.Parameters["@Cedula"].Value = Cedula;
        retornar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return retornar;
    }

    public void Obtener_Imagen(string Cedula, PictureBox Foto)
    {
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand("select Imagen from Imagenes where cedula='"+Cedula+"' ",
Abrir_Conexion);
        SqlDataAdapter dp = new SqlDataAdapter(cmd);
        DataSet ds=new DataSet("Imagenes");
        dp.Fill(ds, "Imagen");
        byte[] bfoto = new byte[0];
        DataRow dr=ds.Tables["imagenes"].Rows[0];
        bfoto = (byte[])dr["Imagen"];
        MemoryStream ms = new MemoryStream(bfoto);
        Foto.Image = Bitmap.FromStream(ms);
        Abrir_Conexion.Close();
    }

    public static int Crear_Usuario(string Cedula, string PNombre_Docente, string SNombre_Docente, string
Primer_Apellido_Docente, string Segundo_Apellido_Docente, string Nombre_Usuario, string Contraseña_Usuario,
string Rol_Usuario, string Direccion, string Telefono, string Correo, string Estado)
    {
        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Insert Into Usuarios (Cedula, Primer_Nombre_Docente,
Segundo_Nombre_Docente, Primer_Apellido_Docente, Segundo_Apellido_Docente, Nombre_Usuario,
Contraseña_Usuario, Rol_Usuario, Direccion, Telefono, Correo, Estado) values ('{0}','{1}','{2}','{3}','{4}','{5}','{6}','{7}',
'{8}','{9}','{10}','{11})," , Cedula, PNombre_Docente, SNombre_Docente, Primer_Apellido_Docente,
Segundo_Apellido_Docente, Nombre_Usuario, Contraseña_Usuario, Rol_Usuario, Direccion, Telefono, Correo,
Estado), Abrir_Conexion);
        retornar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return retornar;
    }

    public static int Crear_Periodo(string Periodo)
    {
        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format ("Insert Into Periodo_Academico
Nombre_Periodo_Academico,Estado) values ('{0}','{1})," ,Periodo, "Activo"), Abrir_Conexion);
        retornar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return retornar;
    }

    public static int Modificar_Periodo(string Periodo)
    {
        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("UPDATE Periodo_Academico SET Estado= 'Inactivo'
WHERE '{0}'=(SELECT '{0}' FROM Periodo_Academico WHERE Estado='Activo')", Periodo), Abrir_Conexion);

```

```

retornar = cmd.ExecuteNonQuery();
Abrir_Conexion.Close();
return retornar;
}
public static int Comprobar_Periodo(string Periodo)
{
int retornar = 0;
SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
SqlCommand cmd = new SqlCommand(string.Format("select * from Periodo_Academico where
(Nombre_Periodo_Academico='{0}']", Periodo), Abrir_Conexion);
SqlDataReader retorna = cmd.ExecuteReader();
while (retorna.Read())
{
returnar++;
}
Abrir_Conexion.Close();
return retornar;
}
public static int Comprobar_Crear_Aula(string Aula)
{
int retornar = 0;
SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
SqlCommand cmd = new SqlCommand(string.Format("select * from Aulas where (Nombre_Aula='{0}']", Aula),
Abrir_Conexion);
SqlDataReader retorna = cmd.ExecuteReader();
while (retorna.Read())
{
returnar++;
}
Abrir_Conexion.Close();
return retornar;
}
public static int Comprobar_Crear_Materia(string Materia)
{
int retornar = 0;
SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
SqlCommand cmd = new SqlCommand(string.Format("select * from Materias where (Nombre_Materia='{0}']",
Materia), Abrir_Conexion);
SqlDataReader retorna = cmd.ExecuteReader();
while (retorna.Read())
{
returnar++;
}
Abrir_Conexion.Close();
return retornar;
}
public static int Comprobar_Crear_Carrera(string Carrera)
{
int retornar = 0;
SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
SqlCommand cmd = new SqlCommand(string.Format("select * from Carrera where (Alias='{0}']", Carrera),
Abrir_Conexion);
SqlDataReader retorna = cmd.ExecuteReader();
while (retorna.Read())
{
returnar++;
}
Abrir_Conexion.Close();
return retornar;
}
public static int Eliminar_Usuario(string CI_Nombre_Apellido)
{
int retornar = 0;
SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
SqlCommand cmd = new SqlCommand(string.Format("Update Usuarios set Estado='Eliminado' where Cedula =
'{0}'", CI_Nombre_Apellido), Abrir_Conexion);
retornar = cmd.ExecuteNonQuery();
Abrir_Conexion.Close();
return retornar;
}
public static int Modificar_Usuario(Datos_Docente PDocente)
{
int retornar = 0;
SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
SqlCommand cmd = new SqlCommand(string.Format("Update Usuarios set Primer_Nombre_Docente='{0}',

```



```

Segundo_Nombre_Docente='{1}',Primer_Apellido_Docente='{2}',Segundo_Apellido_Docente='{3}',
Nombre_Usuario='{4}',Contraseña_Usuario='{5}',Rol_Usuario='{6}',Direccion='{7}',Telefono='{8}',Correo='{9}',
Estado='{10}' where Cedula='{11}'", PDocente.Primer_Nombre_Docente, PDocente.Segundo_Nombre_Docente,
PDocente.Primer_Apellido_Docente, PDocente.Segundo_Apellido_Docente, PDocente.Nombre_Usuario,
PDocente.Contraseña_Usuario, PDocente.Rol_Usuario, PDocente.Direccion_Docente,
PDocente.Telefono_Docente, PDocente.Correo_Docente, PDocente.Estado_Docente, PDocente.Cedula),
Abrir_Conexion);
    retornar = cmd.ExecuteNonQuery();
    Abrir_Conexion.Close();
    return retornar;
}
}
public static int Modificar_Materias(Datos_Materias pMateria)
{
    int retornar = 0;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Update Materias set Nombre_Materia='{0}',Creditos='{1}'
where Nombre_Materia='{2}'", pMateria.Nombre_Materia, pMateria.Creditos, pMateria.Nombre_Materia),
Abrir_Conexion);
    retornar = cmd.ExecuteNonQuery();
    Abrir_Conexion.Close();
    return retornar;
}
}
public static int Modificar_Aulas(Datos_Aulas PAula)
{
    int retornar = 0;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Update Aulas set Capacidad='{0}',Ubicacion='{1}' where
Nombre_Aula='{2}'", PAula.Capacidad, PAula.Ubicacion, PAula.Nombre_Aula), Abrir_Conexion);
    retornar = cmd.ExecuteNonQuery();
    Abrir_Conexion.Close();
    return retornar;
}
}
public static int Modificar_Carrera(Datos_Carrera PCarrera)
{
    int retornar = 0;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Update Carrera set Alias='{0}' where Nombre_Carrera='{1}'",
PCarrera.Alias,PCarrera.Nombre_Carrera), Abrir_Conexion);
    retornar = cmd.ExecuteNonQuery();
    Abrir_Conexion.Close();
    return retornar;
}
}
public static int Modificar_Estado_Aulas_Ocupado(string Cambiar_Estado_Aula)
{
    int retornar = 0;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Update Aulas set Estado='Ocupado' where
Nombre_Aula='{0}'", Cambiar_Estado_Aula), Abrir_Conexion);
    retornar = cmd.ExecuteNonQuery();
    Abrir_Conexion.Close();
    return retornar;
}
}
public static int Modificar_Estado_Aulas_Disponible(string Cambiar_Estado_Aula)
{
    int retornar = 0;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Update Aulas set Estado='Disponible' where
Nombre_Aula='{0}'", Cambiar_Estado_Aula), Abrir_Conexion);
    retornar = cmd.ExecuteNonQuery();
    Abrir_Conexion.Close();
    return retornar;
}
}
public static int Crear_Aula(string Nombre_Aula, int Capacidad_Aula,string Ubicacion_Aula)
{
    int retornar = 0;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Insert Into Aulas (Nombre_Aula, Capacidad, Ubicacion,
Estado) values ('{0}','{1}','{2}','{3}'", Nombre_Aula, Capacidad_Aula, Ubicacion_Aula,"Disponible"), Abrir_Conexion);
    retornar = cmd.ExecuteNonQuery();
    Abrir_Conexion.Close();
    return retornar;
}
}
}
public static int Crear_Materia(string Nombre_Materia, int Numero_creditos)
{
}
}

```

```

        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Insert Into Materias (Nombre_Materia, Creditos, Estado)
        values ({0},{1},{2})", Nombre_Materia, Numero_creditos,"Activo"), Abrir_Conexion);
        retornar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return retornar;
    }
    public static int Crear_Carrera(string Nueva_Carrera ,string Alias_Carrera)
    {
        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Insert Into Carrera (Nombre_Carrera, Alias , Estado) values
        ({0},{1},{2})", Nueva_Carrera, Alias_Carrera, "Activo"), Abrir_Conexion);
        retornar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return retornar;
    }
    public static int Crear_Horario(string Dia, string Docente, string Materia, string Periodo, string Carrera, string Aula,
    string Paralelo, string Nivel, string Hora_Inicio, string Hora_Final, int N_Horas)
    {
        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Insert Into Horario (Dia, Cedula, Cod_Materia,
        Cod_Periodo_Academico, Cod_Carrera, Cod_Aula, Paralelo, Nivel, Hora_Inicial, Hora_Final, N_Horas) values
        ({0},{1},{2},{3},{4},{5},{6},{7},{8},{9},{10})",
        Dia, Docente, Materia, Periodo, Carrera, Aula, Paralelo, Nivel, Hora_Inicio, Hora_Final, N_Horas), Abrir_Conexion);
        retornar = cmd.ExecuteNonQuery();
        Abrir_Conexion.Close();
        return retornar;
    }
    public static int Comprobar_Horario(string Dia, string Aula, string Hora_Inicio, string Hora_Final,string Periodo)
    {
        Convert.ToDateTime(Hora_Inicio);
        Convert.ToDateTime(Hora_Final);
        int retornar = 0;
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("select * from Horario where (Dia='{0}') and (Cod_Aula=(
        select Cod_Aula from Aulas where Nombre_Aula='{1}') and (((select convert(time, (select DATEADD(SECOND,+1,
        '{2}' ))) between Hora_Inicial and Hora_Final) or ((select convert(time, (select DATEADD(SECOND,-1,'{3}' )))
        between Hora_Inicial and Hora_Final)) and (Cod_Periodo_Academico=(select Cod_Periodo_Academico from
        Periodo_Academico where Nombre_Periodo_Academico='{4}'))", Dia, Aula, Hora_Inicio, Hora_Final, Periodo),
        Abrir_Conexion);
        SqlDataReader retorna = cmd.ExecuteReader();
        while (retorna.Read())
        {
            retornar++;
        }
        Abrir_Conexion.Close();
        return retornar;
    }
    public static List<Datos_Docente> Buscar_CI_Docente(string Buscar_CI_Docente)
    {
        List<Datos_Docente> Datos = new List<Datos_Docente>();
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Select Cedula, Primer_Nombre_Docente,
        Segundo_Nombre_Docente, Primer_Apellido_Docente, Segundo_Apellido_Docente, Nombre_Usuario,
        Contraseña_Usuario, Rol_Usuario, Direccion, Telefono, Correo,Estado from Usuarios where Cedula like '%{0}%' ",
        Buscar_CI_Docente), Abrir_Conexion);
        SqlDataReader leer = cmd.ExecuteReader();
        while (leer.Read())
        {
            Datos_Docente pDcatos = new Datos_Docente();
            pDcatos.Cedula = leer.GetString(0);
            pDcatos.Primer_Nombre_Docente = leer.GetString(1);
            pDcatos.Segundo_Nombre_Docente = leer.GetString(2);
            pDcatos.Primer_Apellido_Docente = leer.GetString(3);
            pDcatos.Segundo_Apellido_Docente = leer.GetString(4);
            pDcatos.Nombre_Usuario = leer.GetString(5);
            pDcatos.Contraseña_Usuario = leer.GetString(6);
            pDcatos.Rol_Usuario = leer.GetString(7);
            pDcatos.Direccion_Docente = leer.GetString(8);
        }
    }

```

```

        pDcatos.Telefono_Docente = leer.GetString(9);
        pDcatos.Correo_Docente = leer.GetString(10);
        pDcatos.Estado_Docente = leer.GetString(11);
        Datos.Add(pDcatos);
    }
    Abrir_Conexion.Close();
    return Datos;
}
public static List<Datos_Docente> Buscar_Nombre_Docente(string Buscar_Nombre_Docente)
{
    List<Datos_Docente> Datos = new List<Datos_Docente>();
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Select Cedula, Primer_Nombre_Docente,
Segundo_Nombre_Docente, Primer_Apellido_Docente, Segundo_Apellido_Docente, Nombre_Usuario,
Contraseña_Usuario, Rol_Usuario, Direccion, Telefono, Correo,Estado from Usuarios where
Primer_Nombre_Docente like '%{0}%' ", Buscar_Nombre_Docente), Abrir_Conexion);
    SqlDataReader leer = cmd.ExecuteReader();
    while (leer.Read())
    {
        Datos_Docente pDcatos = new Datos_Docente();
        pDcatos.Cedula = leer.GetString(0);
        pDcatos.Primer_Nombre_Docente = leer.GetString(1);
        pDcatos.Segundo_Nombre_Docente = leer.GetString(2);
        pDcatos.Primer_Apellido_Docente = leer.GetString(3);
        pDcatos.Segundo_Apellido_Docente = leer.GetString(4);
        pDcatos.Nombre_Usuario = leer.GetString(5);
        pDcatos.Contraseña_Usuario = leer.GetString(6);
        pDcatos.Rol_Usuario = leer.GetString(7);
        pDcatos.Direccion_Docente = leer.GetString(8);
        pDcatos.Telefono_Docente = leer.GetString(9);
        pDcatos.Correo_Docente = leer.GetString(10);
        pDcatos.Estado_Docente = leer.GetString(11);
        Datos.Add(pDcatos);
    }
    Abrir_Conexion.Close();
    return Datos;
}
public static List<Datos_Docente> Buscar_PApellido_Docente(string Buscar_PApellido_Docente)
{
    List<Datos_Docente> Datos = new List<Datos_Docente>();
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Select Cedula, Primer_Nombre_Docente,
Segundo_Nombre_Docente, Primer_Apellido_Docente, Segundo_Apellido_Docente, Nombre_Usuario,
Contraseña_Usuario, Rol_Usuario, Direccion, Telefono, Correo,Estado from Usuarios where
Primer_Apellido_Docente like '%{0}%' ", Buscar_PApellido_Docente), Abrir_Conexion);
    SqlDataReader leer = cmd.ExecuteReader();
    while (leer.Read())
    {
        Datos_Docente pDcatos = new Datos_Docente();
        pDcatos.Cedula = leer.GetString(0);
        pDcatos.Primer_Nombre_Docente = leer.GetString(1);
        pDcatos.Segundo_Nombre_Docente = leer.GetString(2);
        pDcatos.Primer_Apellido_Docente = leer.GetString(3);
        pDcatos.Segundo_Apellido_Docente = leer.GetString(4);
        pDcatos.Nombre_Usuario = leer.GetString(5);
        pDcatos.Contraseña_Usuario = leer.GetString(6);
        pDcatos.Rol_Usuario = leer.GetString(7);
        pDcatos.Direccion_Docente = leer.GetString(8);
        pDcatos.Telefono_Docente = leer.GetString(9);
        pDcatos.Correo_Docente = leer.GetString(10);
        pDcatos.Estado_Docente = leer.GetString(11);
        Datos.Add(pDcatos);
    }
    Abrir_Conexion.Close();
    return Datos;
}
public static Datos_Docente Obtener_Docente(string Obtener_Cod_Docente)
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    Datos_Docente pDcatos = new Datos_Docente();
    SqlCommand cmd = new SqlCommand(string.Format("Select Cedula, Primer_Nombre_Docente,
Segundo_Nombre_Docente, Primer_Apellido_Docente, Segundo_Apellido_Docente, Nombre_Usuario,
Contraseña_Usuario, Rol_Usuario, Direccion, Telefono, Correo,Estado from Usuarios where Cedula ='{0}'",
Obtener_Cod_Docente), Abrir_Conexion);
    SqlDataReader leer = cmd.ExecuteReader();

```

```

        while (leer.Read())
        {
            pDcatos.Cedula = leer.GetString(0);
            pDcatos.Primer_Nombre_Docente = leer.GetString(1);
            pDcatos.Segundo_Nombre_Docente = leer.GetString(2);
            pDcatos.Primer_Apellido_Docente = leer.GetString(3);
            pDcatos.Segundo_Apellido_Docente = leer.GetString(4);
            pDcatos.Nombre_Usuario = leer.GetString(5);
            pDcatos.Contraseña_Usuario = leer.GetString(6);
            pDcatos.Rol_Usuario = leer.GetString(7);
            pDcatos.Direccion_Docente = leer.GetString(8);
            pDcatos.Telefono_Docente = leer.GetString(9);
            pDcatos.Correo_Docente = leer.GetString(10);
            pDcatos.Estado_Docente = leer.GetString(11);
        }
        Abrir_Conexion.Close();
        return pDcatos;
    }
    public static List<Datos_Aulas> Obtener_Aulas()
    {
        List<Datos_Aulas> DAulas = new List<Datos_Aulas>();
        SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
        SqlCommand cmd = new SqlCommand(string.Format("Select Nombre_Aula,Capacidad,Estado from Aulas "),
        Abrir_Conexion);
        SqlDataReader leer = cmd.ExecuteReader();
        while (leer.Read())
        {
            Datos_Aulas pAulas = new Datos_Aulas();
            pAulas.Nombre_Aula = leer.GetString(0);
            pAulas.Capacidad = leer.GetInt32(1);
            pAulas.Estado = leer.GetString(2);
            DAulas.Add(pAulas);
        }
        Abrir_Conexion.Close();
        return DAulas;
    }
}
}
}
}

```

//----- FORMULARIO AUTENTICACIÓN -----

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.IO.Ports;
using System.Threading;
using System.Text.RegularExpressions;
using System.IO;
using System.Drawing.Imaging;

namespace Interfaz_Grafica_Reporte
{
    public partial class Form_Autenticacion : Form
    {
        public string CI, P_Nombre_Docente, S_Nombre_Docente, P_Apellido_Docente, S_Apellido_Docente, Usuario,
        Rol_Usuario, Contraseña_Usuario, Direccion, Telefono, Correo, Estado_Usuario;
        public byte [] Foto;
        SqlConnection conexion = new SqlConnection("Data Source =ROSITA-PC ; Initial Catalog = BD_ConAccesoPuertas;
        Integrated Security =true");
        public Form_Autenticacion()
        {
            InitializeComponent();
            Limpiar();
            Importar_Dia_Sistema();
            Importar_Hora_Sistema();
            if (!SerialPort2.IsOpen)
            {
                try

```

```

        {
            serialPort2.Open();
            MessageBox.Show("puerto abierto 2");
        }
        catch (System.Exception ex)
        {
            MessageBox.Show(ex.ToString());
        }
    }
}
void Limpiar()
{
    text_Usuario_Administrador.Clear();
    text_Contraseña_Administrador.Clear();
    text_ConfContraseña_Administrador.Clear();
    text_Usuario_Administrador.Focus();
    text_Usuario_Usuario.Clear();
    text_Contraseña_Usuario.Clear();
    text_Usuario_Usuario.Focus();
    text_CI_Eventual.Clear();
    text_CI_Eventual.Focus();
}
private void btn_Limpiar_Administrador_Click(object sender, EventArgs e)
{
    Limpiar();
}
private void btn_Limpiar_Usuario_Click(object sender, EventArgs e)
{
    Limpiar();
}
private void radioButton_Modo_Administrador_CheckedChanged(object sender, EventArgs e)
{
    groupBox_Autenticacion_Administrador.Visible = true;
    groupBox_Autenticacion_Usuario.Visible = false;
    groupBox_Autenticacion_Eventual.Visible = false;
}
private void radioButton_Modo_Usuario_CheckedChanged(object sender, EventArgs e)
{
    groupBox_Autenticacion_Administrador.Visible = false;
    groupBox_Autenticacion_Usuario.Visible = true;
    groupBox_Autenticacion_Eventual.Visible = false;
}
private void radioButton_Modo_Eventual_CheckedChanged(object sender, EventArgs e)
{
    groupBox_Autenticacion_Administrador.Visible = false;
    groupBox_Autenticacion_Usuario.Visible = false;
    groupBox_Autenticacion_Eventual.Visible = true;
}
private void btn_Ingresar_Administrador_Click(object sender, EventArgs e)
{
    try
    {
        if (text_Usuario_Administrador.Text == "" || text_Contraseña_Administrador.Text == "" ||
text_ConfContraseña_Administrador.Text == "")
        {
            MessageBox.Show("Los campos no están llenos completamente");
        }
    }
    else
    {
        SqlCommand cmd = new SqlCommand("select Cedula, Primer_Nombre_Docente,
Segundo_Nombre_Docente,Primer_Apellido_Docente,Segundo_Apellido_Docente,Nombre_Usuario,
Contraseña_Usuario,Rol_Usuario,Direccion,Telefono,Correo,Estado from Usuarios where Nombre_Usuario
COLLATE Latin1_General_CS_AS=" + text_Usuario_Administrador.Text + " and Contraseña_Usuario
COLLATE Latin1_General_CS_AS=" + text_Contraseña_Administrador.Text + " and Rol_Usuario=" +
radioButton_Modo_Administrador.Text + " ", conexion);
conexion.Open();
        cmd.ExecuteNonQuery();
        DataSet ds = new DataSet();
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(ds, "Usuarios");
        DataRow dro;
        dro = ds.Tables["Usuarios"].Rows[0];
        CI=dro["Cedula"].ToString();
        P_Nombre_Docente = dro["Primer_Nombre_Docente"].ToString();
        S_Nombre_Docente = dro["Segundo_Nombre_Docente"].ToString();
    }
}

```

```

P_Apellido_Docente = dro["Primer_Apellido_Docente"].ToString();
S_Apellido_Docente = dro["Segundo_Apellido_Docente"].ToString();
Usuario = dro["Nombre_Usuario"].ToString();
Contraseña_Usuario = dro["Contraseña_Usuario"].ToString();
Rol_Usuario = dro["Rol_Usuario"].ToString();
Direccion = dro["Direccion"].ToString();
Telefono = dro["Telefono"].ToString();
Correo = dro["Correo"].ToString();
Estado_Usuario = dro["Estado"].ToString();
if (text_Contraseña_Administrador.Text == text_ConfContraseña_Administrador.Text)
{
    if (Estado_Usuario == "Activo")
    {
        MessageBox.Show("Autenticación Correcta. Bienvenido al Modo Administrador");
        serialPort2.Close();
        MessageBox.Show("puerto cerrado 2");
        Form_Administrador Administrador = new Form_Administrador();
        Administrador.textBox_CI_Eventual.Text = CI;
        Administrador.textBox_Docente_Eventual.Text = P_Apellido_Docente + " " + S_Apellido_Docente + " " +
            P_Nombre_Docente;
        Administrador.textBox_Rol_Eventual.Text = Rol_Usuario;
        Administrador.textBox_CI_Asignada.Text = CI;
        Administrador.textBox_Docente_Asignada.Text = P_Apellido_Docente + " " + S_Apellido_Docente + " " +
            P_Nombre_Docente;
        Administrador.textBox_Rol_Asignada.Text = Rol_Usuario;
        Administrador.textBox_CID.Text = CI;
        Administrador.textBox_PND.Text = P_Nombre_Docente;
        Administrador.textBox_SND.Text = S_Nombre_Docente;
        Administrador.textBox_PAD.Text = P_Apellido_Docente;
        Administrador.textBox_SAD.Text = S_Apellido_Docente;
        Administrador.textBox_DD.Text = Direccion;
        Administrador.textBox_TD.Text = Telefono;
        Administrador.textBox_CD.Text = Correo;
        Administrador.textBox_NUD.Text = Usuario;
        Administrador.textBox_CUD.Text = Contraseña_Usuario;
        Administrador.textBox_RUD.Text = Rol_Usuario;
        Administrador.textBox_Fecha_Asignada.Text = datePicker_Fecha.Text;
        Administrador.Show();
        this.Hide();
    }
    else
    {
        MessageBox.Show("El docente fue eliminado por lo que no se encuentra activo en el sistema");
        Limpiar();
    }
}
else
{
    MessageBox.Show("La confirmación de contraseñas es incorrecta, por favor intente nuevamente");
    text_Contraseña_Administrador.Clear();
    text_ConfContraseña_Administrador.Clear();
    text_Contraseña_Administrador.Focus();
}
}
}
catch (Exception)
{
    MessageBox.Show("Autenticación Incorrecta. La base de datos no se conectó con el Sistema de Control de
Acceso");
    Limpiar();
}
finally
{
    conexion.Close();
}
}
private void btn_Ingresar_Usuario_Click(object sender, EventArgs e)
{
    try
    {
        if (text_Usuario_Usuario.Text == "" || text_Contraseña_Usuario.Text == "")
        {
            MessageBox.Show("Los campos no están llenos completamente");
        }
    }
    else

```

```

        {
            SqlCommand cmd = new SqlCommand("select Cedula, Primer_Nombre_Docente,
Segundo_Nombre_Docente,Primer_Apellido_Docente,Segundo_Apellido_Docente,Nombre_Usuario,
Contraseña_Usuario,Rol_Usuario,Direccion,Telefono,Correo,Estado from Usuarios where Nombre_Usuario
COLLATE Latin1_General_CS_AS ='" + text_Usuario_Usuario.Text + "' and Contraseña_Usuario COLLATE
Latin1_General_CS_AS='" + text_Contraseña_Usuario.Text + "' and Rol_Usuario='" +
radioButton_Modo_Usuario.Text + "' ", conexion);
conexion.Open();
            cmd.ExecuteNonQuery();
            DataSet ds = new DataSet();
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            da.Fill(ds, "Usuarios");
            DataRow dro;
            dro = ds.Tables["Usuarios"].Rows[0];
            CI = dro["Cedula"].ToString();
            P_Nombre_Docente = dro["Primer_Nombre_Docente"].ToString();
            S_Nombre_Docente = dro["Segundo_Nombre_Docente"].ToString();
            P_Apellido_Docente = dro["Primer_Apellido_Docente"].ToString();
            S_Apellido_Docente = dro["Segundo_Apellido_Docente"].ToString();
            Usuario = dro["Nombre_Usuario"].ToString();
            Contraseña_Usuario = dro["Contraseña_Usuario"].ToString();
            Rol_Usuario = dro["Rol_Usuario"].ToString();
            Direccion = dro["Direccion"].ToString();
            Telefono = dro["Telefono"].ToString();
            Correo = dro["Correo"].ToString();
            Estado_Usuario = dro["Estado"].ToString();
            if (Estado_Usuario == "Activo")
            {
                MessageBox.Show("Autenticación Correcta. Bienvenido al Modo Usuario");
                serialPort2.Close();
                MessageBox.Show("puerto cerrado 2");
                Form_Administrador Administrador = new Form_Administrador();
                Administrador.textBox_CI_Eventual.Text = CI;
                Administrador.textBox_Docente_Eventual.Text = P_Apellido_Docente + " " + S_Apellido_Docente + " " +
                P_Nombre_Docente;
                Administrador.textBox_Rol_Eventual.Text = Rol_Usuario;
                Administrador.textBox_CI_Asignada.Text = CI;
                Administrador.textBox_Docente_Asignada.Text = P_Apellido_Docente + " " + S_Apellido_Docente + " " +
                P_Nombre_Docente;
                Administrador.textBox_Rol_Asignada.Text = Rol_Usuario;
                Administrador.textBox_CID.Text = CI;
                Administrador.textBox_PND.Text = P_Nombre_Docente;
                Administrador.textBox_SND.Text = S_Nombre_Docente;
                Administrador.textBox_PAD.Text = P_Apellido_Docente;
                Administrador.textBox_SAD.Text = S_Apellido_Docente;
                Administrador.textBox_DD.Text = Direccion;
                Administrador.textBox_TD.Text = Telefono;
                Administrador.textBox_CD.Text = Correo;
                Administrador.textBox_NUD.Text = Usuario;
                Administrador.textBox_CUD.Text = Contraseña_Usuario;
                Administrador.textBox_RUD.Text = Rol_Usuario;
                Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                Administrador.menuStrip1.Items[1].Visible = false;
                Administrador.textBox_Fecha_Asignada.Text = dateTimePicker_Fecha.Text;
                Administrador.Text = "USUARIO";
                Administrador.Show();
                this.Hide();
            }
            else
            {
                MessageBox.Show("El docente fue eliminado por lo que no se encuentra activo en el sistema");
                Limpiar();
            }
        }
        catch (Exception)
        {
            MessageBox.Show("Autenticación Incorrecta. La base de datos no se conectó con el Sistema de Control de
Acceso");
            Limpiar();
        }
        finally

```

```

        {
            conexion.Close();
        }
    }
    private void btn_Ingresar_Eventual_Click(object sender, EventArgs e)
    {
        try
        {
            if (text_CI_Eventual.Text == "")
            {
                MessageBox.Show("El campo no está lleno completamente");
            }
            else
            {
                SqlCommand cmd = new SqlCommand("select Cedula, Primer_Nombre_Docente,
                Segundo_Nombre_Docente,Primer_Apellido_Docente,Segundo_Apellido_Docente,Nombre_Usuario,
                Contraseña_Usuario,Rol_Usuario,Direccion,Telefono,Correo,Estado from Usuarios where Cedula=" +
                text_CI_Eventual.Text + " and Rol_Usuario=" + radioButton_Modo_Eventual.Text + " ", conexion);
                conexion.Open();
                cmd.ExecuteNonQuery();
                DataSet ds = new DataSet();
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                da.Fill(ds, "Usuarios");
                DataRow dro;
                dro = ds.Tables["Usuarios"].Rows[0];
                CI = dro["Cedula"].ToString();
                P_Nombre_Docente = dro["Primer_Nombre_Docente"].ToString();
                S_Nombre_Docente = dro["Segundo_Nombre_Docente"].ToString();
                P_Apellido_Docente = dro["Primer_Apellido_Docente"].ToString();
                S_Apellido_Docente = dro["Segundo_Apellido_Docente"].ToString();
                Usuario = dro["Nombre_Usuario"].ToString();
                Contraseña_Usuario = dro["Contraseña_Usuario"].ToString();
                Rol_Usuario = dro["Rol_Usuario"].ToString();
                Direccion = dro["Direccion"].ToString();
                Telefono = dro["Telefono"].ToString();
                Correo = dro["Correo"].ToString();
                Estado_Usuario = dro["Estado"].ToString();
                if (Estado_Usuario == "Activo")
                {
                    MessageBox.Show("Autenticación Correcta. Bienvenido al Modo Eventual");
                    serialPort2.Close();
                    MessageBox.Show("puerto cerrado 2");
                    Form_Administrador Administrador = new Form_Administrador();
                    Administrador.textBox_CI_Eventual.Text = CI;
                    Administrador.textBox_Docente_Eventual.Text = P_Apellido_Docente + " " + S_Apellido_Docente + " " +
                    P_Nombre_Docente;
                    Administrador.textBox_Rol_Eventual.Text = Rol_Usuario;
                    Administrador.textBox_CI_Asignada.Text = CI;
                    Administrador.textBox_Docente_Asignada.Text = P_Apellido_Docente + " " + S_Apellido_Docente + " " +
                    P_Nombre_Docente;
                    Administrador.textBox_Rol_Asignada.Text = Rol_Usuario;
                    Administrador.textBox_CID.Text = CI;
                    Administrador.textBox_PND.Text = P_Nombre_Docente;
                    Administrador.textBox_SND.Text = S_Nombre_Docente;
                    Administrador.textBox_PAD.Text = P_Apellido_Docente;
                    Administrador.textBox_SAD.Text = S_Apellido_Docente;
                    Administrador.textBox_DD.Text = Direccion;
                    Administrador.textBox_TD.Text = Telefono;
                    Administrador.textBox_CD.Text = Correo;
                    Administrador.textBox_NUD.Text = Usuario;
                    Administrador.textBox_CUD.Text = Contraseña_Usuario;
                    Administrador.textBox_RUD.Text = Rol_Usuario;
                    Administrador.textBox_NUD.Enabled = false;
                    Administrador.textBox_NUD.Text = "NO REQUERIDO";
                    Administrador.textBox_CUD.Enabled = false;
                    Administrador.textBox_CUD.Text = "SU CEDULA";
                    Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                    Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                    Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                    Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                    Administrador.tabControl_Opciones.TabPages.RemoveAt(0);
                    Administrador.menuStrip1.Items[1].Visible = false;
                    Administrador.textBox_Fecha_Asignada.Text = dateTimePicker_Fecha.Text;
                    Administrador.Text = "EVENTUAL";
                    Administrador.Show();
                }
            }
        }
        catch { }
    }
}

```



```

        this.Hide();
    }
    else
    {
        MessageBox.Show("El docente fue eliminado por lo que no se encuentra activo en el sistema");
    }
}
Limpia();
}
}
}
catch (Exception)
{
    MessageBox.Show("Autenticación Incorrecta. La base de datos no se conectó con el Sistema de Control de Acceso");
    Limpia();
}
finally
{
    conexion.Close();
}
}
}
private void text_CI_Eventual_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Numeros_CI_Docente(e);
}
}
char[] broadcast_Hora;
private void timer1_Tick(object sender, EventArgs e)
{
    label_Hora_Sistema.Text = DateTime.Now.ToLongTimeString();
    broadcast_Hora = label_Hora_Sistema.Text.ToArray();
    if (label_Hora_Sistema.Text == "22:10:00")
    {
        Close();
    }
    if ((broadcast_Hora[3] == '5' && broadcast_Hora[4] == '5' && broadcast_Hora[6] == '0' && broadcast_Hora[7] == '0') ||
        (broadcast_Hora[3] == '5' && broadcast_Hora[4] == '8' && broadcast_Hora[6] == '0' && broadcast_Hora[7] == '0'))
    {
        Broadcast_Hora();
    }
}
void Broadcast_Hora()
{
    int h = 0;
    h=Convert.ToInt32(broadcast_Hora[1])-48;
    h = h + 1;
    string hora = Convert.ToString(broadcast_Hora[0] +Convert.ToString(h) + ":00:00");
    if (!serialPort2.IsOpen)
    {
        serialPort2.Open();
    }
    serialPort2.Write("Z+z+1234567890+1+Z+"+hora);
    Thread.Sleep(1000);
}
int Cod_Limpieza;
void Importar_Cod_Limpieza(string Cedula,string Fecha)
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand("SELECT cod_limpieza FROM limpieza WHERE Cedula=" + Cedula+" and Fecha="+Fecha+"", Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
        {
            Cod_Limpieza = lectura.GetInt32(0);
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}
public string rxautenticacion;
private void serialPort2_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)

```

```

    {
        rxautenticacion = this.serialPort2.ReadExisting();
        while ((rxautenticacion.Length != 27) || (rxautenticacion.Length != 35));
        {
            rxautenticacion = "";
            Thread.Sleep(100);
            rxautenticacion = this.serialPort2.ReadExisting();
        }
        int i;
        char[] Cadena = rxautenticacion.ToArray();
        char [] Confirmar = new char [27];
        char[] Cl_Cadena = new char[10];
        for (i = 4; i < 14; i++)
        {
            Cl_Cadena[i - 4] = Cadena[i];
        }
        for (i = 0; i < 27; i++)
        {
            Confirmar[i] = Cadena[i];
        }
        string ClCadena = new string(Cl_Cadena);
        int resultado = Metodos.Comprobar_Cl(ClCadena);
        if (resultado > 0)
        {
            if (Cadena[0] == 'A')
            {
                if (Cadena[2] == 'c')
                {
                    Metodos.Modificar_Estado_Aulas_Disponible("AULA_201");
                    if (Cadena[17] == 'H')
                    {
                        Metodos.Modificar_Reporte_Horario(DateTime.Now.ToLongTimeString(), "HORARIO", "AULA_201",
dateTimePicker_Fecha.Text, ClCadena);
                    }
                    if (Cadena[17] == 'R')
                    {
                        Metodos.Modificar_Reporte_Reserva(DateTime.Now.ToLongTimeString(), "RESERVA", "AULA_201",
dateTimePicker_Fecha.Text, ClCadena);
                    }
                    if (Cadena[17] == 'L')
                    {
                        Metodos.Modificar_Reporte_Limpieza(DateTime.Now.ToLongTimeString(), "LIMPIEZA", "AULA_201",
textBox_Dia.Text, dateTimePicker_Fecha.Text, ClCadena);
                    }
                }
                Confirmar[2] = 'v';
            }
            else if (Cadena[2] == 'd' && Cadena[17] == 'L')
            {
                int resultado_1 = Metodos.Comprobar_Rol(ClCadena, "Conserje");
                if (resultado_1 > 0)
                {
                    int resultado_2 = Metodos.Crear_Limpieza(ClCadena, "AULA_201", textBox_Dia.Text,
dateTimePicker_Fecha.Text, label_Hora_Inicial.Text, label_Hora_Final.Text);
                    if (resultado_2 > 0)
                    {
                        Importar_Cod_Limpieza(ClCadena, dateTimePicker_Fecha.Text);
                        Metodos.Crear_Reporte("LIMPIEZA", Cod_Limpieza, dateTimePicker_Fecha.Text,
label_Hora_Inicial.Text, label_Hora_Final.Text, "LIMPIEZA DEL AULA");
                        Confirmar[2] = 'v';
                        Metodos.Modificar_Estado_Aulas_Ocupado("AULA_201");
                    }
                }
            }
            else
            {
                Confirmar[2] = 'f';
            }
        }
    }
    }
    }
    else
    {
        Confirmar[2] = 'f';
    }
    }
    Thread.Sleep(500);
    string Envio_Confirmacion = new string(Confirmar);

```

```

serialPort2.Write(Convert.ToString(Envio_Confirmacion));
    rxautenticacion = "";
}
void Importar_Dia_Sistema()
{
    string Fecha = DateTime.Now.ToLongDateString();
    char[] Vector_Fecha = Fecha.ToArray();
    for (int i = 0; i <= 9; i++)
    {
        if (Vector_Fecha[i] != Convert.ToChar(","))
        {
            textBox_Dia.CharacterCasing = CharacterCasing.Upper;
            textBox_Dia.Text += Vector_Fecha[i].ToString();
        }
        else
        {
            i = 10;
        }
    }
}
void Importar_Hora_Sistema()
{
    int h1, h2, m1, m2, s1, s2;
    string Hora = DateTime.Now.ToLongTimeString();
    char[] vector_Hora = Hora.ToArray();
    if (vector_Hora.Length == 7)
    {
        h1 = 0;
        h2 = Convert.ToInt32(vector_Hora[0].ToString());
        m1 = Convert.ToInt32(vector_Hora[2].ToString());
        m2 = Convert.ToInt32(vector_Hora[3].ToString());
        s1 = Convert.ToInt32(vector_Hora[5].ToString());
        s2 = Convert.ToInt32(vector_Hora[6].ToString());
    }
    else
    {
        h1 = Convert.ToInt32(vector_Hora[0].ToString());
        h2 = Convert.ToInt32(vector_Hora[1].ToString());
        m1 = Convert.ToInt32(vector_Hora[3].ToString());
        m2 = Convert.ToInt32(vector_Hora[4].ToString());
        s1 = Convert.ToInt32(vector_Hora[6].ToString());
        s2 = Convert.ToInt32(vector_Hora[7].ToString());
    }
    label_Hora_Inicial.Text = Convert.ToString(h1) + Convert.ToString(h2) + ":" + Convert.ToString(m1) +
    Convert.ToString(m2) + ":" + Convert.ToString(s1) + Convert.ToString(s2);
    int h12 = Convert.ToInt32(Convert.ToString(h1) + Convert.ToString(h2)) + 1;
    label_Hora_Final.Text = Convert.ToString(h12) + ":" + Convert.ToString(m1) + Convert.ToString(m2) + ":" +
    Convert.ToString(s1) + Convert.ToString(s2);
}
private void radioButton_Modo_Administrador_MouseClick(object sender, MouseEventArgs e)
{
    text_CI_Eventual.Clear();
    text_Usuario_Usuario.Clear();
    text_Contraseña_Usuario.Clear();
}
private void radioButton_Modo_Usuario_MouseClick(object sender, MouseEventArgs e)
{
    text_CI_Eventual.Clear();
    text_Usuario_Administrador.Clear();
    text_Contraseña_Administrador.Clear();
    text_ConfContraseña_Administrador.Clear();
}
private void radioButton_Modo_Eventual_MouseClick(object sender, MouseEventArgs e)
{
    text_Usuario_Administrador.Clear();
    text_Contraseña_Administrador.Clear();
    text_ConfContraseña_Administrador.Clear();
    text_Usuario_Usuario.Clear();
    text_Contraseña_Usuario.Clear();
}
}
}
}

```

//----- FORMULARIO DEL ADMINISTRADOR, USUARIO Y EVENTUAL -----

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Timers;
using System.Data.SqlClient;
using System.IO.Ports;
using System.Threading;
using System.Text.RegularExpressions;
using System.IO;
using System.Drawing.Imaging;

namespace Interfaz_Grafica_Reporte
{
    public partial class Form_Administrador : Form
    {
        public Form_Administrador()
        {
            InitializeComponent();
            Comparar_CI_Docente();
            Aulas();
            Materias();
            Docentes();
            Aulas_Disponibles();
            Importar_Dia_Sistema();
            Limpiar_Crear_Horario();
            if (!SerialPort1.IsOpen)
            {
                try
                {
                    serialPort1.Open();
                }
                catch (System.Exception ex)
                {
                    MessageBox.Show(ex.ToString());
                }
            }
        }

        //-----
        //-----
        //-----FORMULARIO-----
        private void Form_Administrador_Load(object sender, EventArgs e)
        {
            int h1, h2, m1, m2, s1, s2;
            string hora = DateTime.Now.ToLongTimeString();
            char[] vector_Hora = hora.ToArray();
            if (vector_Hora.Length == 7)
            {
                h1 = 0;
                h2 = Convert.ToInt32(vector_Hora[0].ToString());
                m1 = Convert.ToInt32(vector_Hora[2].ToString());
                m2 = Convert.ToInt32(vector_Hora[3].ToString());
                s1 = Convert.ToInt32(vector_Hora[5].ToString());
                s2 = Convert.ToInt32(vector_Hora[6].ToString());
            }
            else
            {
                h1 = Convert.ToInt32(vector_Hora[0].ToString());
                h2 = Convert.ToInt32(vector_Hora[1].ToString());
                m1 = Convert.ToInt32(vector_Hora[3].ToString());
                m2 = Convert.ToInt32(vector_Hora[4].ToString());
                s1 = Convert.ToInt32(vector_Hora[6].ToString());
                s2 = Convert.ToInt32(vector_Hora[7].ToString());
            }
            label_Hora_Inicial.Text = Convert.ToString(h1) + Convert.ToString(h2) + ":" + Convert.ToString(m1) +
            Convert.ToString(m2) + ":" + Convert.ToString(s1) + Convert.ToString(s2);
            label_Horas_Calculadas.Text = Convert.ToString(h1) + Convert.ToString(h2);
            label_Minutos_Calculados.Text = Convert.ToString(m1) + Convert.ToString(m2);
            textBox_Horas_Asignada.Text = label_Hora_Inicial.Text;
        }
    }
}
```

```

        comboBox_Periodo_Horario.Items.AddRange(new object[] { "MARZO " + (Año_Actual - 1) + " - AGOSTO " +
(Año_Actual - 1) + "", "SEPTIEMBRE " + (Año_Actual - 1) + " - FEBRERO " + Año_Actual });
        comboBox_Periodo_Horario.Items.AddRange(new object[] { "MARZO " + Año_Actual + " - AGOSTO " + Año_Actual
+ "", "SEPTIEMBRE " + Año_Actual + " - FEBRERO " + (Año_Actual + 1) });
        comboBox_Periodo_Academico.Items.AddRange(new object[] { "MARZO " + (Año_Actual - 1) + " - AGOSTO " +
(Año_Actual - 1) + "", "SEPTIEMBRE " + (Año_Actual - 1) + " - FEBRERO " + (Año_Actual) });
        comboBox_Periodo_Academico.Items.AddRange(new object[] { "MARZO " + Año_Actual + " - AGOSTO " +
Año_Actual + "", "SEPTIEMBRE " + Año_Actual + " - FEBRERO " + (Año_Actual + 1) });
        textBox_Periodo_Academico.Text = "SEPTIEMBRE 2014 - FEBRERO 2015";
comboBox_Rol.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Periodo_Academico.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Numero_Horas.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Numero_Aula.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Niveles.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Materia.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Paralelo.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Carreras.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Ubicacion_Aula.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Creditos_Materia.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Seleccion_Nueva_Aula.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Seleccion_Nueva_Materia.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Estado.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Seleccion_Nueva_Carrera.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Hora_Inicial.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Dia_Horario.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Materia_Horario.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Docente_Horario.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Carrera_Horario.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Aula_Horario.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Hora_Inicio_Asignada.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Hora_Final_Asignada.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Nivel_Horario.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Paralelo_Horario.DropDownStyle = ComboBoxStyle.DropDownList;
comboBox_Periodo_Horario.DropDownStyle = ComboBoxStyle.DropDownList;
label12_Informacion.Text = "TODOS LOS CAMPOS DEBEN ESTAR CORRECTAMENTE INGRESADOS Y
LLENOS";
label_Informacion_Aulas_Materias.Text = "TODOS LOS CAMPOS DEBEN ESTAR CORRECTAMENTE
INGRESADOS Y LLENOS EN CADA CASO";
if(textBox_CI_Asignada.Text!="")
{
    Importar_Horario();
    if (textBox_Aula_Asignada.Text == "" )
    {
        label_Titulo1.Visible = false;
        groupBox_Horario.Visible = false;
        Importar_Reserva();
        if (textBox_Aula_Reservada.Text == "")
        {
            label_No_Clases.Visible = true;
            label_Titulo3.Visible = true;
            label_Titulo2.Visible = false;
        }
        groupBox_Reserva.Visible = false;
    }
    else
    {
        label_No_Clases.Visible = false;
        label_Titulo3.Visible = false;
        label_Titulo2.Visible = true;
        groupBox_Reserva.Visible = true;
    }
}
else
{
    label_No_Clases.Visible = false;
    label_Titulo3.Visible = false;
    label_Titulo1.Visible = true;
    label_Titulo2.Visible = false;
    groupBox_Horario.Visible = true;
    groupBox_Reserva.Visible = false;
}
}
}
dataGridView_Aulas.DataSource = Metodos.Obtener_Aulas();
int Total_Aulas = this.dataGridView_Aulas.Rows.Count;
for (int i = 0; i <= Total_Aulas; i++)
{

```

```

        if (dataGridView_Aulas.Rows[j].Cells[3].Value.ToString() == "Ocupado")
        {
            au = dataGridView_Aulas.Rows[j].Cells[0].Value.ToString();
Pintar_Aulas();
        }
    }
}
string au;
void Pintar_Aulas()
{
    if (au == AULA_201.Name)
    {
        AULA_201.BackColor = Color.Blue;
    }
    if (au == AULA_202.Name)
    {
        AULA_202.BackColor = Color.Blue;
    }
    if (au == AULA_203.Name)
    {
        AULA_203.BackColor = Color.Blue;
    }
    if (au == AULA_204.Name)
    {
        AULA_204.BackColor = Color.Blue;
    }
    if (au == AULA_205.Name)
    {
        AULA_205.BackColor = Color.Blue;
    }
    if (au == AULA_206.Name)
    {
        AULA_206.BackColor = Color.Blue;
    }
    if (au == AULA_207.Name)
    {
        AULA_207.BackColor = Color.Blue;
    }
    if (au == AULA_209.Name)
    {
        AULA_209.BackColor = Color.Blue;
    }
    if (au == AULA_302.Name)
    {
        AULA_302.BackColor = Color.Blue;
    }
    if (au == AULA_303.Name)
    {
        AULA_303.BackColor = Color.Blue;
    }
    if (au == AULA_304.Name)
    {
        AULA_304.BackColor = Color.Blue;
    }
    if (au == AULA_305.Name)
    {
        AULA_305.BackColor = Color.Blue;
    }
    if (au == AULA_306.Name)
    {
        AULA_306.BackColor = Color.Blue;
    }
    if (au == AULA_307.Name)
    {
        AULA_307.BackColor = Color.Blue;
    }
    if (au == AULA_308.Name)
    {
        AULA_308.BackColor = Color.Blue;
    }
    if (au == AULA_309.Name)
    {
        AULA_309.BackColor = Color.Blue;
    }
    if (au == AULA_310.Name)

```

```

    {
        AULA_310.BackColor = Color.Blue;
    }
    if (au == AULA_311.Name)
    {
        AULA_311.BackColor = Color.Blue;
    }
}
//-----
//-----
//-----MENÚ PRINCIAL-----
private void SalirToolStripMenuItem_Click(object sender, EventArgs e)
{
    Opcion_Mensaje = MessageBox.Show("Está seguro de querer salir", "PRECAUCIÓN",
    MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (Opcion_Mensaje == DialogResult.Yes)
    {
        serialPort1.Close();
        Form_Autenticacion Autenticacion = new Form_Autenticacion();
        Autenticacion.Show();
        this.Close();
    }
}
private void rPORTESToolStripMenuItem2_Click(object sender, EventArgs e)
{
    Form_Reportes Reporte = new Form_Reportes();
    Reporte.textBox_Cedula_Reporte.Text = textBox_CID.Text;
    if (textBox_RUD.Text!="Administrador")
    {
        Reporte.textBox_Cedula_Reporte.Enabled = false;
    }
    Reporte.Show();
}
//-----
//-----
//-----COMUNICACIÓN SERIAL Y TIEMPO-----
public char[] rxc=new char[35];
private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    rx += this.serialPort1.ReadExisting();
    rxc = rx.ToCharArray();
}
private void timer1_Tick(object sender, EventArgs e)
{
    label_Timer.Text = DateTime.Now.ToLongTimeString();
}
//-----
//-----
//-----CREAR NUEVO DOCENTE -----
string Rol;
string Compara_CI_Docente;
Regex Comparar_Correo = new Regex(@"^w+([-.']w+)*@w+([-.]w+)*\lw+([-.]w+)*");
//-----MÉTODOS-----
public void Bloquear_Nuevo()
{
    textBox_CI_Docente.BackColor = Color.White;
    textBox_PNombre_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))),
    ((int)((byte)(255))), ((int)((byte)(255))));
    textBox_SNombre_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))),
    ((int)((byte)(255))), ((int)((byte)(255))));
    textBox_PApellido_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))),
    ((int)((byte)(255))), ((int)((byte)(255))));
    textBox_SApellido_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))),
    ((int)((byte)(255))), ((int)((byte)(255))));
    textBox_Direccion.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
    ((int)((byte)(255))));
    textBox_Correo.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
    ((int)((byte)(255))));
    textBox_Telefono.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
    ((int)((byte)(255))));
    groupBox_Datos_Nuevo.Enabled = true;
    groupBox_Contor_Nuevo.Enabled = false;
    groupBox_Identificacion_Nuevo.Enabled = false;
    pictureBox_Nuevo_Docente.Visible = true;
    groupBox_Rol_Nuevo.Enabled = false;
}

```

```

textBox_PNombre_Docente.Enabled = false;
textBox_SNombre_Docente.Enabled = false;
textBox_PApellido_Docente.Enabled = false;
textBox_SApellido_Docente.Enabled = false;
textBox_Direccion.Enabled = false;
textBox_Telefono.Enabled = false;
textBox_Correo.Enabled = false;
button_Ingresar_CI.Enabled = true;
textBox_CI_Docente.Enabled = true;
}
public void Activar_Nuevo()
{
    textBox_CI_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
    ((int)((byte)(255))));
    textBox_PNombre_Docente.BackColor=Color.White;
    textBox_SNombre_Docente.BackColor = Color.White;
    textBox_PApellido_Docente.BackColor = Color.White;
    textBox_SApellido_Docente.BackColor = Color.White;
    textBox_Direccion.BackColor = Color.White;
    textBox_Correo.BackColor =Color.White;
    textBox_Telefono.BackColor = Color.White;
    groupBox_Datos_Nuevo.Enabled = true;
    groupBox_Contor_Nuevo.Enabled = true;
    groupBox_Rol_Nuevo.Enabled = true;
    groupBox_Informacion_Nuevo.Enabled = true;
    textBox_PNombre_Docente.Enabled = true;
    textBox_SNombre_Docente.Enabled = true;
    textBox_PApellido_Docente.Enabled = true;
    textBox_SApellido_Docente.Enabled = true;
    textBox_Direccion.Enabled = true;
    textBox_Telefono.Enabled = true;
    textBox_Correo.Enabled = true;
    button_Ingresar_CI.Enabled = false;
    textBox_CI_Docente.Enabled = false;
}
public void Limpiar_Nuevo()
{
    textBox_CI_Docente.Clear();
    textBox_PNombre_Docente.Clear();
    textBox_SNombre_Docente.Clear();
    textBox_PApellido_Docente.Clear();
    textBox_SApellido_Docente.Clear();
    textBox_Direccion.Clear();
    textBox_Correo.Clear();
    textBox_Telefono.Clear();
    textBox_Nombre_Usuario.Clear();
    textBox_Contraseña_Usuario.Clear();
    pictureBox_Administrador.Visible = false;
    pictureBox_Usuario.Visible = false;
    pictureBox_Eventual.Visible = false;
    pictureBox_Conserje.Visible = false;
    pictureBox_Nuevo_Docente.Visible = true;
}
void Comparar_CI_Docente()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("Select Cedula from Usuarios where Cedula like '%{0}%' ",
    textBox_CI_Docente.Text), Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        {
            lectura = cmd.ExecuteReader();
            while (lectura.Read())
            {
                Comparar_CI_Docente = lectura.GetString(0);
            }
            Abrir_Conexion.Close();
        }
    }
    catch (Exception ex)
    {
        {
            MessageBox.Show(ex.Message);
        }
    }
}
//-----CARACTERÍSTICAS DE LOS ELEMENTOS-----
private void radioButton_Administrador_CheckedChanged(object sender, EventArgs e)

```



```

    {
        pictureBox_Administrador.Visible = true;
        pictureBox_Usuario.Visible = false;
        pictureBox_Eventual.Visible = false;
        pictureBox_Conserje.Visible = false;
        pictureBox_Nuevo_Docente.Visible = false;
        Rol = radioButton_Administrador.Text;
    }
private void radioButton_Usuario_CheckedChanged(object sender, EventArgs e)
    {
        pictureBox_Administrador.Visible = false;
        pictureBox_Usuario.Visible = true;
        pictureBox_Eventual.Visible = false;
        pictureBox_Conserje.Visible = false;
        pictureBox_Nuevo_Docente.Visible = false;
        Rol = radioButton_Usuario.Text;
    }
private void radioButton_Eventual_CheckedChanged(object sender, EventArgs e)
    {
        pictureBox_Administrador.Visible = false;
        pictureBox_Usuario.Visible = false;
        pictureBox_Eventual.Visible = true;
        pictureBox_Conserje.Visible = false;
        pictureBox_Nuevo_Docente.Visible = false;
        Rol = radioButton_Eventual.Text;
    }
private void radioButton_Conserje_CheckedChanged(object sender, EventArgs e)
    {
        pictureBox_Administrador.Visible = false;
        pictureBox_Usuario.Visible = false;
        pictureBox_Eventual.Visible = false;
        pictureBox_Conserje.Visible = true;
        pictureBox_Nuevo_Docente.Visible = false;
        Rol = radioButton_Conserje.Text;
    }
private void textBox_CI_Docente_TextChanged(object sender, EventArgs e)
    {
        textBox_Contraseña_Usuario.Text = textBox_CI_Docente.Text;
        label12_Informacion.Text = "Contiene la cédula de identidad del docente";
    }
private void textBox_PNombre_Docente_TextChanged(object sender, EventArgs e)
    {
        textBox_PNombre_Docente.CharacterCasing = CharacterCasing.Upper;
        textBox_Nombre_Usuario.Text = textBox_PNombre_Docente.Text;
        label12_Informacion.Text = "Ingrese el primer nombre del docente, asegúrese de que contenga solo letras";
    }
private void textBox_SNombre_Docente_TextChanged(object sender, EventArgs e)
    {
        textBox_SNombre_Docente.CharacterCasing = CharacterCasing.Upper;
        label12_Informacion.Text = "Ingrese el segundo nombre del docente, asegúrese de que contenga solo letras";
    }
private void textBox_PApellido_Docente_TextChanged(object sender, EventArgs e)
    {
        textBox_PApellido_Docente.CharacterCasing = CharacterCasing.Upper;
        label12_Informacion.Text = "Ingrese el primer apellido del docente, asegúrese de que contenga solo letras";
    }
private void textBox_SApellido_Docente_TextChanged(object sender, EventArgs e)
    {
        textBox_SApellido_Docente.CharacterCasing = CharacterCasing.Upper;
        label12_Informacion.Text = "Ingrese el segundo apellido del docente, asegúrese de que contenga solo letras";
    }
private void textBox_Direccion_TextChanged(object sender, EventArgs e)
    {
        textBox_Direccion.CharacterCasing = CharacterCasing.Upper;
        label12_Informacion.Text = "Ingrese la dirección de residencia del docente";
    }
private void textBox_Telefono_TextChanged(object sender, EventArgs e)
    {
        label12_Informacion.Text = "Se debe ingresar solo un número de teléfono por docente, asegúrese de que contenga solo 10 números";
    }
private void textBox_Correo_TextChanged(object sender, EventArgs e)
    {
        textBox_Correo.CharacterCasing = CharacterCasing.Lower;
        label12_Informacion.Text = "Se debe ingresar el correo institucional de la universidad asignado previamente. Debe

```

```

ingresar solamente los caracteres que se encuentran delante del: @utn.edu.ec";
}
private void textBox_CI_Docente_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Numeros_CI_Docente(e);
}
private void textBox_PNombre_Docente_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_Nombre_Docente(e);
}
private void textBox_SNombre_Docente_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_Nombre_Docente(e);
}
private void textBox_PApellido_Docente_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_PApellido_Docente(e);
}
private void textBox_SApellido_Docente_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_SApellido_Docente(e);
}
private void textBox_Telefono_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Numeros_Telefono(e);
}
private void textBox_Correo_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_Nombre_Docente(e);
}
private void radioButton_Administrador_MouseClick(object sender, MouseEventArgs e)
{
    label12_Informacion.Text = "Establece al docente como administrador del sistema, podrá hacer modificaciones en
él";
}
private void radioButton_Usuario_MouseClick(object sender, MouseEventArgs e)
{
    label12_Informacion.Text = "Establece al docente solo como usuario del sistema, podrá hacer reservaciones de
aulas y ver la disponibilidad de las mismas";
}
private void radioButton_Eventual_MouseClick(object sender, MouseEventArgs e)
{
    label12_Informacion.Text = "Establece al docente como eventual del sistema, solo podrá ver las aulas ocupadas o
disponibles";
}
private void radioButton_Conserje_MouseClick(object sender, MouseEventArgs e)
{
    label12_Informacion.Text = "Establece al docente como conserje, no podrá hacer uso del sistema solo se presentan
registros de aseo a las aulas";
}
//-----BOTONES-----
private void button_Guardar_Click(object sender, EventArgs e)
{
    string Cedula = textBox_CI_Docente.Text;
string PNombre_Docente = textBox_PNombre_Docente.Text;
    string SNombre_Docente = textBox_SNombre_Docente.Text;
    string Primer_Apellido_Docente = textBox_PApellido_Docente.Text;
    string Segundo_Apellido_Docente = textBox_SApellido_Docente.Text;
    string Nombre_Usuario = textBox_Nombre_Usuario.Text;
    string Contraseña_Usuario = textBox_Contraseña_Usuario.Text;
string Rol_Usuario = Rol;
    string Direccion = textBox_Direccion.Text;
string Telefono = textBox_Telefono.Text;
string Correo = textBox_Correo.Text+textBox_utn_edu_ec.Text;
    if (textBox_CI_Docente.Text == "" || textBox_PNombre_Docente.Text == "" || textBox_SNombre_Docente.Text == ""
|| textBox_PApellido_Docente.Text == "" || textBox_SApellido_Docente.Text == "" || textBox_Direccion.Text == "" ||
textBox_Telefono.Text == "" || textBox_Correo.Text=="")
    {
        MessageBox.Show("Campos vacíos. Algún dato no es ingresado correctamente, asegúrese de tener todos los
campos llenos", "DATOS ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if (PNombre_Docente.Length > 15)
    {
        MessageBox.Show("El primer nombre del docente es incorrecto, Ingrese un máximo de 15 caracteres para el
nombre del docente", "NOMBRE DEL DOCENTE INCORRECTO", MessageBoxButtons.OK,

```

```

MessageBoxIcon.Error);
    textBox_PNombre_Docente.Clear();
    textBox_PNombre_Docente.Focus();
}
else if (SNombre_Docente.Length > 15)
{
    MessageBox.Show("El segundo nombre del docente es incorrecto, Ingrese un máximo de 15 caracteres para el
nombre del docente", "NOMBRE DEL DOCENTE INCORRECTO", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    textBox_SNombre_Docente.Clear();
    textBox_SNombre_Docente.Focus();
}
else if (Primer_Apellido_Docente.Length > 15)
{
    MessageBox.Show("El primer apellido del docente es incorrecto, Ingrese un máximo de 15 caracteres para el
primer apellido del docente", "APELLIDO DEL DOCENTE INCORRECTO", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    textBox_PApellido_Docente.Clear();
    textBox_PApellido_Docente.Focus();
}
else if (Segundo_Apellido_Docente.Length > 15)
{
    MessageBox.Show("El segundo apellido del docente es incorrecto, Ingrese un máximo de 15 caracteres para el
segundo apellido del docente", "APELLIDO DEL DOCENTE INCORRECTO", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    textBox_SApellido_Docente.Clear();
    textBox_SApellido_Docente.Focus();
}
else if (Telefono.Length != 10)
{
    MessageBox.Show("El número de teléfono celular del docente es incorrecto. Asegúrese de que este número
contenga específicamente 10 dígitos, verifique el número e intente nuevamente", "TELÉFONO CELULAR
INCORRECTO", MessageBoxButtons.OK, MessageBoxIcon.Error);
    textBox_Telefono.Clear();
    textBox_Telefono.Focus();
}
else if (radioButton_Administrador.Checked == false && radioButton_Usuario.Checked == false &&
radioButton_Eventual.Checked == false && radioButton_Conserje.Checked == false)
{
    MessageBox.Show("Modo de ingreso incorrecto. Asegúrese de elegir la forma de ingreso que se le da al
docente", "MODO DE INGRESO INCORRECTO", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
else
{
    if (Comparar_Correo.IsMatch(Correo))
    {
        int resultado_1 = Metodos.Crear_Usuario(Cedula, PNombre_Docente, SNombre_Docente,
Primer_Apellido_Docente, Segundo_Apellido_Docente, Nombre_Usuario, Contraseña_Usuario, Rol_Usuario,
Direccion, Telefono, Correo, "Activo");
        if (resultado_1 > 0)
        {
            MessageBox.Show("Datos guardados correctamente", "DATOS GUARDADOS", MessageBoxButtons.OK,
MessageBoxIcon.None);
            Limpiar_Nuevo();
            Bloquear_Nuevo();
            textBox_CI_Docente.Focus();
            label12_Informacion.Text = "TODOS LOS CAMPOS DEBEN ESTAR CORRECTAMENTE INGRESADOS Y
LLENOS";
        }
        else
        {
            MessageBox.Show("No se pudo guardar los datos del docente", "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("El correo electrónico no es válido, verifique e intente nuevamente", "CORREO
INCORRECTO", MessageBoxButtons.OK, MessageBoxIcon.Error);
        textBox_Correo.Clear();
        textBox_Correo.Focus();
    }
}
}
private void button_Limpiar_Click(object sender, EventArgs e)

```

```

{
    Limpiar_Nuevo();
    Bloquear_Nuevo();
    textBox_CI_Docente.Focus();
}
private void button_Ingresar_CI_Click(object sender, EventArgs e)
{
    char[] vector = textBox_CI_Docente.Text.ToArray();
    int sumatotal = 0, numero;
    if (textBox_CI_Docente.Text==" " || vector.Length!=10)
    {
        MessageBox.Show("El número de cédula de identidad del docente es incorrecto. Este debe contener
especificamente 10 dígitos, verifique el número e intente nuevamente.", "CÉDULA INCORRECTA",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        textBox_CI_Docente.Clear();
        textBox_Contraseña_Usuario.Clear();
    }
    else
    {
        for (int i = 0; i < vector.Length - 1; i++)
        {
            numero = Convert.ToInt32(vector[i].ToString());
            if ((i + 1) % 2 == 1)
            {
                numero = Convert.ToInt32(vector[i].ToString()) * 2;
                if (numero > 9)
                {
                    numero = numero - 9;
                }
            }
            sumatotal += numero;
        }
        sumatotal = 10 - (sumatotal % 10);
        if (sumatotal == Convert.ToInt32(new string((vector[9], 1)) || sumatotal > 9)
        {
            MessageBox.Show("El número de cédula es correcto, puede continuar.", "CÉDULA CORRECTA",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            Comparar_CI_Docente();
            if (Compara_CI_Docente == textBox_CI_Docente.Text || Compara_CI_Docente == "")
            {
                MessageBox.Show("Ya existe este número de cédula en la base de datos del sistema. Por favor asegúrese
de que el docente no esté registrado anteriormente buscándolo en el sistema.", "CÉDULA EXISTENTE",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                textBox_CI_Docente.Clear();
                textBox_CI_Docente.Focus();
            }
            else
            {
                Activar_Nuevo();
                textBox_PNombre_Docente.Focus();
            }
        }
        else
        {
            MessageBox.Show("Número de cédula es incorrecta, puede continuar", "CÉDULA INCORRECTA",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            textBox_CI_Docente.Text = "";
            textBox_CI_Docente.Focus();
        }
    }
}
//-----
//-----
//-----BUSCAR, ELIMINAR Y MODIFICAR DOCENTE-----
DialogResult Opcion_Mensaje;
public Datos_Docente Docente_Seleccionado { get; set; }
public Datos_Docente Docente_Actual { get; set; }
//----- MÉTODOS-----
public void Limpiar_Buscar()
{
    text_Buscar_CI_Docente.Clear();
    text_Buscar_Nombre_Docente.Clear();
    text_Buscar_Apellido_Docente.Clear();
    text_CI_Docente.Clear();
    text_PNombre_Docente.Clear();
}

```

```

        text_SNombre_Docente.Clear();
        text_PApellido_Docente.Clear();
        text_SApellido_Docente.Clear();
        text_Direccion.Clear();
        text_Telefono.Clear();
        text_Correo.Clear();
        text_Nombre_Usuario.Clear();
        text_Contraseña_Usuario.Clear();
        dataGridView_Datos.ClearSelection();
pictureBox_Modificar.Visible = false;
pictureBox_Eliminar.Visible = false;
pictureBox_Buscar.Visible = false;
pictureBox_Incognita.Visible = true;
comboBox_Rol.Items.Clear();
comboBox_Estado.Items.Clear();
    }
    public void Bloquear_Buscar()
    {
        text_PNombre_Docente.Enabled = false;
        text_PNombre_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        text_SNombre_Docente.Enabled = false;
        text_SNombre_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        text_PApellido_Docente.Enabled = false;
        text_PApellido_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        text_SApellido_Docente.Enabled = false;
        text_SApellido_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        text_Direccion.Enabled = false;
        text_Direccion.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        text_Telefono.Enabled = false;
        text_Telefono.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        text_Correo.Enabled = false;
        text_Correo.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        text_Nombre_Usuario.Enabled = false;
        text_Contraseña_Usuario.Enabled = false;
        comboBox_Rol.Enabled = false;
        comboBox_Rol.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        comboBox_Estado.Enabled = false;
        comboBox_Estado.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
    }
    public void Habilitar_Buscar()
    {
        text_PNombre_Docente.Enabled = true;
        text_PNombre_Docente.BackColor = Color.White;
        text_SNombre_Docente.Enabled = true;
        text_SNombre_Docente.BackColor = Color.White;
        text_PApellido_Docente.Enabled = true;
        text_PApellido_Docente.BackColor = Color.White;
        text_SApellido_Docente.Enabled = true;
        text_SApellido_Docente.BackColor = Color.White;
        text_Direccion.Enabled = true;
        text_Direccion.BackColor = Color.White;
        text_Telefono.Enabled = true;
        text_Telefono.BackColor = Color.White;
        text_Correo.Enabled = true;
        text_Correo.BackColor = Color.White;
        text_Nombre_Usuario.Enabled = false;
        text_Contraseña_Usuario.Enabled = false;
        comboBox_Rol.Enabled = true;
        comboBox_Rol.BackColor = Color.White;
        comboBox_Estado.Enabled = true;
        comboBox_Estado.BackColor = Color.White;
    }
    //-----CARACTERÍSTICAS DE LOS ELEMENTOS-----
    private void radioButton_Buscar_CheckedChanged(object sender, EventArgs e)
    {
        groupBox_Opciones_Busqueda.Enabled = true;
    }

```

```

        radioButton_Modificar.Enabled = false;
        radioButton_Eliminar.Enabled = false;
        radioButton_Buscar.Enabled = false;
    }
    private void radioButton_Modificar_CheckedChanged(object sender, EventArgs e)
    {
        radioButton_Eliminar.Enabled = false;
        groupBox_Opciones_Busqueda.Enabled = false;
        groupBox_Resultados_Busqueda.Enabled = false;
        if (groupBox_Datos.Enabled == true)
        {
            btn_Ok.Enabled = true;
            btn_Cancelar.Enabled = true;
            Habilitar_Buscar();
        }
        else
        {
            MessageBox.Show("No se ha realizado ninguna búsqueda.");
        }
    }
    private void radioButton_Eliminar_CheckedChanged(object sender, EventArgs e)
    {
        radioButton_Modificar.Enabled = false;
        groupBox_Opciones_Busqueda.Enabled = false;
        groupBox_Resultados_Busqueda.Enabled = false;
        if (groupBox_Datos.Enabled == true)
        {
            btn_Ok.Enabled = true;
            btn_Cancelar.Enabled = true;
        }
        else
        {
            MessageBox.Show("No se ha realizado ninguna búsqueda");
        }
    }
    private void radioButton_Buscar_MouseMove(object sender, MouseEventArgs e)
    {
        pictureBox_Buscar.Visible = true;
        pictureBox_Modificar.Visible = false;
        pictureBox_Eliminar.Visible = false;
        pictureBox_Incognita.Visible = false;
    }
    private void radioButton_Modificar_MouseMove(object sender, MouseEventArgs e)
    {
        pictureBox_Buscar.Visible = false;
        pictureBox_Modificar.Visible = true;
        pictureBox_Eliminar.Visible = false;
        pictureBox_Incognita.Visible = false;
    }
    private void radioButton_Eliminar_MouseMove(object sender, MouseEventArgs e)
    {
        pictureBox_Buscar.Visible = false;
        pictureBox_Modificar.Visible = false;
        pictureBox_Eliminar.Visible = true;
        pictureBox_Incognita.Visible = false;
    }
    private void tabPage_Buscar_modificar_Eliminar_MouseMove(object sender, MouseEventArgs e)
    {
        if (radioButton_Buscar.Checked == true || radioButton_Modificar.Checked == true || radioButton_Eliminar.Checked
    == true)
        {
            pictureBox_Incognita.Visible = false;
        }
        else
        {
            pictureBox_Incognita.Visible = true;
        }
    }
    private void text_Buscar_Nombre_Docente_TextChanged(object sender, EventArgs e)
    {
        text_Buscar_Nombre_Docente.CharacterCasing = CharacterCasing.Upper;
    }
    private void text_Buscar_Apellido_Docente_TextChanged(object sender, EventArgs e)
    {
        text_Buscar_Apellido_Docente.CharacterCasing = CharacterCasing.Upper;
    }

```

```

}
private void text_PNombre_Docente_TextChanged(object sender, EventArgs e)
{
    text_PNombre_Docente.CharacterCasing = CharacterCasing.Upper;
}
private void text_SNombre_Docente_TextChanged(object sender, EventArgs e)
{
    text_SNombre_Docente.CharacterCasing = CharacterCasing.Upper;
}
private void text_PApellido_Docente_TextChanged(object sender, EventArgs e)
{
    text_PApellido_Docente.CharacterCasing = CharacterCasing.Upper;
}
private void text_SApellido_Docente_TextChanged(object sender, EventArgs e)
{
    text_SApellido_Docente.CharacterCasing = CharacterCasing.Upper;
}
private void text_Direccion_TextChanged(object sender, EventArgs e)
{
    text_Direccion.CharacterCasing = CharacterCasing.Upper;
}
private void text_Correo_TextChanged(object sender, EventArgs e)
{
    text_Correo.CharacterCasing = CharacterCasing.Lower;
}
private void radioButton_Cod_Docente_CheckedChanged(object sender, EventArgs e)
{
    text_Buscar_CI_Docente.Enabled = true;
    text_Buscar_CI_Docente.Focus();
    text_Buscar_CI_Docente.BackColor = Color.White;
    text_Buscar_Nombre_Docente.Enabled = false;
    text_Buscar_Nombre_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))),
((int)((byte)(255))), ((int)((byte)(255))));
    text_Buscar_Apellido_Docente.Enabled = false;
    text_Buscar_Apellido_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))),
((int)((byte)(255))), ((int)((byte)(255))));
    text_Buscar_Nombre_Docente.Clear();
    text_Buscar_Apellido_Docente.Clear();
    dataGridView_Datos.ClearSelection();
    btn_Limpiar.Enabled = true;
}
private void radioButton_Nombre_Docente_CheckedChanged(object sender, EventArgs e)
{
    text_Buscar_CI_Docente.Enabled = false;
    text_Buscar_CI_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
((int)((byte)(255))));
    text_Buscar_Nombre_Docente.Enabled = true;
text_Buscar_Nombre_Docente.Focus();
    text_Buscar_Nombre_Docente.BackColor = Color.White;
    text_Buscar_Apellido_Docente.Enabled = false;
text_Buscar_Apellido_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))),
((int)((byte)(255))), ((int)((byte)(255))));
    text_Buscar_CI_Docente.Clear();
text_Buscar_Apellido_Docente.Clear();
dataGridView_Datos.ClearSelection();
    btn_Limpiar.Enabled = true;
}
private void radioButton_Apellido_Docente_CheckedChanged(object sender, EventArgs e)
{
    text_Buscar_CI_Docente.Enabled = false;
    text_Buscar_CI_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
((int)((byte)(255))));
    text_Buscar_Nombre_Docente.Enabled = false;
    text_Buscar_Nombre_Docente.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))),
((int)((byte)(255))), ((int)((byte)(255))));
text_Buscar_Apellido_Docente.Enabled = true;
    text_Buscar_Apellido_Docente.BackColor = Color.White;
    text_Buscar_Apellido_Docente.Focus();
text_Buscar_CI_Docente.Clear();
text_Buscar_Nombre_Docente.Clear();
dataGridView_Datos.ClearSelection();
    btn_Limpiar.Enabled = true;
}
private void text_Buscar_CI_Docente_KeyPress(object sender, KeyPressEventArgs e)
{

```

```

        Metodos.Solo_Numeros_CI_Docente(e);
groupBox_Resultados_Busqueda.Enabled = true;
        dataGridView_Datos.DataSource = Metodos.Buscar_CI_Docente(text_Buscar_CI_Docente.Text);
    }
    private void text_Buscar_Nombre_Docente_KeyPress(object sender, KeyPressEventArgs e)
    {
        Metodos.Solo_Letras_Nombre_Docente(e);
groupBox_Resultados_Busqueda.Enabled = true;
dataGridView_Datos.DataSource = Metodos.Buscar_Nombre_Docente(text_Buscar_Nombre_Docente.Text);
    }
    private void text_Buscar_Apellido_Docente_KeyPress(object sender, KeyPressEventArgs e)
    {
        Metodos.Solo_Letras_PAellido_Docente(e);
groupBox_Resultados_Busqueda.Enabled = true;
dataGridView_Datos.DataSource = Metodos.Buscar_PAellido_Docente(text_Buscar_Apellido_Docente.Text);
    }
    private void text_PNombre_Docente_KeyPress(object sender, KeyPressEventArgs e)
    {
        Metodos.Solo_Letras_Nombre_Docente(e);
    }
    private void text_SNombre_Docente_KeyPress(object sender, KeyPressEventArgs e)
    {
        Metodos.Solo_Letras_Nombre_Docente(e);
    }
    private void text_PApellido_Docente_KeyPress(object sender, KeyPressEventArgs e)
    {
        Metodos.Solo_Letras_PAellido_Docente(e);
    }
    private void text_SApellido_Docente_KeyPress(object sender, KeyPressEventArgs e)
    {
        Metodos.Solo_Letras_SApellido_Docente(e);
    }
    private void text_Telefono_KeyPress(object sender, KeyPressEventArgs e)
    {
        Metodos.Solo_Numeros_Telefono(e);
    }
}
//-----BOTONES-----
private void btn_Limpiar_Click(object sender, EventArgs e)
{
    text_Buscar_CI_Docente.Clear();
    text_Buscar_Nombre_Docente.Clear();
    text_Buscar_Apellido_Docente.Clear();
dataGridView_Datos.ClearSelection();
}
private void btn_Cancelar_Click(object sender, EventArgs e)
{
    if (radioButton_Modificar.Checked == true)
    {
        MessageBox.Show("Modificación cancelada");
    }
    else if (radioButton_Eliminar.Checked == true)
    {
        MessageBox.Show("Eliminación cancelada");
    }
    radioButton_Buscar.Enabled = true;
    radioButton_Modificar.Enabled = false;
    radioButton_Eliminar.Enabled = false;
    radioButton_Modificar.Checked = false;
    radioButton_Eliminar.Checked = false;
    groupBox_Opciones_Busqueda.Enabled = false;
    groupBox_Datos.Enabled = false;
groupBox_Resultados_Busqueda.Enabled = false;
    Limpiar_Buscar();
    Bloquear_Buscar();
}
private void btn_Ok_Click(object sender, EventArgs e)
{
    if (radioButton_Modificar.Checked == true)
    {
        if (Convert.ToString(text_Telefono.Text).Length != 10)
        {
            MessageBox.Show("El teléfono del docente no es correcto, verifique e intente nuevamente");
        }
        text_Telefono.Clear();
        text_Telefono.Focus();
    }
}

```



```

else if (!Comparar_Correo.IsMatch(text_Correo.Text))
{
    MessageBox.Show("El correo del docente no es correcto, verifique e intente nuevamente");
text_Correo.Clear();
    text_Correo.Focus();
}
else
{
    Datos_Docente Modificar_Docente = new Datos_Docente();
    Modificar_Docente.Primer_Nombre_Docente = text_PNombre_Docente.Text;
    Modificar_Docente.Segundo_Nombre_Docente = text_SNombre_Docente.Text;
    Modificar_Docente.Primer_Apellido_Docente = text_PApellido_Docente.Text;
    Modificar_Docente.Segundo_Apellido_Docente = text_SApellido_Docente.Text;
    Modificar_Docente.Nombre_Usuario = text_Nombre_Usuario.Text;
    Modificar_Docente.Contraseña_Usuario = text_Contraseña_Usuario.Text;
    Modificar_Docente.Rol_Usuario = comboBox_Rol.Text;
    Modificar_Docente.Direccion_Docente = text_Direccion.Text;
    Modificar_Docente.Telefono_Docente = text_Telefono.Text;
    Modificar_Docente.Correo_Docente = text_Correo.Text;
    Modificar_Docente.Estado_Docente = comboBox_Estado.Text;
    Modificar_Docente.Cedula = Docente_Actual.Cedula;
    int resultado = Metodos.Modificar_Usuario(Modificar_Docente);
    if (resultado >= 0)
    {
        MessageBox.Show("Docente Modificado con éxito");
        Bloquear_Buscar();
        Limpiar_Buscar();
    }
    else
    {
        MessageBox.Show("No se pudo modificar al docente");
    }

    radioButton_Buscar.Enabled = true;
    radioButton_Modificar.Enabled = false;
    radioButton_Eliminar.Enabled = false;
    radioButton_Modificar.Checked = false;
    radioButton_Eliminar.Checked = false;
    groupBox_Opciones_Busqueda.Enabled = false;
    groupBox_Datos.Enabled = false;
    groupBox_Resultados_Busqueda.Enabled = false;
}
}
else if (radioButton_Eliminar.Checked == true)
{
    if (MessageBox.Show("Está seguro(a) que desea eliminar al docente del registro", "PRECAUCIÓN",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        int resultado = Metodos.Eliminar_Usuario(Docente_Actual.Cedula);
        if (resultado > 0)
        {
            MessageBox.Show("Docente eliminado satisfactoriamente", "DOCENTE ELIMINADO",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            Limpiar_Buscar();
        }
        else
        {
            MessageBox.Show("No se pudo eliminar al docente", "ERROR", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }
    }
    else
    {
        MessageBox.Show("Se canceló la eliminación", "CANCELADO", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    radioButton_Buscar.Enabled = true;
    radioButton_Modificar.Enabled = false;
    radioButton_Eliminar.Enabled = false;
    radioButton_Modificar.Checked = false;
    radioButton_Eliminar.Checked = false;
    groupBox_Opciones_Busqueda.Enabled = false;
    groupBox_Datos.Enabled = false;
    groupBox_Resultados_Busqueda.Enabled = false;
}
}
}
}

```

```

private void btn_Aceptar_Click(object sender, EventArgs e)
{
    comboBox_Rol.Items.AddRange(new object[] { "Administrador", "Usuario", "Eventual", "Conserje" });
    comboBox_Estado.Items.AddRange(new object[] { "Activo", "Eliminado" });
    if (dataGridView_Datos.SelectedRows.Count == 1)
    {
        radioButton_Modificar.Enabled = true;
        radioButton_Eliminar.Enabled = true;
        radioButton_Buscar.Enabled = false;
        groupBox_Opciones_Busqueda.Enabled = false;
        btn_Ok.Enabled = false;
        btn_Cancelar.Enabled = false;
        string Obtener_Cod_Docente = Convert.ToString(dataGridView_Datos.CurrentRow.Cells[0].Value);
        Docente_Seleccionado = Metodos.Obtener_Docente(Obtener_Cod_Docente);
        if (Docente_Seleccionado != null)
        {
            groupBox_Datos.Enabled = true;
            Bloquear_Buscar();
            Docente_Actual = Docente_Seleccionado;
            text_CI_Docente.Text = Docente_Seleccionado.Cedula;
            text_PNombre_Docente.Text = Docente_Seleccionado.Primer_Nombre_Docente;
            text_SNombre_Docente.Text = Docente_Seleccionado.Segundo_Nombre_Docente;
            text_PAellido_Docente.Text = Docente_Seleccionado.Primer_Apellido_Docente;
            text_SApellido_Docente.Text = Docente_Seleccionado.Segundo_Apellido_Docente;
            text_Nombre_Usuario.Text = Docente_Seleccionado.Nombre_Usuario;
            text_Contraseña_Usuario.Text = Docente_Seleccionado.Contraseña_Usuario;
            comboBox_Rol.Text = Docente_Seleccionado.Rol_Usuario;
            text_Direccion.Text = Docente_Seleccionado.Direccion_Docente;
            text_Telefono.Text = Docente_Seleccionado.Telefono_Docente;
            text_Correo.Text = Docente_Seleccionado.Correo_Docente;
            comboBox_Estado.Text = Docente_Seleccionado.Estado_Docente;
        }
    }
    else
    {
        MessageBox.Show("No ha seleccionado ningún docente para la consulta", "ADVERTENCIA",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
//-----
//-----
//-----CREAR AULAS O MATERIAS-----

//-----MÉTODOS-----
public void Limpiar_Aulas()
{
    textBox_Nueva_Aula.Clear();
    comboBox_Seleccion_Nueva_Aula.Items.Clear();
    comboBox_Ubicacion_Aula.Items.Clear();
    textBox_Capacidad_Aula.Clear();
    comboBox_Aula_Horario.Items.Clear();
    comboBox_Numero_Aula.Items.Clear();
    comboBox_Ubicacion_Aula.Items.AddRange(new object[] { "Planta Baja", "Primer Piso", "Segundo Piso", "Tercer
Piso", "Cuarto Piso" });
    Aulas();
    Aulas_Disponibles();
}
public void Limpiar_Materia()
{
    textBox_Nueva_Materia.Clear();
    comboBox_Creditos_Materia.Items.Clear();
    comboBox_Seleccion_Nueva_Materia.Items.Clear();
    comboBox_Materia_Horario.Items.Clear();
    comboBox_Materia.Items.Clear();
    Materias();
    comboBox_Creditos_Materia.Items.AddRange(new object[] { "1", "2", "3", "4", "5", "6" });
}
public void Limpiar_Carrera()
{
    textBox_Nueva_Carrera.Clear();
    textBox_Alias_Carrera.Clear();
    comboBox_Seleccion_Nueva_Carrera.Items.Clear();
    comboBox_Carrera_Horario.Items.Clear();
    comboBox_Carreras.Items.Clear();
    Carreras();
}

```

```

}
//-----CARACTERÍSTICAS DE LOS ELEMENTOS-----
private void textBox_Capacidad_Aula_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Numeros_CI_Docente(e);
}
private void textBox_Nueva_Aula_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Numeros_CI_Docente(e);
}
private void textBox_Nueva_Aula_TextChanged(object sender, EventArgs e)
{
    label_Informacion_Aulas_Materias.Text = "El nombre del Aula contiene el formato siguiente: " + " 'AULA-XXX'. " +
    "Asegúrese de que ésta tenga el formato indicado donde 'xxx' es el número correspondiente al aula a crear";
}
private void textBox_Capacidad_Aula_TextChanged(object sender, EventArgs e)
{
    label_Informacion_Aulas_Materias.Text = "Especifica el número máximo de estudiantes que ésta contendrá";
}
private void textBox_Nueva_Materia_TextChanged(object sender, EventArgs e)
{
    textBox_Nueva_Materia.CharacterCasing = CharacterCasing.Upper;
    label_Informacion_Aulas_Materias.Text = "Especifica el nombre de la materia que se crea";
}
private void textBox_Nueva_Carrera_TextChanged(object sender, EventArgs e)
{
    textBox_Nueva_Carrera.CharacterCasing = CharacterCasing.Upper;
    label_Informacion_Aulas_Materias.Text = "Especifica el nombre de la carrera que se crea";
}
private void textBox_Alias_Carrera_TextChanged(object sender, EventArgs e)
{
    textBox_Alias_Carrera.CharacterCasing = CharacterCasing.Upper;
    label_Informacion_Aulas_Materias.Text = "Especifica el alias de la carrera que se crea";
}
private void comboBox_Ubicacion_Aula_SelectedIndexChanged(object sender, EventArgs e)
{
    label_Informacion_Aulas_Materias.Text = "Debe seleccionar el piso donde el aula está ubicada";
}
private void comboBox_Creditos_Materia_SelectedIndexChanged(object sender, EventArgs e)
{
    label_Informacion_Aulas_Materias.Text = "Debe seleccionar el número de créditos y horas a la semana que la
materia contiene";
}
private void comboBox_Seleccion_Nueva_Aula_SelectedIndexChanged(object sender, EventArgs e)
{
    Cap_Ubic_Aula_Nueva();
    textBox_Capacidad_Aula.Text = Convert.ToString(Capacidad_Aula_Selec_Nueva);
    comboBox_Ubicacion_Aula.Text = Ubicacion_Aula_Selec_Nueva;
    label_Informacion_Aulas_Materias.Text = "Seleccione el aula que desea modificar sus características, solamente
se puede modificar la capacidad y la ubicación de la misma";
    button_Modificar_Aula.Enabled = true;
    comboBox_Ubicacion_Aula.Enabled = true;
    textBox_Capacidad_Aula.Enabled = true;
    textBox_Nueva_Aula.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
((int)((byte)(255))));
    textBox_AULA.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
((int)((byte)(255))));
    textBox_Nueva_Aula.Enabled = false;
    textBox_Capacidad_Aula.BackColor = Color.White;
}
private void comboBox_Seleccion_Nueva_Materia_SelectedIndexChanged(object sender, EventArgs e)
{
    Cred_Esta_Materia_Nueva();
    comboBox_Creditos_Materia.Text = Convert.ToString(Creditos);
    label_Informacion_Aulas_Materias.Text = "Seleccione la materia que desea modificar sus características,
solamente se puede modificar el número de créditos y el nivel de la misma";
    textBox_Nueva_Materia.Enabled = false;
    comboBox_Creditos_Materia.Enabled = true;
    button_Crear_Materia.Enabled = false;
    button_Modificar_Materia.Enabled = true;
    textBox_Nueva_Materia.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
((int)((byte)(255))));
}
private void comboBox_Seleccion_Nueva_Carrera_SelectedIndexChanged(object sender, EventArgs e)
{
}

```

```

        Nomb_Alias_Carrera_Nueva());
        textBox_Alias_Carrera.Text = Alias_Carrera_seleccionada;
        label_Informacion_Aulas_Materias.Text = "Seleccione la carrera que desea modificar sus características,
solamente se puede modificar el alias de la misma";
        textBox_Nueva_Carrera.Enabled = false;
        button_Crear_Carrera.Enabled = false;
        button_Modificar_Carrera.Enabled = true;
        textBox_Nueva_Carrera.BackColor = System.Drawing.Color.FromArgb((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        textBox_Alias_Carrera.Enabled = true;
        textBox_Alias_Carrera.BackColor = Color.White;
    }
    private void comboBox_Hora_Inicio_Asignada_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (comboBox_Hora_Inicio_Asignada.Text == "07:00:00")
        {
            horainicio = 7;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "08:00:00")
        {
            horainicio = 8;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "09:00:00")
        {
            horainicio = 9;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "10:00:00")
        {
            horainicio = 10;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "11:00:00")
        {
            horainicio = 11;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "12:00:00")
        {
            horainicio = 12;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "13:00:00")
        {
            horainicio = 13;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "14:00:00")
        {
            horainicio = 14;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "15:00:00")
        {
            horainicio = 15;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "16:00:00")
        {
            horainicio = 16;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "17:00:00")
        {
            horainicio = 17;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "18:00:00")
        {
            horainicio = 18;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "19:00:00")
        {
            horainicio = 19;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "20:00:00")
        {
            horainicio = 20;
        }
        else if (comboBox_Hora_Inicio_Asignada.Text == "21:00:00")
        {
            horainicio = 21;
        }
        comboBox_Hora_Final_Asignada.Enabled = true;
    }

```

```

        comboBox_Hora_Inicio_Asignada.Enabled = false;
    }
    private void comboBox_Hora_Final_Asignada_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (comboBox_Hora_Final_Asignada.Text == "07:00:00")
        {
            horafin = 7;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "08:00:00")
        {
            horafin = 8;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "09:00:00")
        {
            horafin = 9;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "10:00:00")
        {
            horafin = 10;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "11:00:00")
        {
            horafin = 11;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "12:00:00")
        {
            horafin = 12;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "13:00:00")
        {
            horafin = 13;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "14:00:00")
        {
            horafin = 14;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "15:00:00")
        {
            horafin = 15;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "16:00:00")
        {
            horafin = 16;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "17:00:00")
        {
            horafin = 17;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "18:00:00")
        {
            horafin = 18;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "19:00:00")
        {
            horafin = 19;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "20:00:00")
        {
            horafin = 20;
        }
        else if (comboBox_Hora_Final_Asignada.Text == "21:00:00")
        {
            horafin = 21;
        }
        if (horainicio >= horafin)
        {
            MessageBox.Show("La hora final no puede ser igual o menor que la hora inicial, verifique los datos para continuar", "HORAS INCORRECTAS", MessageBoxButtons.OK, MessageBoxIcon.Error);
            textBox_No_Horas_Asignadas.Clear();
            comboBox_Hora_Final_Asignada.Items.Clear();
            comboBox_Hora_Final_Asignada.Items.AddRange(new object[] { "07:00:00", "08:00:00", "09:00:00", "10:00:00", "11:00:00", "12:00:00", "13:00:00", "14:00:00", "15:00:00", "16:00:00", "17:00:00", "18:00:00", "19:00:00", "20:00:00", "21:00:00" });
            comboBox_Hora_Final_Asignada.Enabled = false;
            comboBox_Hora_Inicio_Asignada.Enabled = true;
        }
    }
}

```

```

    }
    else
    {
        textBox_No_Horas_Asignadas.Text = Convert.ToString(horafin - horainicio);
    }
}
private void radioButton_Crear_Aula_CheckedChanged(object sender, EventArgs e)
{
    comboBox_Ubicacion_Aula.Items.Clear();
    comboBox_Ubicacion_Aula.Items.AddRange(new object[] { "Planta Baja", "Primer Piso", "Segundo Piso", "Tercer
Piso", "Cuarto Piso" });
    comboBox_Seleccion_Nueva_Aula.Enabled = false;
    label_Seleccion_Aula.Visible = false;
    comboBox_Seleccion_Nueva_Aula.Visible = false;
    label_Nombre_Aula.Visible = true;
    textBox_Nueva_Aula.Visible = true;
    textBox_Nueva_Aula.Clear();
    textBox_Nueva_Aula.Focus();
    button_Modificar_Aula.Enabled = false;
    textBox_Nueva_Aula.Enabled = true;
    comboBox_Ubicacion_Aula.Enabled = true;
    textBox_Capacidad_Aula.Enabled = true;
    button_Crear_Aula.Enabled = true;
    textBox_Nueva_Aula.BackColor = Color.White;
    textBox_AULA.BackColor = Color.White;
    textBox_Capacidad_Aula.BackColor = Color.White;
    textBox_Capacidad_Aula.Clear();
}
private void radioButton_Modificar_Aula_CheckedChanged(object sender, EventArgs e)
{
    comboBox_Ubicacion_Aula.Items.Clear();
    comboBox_Ubicacion_Aula.Items.AddRange(new object[] { "Planta Baja", "Primer Piso", "Segundo Piso", "Tercer
Piso", "Cuarto Piso" });
    comboBox_Aula_Horario.Items.Clear();
    comboBox_Seleccion_Nueva_Aula.Items.Clear();
    Aulas();
    comboBox_Seleccion_Nueva_Aula.Visible = true;
    label_Seleccion_Aula.Visible = true;
    textBox_Nueva_Aula.Visible = false;
    label_Nombre_Aula.Visible = false;
    comboBox_Seleccion_Nueva_Aula.Enabled = true;
    button_Modificar_Aula.Enabled = true;
    textBox_Nueva_Aula.Enabled = false;
    comboBox_Ubicacion_Aula.Enabled = false;
    textBox_Capacidad_Aula.Enabled = false;
    button_Crear_Aula.Enabled = false;
    textBox_Nueva_Aula.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
((int)((byte)(255))));
    textBox_AULA.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
((int)((byte)(255))));
    textBox_Capacidad_Aula.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
((int)((byte)(255))));
}
private void radioButton_Crear_Materia_CheckedChanged(object sender, EventArgs e)
{
    comboBox_Creditos_Materia.Items.Clear();
    comboBox_Creditos_Materia.Items.AddRange(new object[] { "1", "2", "3", "4", "5", "6" });
    comboBox_Seleccion_Nueva_Materia.Visible = false;
    label_Seleccion_Materia.Visible = false;
    textBox_Nueva_Materia.Visible = true;
    textBox_Nueva_Materia.Clear();
    textBox_Nueva_Materia.Focus();
    label_Nombre_Materia.Visible = true;
    comboBox_Seleccion_Nueva_Materia.Enabled = false;
    textBox_Nueva_Materia.Enabled = true;
    comboBox_Creditos_Materia.Enabled = true;
    button_Crear_Materia.Enabled = true;
    button_Modificar_Materia.Enabled = false;
    textBox_Nueva_Materia.BackColor = Color.White;
}
private void radioButton_Modificar_Materia_CheckedChanged(object sender, EventArgs e)
{
    comboBox_Creditos_Materia.Items.Clear();
    comboBox_Creditos_Materia.Items.AddRange(new object[] { "1", "2", "3", "4", "5", "6" });
    comboBox_Seleccion_Nueva_Materia.Items.Clear();
}

```

```

comboBox_Materia_Horario.Items.Clear();
comboBox_Materia.Items.Clear();
Materias();
    comboBox_Seleccion_Nueva_Materia.Visible = true;
    label_Seleccion_Materia.Visible = true;
    textBox_Nueva_Materia.Visible = false;
    label_Nombre_Materia.Visible = false;
    comboBox_Seleccion_Nueva_Materia.Enabled = true;
    textBox_Nueva_Materia.Enabled = false;
    comboBox_Creditos_Materia.Enabled = false;
    button_Crear_Materia.Enabled = false;
button_Modificar_Materia.Enabled = false;
    textBox_Nueva_Materia.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
    ((int)((byte)(255))));
    }
    private void radioButton_Crear_Carrera_CheckedChanged(object sender, EventArgs e)
    {
        comboBox_Seleccion_Nueva_Carrera.Visible = false;
        label_Seleccion_Carrera.Visible = false;
        textBox_Nueva_Carrera.Visible = true;
        textBox_Nueva_Carrera.Clear();
        textBox_Nueva_Carrera.Focus();
        label_Nombre_Carrera.Visible = true;
        textBox_Alias_Carrera.Clear();
        comboBox_Seleccion_Nueva_Carrera.Enabled = false;
        textBox_Nueva_Carrera.Enabled = true;
        button_Crear_Carrera.Enabled = true;
        button_Modificar_Carrera.Enabled = false;
        textBox_Nueva_Carrera.BackColor = Color.White;
        textBox_Alias_Carrera.Enabled = true;
        textBox_Alias_Carrera.BackColor = Color.White;
    }
    private void radioButton_Modificar_Carrera_CheckedChanged(object sender, EventArgs e)
    {
        comboBox_Carrera_Horario.Items.Clear();
        comboBox_Carreras.Items.Clear();
        comboBox_Seleccion_Nueva_Carrera.Items.Clear();
        Carreras();
        comboBox_Seleccion_Nueva_Carrera.Visible = true;
        label_Seleccion_Carrera.Visible = true;
        textBox_Nueva_Carrera.Visible = false;
        label_Nombre_Carrera.Visible = false;
        comboBox_Seleccion_Nueva_Carrera.Enabled = true;
        textBox_Nueva_Carrera.Enabled = false;
        button_Crear_Carrera.Enabled = false;
        button_Modificar_Carrera.Enabled = false;
        textBox_Nueva_Carrera.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
        textBox_Alias_Carrera.Enabled = false;
        textBox_Alias_Carrera.BackColor = System.Drawing.Color.FromArgb(((int)((byte)(192))), ((int)((byte)(255))),
        ((int)((byte)(255))));
    }
    //-----BOTONES-----
    private void button_Crear_Aula_Click(object sender, EventArgs e)
    {
        if (textBox_Nueva_Aula.Text == "" || comboBox_Ubicacion_Aula.Text == "" || textBox_Capacidad_Aula.Text == "")
        {
            MessageBox.Show("Campos vacíos. Algún dato no es ingresado correctamente, asegúrese de tener todos los
            campos llenos", "DATOS ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            string Nueva_Aula = (textBox_AULA.Text + "_" + textBox_Nueva_Aula.Text);
            int Capacidad_Aula = Convert.ToInt32(textBox_Capacidad_Aula.Text);
            string Ubicacion_Aula = comboBox_Ubicacion_Aula.Text;
            int resultado_1 = Metodos.Comprobar_Crear_Aula(Nueva_Aula);
            if (resultado_1 > 0)
            {
                MessageBox.Show("No se puede crear el aula por que ya existe una aula con estos datos", "ERROR",
                MessageBoxButtons.OK, MessageBoxIcon.None);
            }
            else
            {
                int resultado_2 = Metodos.Crear_Aula(Nueva_Aula, Capacidad_Aula, Ubicacion_Aula);
                if (resultado_2 > 0)
            }
        }
    }

```

```

        {
            MessageBox.Show("Datos guardados correctamente", "DATOS GUARDADOS", MessageBoxButtons.OK,
                MessageBoxIcon.None);
            Limpiar_Aulas();
            textBox_Nueva_Aula.Focus();
            label_Informacion_Aulas_Materias.Text = "TODOS LOS CAMPOS DEBEN ESTAR CORRECTAMENTE
INGRESADOS Y LLENOS";
        }
        else
        {
            MessageBox.Show("No se pudo guardar los datos del aula", "ERROR", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
        }
    }
}
}
private void button_Crear_Materia_Click(object sender, EventArgs e)
{
    if (textBox_Nueva_Materia.Text == "" || comboBox_Creditos_Materia.Text == "")
    {
        MessageBox.Show("Campos vacíos. Algún dato no se ha ingresado correctamente, asegúrese de tener todos
los campos llenos", "DATOS ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        string Nueva_Materia = textBox_Nueva_Materia.Text;
        int Numero_Ceditos = Convert.ToInt32(comboBox_Creditos_Materia.Text);
        int resultado_1 = Metodos.Comprobar_Crear_Materia(Nueva_Materia);
        if (resultado_1 > 0)
        {
            MessageBox.Show("No se puede crear la materia por que ya existe una con estos datos", "ERROR",
                MessageBoxButtons.OK, MessageBoxIcon.None);
        }
        else
        {
            int resultado_2 = Metodos.Crear_Materia(Nueva_Materia, Numero_Ceditos);
            if (resultado_2 > 0)
            {
                MessageBox.Show("Datos guardados correctamente", "DATOS GUARDADOS", MessageBoxButtons.OK,
                    MessageBoxIcon.None);
                Limpiar_Materia();
                textBox_Nueva_Materia.Focus();
                label_Informacion_Aulas_Materias.Text = "TODOS LOS CAMPOS DEBEN ESTAR CORRECTAMENTE
INGRESADOS Y LLENOS";
            }
            else
            {
                MessageBox.Show("No se pudo guardar los datos de la materia", "ERROR", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
            }
        }
    }
}
private void button_Crear_Carrera_Click(object sender, EventArgs e)
{
    if (textBox_Nueva_Carrera.Text == "" || textBox_Alias_Carrera.Text == "")
    {
        MessageBox.Show("Campos vacíos. Algún dato no se ha ingresado correctamente, asegúrese de tener todos
los campos llenos", "DATOS ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        string NuevaCarrera = textBox_Nueva_Carrera.Text;
        string AliasCarrera = textBox_Alias_Carrera.Text;
        int resultado_1 = Metodos.Comprobar_Crear_Carrera(AliasCarrera);
        if (resultado_1 > 0)
        {
            MessageBox.Show("No se puede crear la carrera por que ya existe una con estos datos", "ERROR",
                MessageBoxButtons.OK, MessageBoxIcon.None);
        }
        else
        {
            int resultado_2 = Metodos.Crear_Carrera(NuevaCarrera, AliasCarrera);
            if (resultado_2 > 0)
            {

```



```

        MessageBox.Show("Datos guardados correctamente", "DATOS GUARDADOS", MessageBoxButtons.OK,
        MessageBoxIcon.None);
        Limpiar_Carrera();
        textBox_Nueva_Carrera.Focus();
        label_Informacion_Aulas_Materias.Text = "TODOS LOS CAMPOS DEBEN ESTAR CORRECTAMENTE
INGRESADOS Y LLENOS";
    }
    else
    {
        MessageBox.Show("No se pudo guardar los datos de la carrera", "ERROR", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
}
}
private void button_Modificar_Aula_Click(object sender, EventArgs e)
{
    if (comboBox_Seleccion_Nueva_Aula.Text == "" || comboBox_Ubicacion_Aula.Text == "" ||
    textBox_Capacidad_Aula.Text == "")
    {
        MessageBox.Show("Campos vacíos. Algún dato no se ha ingresado correctamente, asegúrese de tener todos los
campos llenos", "DATOS ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        Datos_Aulas Modificar_Aula = new Datos_Aulas();
        Modificar_Aula.Nombre_Aula = comboBox_Seleccion_Nueva_Aula.Text;
        Modificar_Aula.Ubicacion = comboBox_Ubicacion_Aula.Text;
        Modificar_Aula.Capacidad = Convert.ToInt32(textBox_Capacidad_Aula.Text);
        int resultado = Metodos.Modificar_Aulas(Modificar_Aula);
        if (resultado >= 0)
        {
            MessageBox.Show("Aula modificada con éxito");
            Limpiar_Aulas();
        }
        else
        {
            MessageBox.Show("No se pudo modificar el aula");
        }
    }
}
private void button_Modificar_Materia_Click(object sender, EventArgs e)
{
    if (comboBox_Seleccion_Nueva_Materia.Text == "" || comboBox_Creditos_Materia.Text == "")
    {
        MessageBox.Show("Campos vacíos. Algún dato no se ha ingresado correctamente, asegúrese de tener todos los
campos llenos", "DATOS ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        Datos_Materias Modificar_Materia = new Datos_Materias();
        Modificar_Materia.Nombre_Materia = comboBox_Seleccion_Nueva_Materia.Text;
        Modificar_Materia.Creditos = Convert.ToInt32(comboBox_Creditos_Materia.Text);
        int resultado = Metodos.Modificar_Materias(Modificar_Materia);
        if (resultado >= 0)
        {
            MessageBox.Show("Materia modificada con éxito");
            Limpiar_Materia();
        }
        else
        {
            MessageBox.Show("No se pudo modificar la materia seleccionada");
        }
    }
}
private void button_Modificar_Carrera_Click(object sender, EventArgs e)
{
    if (comboBox_Seleccion_Nueva_Carrera.Text == "" || textBox_Alias_Carrera.Text == "")
    {
        MessageBox.Show("Campos vacíos. Algún dato no se ha ingresado correctamente, asegúrese de tener todos los
campos llenos", "DATOS ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        Datos_Carrera Modificar_Carrera = new Datos_Carrera();

```

```

Modificar_Carrera.Nombre_Carrera = comboBox_Seleccion_Nueva_Carrera.Text;
Modificar_Carrera.Alias = textBox_Alias_Carrera.Text;
int resultado = Metodos.Modificar_Carrera(Modificar_Carrera);
if (resultado >= 0)
{
    MessageBox.Show("Carrera modificada con éxito");
    Limpiar_Materia();
}
else
{
    MessageBox.Show("No se pudo modificar la carrera seleccionada");
}
}
}
//-----
//-----
//-----CREAR HORARIOS-----
int Año_Actual = Convert.ToInt32(DateTime.Now.Year.ToString());
//-----MÉTODOS-----
void Limpiar_Crear_Horario()
{
    comboBox_Dia_Horario.Items.Clear();
    comboBox_Carrera_Horario.Items.Clear();
    comboBox_Carreras.Items.Clear();
    comboBox_Docente_Horario.Items.Clear();
comboBox_Materia.Items.Clear();
    comboBox_Materia_Horario.Items.Clear();
comboBox_Aula_Horario.Items.Clear();
    comboBox_Paralelo_Horario.Items.Clear();
    comboBox_Nivel_Horario.Items.Clear();
    comboBox_Hora_Inicio_Asignada.Items.Clear();
    comboBox_Hora_Final_Asignada.Items.Clear();
    textBox_No_Horas_Asignadas.Clear();
    Docentes();
    Materias();
    Aulas();
    Carreras();
    comboBox_Dia_Horario.Items.AddRange(new object[] { "LUNES", "MARTES", "MIÉRCOLES", "JUEVES",
"VIERNES", "SÁBADO" });
    comboBox_Paralelo_Horario.Items.AddRange(new object[] { "A", "B" });
    comboBox_Nivel_Horario.Items.AddRange(new object[] { "1ro", "2do", "3ro", "4to", "5to", "6to", "7mo", "8vo", "9no",
"10mo" });
    comboBox_Hora_Inicio_Asignada.Items.AddRange(new object[] { "07:00:00", "08:00:00", "09:00:00", "10:00:00",
"11:00:00", "12:00:00", "13:00:00", "14:00:00", "15:00:00", "16:00:00", "17:00:00", "18:00:00", "19:00:00", "20:00:00",
"21:00:00" });
    comboBox_Hora_Final_Asignada.Items.AddRange(new object[] { "07:00:00", "08:00:00", "09:00:00", "10:00:00",
"11:00:00", "12:00:00", "13:00:00", "14:00:00", "15:00:00", "16:00:00", "17:00:00", "18:00:00", "19:00:00", "20:00:00",
"21:00:00" });
    groupBox_Horas_Asignadas.Enabled = false;
}
//-----CARACTERÍSTICAS DE LOS ELEMENTOS-----

//-----BOTONES-----
private void button_Crear_Horario_Click(object sender, EventArgs e)
{
    if (comboBox_Dia_Horario.Text == "" || comboBox_Docente_Horario.Text == "" || comboBox_Materia_Horario.Text
== "" || comboBox_Carrera_Horario.Text == "" || comboBox_Aula_Horario.Text == "" ||
comboBox_Hora_Inicio_Asignada.Text == "" || comboBox_Hora_Final_Asignada.Text == "" ||
comboBox_Paralelo_Horario.Text == "" || comboBox_Nivel_Horario.Text == "" ||
textBox_No_Horas_Asignadas.Text=="")
    {
        MessageBox.Show("Campos vacíos. Algún dato no se ha ingresado correctamente, asegúrese de tener todos los campos
llenos", "DATOS ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        string Dia = comboBox_Dia_Horario.Text;
        string Docente = comboBox_Docente_Horario.Text;
        string Materia = comboBox_Materia_Horario.Text;
        string Periodo = comboBox_Periodo_Horario.Text;
        string Carrera = comboBox_Carrera_Horario.Text;
string Aula = comboBox_Aula_Horario.Text;
        string Paralelo = comboBox_Paralelo_Horario.Text;
        string Nivel = comboBox_Nivel_Horario.Text;
        string Hora_Inicio = comboBox_Hora_Inicio_Asignada.Text;
        string Hora_Final =comboBox_Hora_Final_Asignada.Text;
    }
}
}
}

```

```

        int N_Horas = Convert.ToInt32(textBox_No_Horas_Asignadas.Text);
        int resultado = Metodos.Comprobar_Horario(Dia, Aula, Hora_Inicio, Hora_Final, Periodo);
        if (resultado > 0)
        {
            MessageBox.Show("No se puede crear el horario, ya existe un horario con estos parámetros", "ERROR",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
            Limpiar_Crear_Horario();
        }
        else
        {
            int resultado_1 = Metodos.Crear_Horario(Dia, Docente, Materia, Periodo, Carrera, Aula, Paralelo, Nivel,
            Hora_Inicio, Hora_Final, N_Horas);
            if (resultado_1 > 0)
            {
                MessageBox.Show("Datos guardados correctamente", "DATOS GUARDADOS", MessageBoxButtons.OK,
                MessageBoxIcon.None);
                Limpiar_Crear_Horario();
            }
            else
            {
                MessageBox.Show("No se pudo guardar los datos del docente", "ERROR", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
            }
        }
    }
}
private void button_Crear_Periodo_Academico_Click(object sender, EventArgs e)
{
    if (comboBox_Periodo_Horario.Text != "")
    {
        if (MessageBox.Show("Está seguro de crear un nuevo horario.", "SUGERENCIA",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)
        {
            int resultado_1 = Metodos.Comprobar_Periodo(comboBox_Periodo_Horario.Text);
            if (resultado_1 > 0)
            {
                MessageBox.Show("No se pudo crear el periodo seleccionado por que ya existe uno igual.", "ERROR",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
                groupBox_Crear_Horario.Enabled = false;
                groupBox_Crear_Periodo_Academico.Enabled = true;
                comboBox_Periodo_Horario.Text = null;
            }
            else
            {
                if (MessageBox.Show("Para poder crear un horario nuevo, debe inactivar el antiguo horario. ¿Desea realizar
                esta acción?", "SUGERENCIA", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) ==
                DialogResult.OK)
                {
                    int resultado_3 = Metodos.Modificar_Periodo(comboBox_Periodo_Horario.Text);
                    if (resultado_3 > 0)
                    {
                        int resultado_2 = Metodos.Crear_Periodo(comboBox_Periodo_Horario.Text);
                        if (resultado_2 > 0)
                        {
                            MessageBox.Show("El antiguo periodo académico fue modificado con éxito y el nuevo periodo
                            académico fue creado correctamente, puede continuar", "MODIFICACIÓN Y CREACIÓN DE
                            PERIODOS", MessageBoxButtons.OK, MessageBoxIcon.None);
                            groupBox_Crear_Horario.Enabled = true;
                            groupBox_Crear_Periodo_Academico.Enabled = false;
                        }
                        else
                        {
                            MessageBox.Show("No se pudo crear el periodo seleccionado, error en sintaxis", "ERROR",
                            MessageBoxButtons.OK, MessageBoxIcon.Error);
                        }
                    }
                    else
                    {
                        MessageBox.Show("No se pudo modificar el periodo seleccionado, error en sintaxis", "ERROR",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
            }
        }
    }
}

```

```

MessageBox.Show("No podrá crear un periodo académico nuevo mientras se encuentre activo el anterior.", "PERIODO",
MessageBoxButtons.OK, MessageBoxIcon.None);
groupBox_Crear_Horario.Enabled = false;
    groupBox_Crear_Periodo_Academico.Enabled = true;
    comboBox_Periodo_Horario.Text = null;
    }
    }
    }
    else
    {
groupBox_Crear_Horario.Enabled = false;
    comboBox_Periodo_Horario.Text = null;
    }
    }
    else
    {
    MessageBox.Show("Aún no ha seleccionado un periodo académico. Verifique los datos e intente nuevamente");
    }
    }
}
private void button_Limpiar_Horario_Click(object sender, EventArgs e)
{
    Limpiar_Crear_Horario();
}
//-----
//-----MODIFICAR DATOS PERSONALES-----
//-----
private void textBox_PND_TextChanged(object sender, EventArgs e)
{
    textBox_PND.CharacterCasing = CharacterCasing.Upper;
}
private void textBox_SND_TextChanged(object sender, EventArgs e)
{
    textBox_SND.CharacterCasing = CharacterCasing.Upper;
}
private void textBox_PAD_TextChanged(object sender, EventArgs e)
{
    textBox_PAD.CharacterCasing = CharacterCasing.Upper;
}
private void textBox_SAD_TextChanged(object sender, EventArgs e)
{
    textBox_SAD.CharacterCasing = CharacterCasing.Upper;
}
private void textBox_DD_TextChanged(object sender, EventArgs e)
{
    textBox_DD.CharacterCasing = CharacterCasing.Upper;
}
private void textBox_CD_TextChanged(object sender, EventArgs e)
{
    textBox_CD.CharacterCasing = CharacterCasing.Lower;
}
private void textBox_PND_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_Nombre_Docente(e);
}
private void textBox_SND_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_Nombre_Docente(e);
}
private void textBox_PAD_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_PApellido_Docente(e);
}
private void textBox_SAD_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Letras_SApellido_Docente(e);
}
private void textBox_TD_KeyPress(object sender, KeyPressEventArgs e)
{
    Metodos.Solo_Numeros_Telefono(e);
}
private void button_OKD_Click(object sender, EventArgs e)
{
    string TD=textBox_TD.Text;
    if (textBox_PND.Text == "" || textBox_PAD.Text == "" || textBox_SAD.Text == "" || textBox_DD.Text == "" ||
textBox_TD.Text == "" || textBox_CD.Text == "" || textBox_NUD.Text == "" || textBox_CUD.Text == "")

```

```

{
    MessageBox.Show("Todos los campos deben estar completamente llenos");
}
else if (TD.Length != 10)
{
    MessageBox.Show("Los datos del teléfono no son correctos");
textBox_TD.Clear();
textBox_TD.Focus();
}
else
{
    if (Comparar_Correo.IsMatch(textBox_CD.Text))
    {
        Datos_Docente Modificar_Docente = new Datos_Docente();
        Modificar_Docente.Primer_Nombre_Docente = textBox_PND.Text;
        Modificar_Docente.Segundo_Nombre_Docente = textBox_SND.Text;
        Modificar_Docente.Primer_Apellido_Docente = textBox_PAD.Text;
        Modificar_Docente.Segundo_Apellido_Docente = textBox_SAD.Text;
        Modificar_Docente.Nombre_Usuario = textBox_NUD.Text;
        Modificar_Docente.Contraseña_Usuario = textBox_CUD.Text;
        Modificar_Docente.Rol_Usuario = textBox_RUD.Text;
        Modificar_Docente.Direccion_Docente = textBox_DD.Text;
        Modificar_Docente.Telefono_Docente = textBox_TD.Text;
        Modificar_Docente.Correo_Docente = textBox_CD.Text;
        Modificar_Docente.Estado_Docente = "Activo";
        Modificar_Docente.Cedula = textBox_CID.Text; //Docente_Actual.CI_Docente;
        int resultado = Metodos.Modificar_Usuario(Modificar_Docente);
        if (resultado >= 0)
        {
            MessageBox.Show("Datos personales modificado con éxito");
        }
        else
        {
            MessageBox.Show("No se pudo modificar los datos personales. Póngase en contacto con el administrador
del sistema");
        }
    }
    else
    {
        MessageBox.Show("El correo del docente no es correcto, verifique e intente nuevamente");
    }
}
}
//-----
//-----
//-----UTILIZAR AULAS-----
int horainicio = 0, horafin = 0, No_Horas_Importdo, Cod_Horario_Importado;
string Dia_Importado, Materia_Importado, Periodo_Importado, Carrera_Importado, Aula_Importado,
Paralelo_Importado, Nivel_Importado,Detalle_Reserva_Importado;
string Hora_Inicio_Importado, Hora_Fin_Importado;
//-----MÉTODOS-----
void Importar_Horario()
{
    string hora1 = textBox_Horas_Asignada.Text;
SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand("SELECT H.COD_HORARIO,H.DIA,H.CEDULA,M.NOMBRE_MATERIA,
PA.NOMBRE_PERIODO_ACADEMICO,C.ALIAS,A.NOMBRE_AULA,H.PARALELO,H.NIVEL,H.HORA_INICIAL,
H.HORA_FINAL,H.N_HORAS FROM HORARIO H, MATERIAS M,AULAS A, CARRERA C,
PERIODO_ACADEMICO PA WHERE H.DIA='" + textBox_Dia_Asignada.Text + "' AND H.COD_MATERIA=
M.COD_MATERIA AND H.COD_AULA=A.COD_AULA AND H.COD_CARRERA=C.COD_CARRERA AND '" +
hora1 + "' between H.Hora_Inicial and H.Hora_Final and H.Cedula ='" + textBox_CI_Asignada.Text + "'",
Abrir_Conexion);
SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
        {
            Cod_Horario_Importado = lectura.GetInt32(0);
            Dia_Importado = lectura.GetString(1);
            Materia_Importado = lectura.GetString(3);
            Periodo_Importado = lectura.GetString(4);
            Carrera_Importado = lectura.GetString(5);
            Aula_Importado = lectura.GetString(6);
            Paralelo_Importado = lectura.GetString(7);

```

```

        Nivel_Importado = lectura.GetString(8);
        Hora_Inicio_Importado = lectura.GetString(9);
        Hora_Fin_Importado = lectura.GetString(10);
        No_Horas_Importdo = lectura.GetInt32(11);
        textBox_Materia_Asignada.Text = Materia_Importado;
        textBox_Periodo_Academico_Asignada.Text = Periodo_Importado;
        textBox_Carrera_Asignada.Text = Carrera_Importado;
        textBox_Aula_Asignada.Text = Aula_Importado;
        textBox_Paralelo_Asignada.Text = Paralelo_Importado;
        textBox_Nivel_Asignada.Text = Nivel_Importado;
        textBox3.Text = Convert.ToString(Hora_Inicio_Importado);
        textBox4.Text = Convert.ToString(Hora_Fin_Importado);
        textBox5.Text = Convert.ToString(No_Horas_Importdo);
    }
    Abrir_Conexion.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
int Cod_Reserva_Importado;
string Fecha_Reserva_Importado;
void Importar_Reserva()
{
    string hora1 = textBox_Horas_Asignada.Text;
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand("SELECT R.COD_RESERVA,R.DIA,R.FECHA,M.NOMBRE_MATERIA,
PA.NOMBRE_PERIODO_ACADEMICO,C.ALIAS,A.NOMBRE_AULA,R.PARALELO,R.NIVEL,R.HORA_INICIAL,
R.HORA_FINAL,R.N_HORAS,R.DETALLE FROM RESERVA R, MATERIAS M,AULAS A, CARRERA
C,PERIODO_ACADEMICO PA WHERE R.DIA='" + textBox_Dia_Asignada.Text + "' AND R.FECHA='" +
textBox_Fecha_Asignada.Text + "' AND R.COD_MATERIA=M.COD_MATERIA AND R.COD_AULA=A.COD_AULA
AND R.COD_CARRERA=C.COD_CARRERA AND '" + hora1 + "' between R.Hora_Inicial and R.Hora_Final and
R.Cedula='" + textBox_CI_Asignada.Text + "'", Abrir_Conexion);
SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
        {
            Cod_Reserva_Importado = lectura.GetInt32(0);
            Dia_Importado = lectura.GetString(1);
            Fecha_Reserva_Importado=lectura.GetString(2);
            Materia_Importado = lectura.GetString(3);
            Periodo_Importado = lectura.GetString(4);
            Carrera_Importado = lectura.GetString(5);
            Aula_Importado = lectura.GetString(6);
            Paralelo_Importado = lectura.GetString(7);
            Nivel_Importado = lectura.GetString(8);
            Hora_Inicio_Importado = lectura.GetString(9);
            Hora_Fin_Importado = lectura.GetString(10);
            No_Horas_Importdo = lectura.GetInt32(11);
            Detalle_Reserva_Importado = lectura.GetString(12);
            textBox_Materia_Reservada.Text = Materia_Importado;
            textBox_Periodo_Academico_Reservada.Text = Periodo_Importado;
            textBox_Carrera_Reservada.Text = Carrera_Importado;
            textBox_Aula_Reservada.Text = Aula_Importado;
            textBox_Paralelo_Reservada.Text = Paralelo_Importado;
            textBox_Nivel_Reservada.Text = Nivel_Importado;
            textBox_Detalle_Reservada.Text = Detalle_Reserva_Importado;
            textBox_Fecha_Reservada.Text = Fecha_Reserva_Importado;
            textBox6.Text = Convert.ToString(Hora_Inicio_Importado);
            textBox7.Text = Convert.ToString(Hora_Fin_Importado);
            textBox8.Text = Convert.ToString(No_Horas_Importdo);
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
void Importar_Dia_Sistema()
{
    string Fecha = DateTime.Now.ToLongDateString();

```

```

char[] Vector_Fecha = Fecha.ToArray();
for (int i = 0; i <= 9; i++)
    {
        if (Vector_Fecha[i] != Convert.ToChar(","))
            {
                textBox_Dia_Asignada.CharacterCasing = CharacterCasing.Upper;
                textBox_Dia.CharacterCasing = CharacterCasing.Upper;
                textBox_Dia_Asignada.Text += Vector_Fecha[i].ToString();
            }
        else
            {
                i = 10;
            }
    }
}
void Guardar_Reporte_Asignado()
{
    string Tipo = "HORARIO";
    string CI_Docente = textBox_CI_Asignada.Text;
    string Docente = textBox_Docente_Asignada.Text;
    string Rol_Usuario = textBox_Rol_Asignada.Text;
string Periodo_Academico = textBox_Periodo_Academico_Asignada.Text;
    string Carrera = textBox_Carrera_Asignada.Text;
    string Nivel = textBox_Nivel_Asignada.Text;
    string Materia = textBox_Materia_Asignada.Text;
    string Aula = textBox_Aula_Asignada.Text;
    string Fecha = textBox_Fecha_Asignada.Text;
    string Dia = textBox_Dia_Asignada.Text;
    string Numero_Horas = textBox5.Text;
    string Paralelo = textBox_Paralelo_Asignada.Text;
    string Hora_Inicial = textBox_Horas_Asignada.Text;
    string Hora_Final = textBox4.Text;
    string Detalle = "HORARIO ASIGNADO POR EL DISTRIBUTIVO DE AULAS";
    int resultado = Metodos.Crear_Reporte(Tipo, Cod_Horario_Importado, Fecha, Hora_Inicial, Hora_Final, Detalle);
    if (resultado > 0)
        {
            MessageBox.Show("Reporte guardado correctamente", "REPORTE GUARDADO", MessageBoxButtons.OK,
MessageBoxIcon.None);
        }
    else
        {
            MessageBox.Show("No se pudo guardar los datos del reporte", "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
}
void Guardar_Reporte_Reservada()
{
    string Tipo = "RESERVA";
    string CI_Docente = textBox_CI_Asignada.Text;
    string Docente = textBox_Docente_Asignada.Text;
string Rol_Usuario = textBox_Rol_Asignada.Text;
string Periodo_Academico = textBox_Periodo_Academico_Reservada.Text;
    string Carrera = textBox_Carrera_Reservada.Text;
    string Nivel = textBox_Nivel_Reservada.Text;
    string Materia = textBox_Materia_Reservada.Text;
    string Aula = textBox_Aula_Reservada.Text;
    string Fecha = textBox_Fecha_Reservada.Text;
    string Dia = textBox_Dia_Asignada.Text;
    string Numero_Horas = textBox8.Text;
    string Paralelo = textBox_Paralelo_Reservada.Text;
    string Hora_Inicial = textBox_Horas_Asignada.Text;
    string Hora_Final = textBox7.Text;
    string Detalle = textBox_Detalle_Reservada.Text;
    int resultado = Metodos.Crear_Reporte(Tipo, Cod_Reserva_Importado, Fecha, Hora_Inicial, Hora_Final, Detalle);
    if (resultado > 0)
        {
            MessageBox.Show("Reporte guardado correctamente", "REPORTE GUARDADO", MessageBoxButtons.OK,
MessageBoxIcon.None);
        }
    else
        {
            MessageBox.Show("No se pudo guardar los datos del reporte", "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
}
}

```

```

//-----CARACTERÍSTICAS DE LOS ELEMENT-----

//-----BOTONES-----
private void button_Aula_Asignada_Click(object sender, EventArgs e)
{
    if (textBox_Dia_Asignada.Text == "" || textBox_Horas_Asignada.Text == "" || textBox_CI_Asignada.Text == "" ||
        textBox_Docente_Asignada.Text == "" || textBox_Rol_Asignada.Text == "" || textBox_Carrera_Asignada.Text == "" ||
        textBox_Materia_Asignada.Text == "" || textBox_Nivel_Asignada.Text == "" || textBox_Aula_Asignada.Text == "" ||
        textBox_Periodo_Academico_Asignada.Text == "" || textBox_Paralelo_Asignada.Text == "")
    {
        MessageBox.Show("Usted no tiene clases a esta hora, puede utilizar un aula disponible si desea", "DATOS
ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        Envio_Caracter();
        tx = Caracter_Aula + "+" + "a" + "+" + textBox_CI_Asignada.Text + "+" + textBox5.Text + "+H+" + textBox4.Text;
        textBox1.Text = tx;
        for (int Contador = 1; Contador <= 10; )
        {
            serialPort1.Write(tx);
            Thread.Sleep(3000);
            if (rxc[2] == 'k')
            {
                Contador = 11;
            }
            else
            {
                Contador++;
            }
        }
        if (rxc[2] != 'k')
        {
            MessageBox.Show("No se tuvo respuesta del receptor, intente nuevamente");
            groupBox_Opcion_Horario.Enabled = true;
        }
    }
    else
    {
        textBox2.Text = rx;
        Guardar_Reporte_Asignado();
        int resultado = Metodos.Modificar_Estado_Aulas_Ocupado(textBox_Aula_Asignada.Text);
        if (resultado > 0)
        {
            MessageBox.Show("El aula pasará a un estado de ocupado", "ESTADO DEL AULA",
            MessageBoxButtons.OK, MessageBoxIcon.None);
            groupBox_Opcion_Horario.Enabled = false;
        }
    }
}

private void button_Aula_Reservada_Click(object sender, EventArgs e)
{
    if (textBox_Dia_Asignada.Text == "" || textBox_Horas_Asignada.Text == "" || textBox_CI_Asignada.Text == "" ||
        textBox_Docente_Asignada.Text == "" || textBox_Rol_Asignada.Text == "" || textBox_Carrera_Reservada.Text == "" ||
        textBox_Materia_Reservada.Text == "" || textBox_Nivel_Reservada.Text == "" || textBox_Aula_Reservada.Text ==
        "" || textBox_Periodo_Academico_Reservada.Text == "" || textBox_Paralelo_Reservada.Text == "")
    {
        MessageBox.Show("Usted no tiene clases a esta hora, puede utilizar una aula disponible si desea", "DATOS
ERRÓNEOS", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        Envio_Caracter();
        tx = Caracter_Aula + "+" + "a" + "+" + textBox_CI_Asignada.Text + "+" + textBox8.Text + "+R+" + textBox7.Text;
        textBox1.Text = tx;
        for (int Contador = 1; Contador <= 10; )
        {
            serialPort1.Write(tx);
            Thread.Sleep(3000);
            if (rxc[2] == 'k')
            {
                Contador = 11;
            }
            else
            {
                Contador++;
            }
        }
    }
}

```



```

Contador++;
    }
    }
    if (rxc[2] != 'k')
    {
        MessageBox.Show("No se tuvo respuesta del receptor, intente nuevamente");
        groupBox_Opcion_Reserva.Enabled = true;
    }
    else
    {
        textBox2.Text = rx;
        Guardar_Reporte_Reservada();
        int resultado = Metodos.Modificar_Estado_Aulas_Ocupado(textBox_Aula_Reservada.Text);
        if (resultado > 0)
        {
            MessageBox.Show("El aula pasará a un estado de opupado", "ESTADO DEL AULA",
            MessageBoxButtons.OK, MessageBoxIcon.None);
            groupBox_Opcion_Reserva.Enabled = false;
        }
    }
}
}
}
}
//-----
//-----
//-----UTILIZACION DEL AULA DISPONIBLE-----
public int hora_final, Capacidad_Aula_Select, Capacidad_Aula_Select_Nueva;
public string Caracter_Aula, Clave_Actual, rx, tx;
public string Aula_Select, Ubicacion_Aula_Select, Estado_Aula_Select, Aula_Disponible,
Ubicacion_Aula_Select_Nueva, Estado_Aula_Select_Nueva;
//-----MÉTODOS-----
void Limpiar_Aula_Disponible()
{
    comboBox_Numero_Aula.Items.Clear();
    comboBox_Periodo_Academico.Items.Clear();
    comboBox_Materia.Items.Clear();
    comboBox_Materia_Horario.Items.Clear();
    comboBox_Carrera_Horario.Items.Clear();
    comboBox_Carreras.Items.Clear();
    comboBox_Niveles.Items.Clear();
    comboBox_Numero_Horas.Items.Clear();
    comboBox_Paralelo.Items.Clear();
    label_Hora_Final.Text = "00:00:00";
    textBox_Ubicacion_Aula.Clear();
    textBox_Capacidad_Aula.Clear();
    textBox_Detalle.Clear();
    comboBox_Hora_Inicial.Items.Clear();
    Aulas_Disponibles();
    Materias();
    Carreras();
    comboBox_Periodo_Academico.Items.AddRange(new object[] { "MARZO " + Año_Actual + " - AGOSTO " +
Año_Actual + "", "SEPTIEMBRE " + Año_Actual + " - FEBRERO " + (Año_Actual + 1) });
    comboBox_Periodo_Academico.Items.AddRange(new object[] { "MARZO " + (Año_Actual - 1) + " - AGOSTO " +
(Año_Actual - 1) + "", "SEPTIEMBRE " + (Año_Actual - 1) + " - FEBRERO " + (Año_Actual) });
    comboBox_Niveles.Items.AddRange(new object[] { "1ro", "2do", "3ro", "4to", "5to", "6to", "7mo", "8vo", "9no", "10mo" });
    comboBox_Numero_Horas.Items.AddRange(new object[] { "1", "2", "3" });
    comboBox_Paralelo.Items.AddRange(new object[] { "A", "B" });
    comboBox_Hora_Inicial.Items.AddRange(new object[] { "07:00:00", "08:00:00", "09:00:00", "10:00:00", "11:00:00",
"12:00:00", "13:00:00", "14:00:00", "15:00:00", "16:00:00", "17:00:00", "18:00:00", "19:00:00", "20:00:00", "21:00:00" });
    groupBox_Parametros_Reserva.Enabled = false;
    groupBox_Tiempo.Enabled = false;
    groupBox_Detalle.Enabled = false;
    textBox_Detalle.Text = "MOTIVO DE LA RESERVA: ";
    groupBox_Controles_Eventual.Enabled = false;
}
void Envio_Caracter()
{
    if (comboBox_Numero_Aula.Text == "AULA_201" || textBox_Aula_Asignada.Text == "AULA_201" ||
textBox_Aula_Reservada.Text == "AULA_201")
    {
        AULA_201.BackColor = Color.Blue;
        Caracter_Aula = "A";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_202" || textBox_Aula_Asignada.Text == "AULA_202" ||
textBox_Aula_Reservada.Text == "AULA_202")
    {

```

```

        AULA_202.BackColor = Color.Blue;
        Caracter_Aula = "B";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_203" || textBox_Aula_Asignada.Text == "AULA_203" ||
    textBox_Aula_Reservada.Text == "AULA_203")
    {
        AULA_203.BackColor = Color.Blue;
        Caracter_Aula = "C";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_204" || textBox_Aula_Asignada.Text == "AULA_204" ||
    textBox_Aula_Reservada.Text == "AULA_204")
    {
        AULA_204.BackColor = Color.Blue;
        Caracter_Aula = "D";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_205" || textBox_Aula_Asignada.Text == "AULA_205" ||
    textBox_Aula_Reservada.Text == "AULA_205")
    {
        AULA_205.BackColor = Color.Blue;
        Caracter_Aula = "E";
    }
}
    else if (comboBox_Numero_Aula.Text == "AULA_206" || textBox_Aula_Asignada.Text == "AULA_206" ||
    textBox_Aula_Reservada.Text == "AULA_206")
    {
        AULA_206.BackColor = Color.Blue;
        Caracter_Aula = "F";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_207" || textBox_Aula_Asignada.Text == "AULA_207" ||
    textBox_Aula_Reservada.Text == "AULA_207")
    {
        AULA_207.BackColor = Color.Blue;
        Caracter_Aula = "G";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_209" || textBox_Aula_Asignada.Text == "AULA_209" ||
    textBox_Aula_Reservada.Text == "AULA_209")
    {
        AULA_209.BackColor = Color.Blue;
        Caracter_Aula = "H";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_302" || textBox_Aula_Asignada.Text == "AULA_302" ||
    textBox_Aula_Reservada.Text == "AULA_302")
    {
        AULA_302.BackColor = Color.Blue;
        Caracter_Aula = "I";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_303" || textBox_Aula_Asignada.Text == "AULA_303" ||
    textBox_Aula_Reservada.Text == "AULA_303")
    {
        AULA_303.BackColor = Color.Blue;
        Caracter_Aula = "J";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_304" || textBox_Aula_Asignada.Text == "AULA_304" ||
    textBox_Aula_Reservada.Text == "AULA_304")
    {
        AULA_304.BackColor = Color.Blue;
        Caracter_Aula = "K";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_305" || textBox_Aula_Asignada.Text == "AULA_305" ||
    textBox_Aula_Reservada.Text == "AULA_305")
    {
        AULA_305.BackColor = Color.Blue;
        Caracter_Aula = "L";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_306" || textBox_Aula_Asignada.Text == "AULA_306" ||
    textBox_Aula_Reservada.Text == "AULA_306")
    {
        AULA_306.BackColor = Color.Blue;
        Caracter_Aula = "M";
    }
    else if (comboBox_Numero_Aula.Text == "AULA_307" || textBox_Aula_Asignada.Text == "AULA_307" ||
    textBox_Aula_Reservada.Text == "AULA_307")
    {
        AULA_307.BackColor = Color.Blue;
        Caracter_Aula = "N";
    }
}

```

```

        else if (comboBox_Numero_Aula.Text == "AULA_308" || textBox_Aula_Asignada.Text == "AULA_308" ||
textBox_Aula_Reservada.Text == "AULA_308")
        {
            AULA_308.BackColor = Color.Blue;
            Caracter_Aula = "O";
        }
        else if (comboBox_Numero_Aula.Text == "AULA_309" || textBox_Aula_Asignada.Text == "AULA_309" ||
textBox_Aula_Reservada.Text == "AULA_309")
        {
            AULA_309.BackColor = Color.Blue;
            Caracter_Aula = "P";
        }
        else if (comboBox_Numero_Aula.Text == "AULA_310" || textBox_Aula_Asignada.Text == "AULA_310" ||
textBox_Aula_Reservada.Text == "AULA_310")
        {
            AULA_310.BackColor = Color.Blue;
            Caracter_Aula = "Q";
        }
        else if (comboBox_Numero_Aula.Text == "AULA_311" || textBox_Aula_Asignada.Text == "AULA_311" ||
textBox_Aula_Reservada.Text == "AULA_311")
        {
            AULA_311.BackColor = Color.Blue;
            Caracter_Aula = "R";
        }
    }
    void hora_Inicial_Final()
    {
        if (comboBox_Numero_Horas.Text == "")
        {
            MessageBox.Show("Debe seleccionar un determinado número de horas");
        }
        else
        {
            hora_final = Convert.ToInt32(comboBox_Numero_Horas.Text);
            if (comboBox_Hora_Inicial.Text == "07:00:00")
            {
                hora_final = hora_final + 7;
            }
            else if (comboBox_Hora_Inicial.Text == "08:00:00")
            {
                hora_final = hora_final + 8;
            }
            else if (comboBox_Hora_Inicial.Text == "09:00:00")
            {
                hora_final = hora_final + 9;
            }
            else if (comboBox_Hora_Inicial.Text == "10:00:00")
            {
                hora_final = hora_final + 10;
            }
            else if (comboBox_Hora_Inicial.Text == "11:00:00")
            {
                hora_final = hora_final + 11;
            }
            else if (comboBox_Hora_Inicial.Text == "12:00:00")
            {
                hora_final = hora_final + 12;
            }
            else if (comboBox_Hora_Inicial.Text == "13:00:00")
            {
                hora_final = hora_final + 13;
            }
            else if (comboBox_Hora_Inicial.Text == "14:00:00")
            {
                hora_final = hora_final + 14;
            }
            else if (comboBox_Hora_Inicial.Text == "15:00:00")
            {
                hora_final = hora_final + 15;
            }
            else if (comboBox_Hora_Inicial.Text == "16:00:00")
            {
                hora_final = hora_final + 16;
            }
            else if (comboBox_Hora_Inicial.Text == "17:00:00")

```

```

{
    hora_final = hora_final + 17;
}
else if (comboBox_Hora_Inicial.Text == "18:00:00")
{
    hora_final = hora_final + 18;
}
else if (comboBox_Hora_Inicial.Text == "19:00:00")
{
    hora_final = hora_final + 19;
}
else if (comboBox_Hora_Inicial.Text == "20:00:00")
{
    hora_final = hora_final + 20;
}
else if (comboBox_Hora_Inicial.Text == "21:00:00")
{
    hora_final = hora_final + 21;
}
label_Hora_Final.Text = Convert.ToString(hora_final) + ":00:00";
}
}
void Aulas()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand("select * from Aulas", Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
        {
            Aula_Select = lectura.GetString(1);
            comboBox_Seleccion_Nueva_Aula.Items.Add(Aula_Select);
            comboBox_Aula_Horario.Items.Add(Aula_Select);
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
void Cap_Ubic_Aula_Disponible()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("select Capacidad,Ubicacion,Estado from Aulas where
Nombre_Aula like '{0}%' ",comboBox_Numero_Aula.Text), Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
        {
            Capacidad_Aula_Select = lectura.GetInt32(0);
            Ubicacion_Aula_Select = lectura.GetString(1);
            Estado_Aula_Select=lectura.GetString(2);
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
void Cap_Ubic_Aula_Nueva()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("select Capacidad,Ubicacion,Estado from Aulas where
Nombre_Aula like '{0}%' ", comboBox_Seleccion_Nueva_Aula.Text), Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
    }
}

```

```

    {
        Capacidad_Aula_Selec_Nueva = lectura.GetInt32(0);
        Ubicacion_Aula_Selec_Nueva = lectura.GetString(1);
        Estado_Aula_Selec_Nueva = lectura.GetString(2);
    }
    Abrir_Conexion.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
string Materia,Estado_Materia;
int Creditos;
void Materias()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand("select * from Materias", Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
        {
            Materia = lectura.GetString(1);
            comboBox_Materia.Items.Add(Materia);
            comboBox_Seleccion_Nueva_Materia.Items.Add(Materia);
            comboBox_Materia_Horario.Items.Add(Materia);
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
void Cred_Esta_Materia_Nueva()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("select Creditos,Estado from Materias where
Nombre_Materia like '%{0}%' "), comboBox_Seleccion_Nueva_Materia.Text), Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
        {
            Creditos = lectura.GetInt32(0);
            Estado_Materia = lectura.GetString(1);
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
public string Nombre_Carrera, Alias_Carrera_seleccionada, Estado_Carrera_seleccionada;
public string Alias_Carrera;
void Carreras()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand("select * from Carrera", Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        lectura = cmd.ExecuteReader();
        while (lectura.Read())
        {
            Nombre_Carrera = lectura.GetString(1);
            Alias_Carrera = lectura.GetString(2);
            comboBox_Carreras.Items.Add(Alias_Carrera);
            comboBox_Seleccion_Nueva_Carrera.Items.Add(Nombre_Carrera);
            comboBox_Carrera_Horario.Items.Add(Alias_Carrera);
        }
    }
}

```

```

        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

void Nomb_Alias_Carrera_Nueva()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand(string.Format("select Nombre_Carrera, Alias, Estado from Carrera where
Nombre_Carrera like '%{0}%' "), comboBox_Seleccion_Nueva_Carrera.Text), Abrir_Conexion);
SqlDataReader lectura;
    try
    {
        {
            lectura = cmd.ExecuteReader();
            while (lectura.Read())
            {
                Alias_Carrera_seleccionada = lectura.GetString(1);
                Estado_Carrera_seleccionada = lectura.GetString(2);
            }
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

void Docentes()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand("select * from Usuarios", Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        {
            lectura = cmd.ExecuteReader();
            while (lectura.Read())
            {
                string N_Docente = lectura.GetString(1);
                string P_Apellido = lectura.GetString(3);
                string S_Apellido = lectura.GetString(4);
                comboBox_Docente_Horario.Items.Add(P_Apellido + " " + S_Apellido + " " + N_Docente);
            }
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

void Aulas_Disponibles()
{
    SqlConnection Abrir_Conexion = Conexion.Obtener_Conexion();
    SqlCommand cmd = new SqlCommand("select * from Aulas", Abrir_Conexion);
    SqlDataReader lectura;
    try
    {
        {
            lectura = cmd.ExecuteReader();
            while (lectura.Read())
            {
                Aula_Disponible = lectura.GetString(1);
                comboBox_Numero_Aula.Items.Add(Aula_Disponible);
            }
        }
        Abrir_Conexion.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

void Guardar_Reserva_Disponible()
{
    string CI_Docente = textBox_CI_Eventual.Text;
    string Docente = textBox_Docente_Eventual.Text;
    string Rol_Usuario = textBox_Rol_Eventual.Text;
}

```

```

string Periodo_Academico = textBox_Periodo_Academico.Text;
string Carrera = comboBox_Carreras.Text;
string Nivel = comboBox_Niveles.Text;
string Materia = comboBox_Materia.Text;
string Aula = comboBox_Numero_Aula.Text;
string Fecha = dateTimePicker1.Text;
string Dia= textBox_Dia.Text;
string Numero_Horas = comboBox_Numero_Horas.Text;
string Paralelo = comboBox_Paralelo.Text;
string Hora_Inicial = comboBox_Hora_Inicial.Text;
string Hora_Final = label_Hora_Final.Text;
string Detalle = textBox_Detalle.Text;
int resultado_1 = Metodos.Comprobar_Horario(Dia, Aula, Hora_Inicial, Hora_Final, Periodo_Academico);
if (resultado_1 > 0)
{
    MessageBox.Show("Ya hay un horario con estos datos", "ERROR", MessageBoxButtons.OK,
    MessageBoxIcon.None);
}
else
{
    int resultado_2 = Metodos.Comprobar_Reserva(Dia, Fecha, Aula, Hora_Inicial, Hora_Final, Periodo_Academico);
    if (resultado_2 > 0)
    {
        MessageBox.Show("Ya hay una reserva con estos datos", "ERROR", MessageBoxButtons.OK,
        MessageBoxIcon.None);
    }
    else
    {
        int resultado_3 = Metodos.Crear_Reserva(Dia, Fecha, Cl_Docente, Materia, Periodo_Academico, Carrera,
        Aula, Paralelo, Nivel, Hora_Inicial, Hora_Final, Numero_Horas, Detalle);
        if (resultado_3 > 0)
        {
            MessageBox.Show("Reporte guardado correctamente", "RESERVA GUARDADA",
            MessageBoxButtons.OK, MessageBoxIcon.None);
        }
        else
        {
            MessageBox.Show("No se pudo guardar los datos de la reserva", "ERROR", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        }
    }
}
}
}
//-----CARACTERÍSTICAS DE LOS ELEMENTOS-----
private void textBox_Detalle_TextChanged(object sender, EventArgs e)
{
    textBox_Detalle.CharacterCasing = CharacterCasing.Upper;
}
private void comboBox_Numero_Aula_SelectedIndexChanged(object sender, EventArgs e)
{
    Cap_Ubic_Aula_Disponible();
    textBox_Capacidad_Aula.Text = Convert.ToString(Capacidad_Aula_Selec);
    textBox_Ubicacion_Aula.Text = Ubicacion_Aula_Selec;
    groupBox_Parametros_Reserva.Enabled = true;
    groupBox_Detalle.Enabled = true;
}
private void comboBox_Numero_Horas_SelectedIndexChanged(object sender, EventArgs e)
{
    hora_Inicial_Final();
}
//-----BOTONES-----
private void btn_OK_Click(object sender, EventArgs e)
{
    if (comboBox_Numero_Aula.Text != "" && textBox_Periodo_Academico.Text != "" && comboBox_Carreras.Text != ""
    && comboBox_Niveles.Text != "" && comboBox_Materia.Text != "" && comboBox_Numero_Horas.Text != "" &&
    comboBox_Paralelo.Text != "" && textBox_Detalle.Text != "" && comboBox_Hora_Inicial.Text != "" &&
    textBox_Dia.Text != "")
    {
        if (textBox_Detalle.Text == "MOTIVO DE LA RESERVA: ")
        {
            MessageBox.Show("El campo detalle no ha sido ingresado correctamente, asegúrese de informar el motivo
            exacto de la reserva", "ERROR", MessageBoxButtons.OK);
            textBox_Detalle.Clear();
            textBox_Detalle.Focus();
        }
    }
}

```

```

        else
        {
            if (MessageBox.Show("Para confirmar su reserva haga click en Aceptar o Cancelar para intentarlo nuevamente", "SUGERENCIA", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)
            {
                Guardar_Reserva_Disponible();
                Limpiar_Aula_Disponible();
            }
            else
            {
                MessageBox.Show("Usted canceló la reserva. Intente nuevamente");
                Limpiar_Aula_Disponible();
            }
        }
    }
    else
    {
        MessageBox.Show("Todos los campos deben estar completamente llenos", "INFORMACIÓN", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void btn_Limpiar_Eventual_Click(object sender, EventArgs e)
{
    Limpiar_Aula_Disponible();
}

private void comboBox_Aula_Horario_SelectedIndexChanged(object sender, EventArgs e)
{
    groupBox_Horas_Asignadas.Enabled = true;
    comboBox_Hora_Inicio_Asignada.Enabled = true;
    comboBox_Hora_Final_Asignada.Enabled = false;
}

private void comboBox_Paralelo_SelectedIndexChanged(object sender, EventArgs e)
{
    groupBox_Tiempo.Enabled = true;
}

private void textBox_Detalle_MouseClick(object sender, MouseEventArgs e)
{
    textBox_Detalle.Text = " ";
}

private void comboBox_Hora_Inicial_SelectedIndexChanged(object sender, EventArgs e)
{
    hora_Inicial_Final();
    groupBox_Controles_Eventual.Enabled = true;
}

private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
    string dia = dateTimePicker1.Value.DayOfWeek.ToString();
    if (dia == "Monday")
    {
        textBox_Dia.Text = "LUNES";
    }
    else if (dia == "Tuesday")
    {
        textBox_Dia.Text = "MARTES";
    }
    else if (dia == "Wednesday")
    {
        textBox_Dia.Text = "MIÉRCOLES";
    }
    else if (dia == "Thursday")
    {
        textBox_Dia.Text = "JUEVES";
    }
    else if (dia == "Friday")
    {
        textBox_Dia.Text = "VIERNES";
    }
    else if (dia == "Saturday")
    {
        textBox_Dia.Text = "SÁBADO";
    }
    else if (dia == "Sunday")
    {
        textBox_Dia.Text = "DOMINGO";
    }
}

```



```
}  
  }  
}  
}
```

//----- REPORTES-----

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
using CrystalDecisions.CrystalReports.Engine;  
using CrystalDecisions.Shared;  
  
namespace Interfaz_Grafica_Reporte  
{  
    public partial class Form_Reportes: Form  
    {  
        public Form_Reportes()  
        {  
            InitializeComponent();  
        }  
        ReportDocument Reporte_rpt = new ReportDocument();  
        private void Form_Reportes_Load(object sender, EventArgs e)  
        {  
            Reporte_rpt.Load(@"C:\Users\ROSITA\Desktop\Final\Antonio Bolívar Guerrero Aseicha\Interfaz Grafica\Interfaz  
Grafica Reporte\Interfaz Grafica Reporte\Reportes\Reporte.rpt");  
            crystalReportViewer1.ReportSource = Reporte_rpt;  
            crystalReportViewer1.Refresh();  
        }  
        private void button_Atras_Reporte_Click(object sender, EventArgs e)  
        {  
            this.Close();  
        }  
    }  
}
```

## **ANEXO E**

### **7.9 MANUALES DEL SISTEMA DE CONTROL DE ACCESO (SOFTWARE) PARA LAS AULAS DE LA FACULTAD**

## 7.9.1 MANUAL DEL ADMINISTRADOR

### 7.9.1.1 INTRODUCCIÓN

El sistema de control de acceso para la Facultad de Ingeniería en Ciencias Aplicadas es un software diseñado para mejorar el proceso de utilizar y controlar el ingreso a las diferentes aulas por parte de los docentes y funcionarios que laboran en dicha facultad.

### 7.9.1.2 MODOS DE INGRESO

El sistema permite tres modos de ingreso, dependiendo de la función que cada uno de los docentes o funcionarios realizan en las aulas, los cuales son:

- ✓ Administrador
- ✓ Usuario y
- ✓ Eventual

*Administrador:* Es el encargado de gestionar la aplicación y la base de datos según sea necesario, el cual permitirá realizar acciones como eliminar, crear, modificar ya sea los datos de un docente como los datos de las aulas y materias, permitiéndole cambiar diferentes parámetros.

*Usuario:* Es aquel que tendrá ciertos permisos en la aplicación, es decir, solamente se le permite modificar y verificar los datos propios, así como ver las aulas que se encuentran disponibles en ese momento y utilizar un aula disponible en caso de necesitarla fuera de su horario normal.

*Eventual:* Es aquel que puede ver las diferentes aulas que se encuentran disponibles u ocupadas previamente en un periodo determinado, y realizar las respectivas reservas en caso de requerir un aula.

En este manual se especifica el modo de ingreso al sistema denominado “*Administrador*”, permitiendo dar a conocer los pasos a seguir en cada uno de los parámetros de forma general para la creación, utilización o modificación de las diferentes aulas de la facultad, así como reservar un aula disponible, generar un reporte de la utilización de las aulas, etc.

### **7.9.1.3 FORMULARIOS DEL SISTEMA**

De igual forma el sistema presenta tres formularios los cuales se dividen, dependiendo de la función que cumple, en:

- ✓ Formulario de autenticación.
- ✓ Formulario de procesos.
- ✓ Formulario de reportes.

#### **7.9.1.3.1 Formulario de autenticación**

Permite verificar si el docente o funcionario que utilice el sistema se encuentra registrado como administrador, además de recopilar información como el día, fecha y hora actual del sistema, esta interfaz se presenta en la figura siguiente:

Figura 148. Formulario de autenticación del administrador

The image shows a web-based authentication form titled "AUTENTICACIÓN". The form is for the "UNIVERSIDAD TÉCNICA DEL NORTE FICA - CIERCOM SISTEMA DE CONTROL DE ACCESO". It is divided into three main sections:

- Modo de ingreso:** Contains three radio buttons: "Administrador" (selected), "Usuario", and "Eventual".
- Datos del Sistema:** Displays system information including "Fecha" (30/05/2015), "Dia" (SÁBADO), and "Hora Actual" (16:34:09).
- Autenticación:** Contains three input fields: "Ingrese Usuario", "Ingrese Contraseña", and "Confirmar Contraseña". Below these fields are two buttons: "LIMPIAR" and "INGRESAR".

La interfaz de autenticación del administrador presenta tres secciones que son:

- A. Modo de Ingreso: Permite seleccionar el rol de usuario con el cual se autenticará el docente para ingresar al sistema.
- B. Datos del Sistema: Visualiza los datos provenientes de la computadora con la que se encuentre trabajando el sistema.
- C. Autenticación: Permite ingresar la información al usuario con la que se registrará en el sistema. Esta sección depende de la sección "Modo de Ingreso", debido a que contiene campos que se habilitan o deshabilitan según el rol que se escoja.

Figura 149. Identificación de las secciones en el formulario de autenticación del administrador.

The image shows a web form titled 'AUTENTICACIÓN' for 'UNIVERSIDAD TÉCNICA DEL NORTE FICA - CIERCOM SISTEMA DE CONTROL DE ACCESO'. The form is divided into three sections marked with red boxes and arrows:

- Sección A:** 'Modo de ingreso' with radio buttons for 'Administrador' (selected), 'Usuario', and 'Eventual'.
- Sección B:** 'Datos del Sistema' with a date field 'Fecha' (30/05/2015), a day field 'Dia' (SÁBADO), and a time field 'Hora Actual' (16:34:09).
- Sección C:** 'Autenticación' with input fields for 'Ingreso Usuario', 'Ingreso Contraseña', and 'Confirmar Contraseña', and buttons for 'LIMPIAR' and 'INGRESAR'.

*Nota:*

*La Sección C tiene su propio control que realiza una tarea determinada, las mismas que se explican en el proceso de utilización más adelante.*

#### **7.9.1.3.2 Formulario de procesos**

En este formulario se presentan ciertos campos a seleccionar dependiendo de las prioridades del administrador, tales como la creación, búsqueda, modificación o eliminación de un docente en el sistema, la creación o modificación de una aula, materia o carrera, así como también le permite utilizar o reservar un aula al docente y generar un reporte de todas las aulas utilizadas en la facultad. Estas prioridades se encuentran especificadas más detalladamente en el trabajo

de titulación de este proyecto. A continuación se presenta la interfaz de este formulario en la siguiente figura.

Figura 150. Formulario de procesos del administrador.

Además, esta interfaz presenta dos menús, uno principal y otro selectivo, los mismos que cuentan con los siguientes campos habilitados:

- ✓ **Menú Principal:** Permite abrir o cerrar otra interfaz dependiendo de la función, estas funciones pueden ser:
  - *Salir:* Cierra la interfaz actual del administrador y abre la interfaz de autenticación.

- *Reportes*: Abre una nueva interfaz para mostrar el reporte de las aulas utilizadas que genera el sistema. Esta interfaz se denomina "Reportes".
- ✓ **Menú Selectivo**: Permite moverse dentro de la misma interfaz, este campo tiene submenús, los cuales realizan diferentes funciones tales como:
  - *Nuevo Docente*: Permite crear un nuevo docente y guardar los datos en el sistema.
  - *Buscar Modificar y Eliminar Docente*: Este campo busca a un docente determinado para realizar las funciones de modificar datos o eliminarlo del sistema para no permitir el ingreso al mismo.
  - *Crear, Modificar Materia, Aula o Carrera*: Permite crear una materia o carrera en caso de ser necesario, o modifica los parámetros de los mismos si así lo requiere el administrador.
  - *Crear Horario*: Permite crear un periodo académico nuevo con su respectivo horario de clases de cada una de las carreras de la facultad.
  - *Modificar Datos Personales*: Permite realizar modificaciones en los datos personales del docente.
  - *Utilizar Aula Asignada o Reservada*: Este campo permite utilizar una aula que se encuentre ya sea reservada con anticipación o se encuentre en el horario de clases del docente.
  - *Reservar Aula Disponible*: Permite reservar una aula de clases que se encuentre disponible por un periodo determinado.
  - *Ver Aulas*: Visualiza las aulas que se encuentran utilizadas y disponibles en la facultad.



Figura 151. Identificación de los menús en el formulario de procesos del administrador.



#### 7.9.1.3.2.1 Secciones del menú selectivo

En la interfaz de proceso, cada uno de los elementos del “menús selectivos”, presentan secciones diferentes que se acoplan a cada función que cumplen, las mismas que se detallan a continuación:

##### 7.9.1.3.2.1.1 Nuevo Docente

- A. Datos Personales: Permite ingresar los datos personales de un nuevo docente o funcionario.
- B. Información: Genera una ayuda para el administrador, informando lo que se debe hacer en cada uno de los campos que se debe ingresar o seleccionar.

- C. Identificación de Registro: Visualiza los datos del docente o funcionario con los cuales se deben registrar inicialmente. Estos datos pueden ser modificados solamente por el usuario correspondiente.
- D. Rol de Usuario: Permite al administrador dar una función al docente que va a registrar. Este parámetro es importante ya que permite dar las prioridades al docente o funcionario en el sistema.
- E. Controles: Permite guardar o borrar la información que se encuentre ingresando en este proceso.

Figura 152. Identificación de las secciones en el formulario de procesos del administrador del submenú nuevo docente.

The screenshot shows a web application window titled 'ADMINISTRADOR' with a menu bar containing 'SALIR' and 'REPORTES'. Below the menu is a navigation bar with options: 'Nuevo Docente', 'Buscar Modificar y Eliminar Docentes', 'Crear, Modificar Materia Aula o Carrera', 'Crear Horario', 'Modificar Datos Personales', and 'Utilizar Au...'. The main content area is titled 'DATOS DEL DOCENTE' and is divided into five sections marked with red boxes and arrows:

- Sección A:** 'Datos Personales' containing fields for C.I.\_Docente (with a 'VALIDAR' button), Primer\_Nombre\_Docente, Segundo\_Nombre\_Docente, Primer\_Apellido\_Docente, Segundo\_Apellido\_Docente, Dirección, Teléfono\_Celular\_Principal, and Correo (with a '@utn.edu.ec' suffix).
- Sección B:** 'Información' containing the text: 'TODOS LOS CAMPOS DEBEN ESTAR CORRECTAMENTE INGRESADOS Y LLENOS'.
- Sección C:** 'Identificación de Registro' containing fields for Nombre\_Usuario and Contraseña\_Usuario.
- Sección D:** 'Rol\_Usuario' containing radio buttons for 'Administrador', 'Usuario', 'Eventual', and 'Conserje', along with the logo of 'UNIVERSIDAD TÉCNICA DEL TIDESTE'.
- Sección E:** 'Controles' containing 'GUARDAR' and 'LIMPIAR' buttons.

At the bottom of the form is a banner image of a building with the text 'Facultad de Ingeniería en Ciencias Aplicadas'.

#### 7.9.1.3.2.1.2 Buscar, Modificar y Eliminar Docente.

- A. Opciones: Permite escoger el tipo de trabajo que se va a realizar en este proceso.

- B. Buscar Docente por: Realiza una búsqueda de un docente determinado dependiendo del tipo de dato que se vaya a ingresar.
- C. Resultados de la Búsqueda: Visualiza la búsqueda del docente o funcionario que se esté realizando en la “sección B” y permite seleccionarlo para cargar los datos en la “sección D”.
- D. Datos Personales: Carga los datos del docente o funcionario del que se realizó la búsqueda.
- E. Identificación de Registro: Visualiza los datos del docente o funcionario con los cuales se deben registrar inicialmente. Estos datos pueden ser modificados solamente por el usuario correspondiente.

Figura 153. Identificación de las secciones en el formulario de procesos del administrador del submenú buscar, modificar y eliminar docente.

The screenshot shows a web application window titled "ADMINISTRADOR". At the top, there are navigation links: "SALIR" and "REPORTES". Below that is a menu bar with options: "Nuevo Docente", "Buscar Modificar y Eliminar Docentes", "Crear, Modificar Materia Aula o Carrera", "Crear Horario", "Modificar Datos Personales", and "Utilizar Au...".

The main content area is divided into several sections, each highlighted with a red box and labeled with an arrow:

- Sección A:** Located in the top right, it contains an "Opciones" section with radio buttons for "Buscar", "Modificar", and "Eliminar", and a 3D graphic of a person thinking.
- Sección B:** Located in the top left, it contains a "Buscar Docente por:" section with radio buttons for "C.I.\_Docente", "Nombre\_Docente", and "Apellido\_Docente", and a "LIMPIAR" button.
- Sección C:** Located at the bottom, it contains a "Resultados de la Búsqueda" section with a large grey rectangular area representing a table and an "ACEPTAR" button.
- Sección D:** Located in the middle left, it contains a "Datos Personales" section with input fields for "C.I.\_Docente", "Primer\_Nombre\_Docente", "Segundo\_Nombre\_Docente", "Primer\_Apellido\_Docente", "Segundo\_Apellido\_Docente", "Dirección", "Teléfono\_Celular\_Principal", and "Correo".
- Sección E:** Located in the middle right, it contains an "Identificación de Registro" section with input fields for "Nombre\_Usuario", "Contraseña\_Usuario", "Rol\_Usuario", and "Estado", and "Ok" and "CANCELAR" buttons.

In the center of the form, there is a logo for "UNIVERSIDAD TÉCNICA DEL NORTE" with the text "ABARRA - ECUADOR" below it.

*Nota:*

*En lo referente a las secciones B, C y D estas tienen sus propios controles que realizan una tarea determinada, las mismas que se explican en el proceso de utilización más adelante.*

*7.9.1.3.2.1.3 Crear, Modificar Materia, Aula o Carrera.*

- A. Datos del Aula: Permite ingresar datos de un aula nueva o modificar los datos de ésta.
- B. Información: Genera una ayuda para el administrador, informando lo que se debe hacer en cada uno de los campos que se debe ingresar o seleccionar.
- C. Datos de la Materia: Permite ingresar datos de una materia nueva o modificar los datos de ésta.
- D. Datos de la Carrera: Permite ingresar datos de una carrera nueva o modificar los datos de ésta.

Figura 154. Identificación de las secciones en el formulario de procesos del administrador del submenú crear, modificar materia, aula o carrera.

*Nota:*

*Las secciones A, C y D tienen sus propios controles que realizan una tarea determinada las mismas que se explican en el proceso de utilización más adelante.*

#### 7.9.1.3.2.1.4 Crear Horario.

- A. Opciones: Permite seleccionar una de las tres opciones (Nuevo, Modificar o Duplicar un periodo académico) según crea el administrador.
- B. Acción: Según la “Opción” seleccionada este campo realiza el proceso seleccionado habilitando o deshabilitando los controles.

- C. Datos del Horario: Permite seleccionar los datos de un aula para asignarle a un docente determinado en un horario correspondiente al distributivo de aulas proporcionado por el departamento de informática de la universidad.
- D. Horas: Permite asignar una hora inicial y final al docente en ese horario seleccionado.
- E. Horarios del Periodo Académico: Permite seleccionar un horario de un determinado periodo académico para ser modificado con anterioridad.

Figura 155. Identificación de las secciones en el formulario de procesos del administrador del submenú crear horario.

*Nota:*

*Las secciones B, C y E tienen sus propios controles que realizan una tarea determinada las mismas que se explican en el proceso de utilización más adelante.*



7.9.1.3.2.1.5 *Modificar Datos Personales.*

- A. Datos Personales: Permite modificar los datos personales de un docente o funcionario que se haya registrado previamente en el sistema.
- B. Identificación de Registro: Visualiza los datos del docente o funcionario con los cuales se registró inicialmente.
- C. Control: Permite guardar la información que se haya modificado en este proceso.

Figura 156. Identificación de las secciones en el formulario de procesos del administrador del submenú modificar datos personales.

ADMINISTRADOR

SALIR REPORTE

Nuevo Docente | Buscar, Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Au...

### MODIFICAR DATOS PERSONALES Y DE REGISTRO DEL SISTEMA

**Sección A** → **Datos Personales**

C.I._Docente	1003788401
Primer_Nombre_Docente	ANTONIO
Segundo_Nombre_Docente	BOLÍVAR
Primer_Apellido_Docente	GUERRERO
Segundo_Apellido_Docente	ASEICHA
Dirección	MIGUEL SANCHEZ Y TOBÍAS MENA
Teléfono_Celular_Principal	0981162999
Correo	abguerreroa@utn.edu.ec

**Identificación de Registro** → **Sección B**

Nombre_Usuario	t
Contraseña_Usuario	5
Rol_Usuario	Administrador

**Control** → **Sección C**

OK

CIERCOM INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

UNIVERSIDAD TÉCNICA DEL NORTE UIN AUTÓNOMA DE IBARRA - ECUADOR

Facultad de Ingeniería en Ciencias Aplicadas

7.9.1.3.2.1.6 *Utilizar Aula Asignada o Reservada.*

- A. Datos del Sistema: Visualiza los datos provenientes de la computadora con la que se encuentre trabajando el sistema.

B. Datos del Docente: Visualiza los datos del docente con los cuales se registró en el sistema.

C. Horario/Reserva: Permite visualizar la información correspondiente al horario o a la reserva del aula a utilizar por parte del docente en ese momento.

Esta sección depende del horario de clases del docente o de una reserva realizada con anticipación para poder visualizar los datos del aula a utilizar, en caso de no existir estos parámetros se visualiza un mensaje informando al docente que no tiene clases a esa hora.

a. Utilizar Aula: Permite comunicarse con el hardware con el botón "OK" para abrir la puerta del aula correspondiente.



Figura 157. Identificación de las secciones en el formulario de procesos del administrador del submenú utilizar aula asignada o reservada.

ADMINISTRADOR

SALIR REPORTES

Crear Horario Modificar Datos Personales Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

**AULA NO ASIGNADA NI RESERVADA**

**Sección A** Datos del Sistema  
Fecha 01/06/2015 Día LUNES Hora 10:11:44

**Sección B** Datos del Docente  
Cédula 1003788401 Docente GUERRERO ASEICHA ANTONIO Rol Administrador

**Sección C** USTED NO TIENE CLASES A ESTA HORA...!!!

*Nota:*

*En caso de no tener clases el docente en ese momento, no se cargan los datos y se visualiza un mensaje informándole que no tiene clases a esa hora.*

*La sección C tiene su propio control que realiza una tarea determinada, la misma que se explica en el proceso de utilización más adelante.*

#### *7.9.1.3.2.1.7 Reservar Aula Disponible.*

- A. Datos del Aula: Permite seleccionar un aula disponible a reservar por el docente.
- B. Detalle: Permite ingresar información del motivo por el cual el aula seleccionada en la sección A se va a utilizar.

- C. Parámetros de Reserva: Permite seleccionar parámetros de la materia que se va impartir en una fecha determinada.
- D. Datos Docente: Visualiza los datos del docente con los cuales se registró en el sistema.
- E. Tiempo: Permite seleccionar el tiempo en el cual el aula será utilizada por el docente.
- F. Controles: Permite guardar o borrar la información que se encuentre ingresando en este proceso.

Figura 158. Identificación de las secciones en el formulario de procesos del administrador del submenú reservar aula disponible.

The screenshot shows a web application window titled 'ADMINISTRADOR' with a menu bar containing 'SALIR' and 'REPORTES'. Below the menu bar is a navigation bar with buttons: 'Crear, Modificar Materia Aula o Carrera', 'Crear Horario', 'Modificar Datos Personales', 'Utilizar Aula Asignada o Reservada', and 'Reservar Aula Disponible'. The main content area is titled 'UTILIZAR AULA DISPONIBLE' and contains the following sections:

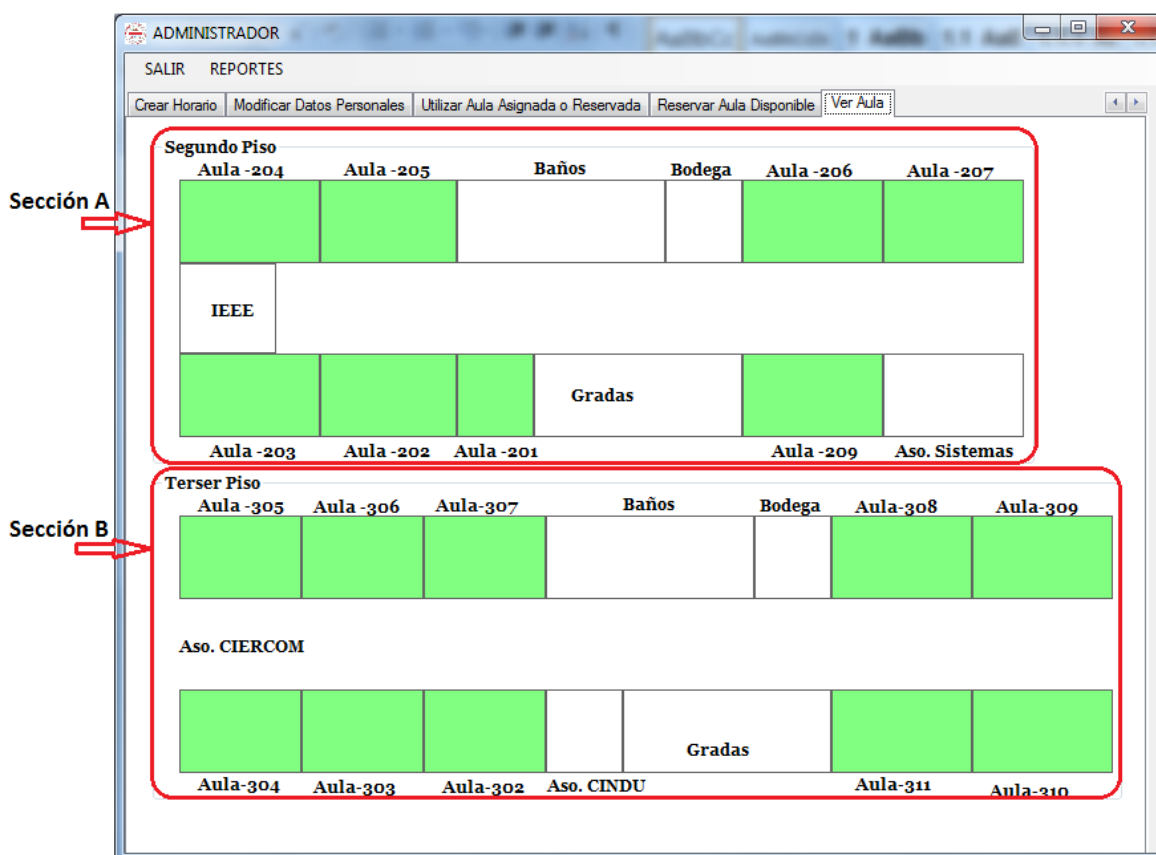
- Sección A:** 'Datos del Aula' with fields for 'Número de Aula' (dropdown), 'Uvicación' (text), and 'Capacidad de Estudiantes' (text).
- Sección D:** 'Datos Docente' with fields for 'C.I.\_Docente' (text, value: 1003788401), 'Docente' (text, value: GUERRERO ASEICHA ANTONIO), and 'Rol\_Usuario' (text, value: Administrador).
- Sección B:** 'Detalle:' with a text area for 'MOTIVO DE LA RESERVA:'.
- Sección C:** 'Parámetros de Reserva' with fields for 'Día' (text, value: LUNES), 'Fecha de Reserva' (calendar, value: 01/06/2015), 'Periodo Académico' (text, value: MARZO 2015 - AGOSTO 2015), 'Materia' (dropdown), 'Carrera' (dropdown), 'Nivel' (dropdown), and 'Paralelo' (dropdown).
- Sección E:** 'Tiempo' with fields for 'Número de Horas' (dropdown), 'Hora Actual' (text, value: 10:33:38), 'Hora Inicial' (dropdown), and 'Hora Final' (text, value: 00:00:00).
- Sección F:** 'Controles' with buttons for 'LIMPIAR' and 'OK'.

At the bottom of the form, there are two small images showing classroom interiors.

#### 7.9.1.3.2.1.8 Ver Aula.

- A. Segundo Piso: Identifica a todas las aulas de la facultad que se encuentran en el segundo piso de la facultad.
- B. Tercer Piso: Identifica a todas las aulas de la facultad que se encuentran en el tercer piso de la facultad.

Figura 159. Identificación de las secciones en el formulario de procesos del administrador del submenú ver aula.



Para identificar si un aula se encuentra ocupada o disponible se visualiza dependiendo de los siguientes colores:

- ✓ Color Azul: El aula se encuentra Ocupada.
- ✓ Color Verde: El aula se encuentra disponible.

*Nota:*

En lo referente al menú selectivo, para el administrador, todos los campos son habilitados, esto depende del modo de ingreso con el cual se registra el docente o funcionario ya que cada campo realiza una función determinada por ende se tiene derecho a todo en el sistema

### 7.9.1.3.3 Formulario de reportes.

Este formulario es utilizado únicamente por el administrador del sistema, el mismo que es un informe presentado por el sistema con parámetros establecidos previamente de las aulas utilizadas. A continuación se presenta dicha interfaz en la siguiente figura.

Figura 160. Formulario de reportes del administrador.

REPORTES

SAP CRYSTAL REPORTS\*

Informe principal

**SISTEMA DE CONTROL DE ACCESO  
INFORME DEL ADMINISTRADOR  
UTN - FICA - CIERCOM**

Cédula: 1003098215  
Docente: BENAVIDES CHAMORRO KATHERINE

Tipo	Día	Fecha	Carrera	Materia	Paralelo	Nivel	Aula	Horas	Hora_Inicio	Hora_Final	Detalle
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A	1º	AULA_201	3	08:28:47	08:30:37	DSFSDFSD
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A	1º	AULA_201	3	08:41:03	08:42:01	DSFSDFSD
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A	1º	AULA_201	3	08:45:27	08:46:45	DSFSDFSD
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A	1º	AULA_201	3	08:51:33	08:52:26	DSFSDFSD
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	A	1º	AULA_201	3	08:54:18	08:56:41	DSFSDFSD
RESERVA	SÁBADO	23/05/2015	CIERCOM	ALGEBRA LINEAL	B	1º	AULA_201	3	10:10:55	10:12:05	FB

Cédula: 1003765565  
Docente: ANDRADE VALENZUELA CARLOS




Tipo	Día	Fecha	Carrera	Materia	Paralelo	Nivel	Aula	Horas	Hora_Inicio	Hora_Final	Detalle
LIMPIEZA	SÁBADO	23/05/2015	-	-	-	-	AULA_201	0	11:10:41	11:26:55	LIMPIEZA DELAULA


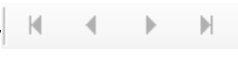


Autor: Guerrero Asecha Antonio

Página 1 de 1

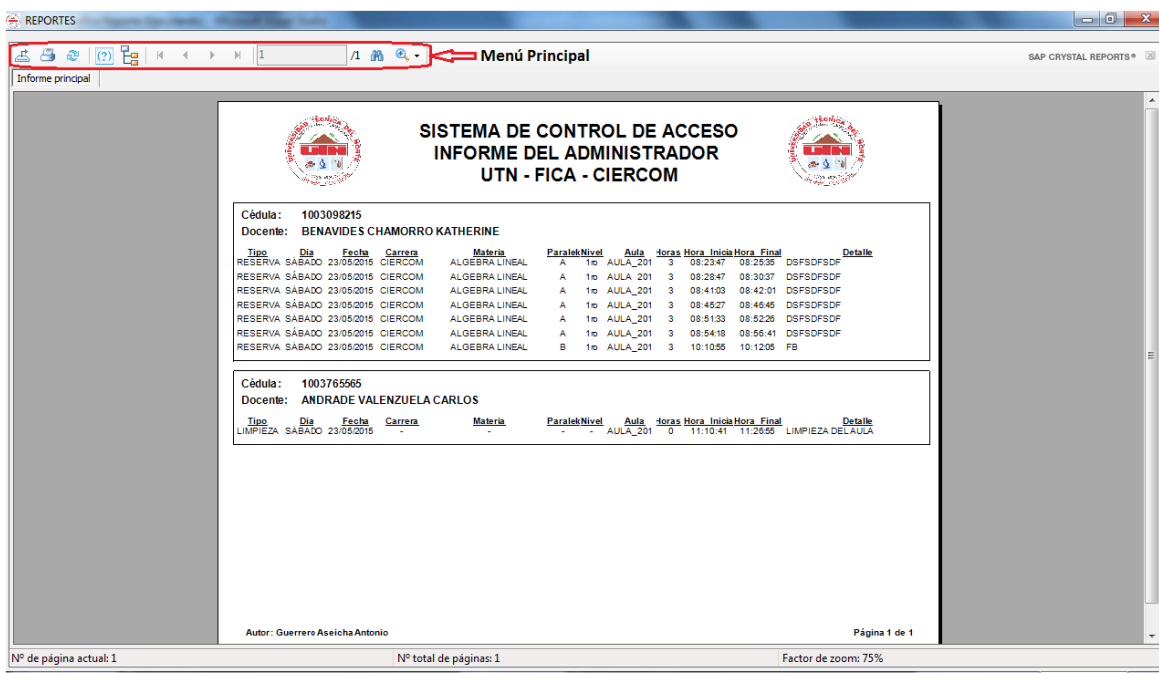
Nº de página actual: 1      Nº total de páginas: 1      Factor de zoom: 75%

Además, esta interfaz presenta un menú principal que cuentan con los siguientes campos habilitados:

- ✓ **Menú Principal:** Permite realizar varias funciones :
  - *Exportar Informe*  : Permite guardar el documento del reporte generado.
  - *Imprimir Informe*  : Imprime el documento del reporte generado.
  - *Actualizar*  : Permite actualizar o refrescar los datos indicados en el reporte que es generado.

- **Mostrar Árbol de Grupo** : Permite desplegar una lista de los docentes que han realizado un trabajo en una aula determinada por medio de la cédula del mismo.
- **Desplazar** : Permite desplazarse de una hoja a otra en el documento del reporte generado.
- **Paginación**  / 1: Identifica la página en la que se encuentra en ese momento del reporte.
- **Buscar** : Realiza una búsqueda de un parámetro en específico.
- **Zoom** : Aumenta o disminuye la visión del documento del reporte generado.

.Figura 161. Identificación del menú en el formulario de reportes del administrador



### 7.9.1.4 Proceso de utilización del sistema del administrador

A continuación se especifica todos los pasos que debe seguir el usuario que cumple las funciones de administrador en el sistema de control de acceso de las aulas de la facultad, de igual forma se presenta en tres etapas según los formularios que el sistema contiene.

#### **7.9.1.4.1 Pasos a seguir en el formulario de autenticación.**

Para poder ingresar al sistema mediante el modo administrador es necesario seguir los siguientes pasos:

1. Seleccionar en la sección “*Modo de ingreso*” la casilla que identifica este modo.
2. Una vez seleccionado el modo de ingreso se habilitan los campos de texto en la sección “*Autenticación*” en los cuales se debe proceder a ingresar la información que los mismos requieren con sus respectivos datos del docente:
  - ✓ *Ingrese Usuario*: Se debe escribir el nombre con el cual esté registrado en el sistema.
  - ✓ *Ingrese Contraseña*: Se debe escribir la contraseña con la cual esté registrado en el sistema.
  - ✓ *Confirmar Contraseña*: Se debe volver a escribir la contraseña con la que se encuentra registrado en el sistema.

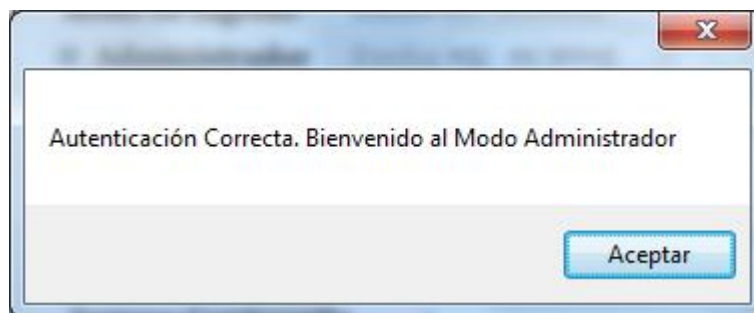
De igual forma se habilitan los botones con los cuales se puede:

- ✓ *Limpiar*: Borra el texto introducido en los campos usuario, contraseña y la confirmación de la contraseña.
  - ✓ *Ingresar*: En caso de que los datos se encuentren correctamente ingresados en los anteriores campos el usuario puede acceder al sistema en lo cual se abre una nueva ventada, caso contrario el sistema deniega el ingreso mostrando previamente el error suscitado.
3. Finalmente se hace click en el botón “Ingresar” y debe esperar el mensaje de confirmación.

Figura 162. Pasos a seguir en el formulario autenticación del administrador.

The screenshot shows a window titled 'AUTENTICACIÓN' for the 'UNIVERSIDAD TÉCNICA DEL NORTE FICA - CIERCOM SISTEMA DE CONTROL DE ACCESO'. It features three main sections: 'Modo de ingreso' with radio buttons for 'Administrador', 'Usuario', and 'Eventual'; 'Datos del Sistema' with fields for 'Fecha' (01/06/2015), 'Dia' (LUNES), and 'Hora Actual' (18:26:00); and 'Autenticación' with input fields for 'Ingreso Usuario' (tonny), 'Ingreso Contraseña' (\*\*), and 'Confirmar Contraseña' (\*\*). At the bottom are 'LIMPIAR' and 'INGRESAR' buttons. Three yellow arrows and labels indicate the steps: 'Paso 1' points to the 'Administrador' radio button, 'Paso 2' points to the password input fields, and 'Paso 3' points to the 'INGRESAR' button.

Figura 163. Mensaje de confirmación del ingreso al sistema del usuario administrador.



*Nota:*

*En caso de existir algún error en los datos ingresados por el usuario, el sistema informará mediante un mensaje el motivo por el cual no se pudo dar acceso al sistema.*

#### **7.9.1.4.2 Pasos a seguir en el formulario procesos.**

A continuación se profundiza las funciones y el uso de cada uno de los submenús descritos anteriormente en el formulario de procesos según los parámetros correspondientes al usuario administrador.

##### **7.9.1.4.2.1 Nuevo Docente.**

Para poder ingresar a un nuevo docente al sistema es necesario verificar o validar el número de cédula del mismo, para ello se siguen los siguientes pasos tal como se menciona a continuación:

1. Ingresar en el cuadro de texto “C.I. Docente” el número de cedula del docente o funcionario.
2. Hacer click en el botón “Validar” y esperar al mensaje de verificación.

En el caso que el número de cédula sea correcto los campos que se encuentran bloqueados en la sección “*Datos Personales*” se habilitarán, lo cual permite pasar al siguiente paso, de lo contrario, estos datos ingresados se borrarán de los cuadros de texto previo a un mensaje de error lo que lleva nuevamente al primer paso.



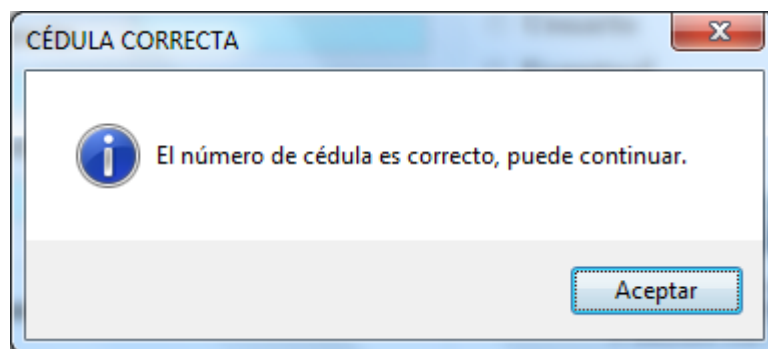
Figura 164. Pasos para validar el número de cedula del docente.

The screenshot shows a web application window titled 'ADMINISTRADOR'. The main content area is titled 'DATOS DEL DOCENTE'. It is divided into several sections:

- Datos Personales:** Includes fields for 'C.I. Docente' (containing '2100721774'), 'Primer\_Nombre\_Docente', 'Segundo\_Nombre\_Docente', 'Primer\_Apellido\_Docente', 'Segundo\_Apellido\_Docente', 'Dirección', 'Teléfono\_Celular\_Principal', and 'Correo' (with a placeholder '@utn.edu.ec'). A yellow box highlights the 'VALIDAR' button next to the ID number, with a yellow arrow labeled 'Paso 2' pointing to it. Another yellow arrow labeled 'Paso 1' points to the ID number field.
- Identificación de Registro:** Includes fields for 'Nombre\_Usuario' and 'Contraseña\_Usuario' (masked with asterisks).
- Rol\_Usuario:** Includes radio buttons for 'Administrador', 'Usuario', 'Eventual', and 'Conserje'. A circular logo for 'UNIVERSIDAD TÉCNICA DEL NORTE' is visible.
- Información:** A section with the text 'Contiene la cédula de identidad del docente'.
- CIERCOM:** Logo for 'INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN'.
- Controles:** Includes 'GUARDAR' and 'LIMPIAR' buttons.

At the bottom of the form, there is a banner image of a building with the text 'Presencial de Ingeniería en Ciencias Aplicadas'.

Figura 165. Confirmación de la validación del número de cédula del docente.



3. Una vez habilitados los cuadros de texto de la sección " *Datos Personales*" se procede a llenar con la información correspondiente a cada uno de estos campos, los mismos que identifican lo siguiente:
  - ✓ *Primer Nombre Docente*: Se debe ingresar solo el primer nombre del docente.

- ✓ *Segundo Nombre Docente:* Se debe ingresar solo el segundo nombre del docente.
  - ✓ *PrimerApellido Docente:* Se debe ingresar el apellido paterno del docente.
  - ✓ *SegundoApellido Docente:* Se debe ingresar el apellido materno del docente.
  - ✓ *Dirección:* Se debe ingresar la dirección del domicilio del docente.
  - ✓ *Teléfono Celular Personal:* Se debe ingresar el número de teléfono personal del docente con el cual se le pueda localizar.
  - ✓ *Correo:* Se debe ingresar el correo electrónico del docente proporcionado por la Universidad Técnica del Norte.
4. Elegir en la sección “*Rol Usuario*” la función que este docente desempeñará en la facultad. Estas funciones permiten al docente registrado ingresar al sistema con las prioridades de cada una de ellas.
  5. Finalmente hacer click en el botón “Guardar” y esperar el mensaje de confirmación. Este guardará en la base de datos la información ingresada.

Figura 166. Pasos para la creación de un docente en el formulario procesos del administrador.

**ADMINISTRADOR**

SALIR REPORTES

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar A...

### DATOS DEL DOCENTE

**Datos Personales**

C.I. Docente  8100721774

Primer\_Nombre\_Docente JOSE

Segundo\_Nombre\_Docente LUIS

Primer\_Apellido\_Docente PEREZ

Segundo\_Apellido\_Docente VALVERDE

Dirección LOS CEIBOS

Teléfono\_Celular\_Principal 0987654321

Correo jperezl @utn.edu.ec

**Identificación de Registro**

Nombre\_Usuario JOSE

Contraseña\_Usuario \*\*\*\*\*

**Rol Usuario**

Administrador

Usuario **Paso 4**

Eventual

Conserje

**Información**

Establece al docente solo como usuario del sistema, podrá hacer reservaciones de aulas y ver disponibilidad de las mismas

**Controles**

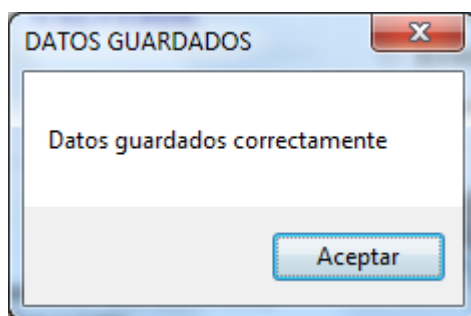
**Paso 3**

**Paso 4**

**Paso 5**

CIERCOM  
INGENIERIA EN ELECTRONICA  
Y REDES DE COMUNICACION

Figura 167. Confirmación de los datos guardados en el sistema del nuevo docente.



*Notas:*

*En los parámetros para ingresar datos, en caso de existir errores, el sistema informará por medio de un mensaje de error el motivo por el cual ese campo no puede ser completado con la información proporcionada, por lo que hay que tener en cuenta cada uno de ellos.*

*De igual forma en la sección “Identificación Registro” los campos “Nombre Usuario y Contraseña” son generados automáticamente por el sistema siendo estos campos el primer nombre del docente y la cédula del mismo respectivamente. Estos datos pueden ser modificados en el submenú **Modificar datos personales** (se explica más adelante en este manual).*

*7.9.1.4.2.2 Buscar Modificar y Eliminar Docente.*

Este parámetro ayuda al administrador a buscar a un docente específico para modificar datos del mismo o eliminarlo del sistema en caso de ya no pertenecer a la facultad. Para utilizar este método se sigue dos procesos específicamente:

- A. Buscar
- B. Modificar o Eliminar

#### 7.9.1.4.2.2.1 Proceso A (Buscar)

Inicialmente todas las secciones se encuentran deshabilitadas por defecto. Conforme se realice un proceso cada una de ellas se habilitará dependiendo del trabajo a realizar por el administrador. Para ello se debe realizar los siguientes pasos:

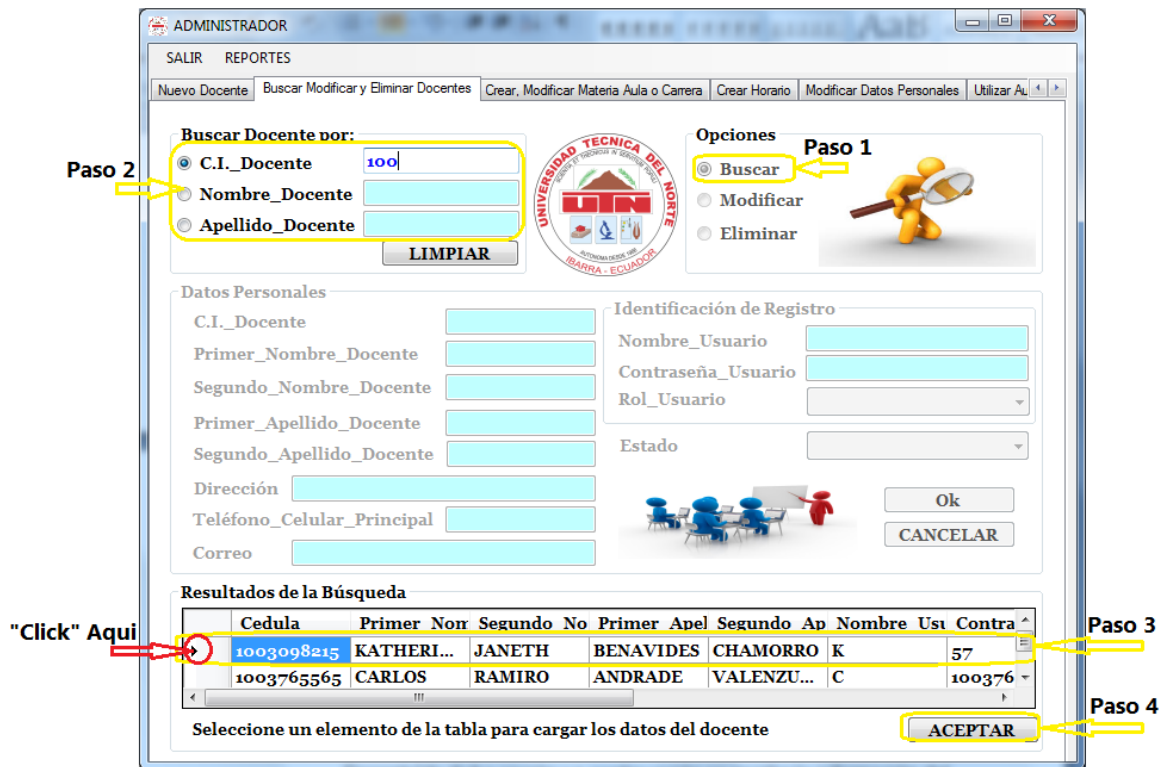
1. Dirigir a la sección de “*Opciones*” y hacer click en la casilla de “*Buscar*”.
2. A continuación se habilita la sección de “*Buscar Docente por*”, dependiendo del tipo de búsqueda que se desee realizar se puede escoger tres opciones:
  - a. *C.I. Docente*: Se buscará al docente por medio de su cédula de identidad, luego se habilita la casilla de texto correspondiente a “*C.I. Docente*” de dicha opción y a continuación se escribe la información del docente.
  - b. *Nombre Docente*: Se realiza la búsqueda por medio del primer nombre del docente. luego se habilita la casilla de texto correspondiente al “*Nombre Docente*” de dicha opción y a continuación se escribe la información del docente.
  - c. *Apellido Docente*: Se puede realizar la búsqueda por medio del primer apellido del docente. luego se habilita la casilla de texto correspondiente al “*Apellido Docente*” de dicha opción y a continuación se escribe la información del docente.

La búsqueda del docente se realiza progresivamente, esto se puede visualizar en la sección “*Resultados de la Búsqueda*”. Una vez identificado al docente que se desee se procede a:

3. Seleccionar al docente haciendo click en la parte izquierda de la tabla.

4. Hacer click en le botón “Aceptar” de la misma sección.

Figura 168. Pasos para la búsqueda de un docente en el formulario procesos del administrador.



A continuación los datos seleccionados del docente se cargan en cada una de las casillas de la sección “*Datos Personales*” y una vez cargados estos datos, las opciones “Modificar y Eliminar” de la sección “*Opciones*” se habilitan dando paso al siguiente proceso.

#### 7.9.1.4.2.2 Proceso B (Modificación o Eliminación)

Este proceso permite realizar dos trabajos diferentes al administrador:

- Modificar datos del docente seleccionado.
- Eliminar al docente seleccionado.

#### 7.9.1.4.2.2.2.1 Modificar datos del docente seleccionado.

El administrador tiene la capacidad de modificar ciertos datos de un docente determinado. Este proceso es implementado para el administrador debido a que pueden ocurrir errores ya sea en el registro de un docente con sus nombres, dirección, correo entre otros parámetros los mismos que pueden ser corregidos oportunamente. A continuación se enumeran los pasos para realizar esta tarea:

1. En la sección “*Opciones*” se debe seleccionar la opción “Modificar”. Una vez seleccionado se habilitan los campos de texto de la sección “*Datos Personales*” así como el “Rol Usuario” y el “Estado” del docente.
2. Habilitados los campos de la sección “*Datos Personales*” también se habilitan los botones “OK” y “Cancelar” que permiten realizar las siguientes funciones:
  - a) Botón OK: Verifica si la información modificada se encuentra correctamente ingresada y procede a guardar los cambios en la base de datos del sistema.
  - b) Botón Cancelar: limpia la información del docente y no realiza ninguna notificación en el sistema e inhabilita todas las secciones y las reubica en el estado por defecto.
3. Se procede a cambiar los datos que se desee modificar del docente.
4. Posteriormente se debe hacer click en el botón “OK” y esperar el mensaje de confirmación.

Figura 169. Pasos para modificar datos del docente seleccionado.

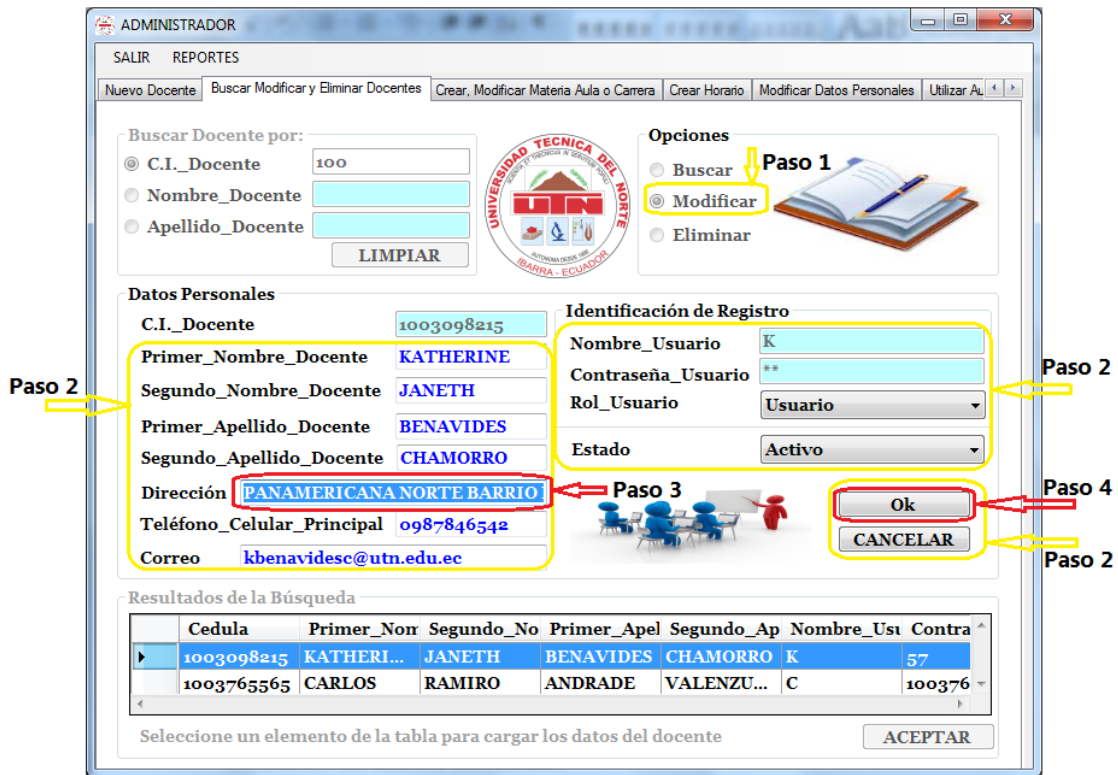
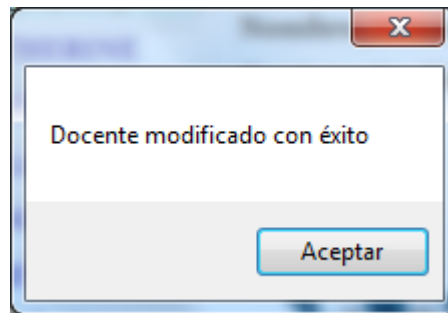


Figura 170. Confirmación de los datos modificados del docente.



#### 7.9.1.4.2.2.2 Eliminación del docente seleccionado.

De igual forma, el administrador tiene la capacidad de eliminar un docente del sistema. Este proceso es implementado para el administrador debido a que un docente ya no puede pertenecer a la facultad por ende mantener sus datos sería irrelevante. A continuación se enumeran los pasos para realizar esta tarea.

1. En la sección "Opciones" se debe seleccionar la opción "Eliminar".



2. Una vez seleccionado se cargan los datos del docente en la sección “Datos Personales” y se habilitan los botones “Ok” y “Cancelar”.

3. Posteriormente se hace click en el botón “OK” y se espera el mensaje de confirmación.

Se debe tener en cuenta que para eliminar a un docente del registro debe estar seguro de realizar esta acción. Para ello el sistema visualiza un mensaje adicional de precaución.

Figura 171. Pasos para eliminar a un docente seleccionado.

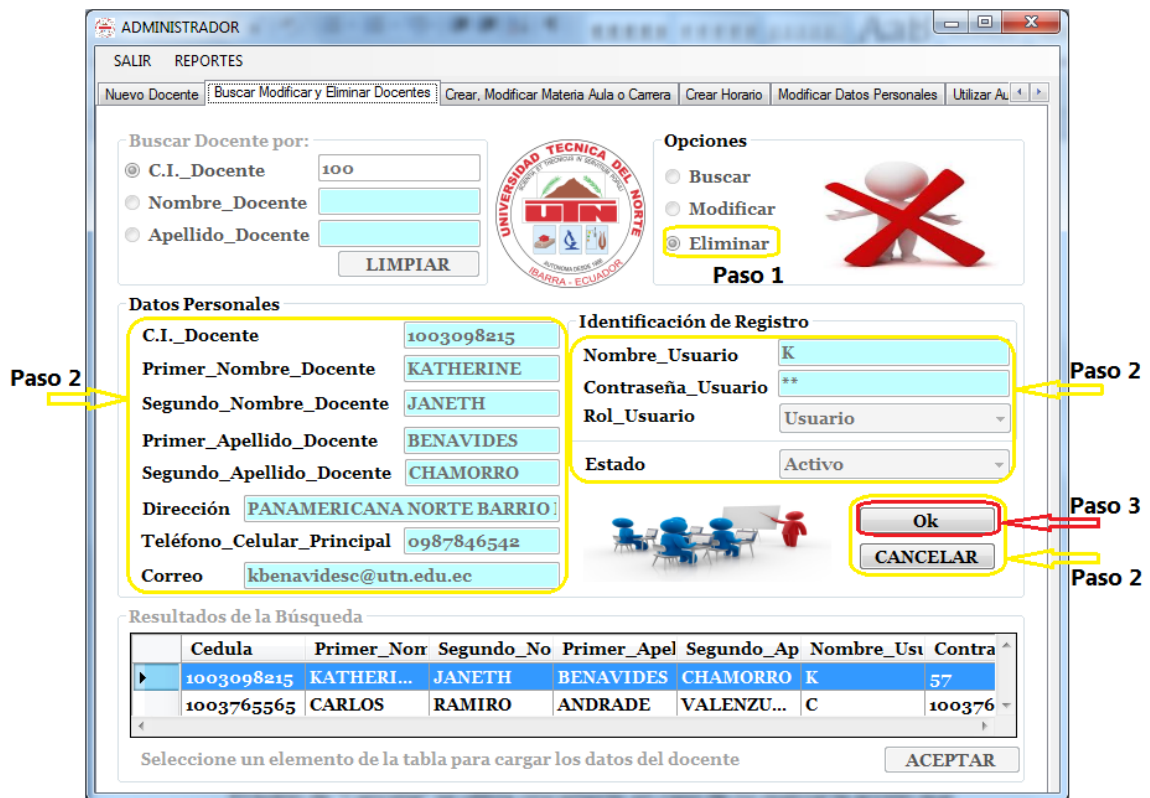




Figura 172. Mensaje de precaución antes de eliminar a un docente.

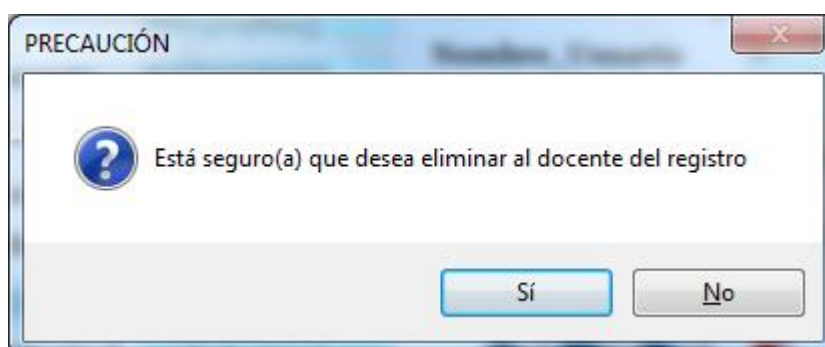
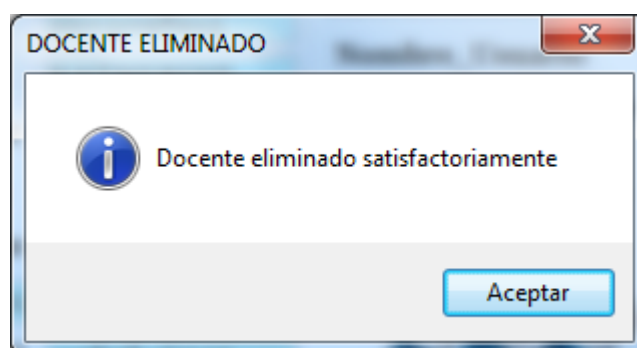


Figura 173. Confirmación del docente eliminado del sistema.



*Notas:*

*La sección “Identificación de Registro” no se habilita en ningún momento durante el proceso debido a que estos datos son únicamente modificables por el docente que le corresponda. Estos datos pueden ser modificados en el submenú **Modificar datos personales** (se explica más adelante en este manual).*

*El botón de “Cancelar” se utiliza únicamente en caso de no realizar la acción que se esté ejecutando en ese momento.*

*La eliminación del docente en el sistema no se realiza completamente, es decir, al docente se le pasa a un estado de “inactivo” ya que la eliminación completa de datos podría borrar completamente los registros del docente y ello podría traer inconvenientes, es por ello que simplemente se le ubica en ese*

*estado, en caso de requerir los servicios nuevamente del docente simplemente se modifica esta opción.*

#### *7.9.1.4.2.3 Crear o Modificar Materia, Aula o Carrera.*

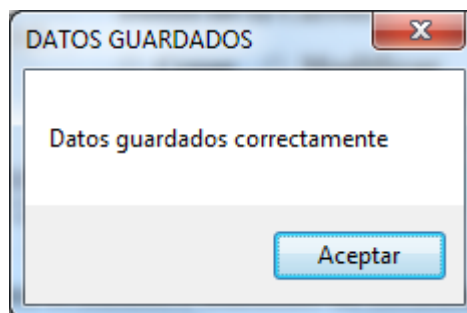
Este proceso permite la creación o modificación de los datos ya sea de un aula, materia o carrera permitiéndole al administrador realizar estas opciones dependiendo de las modificaciones de infraestructura o la modificación de la malla curricular de cada una de las diferentes carreras que se imparten en la facultad. Para hacer uso de este proceso se ha dividido en tres secciones diferentes los cuales son:

- A. Datos del Aula: En él se pueden realizar las funciones de “Crear” o “Modificar” parámetros de un aula determinada. Dependiendo de la selección de estas opciones, se habilitan o deshabilitan los campos que contiene esta sección.
  - a. Crear Aula: para crear un aula determinada se siguen los siguientes pasos:
    1. Seleccionar la opción “Crear “en la sección “*Datos del Aula* “en la cual se habilitan los campos pertenecientes a esta función.
    2. En el campo “Nombre del Aula” escribir un número que identifique al aula a crear.
    3. En el campo “Ubicación” seleccionar el lugar donde se encuentra físicamente esta aula en la facultad.
    4. En el campo “Capacidad de estudiantes” se debe ingresar el número máximo de estudiantes en la cual esta aula puede admitir.

5. Hacer click en el botón “Crear” de esta sección y esperar el mensaje de confirmación.

Figura 174. Pasos para crear un aula en el sistema.

Figura 175. Confirmación de la creación de una nueva aula.



- b. Modificar Aula: Para modificar un aula determinada se siguen los siguientes pasos:

1. Seleccionar la opción “Modificar” en la sección “*Datos del Aula*” en la cual se habilitan los campos pertenecientes a esta función.
2. En el campo “Seleccione Aula” escoger el aula a la cual se le va a modificar los parámetros de “Ubicación” o “Capacidad de estudiantes”, los mismos que serán cargados en los campos correspondientes a estos parámetros.
3. En el campo “Ubicación” seleccionar el lugar donde se encuentra físicamente esta aula en la facultad.
4. En el campo “Capacidad de estudiantes” se debe modificar el número máximo de estudiantes en la cual esta aula puede admitir.
5. Hacer click en el botón “Modificar” de esta sección y esperar el mensaje de confirmación.

Figura 176. Pasos para modificar los datos de un aula.

**ADMINISTRADOR**

SALIR REPORTE

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Au...

### CREAR O MODIFICAR AULAS, MATERIAS O CARRERAS

**Datos del Aula** **Paso 1** **Paso 2**

Crear  **Modificar**

Seleccione Aula **AULA\_202**

Ubicación **Segundo Piso**

Capacidad de Estudiantes **20** **Paso 4**

**Paso 5**

**Datos de la Materia**

Crear  Modificar

Nombre de la Materia

Número de Créditos

**Datos de la Carrera**

Crear  Modificar

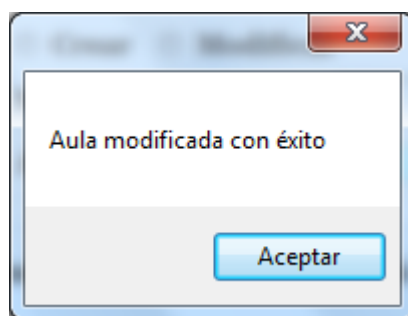
Nombre de la Carrera

Alias de la Carrera

**Información**

La capacidad del aula especifica el número máximo de estudiantes que ésta contendrá

Figura 177. Confirmación de la modificación de los datos del aula.



B. Datos de la Materia: En él se pueden realizar las funciones de “Crear” o “Modificar” parámetros de una materia determinada. Dependiendo de la selección de estas opciones se habilitan o deshabilitan los campos que contiene esta sección.

a. Crear Materia: Para crear una materia determinada se siguen los siguientes pasos:

1. Seleccionar la opción “Crear” en la sección “*Datos de la Materia*” en la cual se habilitan los campos pertenecientes a esta función.
2. En el campo “Nombre de la Materia” escribir el nombre completo de la materia a crear.
3. En el campo “Número de Créditos” se debe seleccionar el número de horas que esta materia tiene en el pénsum académico.
4. Hacer click en el botón “Crear” de esa sección y esperar el mensaje de confirmación

Figura 178. Pasos para crear una materia en el sistema.

ADMINISTRADOR

SALIR REPORTE

Nuevo Docente Buscar Modificar y Eliminar Docentes Crear, Modificar Materia Aula o Carrera Crear Horario Modificar Datos Personales Utilizar Au

### CREAR O MODIFICAR AULAS, MATERIAS O CARRERAS

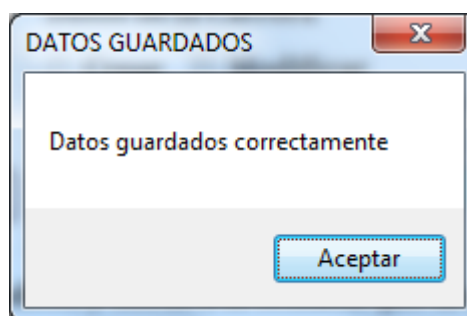
**Datos del Aula**  
 Crear  Modificar **Paso 1**  
Seleccione Aula   
Ubicación   
Capacidad de Estudiantes

**Datos de la Materia**  
 Crear  Modificar **Paso 2**  
Nombre de la Materia   
Número de Créditos     
**Paso 3** **Paso 4**

**Datos de la Carrera**  
 Crear  Modificar  
Nombre de la Carrera   
Alias de la Carrera

**Información**  
Debe seleccionar el número de créditos y horas a la semana que la materia contiene

Figura 179. Confirmación de la creación de una nueva materia.



- b. Modificar Materia: Para modificar una materia determinada se siguen los siguientes pasos:
1. Seleccionar la opción “Modificar” en la sección “*Datos de la Materia*” en la cual se habilitan los campos pertenecientes a esta función.

2. En el campo “Seleccione Materia” escoger la materia a la cual se va a modificar los parámetros de “Créditos”, el mismo que es cargado en el campo correspondiente a estos parámetros.
3. En el campo “Créditos” modificar este valor en caso de requerirlo.
4. Hacer click en el botón “Modificar” de esa sección y esperar el mensaje de confirmación.

Figura 180. Pasos para modificar los datos de una materia.

ADMINISTRADOR

SALIR REPORTES

Nuevo Docente Buscar Modificar y Eliminar Docentes Crear, Modificar Materia Aula o Carrera Crear Horario Modificar Datos Personales Utilizar Au

### CREAR O MODIFICAR AULAS, MATERIAS O CARRERAS

**Datos del Aula**  
 Crear  Modificar  
Seleccione Aula   
Ubicación   
Capacidad de Estudiantes   
CREAR MODIFICAR

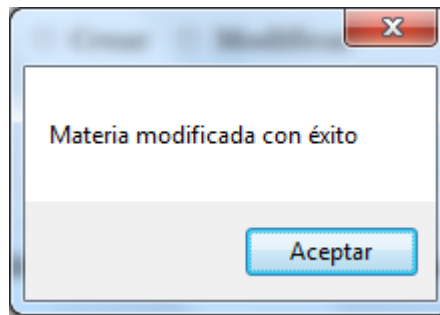
**Datos de la Materia**  
 Crear  Modificar ← Paso 1  
Seleccione Materia MATEMÁTICAS APLICADAS Paso 2  
Número de Créditos 4 Paso 3 CREAR MODIFICAR Paso 4

**Datos de la Carrera**  
 Crear  Modificar  
Nombre de la Carrera   
Alias de la Carrera  CREAR MODIFICAR

**CIERCOM**  
INGENIERÍA EN ELECTRÓNICA  
Y REDES DE COMUNICACIÓN

**Información**  
Seleccione la materia que desea modificar sus características, solamente se puede modificar el número de créditos y el nivel de la misma

Figura 181. Confirmación de la modificación de los datos de la materia.



C. Datos de la Carrera: En él se pueden realizar las funciones de “Crear” o “Modificar” parámetros de una carrera determinada. Dependiendo de la selección de estas opciones se habilitan o deshabilitan los campos que contiene esta sección.

a. Crear Carrera: para crear una carrera determinada se siguen los siguientes pasos:

1. Seleccionar la opción “Crear” en la sección “*Datos de la Carrera*” en la cual se habilitan los campos pertenecientes a esta función.
2. En el campo “Nombre de la Carrera” escribir el nombre completo de la Carrera a crear.
3. En el campo “Alias” se escribe las iniciales de la carrera a crear.
4. Hacer click en el botón “Crear” de esa sección y esperar el mensaje de confirmación



Figura 182. Pasos para crear una carrera en el sistema.

ADMINISTRADOR

SALIR REPORTE

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Au

### CREAR O MODIFICAR AULAS, MATERIAS O CARRERAS

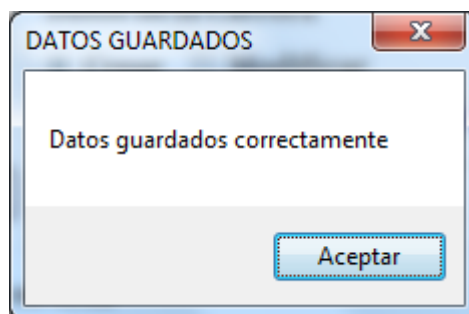
**Datos del Aula**  
 Crear  Modificar  
Nombre del Aula    
Ubicación   
Capacidad de Estudiantes

**Datos de la Materia**  
 Crear  Modificar  
Nombre de la Materia   
Número de Créditos

**Datos de la Carrera**  
 Crear  Modificar Paso 2  
Nombre de la Carrera   
Alias de la Carrera     
Paso 3 Paso 4

**Información**  
Especifica el alias de la carrera que se crea

Figura 183. Confirmación de la creación de una nueva carrera.



- b. Modificar Carrera: Para modificar una carrera determinada se siguen los siguientes pasos:
1. Seleccionar la opción “Modificar “en la sección “*Datos de la Carrera*” en la cual se habilitan los campos pertenecientes a esta función.

2. En el campo “Seleccionar Carrera” escoger la carrera a la cual se le va a modificar los parámetros de “Alias”, el mismo que es cargado en el campo correspondiente a estos parámetros.
3. Hacer click en el botón “Modificar” de esa sección y esperar el mensaje de confirmación.

Figura 184. Pasos para crear una carrera en el sistema.

ADMINISTRADOR

SALIR REPORTE

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Au

### CREAR O MODIFICAR AULAS, MATERIAS O CARRERAS

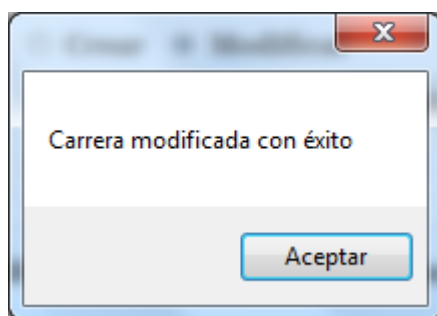
**Datos del Aula**  
 Crear  Modificar  
Nombre del Aula  -   
Ubicación   
Capacidad de Estudiantes

**Datos de la Materia**  
 Crear  Modificar  
Nombre de la Materia   
Número de Créditos

**Datos de la Carrera**  
 Crear  Modificar **Paso 1**  
Seleccione Carrera  **Paso 2**  
Alias de la Carrera    **Paso 3** **Paso 4**

**Información**  
Seleccione la carrera que desea modificar sus características, solamente se puede modificar el alias de la misma

Figura 185. Confirmación de la modificación de los datos de la carrera.



*Nota:*

*La sección de “Información” genera una ayuda para el administrador, informando lo que se debe hacer en cada uno de los campos que se debe ingresar o seleccionar.*

#### *7.9.1.4.2.4 Crear Horario.*

Este proceso se utiliza para crear un horario de clases dependiendo del periodo académico que se imparte en la facultad. Este horario es asignado por el departamento de informática de la universidad el mismo que debe ser ingresado por el administrador al sistema para que las aulas funcionen dependiendo del horario correspondiente a cada docente.

Existen tres opciones para utilizar este campo por el administrador los mismos que son:

- ✓ La creación de un nuevo horario académico.
- ✓ La modificación de un periodo académico ya existente.
- ✓ La duplicación de datos de un periodo académico existente.

#### 7.9.1.4.2.4.1 Creación de un nuevo horario académico.

A continuación se menciona los pasos a seguir para la creación de un nuevo periodo académico con su respectivo horario de clases para los docentes de la facultad.

1. Seleccionar la opción Nuevo de la sección “Opciones”.

Una vez realizada esta acción se habilita la sección “Acción” con el nombre “Periodo Académico” con dos campos extras:

- Nuevo: Permite seleccionar de una lista, el periodo académico que se vaya a crear.
- Vigente: Visualiza el periodo académico que se encuentra en proceso actualmente.

2. Hacer click en el campo Nuevo que se encuentra habilitado de la sección “Periodo Académico” y seleccionar un periodo académico que se desee crear.

3. Hacer click en el botón “OK” de la sección “Periodo Académico” y esperar el mensaje de confirmación.

Antes de crear un horario diferente se debe modificar los periodos académicos, estos se realizan internamente en el sistema por lo que el administrador es advertido de las acciones a tomar.

Figura 186. Pasos para seleccionar un periodo académico a crear en el sistema.

ADMINISTRADOR

SALIR REPORTE

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Au...

### HORARIO ASIGNADO SEGÚN EL DEPARTAMENTO DE INFORMÁTICA DE LA UTN

**Paso 1**

Opciones

- Nuevo
- Modificar
- Duplicar

Periodo Académico

Nuevo: SEPTIEMBRE 2016 - FEBRERO 2017 **Paso 3** OK

Vigente: MARZO 2016 - AGOSTO 2016 **Paso 2**

Datos del Horario

Día: [dropdown] Carrera: [dropdown] Aula: [dropdown] Horas Inicio: [dropdown]

Docente: [dropdown] Paralelo: [dropdown] Final: [dropdown]

Materia: [dropdown] Nivel: [dropdown] Nº Horas: [input type="text"]

CREAR MODIFICAR LIMPIAR

Horarios del Periodo Académico

Total de registros: [input type="text"] Seleccione un elemento de la tabla para cargar los datos CARGAR

Figura 187. Sugerencia antes de crear un periodo académico nuevo.

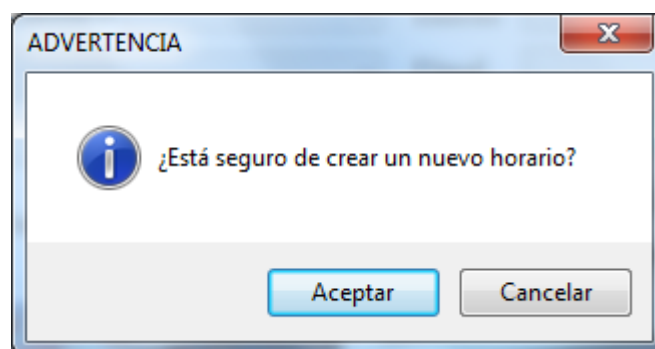


Figura 188. Sugerencia antes de continuar con la modificación del periodo académico.

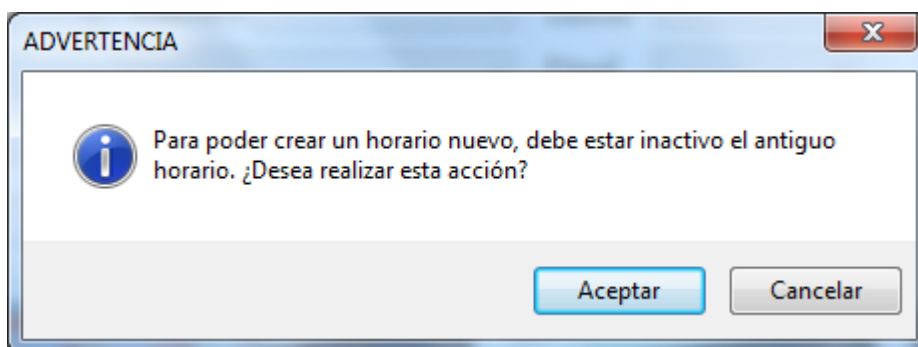
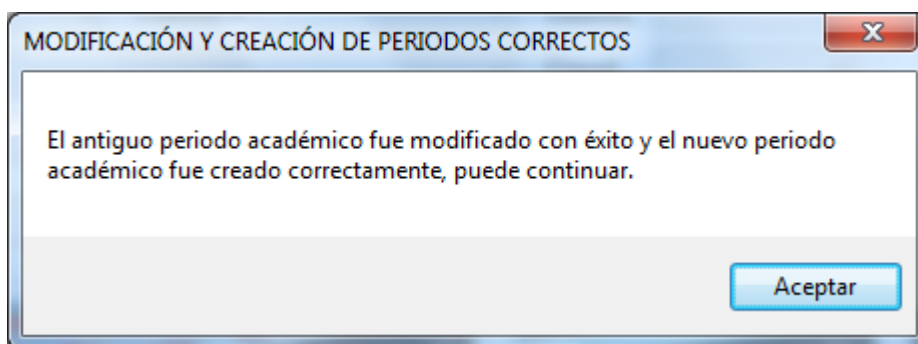


Figura 189. Confirmación de la modificación y creación de un nuevo periodo académico.



Una vez realizado este paso se habilitan los campos de la sección “*Datos del Horario*” y a continuación se eligen los parámetros para crear el horario de un determinado docente. Estos parámetros se mencionan en los siguientes pasos:

4. Día: Corresponde al día que el docente debe impartir la clase a una determinada materia.
5. Carrera: Corresponde la carrera a la que el docente va a dar clases.
6. Docente: Corresponde el nombre del docente al que se va a asignar un aula determinada para impartir clases.
7. Materia: Corresponde la materia a la cual el docente debe impartir en una aula determinada.
8. Aula: Corresponde al aula asignada para el docente.
9. Paralelo: Corresponde al paralelo de la materia a dar clases.

10. Nivel: Corresponde el nivel de la materia a dar clases.
11. Hora Inicio: Corresponde al inicio de la hora en la cual la materia debe ser impartida.
12. Hora final: Corresponde al final de la hora en la cual el docente terminará de dar clases.

Para poder crear el horario de clases o corregir algún error se puede utilizar los siguientes botones que cumplen determinada función.

- ✓ Crear: Permite crear el horario de clases previa selección.
- ✓ Limpiar: Permite limpiar los campos de la sección “*Datos del Aula*” para corregir posibles errores.

13. Finalmente, una vez seleccionados todos los campos de la sección “*Datos del Horario*” se debe seleccionar el botón “Crear” y esperar el mensaje de confirmación.

Una vez realizado estos pasos el horario creado aparecerá en la sección “Horario del Periodo Académico” conjuntamente con el nombre del periodo académico creado y los campos de la sección “*Datos del Horario*” se borrarán dando paso a un nuevo horario a crear, este proceso se realizará cada vez que un horario nuevo se cree, el mismo que irá apareciendo en la lista de la sección “Horario del Periodo Académico”.

Este proceso no terminara mientras el administrador no culmine su proceso de autenticación, si el administrador decide finalizar la estadía en el sistema el periodo académico se creara hasta el punto donde culmine el mismo.

Figura 190. Creación de un horario en un periodo académico en el sistema.

ADMINISTRADOR

SALIR REPORTES

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | **Crear Horario** | Modificar Datos Personales | Utilizar Au...

### HORARIO ASIGNADO SEGÚN EL DEPARTAMENTO DE INFORMÁTICA DE LA UTN

Opciones  
 Nuevo  
 Modificar  
 Duplicar

Periodo Académico  
Nuevo: SEPTIEMBRE 2016 - FEBRERO 2017 OK  
Anterior: MARZO 2016 - AGOSTO 2016

**Datos del Horario**

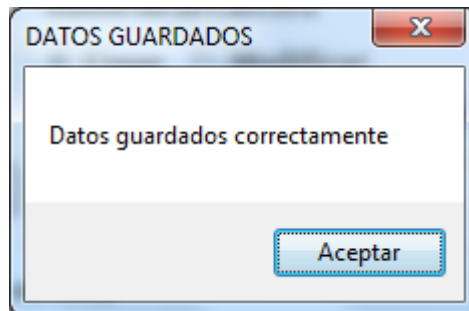
Paso 4: Día: LUNES  
Paso 5: Carrera: CIERCOM  
Paso 8: Aula: AULA\_201  
Paso 6: Docente: GUERRERO ASEICHA ANTONIO  
Paso 9: Paralelo: A  
Paso 7: Materia: MATEMÁTICAS APLICADAS  
Paso 10: Nivel: 4to  
Paso 11: Inicio: 07:00:00  
Paso 12: Final: 10:00:00  
N° Horas: 3

Paso 13: **CREAR** | MODIFICAR | LIMPIAR

Horarios del Periodo Académico SEPTIEMBRE 2016 - FEBRERO 2017

Total de registros: [ ] Seleccione un elemento de la tabla para cargar los datos [ CARGAR ]

Figura 191. Confirmación de la creación de un nuevo horario académico.



Nota:

*El campo "N° Horas" de la sección "Horas" no se habilita debido a que este campo es calculado automáticamente por el sistema.*

*El botón limpiar borra todos los datos que se ingresaron en la sección "Datos del Horario" permitiéndole al docente corregir un posible error.*



#### 7.9.1.4.2.4.2 *Modificación de un periodo académico ya existente.*

A continuación se menciona los pasos a seguir para la modificación de un periodo académico seleccionado.

1. Seleccionar la opción Modificar de la sección “Opciones”.

Una vez realizada esta acción se habilita la sección “Acción” con el nombre “Periodo Académico” con dos campos extras:

- Modificar: Permite seleccionar de una lista, el periodo académico al cual se va a modificar ciertos parámetros del horario.
  - Vigente: Visualiza el periodo académico que se encuentra en proceso actualmente.
2. Hacer click en el campo Modificar que se encuentra habilitado de la sección “Periodo Académico” y seleccionar un periodo académico que se desee modificar.
  3. Hacer click en el botón “OK” de la sección “Periodo Académico”.

Figura 192. Pasos para seleccionar un periodo académico a modificar en el sistema.

ADMINISTRADOR

SALIR REPORTES

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Au...

### HORARIO ASIGNADO SEGÚN EL DEPARTAMENTO DE INFORMÁTICA DE LA UTN

**Paso 1**

Opciones

Nuevo

**Modificar**

Duplicar

**Periodo Académico**

Modificar: SEPTIEMBRE 2016 - FEBRERO 2017

Vigente: SEPTIEMBRE 2016 - FEBRERO 2017

**Paso 3**

OK

**Datos del Horario**

Día: [dropdown] Carrera: [dropdown] Aula: [dropdown]

Docente: [input] Paralelo: [dropdown]

Materia: [dropdown] Nivel: [dropdown]

Horas

Inicio: [dropdown]

Final: [dropdown]

Nº Horas: [input]

CREAR MODIFICAR LIMPIAR

Horarios del Periodo Académico

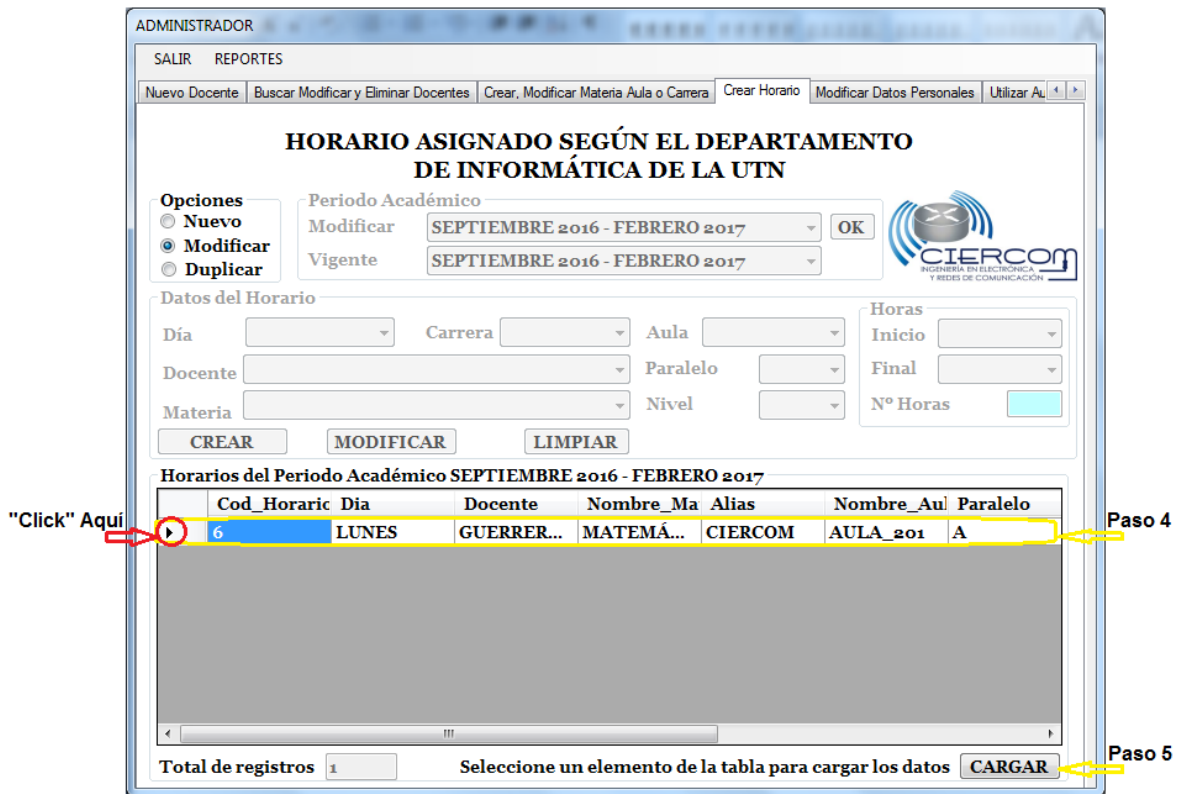
Total de registros: [input] Seleccione un elemento de la tabla para cargar los datos CARGAR

Una vez realizado este proceso, la sección “Horarios del Periodo Académico” con se respectivo nombre se habilita cargando todos los horarios creados en ese periodo académico dando paso a los siguientes pasos.

Para modificar parámetros en un horario académico se deben seguir los siguientes pasos:

4. Seleccionar el horario académico haciendo click en la parte izquierda de la tabla.
5. Hacer click en le botón “Cargar” de la misma sección.

Figura 193. Pasos para cargar los datos de un periodo académico a modificar en el sistema.



A continuación los datos seleccionados del horario se cargan en cada una de las casillas de la sección “*Datos del Horario*” y una vez cargados estos datos, el botón “Modificar” de la misma sección se habilita permitiendo realizar las modificaciones pertinentes de ese horario.

6. Una vez realizada las modificaciones en las casillas de la sección “*Datos del Horario*” se procede hacer click en el botón Modificar y finalmente esperar el mensaje de confirmación.

Figura 194. Modificación de un horario en un periodo académico en el sistema.

ADMINISTRADOR

SALIR REPORTES

Nuevo Docente Buscar Modificar y Eliminar Docentes Crear, Modificar Materia Aula o Carrera Crear Horario Modificar Datos Personales Utilizar Au

### HORARIO ASIGNADO SEGÚN EL DEPARTAMENTO DE INFORMÁTICA DE LA UTN

**Opciones**

Nuevo

Modificar

Duplicar

**Periodo Académico**

Modificar: SEPTIEMBRE 2016 - FEBRERO 2017 OK

Vigente: SEPTIEMBRE 2016 - FEBRERO 2017

**Datos del Horario**

Día: MIÉRCOLES Carrera: CIERCOM Aula: AULA\_201

Docente: GUERRERO ASEICHA ANTONIO Paralelo: A

Materia: MATEMÁTICAS APLICADAS Nivel: 4to

Horas: Inicio: 07:00:00 Final: 10:00:00 N° Horas: 3

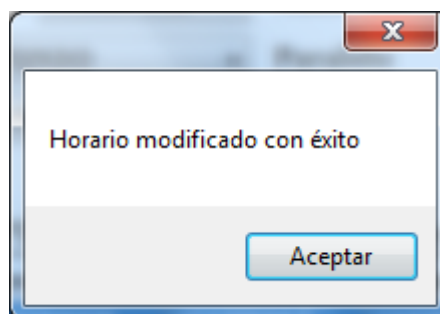
CREAR MODIFICAR Paso 6 LIMPIAR

**Horarios del Periodo Académico SEPTIEMBRE 2016 - FEBRERO 2017**

Cod_Horari	Dia	Docente	Nombre_Ma	Alias	Nombre_Au	Paralelo
6	LUNES	GUERRER...	MATEMÁ...	CIERCOM	AULA_201	A

Total de registros: 1 Seleccione un elemento de la tabla para cargar los datos CARGAR

Figura 195. Confirmación de la modificación del horario académico en el sistema.



Nota:

Los botones *Crear* y *Limpiar* de la sección “*Datos del horario*” no se habilitan ya que esta función no pertenece a este proceso.

#### *7.9.1.4.2.4.3 Duplicación de datos de un periodo académico existente.*

A continuación se menciona los pasos a seguir para la duplicación de un periodo académico seleccionado.

1. Seleccionar la opción Duplicar de la sección “Opciones”.

Una vez realizada esta acción se habilita la sección “Acción” con el nombre “Duplicar Periodo Académico” con dos campos extras:

- Nuevo: Permite seleccionar de una lista, el periodo académico que se vaya a crear.
  - A Duplicar: Permite seleccionar de una lista, el periodo académico al cual se tomará como referencia el periodo académico y se extraerá los horarios de ese periodo académico.
2. Hacer click en el campo A Duplicar que se encuentra habilitado de la sección “Duplicar Periodo Académico” y seleccionar un periodo académico que será duplicado.
  3. Hacer click en el campo Nuevo que se encuentra habilitado de la sección “Duplicar Periodo Académico” y seleccionar un periodo académico que se desee Crear.
  4. Hacer click en el botón “OK” de la sección “Duplicar Periodo Académico” y esperar el mensaje de confirmación.

Figura 196. Pasos para seleccionar un periodo académico a duplicar en el sistema

ADMINISTRADOR

SALIR REPORTES

Nuevo Docente | Buscar Modificar y Eliminar Docentes | Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Al...

### HORARIO ASIGNADO SEGÚN EL DEPARTAMENTO DE INFORMÁTICA DE LA UTN

**Opciones**

- Nuevo
- Modificar
- Duplicar

**Duplicar Periodo Académico**

Nuevo: SEPTIEMBRE 2016 - FEBRERO 2017

A Duplicar: MARZO 2016 - AGOSTO 2016

**Datos del Horario**

**Paso 1** Día  Carrera  Aula  **Paso 2** **Paso 3** **Paso 4**

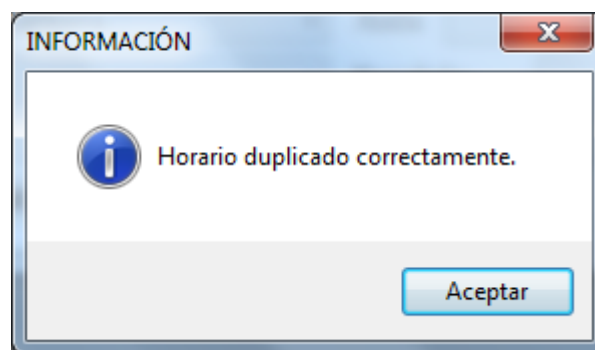
Docente  Paralelo  Horas Inicio  Final  N° Horas

Materia  Nivel

**Horarios del Periodo Académico**

Total de registros  Seleccione un elemento de la tabla para cargar los datos

Figura 197. Confirmación de la duplicación del horario académico en el sistema.



Una vez realizado este proceso, la sección “Horarios del Periodo Académico” con se respectivo nombre se habilita cargando todos los horarios duplicados en ese periodo académico dando paso a los siguientes pasos.

Para modificar parámetros en un horario académico duplicado se deben seguir los siguientes pasos:

5. Seleccionar el horario académico haciendo click en la parte izquierda de la tabla.
6. Hacer click en el botón “Cargar” de la misma sección.

Figura 198. Pasos para cargar los datos de un periodo académico duplicado en el sistema.

Cod_Horari	Dia	Docente	Nombre_Ma	Alias	Nombre_Au	Paralelo
13	JUEVES	GUERRER...	ALGEBRA...	CIERCOM	AULA_201	A
14	VIERNES	GUERRER...	FÍSICA I	CIERCOM	AULA_202	A
15	LUNES	GUERRER...	GEOMETR...	CIERCOM	AULA_302	B

A continuación los datos seleccionados del horario duplicado se cargan en cada una de las casillas de la sección “Datos del Horario” y una vez cargados estos datos, el botón “Modificar” de la misma sección se habilita permitiendo realizar las modificaciones pertinentes de ese horario si así lo requiere.

- Una vez realizada las modificaciones en las casillas de la sección “*Datos del Horario*” se procede hacer click en el botón Modificar y finalmente esperar el mensaje de confirmación.

Figura 199. Modificación de la duplicación de un horario en un periodo académico en el sistema.

**HORARIO ASIGNADO SEGÚN EL DEPARTAMENTO DE INFORMÁTICA DE LA UTN**

**Opciones**  
 Nuevo  
 Modificar  
 Duplicar

**Duplicar Periodo Académico**  
 Nuevo: SEPTIEMBRE 2016 - FEBRERO 2017 [OK]  
 A Duplicar: MARZO 2016 - AGOSTO 2016

**Datos del Horario**  
 Día: MARTES Carrera: CIERCOM Aula: AULA\_205  
 Docente: GUERRERO ASEICHA ANTONIO Paralelo: A  
 Materia: ELECTRÓNICA Nivel: 1ro Nº Horas: 4

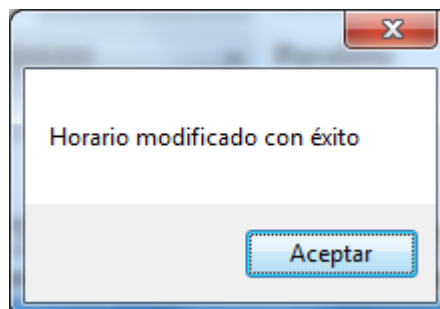
CREAR MODIFICAR **Paso 7** LIMPIAR

**Horarios del Periodo Académico SEPTIEMBRE 2016 - FEBRERO 2017**

Cod_Horari	Dia	Docente	Nombre_Ma	Alias	Nombre_Au	Paralelo
13	JUEVES	GUERRER...	ALGEBRA ...	CIERCOM	AULA_201	A
14	VIERNES	GUERRER...	FÍSICA I	CIERCOM	AULA_202	A
15	LUNES	GUERRER...	GEOMETR...	CIERCOM	AULA_302	B

Total de registros: 3 Seleccione un elemento de la tabla para cargar los datos CARGAR

Figura 200. Confirmación de la modificación del horario académico duplicado en el sistema.





#### 7.9.1.4.2.5 Modificar Datos Personales.

Este proceso se utiliza para modificar los datos del docente que se registra previamente, es decir, estos campos se actualizan solamente con los datos personales del administrador, de la misma forma que los otros procesos.

Para poder modificar los datos personales y de registro de un docente se siguen los siguientes pasos:

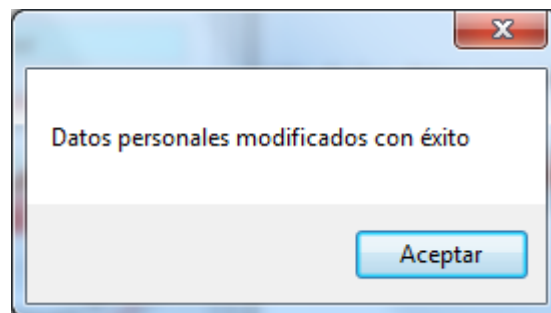
1. Seleccionar el campo requerido y modificar la información. Estos campos tienen la siguiente información:
  - ✓ Primer Nombre Docente: Se debe modificar solamente el primer nombre del docente.
  - ✓ Segundo Nombre Docente: Se debe modificar solamente el segundo nombre del docente.
  - ✓ Primer Apellido Docente: Se debe modificar el apellido paterno del docente.
  - ✓ Segundo Apellido Docente: Se debe modificar el apellido materno del docente.
  - ✓ Dirección: Se debe modificar la dirección del domicilio del docente.
  - ✓ Teléfono Celular Personal: Se debe modificar el número de teléfono personal del docente con el cual se le pueda localizar.
  - ✓ Correo: Se debe modificar el correo electrónico del docente proporcionado por la Universidad Técnica del Norte.
  - ✓ Nombre Usuario: Se debe modificar un nombre con el cual se deberá identificar en el formulario "Autenticación" para ingresar al sistema.

- ✓ Contraseña: Se debe modificar la contraseña con la cual se deberá identificar en el formulario “Autenticación” para ingresar al sistema.
2. Una vez modificados los datos, se procede hacer click en el botón “OK” y esperar el mensaje de confirmación.

Figura 201. Pasos para modificar los datos personales del administrador en el sistema.



Figura 202. Confirmación de la modificación de los datos personales del administrador.



*Nota:*

*El campo “C.I. Docente” de la sección “Datos del Docente” no puede ser modificado en ningún momento por el administrador por lo tanto no se habilita.*

*El campo “Rol Usuario” de la sección “Identificación de Registro” no se habilita debido a que este parámetro no puede ser modificado en esta sección. Para ello debe dirigirse a la “sección 1.4.2.2” y realizar los pasos mencionados en esa sección. En caso de ser modificado este parámetro, tomar en cuenta que para ingresar al sistema en “modo administrador” debe permanecer un usuario con este rol.*

#### *7.9.1.4.2.6 Utilizar Aula Asignada o Reservada.*

Este proceso permite utilizar un aula asignada por el horario o reservada por el docente el cual carga los datos provenientes del sistema verificando si el docente tiene clases a esa hora (en la que se registró) o no. Debido a esta eventualidad este proceso se presenta en tres etapas que son:

1. Cuando el docente tiene clases por medio de un horario.
2. Cuando el docente tiene clases por medio de una reserva.
3. Cuando el docente no tiene clases.

##### *7.9.1.4.2.6.1 Cuando el docente tiene clases por medio de un horario.*

El sistema calcula si en el horario de clases de ese periodo académico en curso, el docente autenticado tiene clases a esa hora, el formulario se presenta de la siguiente forma.

Figura 203. Formulario de clases por medio de un horario.

ADMINISTRADOR

SALIR REPORTES

Crear, Modificar Materia Aula o Carrera | Crear Horario | Modificar Datos Personales | Utilizar Aula Asignada o Reservada | Reservar Aula Disponible | Ver

### AULA ASIGNADA POR EL DISTRIBUTIVO DE AULAS

**Datos del Sistema**

Fecha: 02/06/2015 | Día: MARTES | Hora: 14:39:03

**Datos del Docente**

Cédula: 1003788401 | Docente: GUERRERO ASEICHA ANTONIO | Rol: Administrador

**Horario**

Materia: MATEMÁTICAS APLICADAS | Aula: AULA\_202

Periodo\_Académico: MARZO 2015 - AGOSTO 2015 | Nivel: 1ro | Utilizar\_Aula: OK

Carrera: CIERCOM | Paralelo: A

7.9.1.4.2.6.2 Cuando el docente tiene clases por medio de una reserva.

El sistema calcula si el docente ha reservado un aula para esa fecha y hora en el periodo académico en curso, el formulario se presenta de la siguiente forma.

Figura 204. Formulario de clases por medio de una reserva.

ADMINISTRADOR

SALIR REPORTES

Crear, Modificar Materia Aula o Carrera Crear Horario Modificar Datos Personales Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver

### AULA RESERVADA POR EL DOCENTE

**Datos del Sistema**

Fecha 02/06/2015 Día MARTES Hora 17:24:37

**Datos del Docente**

Cédula 1003788401 Docente GUERRERO ASEICHA ANTONIO Rol Administrador

**Reserva**

Materia ALGEBRA LINEAL Detalle PRUEBAS DE LA TESIS

Periodo\_Académico MARZO 2015 - AGOSTO 2015

Carrera CIERCOM

Fecha 02/06/2015 Nivel 2do

Aula AULA\_201 Paralelo A

Utilizar\_Aula OK

7.9.1.4.2.6.3 Cuando el docente no tiene clases.

En caso de no tener clases a esa hora el formulario se presenta de la siguiente forma.

Figura 205. Formulario de no tener clases.

ADMINISTRADOR

SALIR REPORTES

Crear, Modificar Materia Aula o Camera | Crear Horario | Modificar Datos Personales | Utilizar Aula Asignada o Reservada | Reservar Aula Disponible | Ver

### AULA NO ASIGNADA NI RESERVADA


**Datos del Sistema**

Fecha  Día  Hora

**Datos del Docente**

Cédula  Docente  Rol

**USTED NO TIENE CLASES A ESTA HORA...!!!**



Los pasos a seguir en caso de tener clases, ya sea por medio del horario o la reserva son los siguientes:

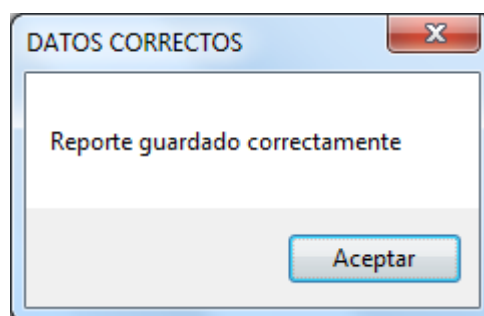
1. Hacer click en el botón "OK" y esperar el mensaje de confirmación.

Una vez realizada esta acción la trama de datos es enviada hacia el hardware de este sistema y guardados los datos de este proceso en el sistema para generar el reporte de la utilización del aula correspondiente. Posteriormente se visualiza el mensaje de confirmación de los datos guardados.

Figura 206. Pasos para utilizar el aula asignada o reservada para el docente.

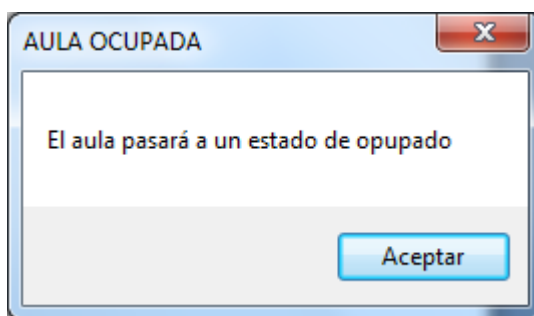


Figura 207. Confirmación de los datos guardados en el reporte del sistema



Una vez guardado el reporte y realizada la apertura de la puerta en el hardware, el sistema pone al aula correspondiente en un estado de “ocupado” por lo que se genera un mensaje informando este evento.

Figura 208. Confirmación del estado del aula actual.



*Nota:*

*La información indicada en esta sección no se modifica debido a que los datos son extraídos de la base de datos del sistema.*

*En este proceso, debido a que se comunica el sistema con el hardware creado para las puertas de la facultad, se envía una trama de datos durante un periodo de tiempo específico por lo cual puede tardar un poco la confirmación de la apertura de la puerta.*

#### *7.9.1.4.2.7 Reservar Aula Disponible.*

Este proceso le permite al docente realizar una reserva de un aula determinada en un tiempo establecido en un periodo diferente al creado en su horario de clases especificando el motivo de la misma.

Para utilizar este método se siguen los siguientes pasos:

1. Seleccionar el "Número de Aula" de la sección "*Datos del Aula*" la misma que contiene los siguientes campos:
  - ✓ Nombre del Aula: identifica el aula a realizar la reserva.
  - ✓ Ubicación: Identifica el lugar donde se encuentra físicamente esa aula.



- ✓ Capacidad: informa el número máximo de estudiantes que pueden ingresar en esa aula seleccionada.
2. Escribir el motivo por el cual se está realizando la reserva de esa aula en el campo “Detalle”.
  3. Identificar la fecha en la cual se desea utilizar el aula seleccionada.
  4. Identificar qué materia se va a impartir en esa aula.
  5. Identificar la carrera a la que pertenece esa materia seleccionada.
  6. Identificar el nivel al que pertenece esa materia seleccionada.
  7. Seleccionar el paralelo a la cual se le va a impartir la materia seleccionada.
  8. Especificar qué tiempo se va a utilizar el aula seleccionada.
  9. Indicar la hora en la que se va a iniciar a impartir clases en esa aula seleccionada.
  10. Finalmente, hacer click en el botón “OK” y esperar el mensaje de confirmación.

Figura 209. Pasos para realizar una reserva de un aula en el sistema.



Figura 210. Sugerencia para reservar el aula.

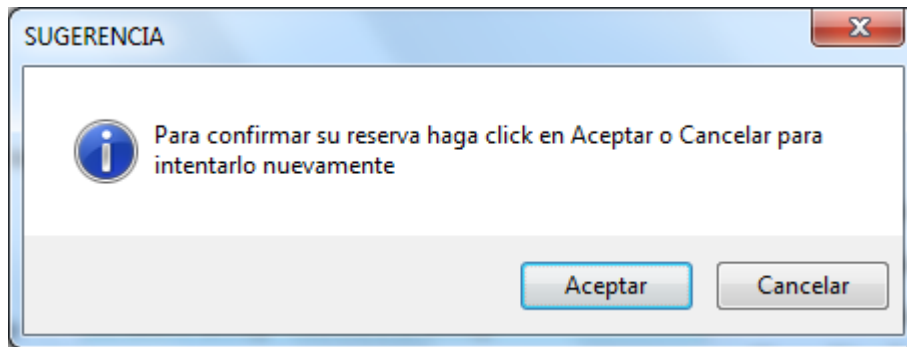
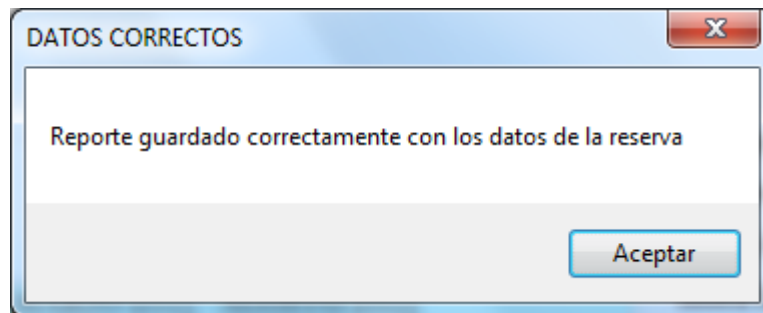


Figura 211. Confirmación de la reserva del aula realizada correctamente.



#### 7.9.1.4.2.8 Ver Aulas.

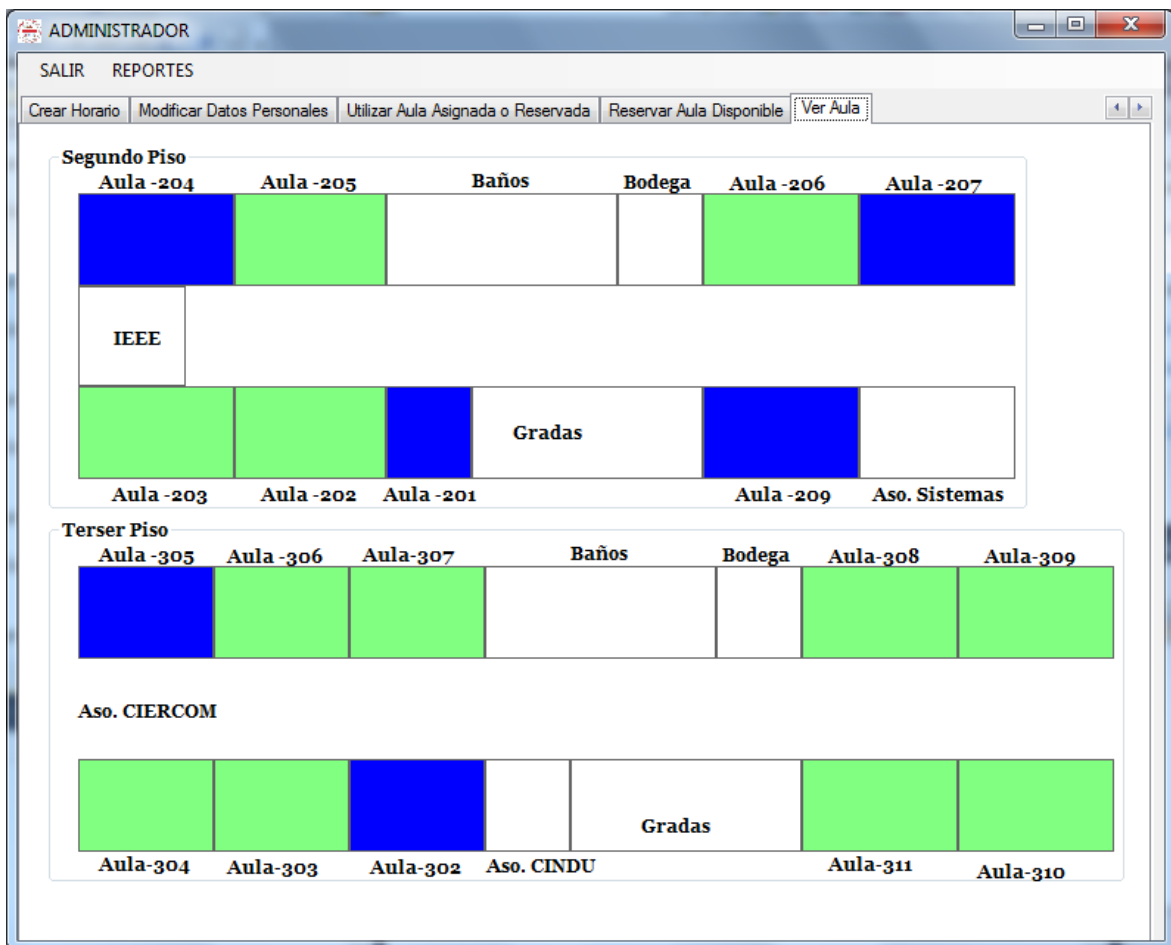
Este proceso permite visualizar el estado de las aulas de la facultad correspondiente a la disponibilidad de las mismas, es decir, es un parámetro que le permite al docente identificar de forma gráfica que aulas se encuentran disponibles u ocupadas.

Para identificar uno u otro estado se identifica con los siguientes colores:

- ✓ Color Azul: El aula se encuentra ocupada.
- ✓ Color Verde: El aula se encuentra disponible.

En la figura siguiente se visualiza el mapa de las aulas de la facultad, unas disponibles y otras ocupadas.

Figura 212. Identificación del estado de las aulas de la facultad.



#### 7.9.1.5 Recomendaciones:

1. Leer detalladamente este manual para evitar cometer errores en la ejecución del sistema.
2. La información proporcionada como una “**Nota**” que se incluye en las diferentes secciones de este manual es de mucha importancia, hay que tener en cuenta, ya que de esto depende el buen funcionamiento del sistema creado.
3. En caso de existir algún error al ingresar o seleccionar los datos, el sistema genera un mensaje de “**datos erróneos o sugerencia**” para evitar los posibles errores por lo tanto se debe tomar en cuenta dichos mensajes.
4. La utilización de este manual es precisamente para el administrador del sistema, por ende la información proporcionada en él es de carácter reservado.

## **7.9.2 MANUAL DE USUARIO**

### **7.9.2.1 INTRODUCCIÓN**

El sistema de control de acceso para la Facultad de Ingeniería en Ciencias Aplicadas es un software diseñado para mejorar el proceso de utilizar y controlar el ingreso a las diferentes aulas por parte de los docentes y funcionarios que laboran de dicha facultad.

En este manual se especifica el modo de ingreso al sistema denominado “*Usuario*” permitiendo dar a conocer los pasos a seguir en cada uno de los parámetros de forma general para la utilización de las aulas de la facultad, así como reservar un aula disponible.

### **7.9.2.2 FORMULARIOS DEL SISTEMA**

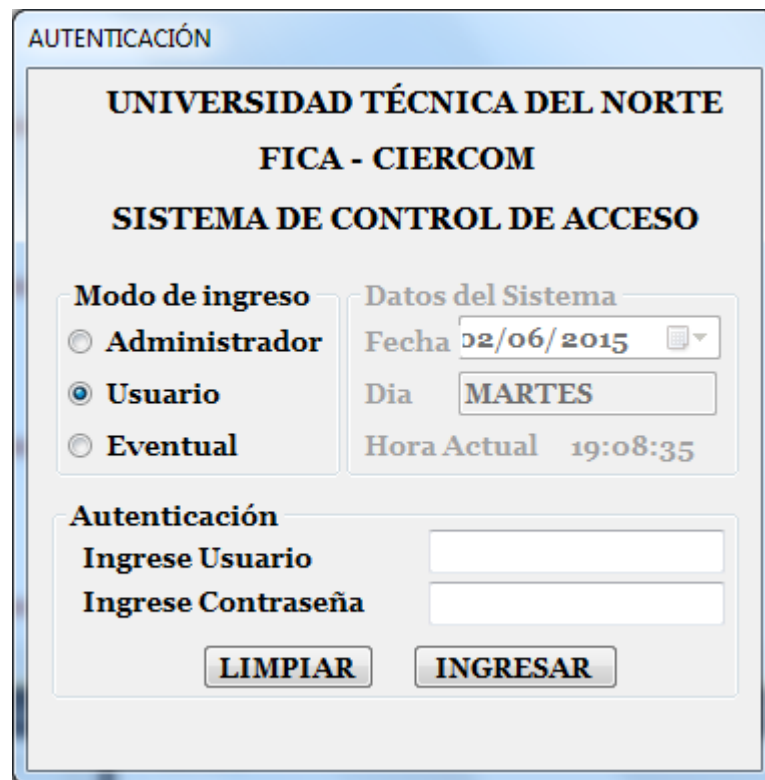
El sistema presenta tres formularios los cuales se dividen, dependiendo de la función que cumple, en:

- ✓ Formulario de autenticación.
- ✓ Formulario de procesos.

#### ***7.9.2.2.1 Formulario de autenticación***

Permite verificar si el docente o funcionario que utilice el sistema se encuentra registrado como usuario, además de recopilar información como el día, fecha y hora actual del sistema, esta interfaz se presenta en la figura siguiente:

Figura 213. Formulario de autenticación del usuario.



**AUTENTICACIÓN**

**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FICA - CIERCOM**  
**SISTEMA DE CONTROL DE ACCESO**

**Modo de ingreso**

Administrador  
 Usuario  
 Eventual

**Datos del Sistema**

Fecha 02/06/2015  
Día MARTES  
Hora Actual 19:08:35

**Autenticación**

Ingrese Usuario  
Ingrese Contraseña

LIMPIAR    INGRESAR

La interfaz de autenticación del usuario presenta tres secciones que son:

- A. Modo de Ingreso: Permite seleccionar el rol de usuario con el cual se autenticará el docente para ingresar al sistema.
- B. Datos del Sistema: Visualiza los datos provenientes de la computadora con la que se encuentre trabajando el sistema.
- C. Autenticación: Permite ingresar la información al usuario con la que se registrará en el sistema. Esta sección depende de la sección “Modo de Ingreso” debido a que contiene campos que se habilitan o deshabilitan según el rol que se escoja.

Figura 214. Identificación de las secciones en el formulario de autenticación del usuario.

The image shows a screenshot of a web application window titled "AUTENTICACIÓN". The header text reads "UNIVERSIDAD TÉCNICA DEL NORTE", "FICA - CIERCOM", and "SISTEMA DE CONTROL DE ACCESO". The form is divided into three sections highlighted with red boxes and labeled with arrows:

- Sección A:** "Modo de ingreso" with radio buttons for "Administrador", "Usuario" (selected), and "Eventual".
- Sección B:** "Datos del Sistema" with fields for "Fecha" (02/06/2015), "Dia" (MARTES), and "Hora Actual" (19:08:35).
- Sección C:** "Autenticación" with input fields for "Ingreso Usuario" and "Ingreso Contraseña", and buttons for "LIMPIAR" and "INGRESAR".

*Nota:*

*La Sección C tiene su propio control que realiza una tarea determinada, las mismas que se explican en el proceso de utilización más adelante.*

#### **7.9.2.2 Formulario de procesos**

En este formulario se presentan ciertos campos a seleccionar dependiendo de las prioridades del usuario, al cual le permite modificar sus datos personales, utilizar o reservar un aula al docente y visualizar las aulas de la facultad. Estas prioridades se encuentran especificadas más detalladamente en el trabajo de titulación de este proyecto. A continuación se presenta la interfaz de este formulario en la siguiente figura.

Figura 215. Formulario de procesos del usuario.

USUARIO

SALIR

Modificar Datos Personales Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

### MODIFICAR DATOS PERSONALES Y DE REGISTRO DEL SISTEMA

**Datos Personales**

C.I. Docente: 1003098215

Primer\_Nombre\_Docente: KATHERINE

Segundo\_Nombre\_Docente: JANETH

Primer\_Apellido\_Docente: BENAVIDES

Segundo\_Apellido\_Docente: CHAMORRO

Dirección: PANAMERICANA NORTE

Teléfono\_Celular\_Principal: 0987846542

Correo: kjbenavidesc@utn.edu.ec

**Identificación de Registro**

Nombre\_Usuario: kat

Contraseña\_Usuario: kat57

Rol\_Usuario: Usuario

Control: OK

CIERCOM  
INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

UNIVERSIDAD TÉCNICA DEL NORTE  
IBARRA - ECUADOR

Facultad de Ingeniería en Ciencias Aplicadas

Además, esta interfaz presenta dos menús, uno principal y otro selectivo, los mismos que cuentan con los siguientes campos habilitados:

- ✓ **Menú Principal:** Permite abrir o cerrar otra interfaz dependiendo de la función, estas funciones pueden ser:
  - *Salir:* Cierra la interfaz actual del administrador y abre la interfaz de autenticación.
- ✓ **Menú Selectivo:** Permite moverse dentro de la misma interfaz, este campo tiene submenús los cuales realizan diferentes funciones tales como:
  - *Modificar Datos Personales:* Permite realizar modificaciones en los datos personales del docente.



- *Utilizar Aula Asignada o Reservada:* Este campo permite utilizar un aula que se encuentre ya sea reservada con anticipación o se encuentre en el horario de clases del docente.
- *Reservar Aula Disponible:* Permite reservar una aula de clases que se encuentre disponible por un periodo determinado.
- *Ver Aulas:* Visualiza las aulas que se encuentran utilizadas y disponibles en la facultad.

Figura 216. Identificación de los menús en el formulario de procesos del usuario.



### 7.9.2.2.2.1 Secciones del menú selectivo

En la interfaz de proceso, cada uno de los elementos del “menús selectivos”, presentan secciones diferentes que se acoplan a cada función que cumplen, las mismas que se detalla a continuación:

#### 7.9.2.2.2.1.1 Modificar Datos Personales.

- A. Datos Personales: Permite modificar los datos personales de un docente o funcionario que se haya registrado previamente en el sistema.
- B. Identificación de Registro: Visualiza los datos del docente o funcionario con los cuales se registró inicialmente.
- C. Control: Permite guardar la información que se haya modificado en este proceso.

Figura 217. Identificación de las secciones en el formulario de procesos del usuario del submenú modificar datos personales.

The screenshot shows a web application window titled "USUARIO" with a menu bar containing "SALIR", "Modificar Datos Personales", "Utilizar Aula Asignada o Reservada", "Reservar Aula Disponible", and "Ver Aula". The main content area is titled "MODIFICAR DATOS PERSONALES Y DE REGISTRO DEL SISTEMA".

**Sección A: Datos Personales**

C.I._Docente	1003098215
Primer_Nombre_Docente	KATHERINE
Segundo_Nombre_Docente	JANETH
Primer_Apellido_Docente	BENAVIDES
Segundo_Apellido_Docente	CHAMORRO
Dirección	PANAMERICANA NORTE BARRIO PR
Teléfono_Celular_Principal	0987846542
Correo	kbenavidesc@utn.edu.ec

**Sección B: Identificación de Registro**

Nombre_Usuario	Kat
Contraseña_Usuario	kat57
Rol_Usuario	Usuario

**Sección C: Control**

Ok

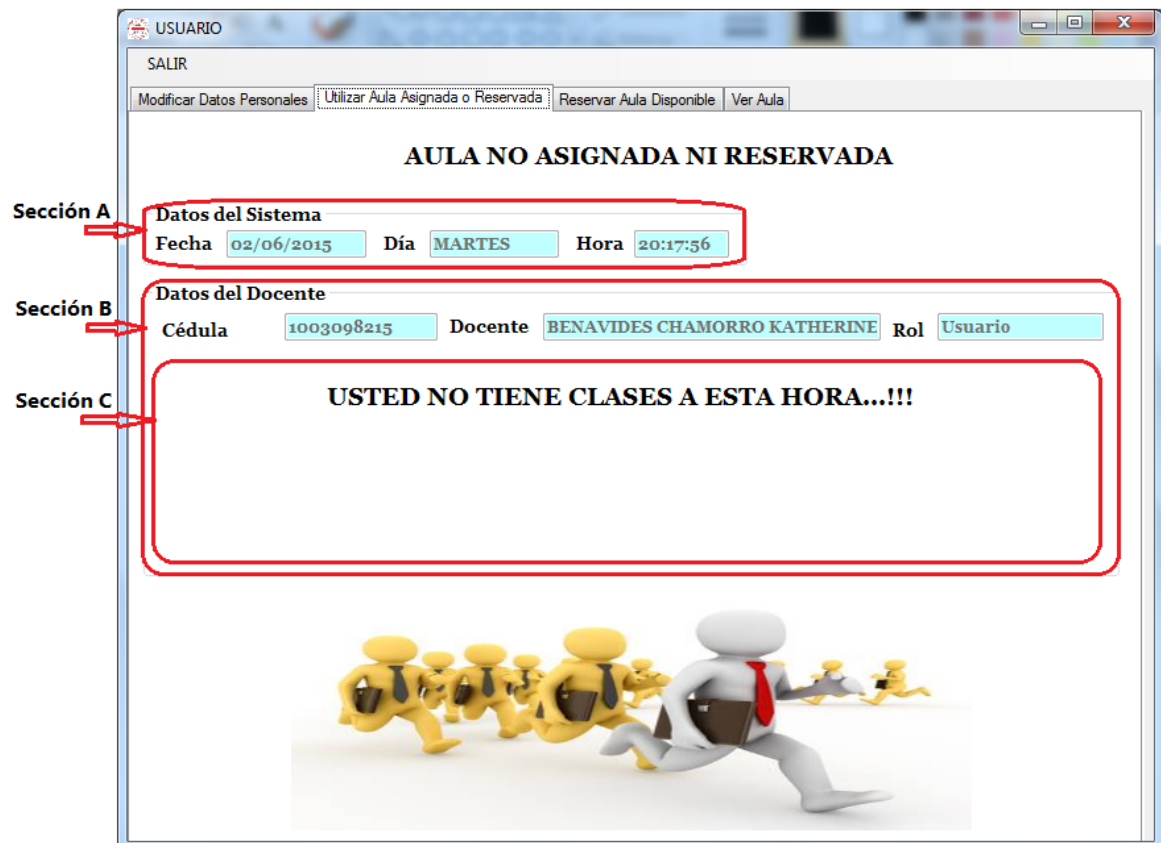
Logos: CIERCOM (Instituto de Investigación y Desarrollo en Ingeniería y Ciencias de la Comunicación) and Universidad Técnica del Norte (UTN) - BARBA - ECUADOR.

7.9.2.2.1.2 Utilizar Aula Asignada o Reservada.

- A. Datos del Sistema: Visualiza los datos provenientes de la computadora con la que se encuentre trabajando el sistema.
- B. Datos del Docente: Visualiza los datos del docente con los cuales se registró en el sistema.
- C. Horario/Reserva: Permite visualizar la información correspondiente al horario o a la reserva del aula a utilizar.

Esta sección depende del horario de clases del docente o de una reserva realizada con anticipación para poder visualizar los datos del aula para utilizar, en caso de no existir estos parámetros se visualiza un mensaje informando al docente, que no tiene clases a esa hora.

Figura 218. Identificación de las secciones en el formulario de procesos del usuario del submenú utilizar aula asignada o reservada.



*Nota:*

*En caso de no tener clases el docente, en ese momento no se carga los datos y se visualiza un mensaje informándole que no tiene clases a esa hora.*

*La sección C tiene su propio control que realiza una tarea determinada la misma que se explica en el proceso de utilización más adelante.*

#### *7.9.2.2.1.3 Reservar Aula Disponible.*

- A. Datos del Aula: Permite seleccionar una aula disponible a reservar por el docente.
- B. Detalle: Permite ingresar información del motivo por el cual el aula seleccionada en la sección A se va a utilizar.
- C. Parámetros de Reserva: Permite seleccionar parámetros de la materia que se va impartir en una fecha determinada.
- D. Datos Docente: Visualiza los datos del docente con los cuales se registró en el sistema.
- E. Tiempo: Permite seleccionar el tiempo en el cual el aula será utilizada por el docente.
- F. Controles: Permite guardar o borrar la información que se encuentre ingresando en este proceso.

Figura 219. Identificación de las secciones en el formulario de procesos del usuario del submenú reservar aula disponible.

The screenshot shows a web application window titled "USUARIO" with a sub-menu "UTILIZAR AULA DISPONIBLE". The window contains several sections labeled A through F:

- Sección A:** Datos del Aula (Número de Aula, Ubicación, Capacidad de Estudiantes) and Datos Docente (C.I.\_Docente, Docente, Rol\_Usuario).
- Sección B:** Detalle: MOTIVO DE LA RESERVA:
- Sección C:** Parámetros de Reserva (Día, Fecha de Reserva, Período Académico, Materia, Carrera, Nivel, Paralelo).
- Sección E:** Tiempo (Número de Horas, Hora Actual, Hora Inicial, Hora Final).
- Sección F:** Controles (LIMPIAR, OK).

At the bottom of the window, there are two images of classrooms.

#### 7.9.2.2.2.1.4 Ver Aula.

- A. Segundo Piso: Identifica a todas las aulas de la facultad que se encuentran en el segundo piso de la facultad.
- B. Tercer Piso: Identifica a todas las aulas de la facultad que se encuentran en el tercer piso de la facultad.

Figura 220. Identificación de las secciones en el formulario de procesos del usuario del submenú ver aula.



Para identificar si un aula se encuentra ocupada o disponible se visualiza dependiendo de los siguientes colores:

- ✓ Color Azul: El aula se encuentra Ocupada.
- ✓ Color Verde: El aula se encuentra disponible.

Nota:

En lo referente al menú selectivo, para el usuario, no todos los campos son habilitados, esto depende del modo de ingreso con el cual se registra el docente o funcionario ya que cada campo realiza una función determinada, por ende solamente se habilitan los campos en los cuales se tiene acceso a modificar o utilizar.

### **7.9.2.3 Proceso de utilización del sistema del usuario.**

A continuación se especifica todos los pasos que debe seguir el docente que cumple las funciones de usuario en el sistema de control de acceso de las aulas de la facultad, de igual forma se presenta en tres etapas según los formularios que el sistema contiene.

#### **7.9.2.3.1 Pasos a seguir en el formulario de autenticación.**

Para poder ingresar al sistema mediante el modo usuario es necesario seguir los siguientes pasos:

1. Seleccionar en la sección "*Modo de ingreso*" la casilla que identifica este modo.
2. Una vez seleccionado el modo de ingreso se habilitan los campos de texto en la sección "*Autenticación*" en los cuales se debe proceder a ingresar la información que los mismos requieren con sus respectivos datos del docente:
  - ✓ *Ingrese Usuario*: Se debe escribir el nombre con el cual esté registrado en el sistema.
  - ✓ *Ingrese Contraseña*: Se debe escribir la contraseña con la cual esté registrado en el sistema.

De igual forma se habilitan los botones con los cuales se puede:

- ✓ *Limpiar*: Borra el texto introducido en los campos usuario, contraseña y la confirmación de la contraseña.
- ✓ *Ingresar*: En caso de que los datos se encuentren correctamente ingresados en los anteriores campos, el usuario puede acceder al

sistema en el cual se abre una nueva ventana, caso contrario el sistema deniega el ingreso, mostrando previamente el error suscitado.

3. Finalmente se hace click en el botón “Ingresar” y esperar el mensaje de confirmación.

Figura 221. Pasos a seguir en el formulario autenticación del usuario.

**AUTENTICACIÓN**

**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FICA - CIERCOM**  
**SISTEMA DE CONTROL DE ACCESO**

**Modo de ingreso**

Administrador  
 **Usuario**  
 Eventual

**Datos del Sistema**

Fecha 02/06/2015  
Dia MARTES  
Hora Actual 20:46:20

**Autenticación**

Ingrese Usuario   
Ingrese Contraseña

LIMPIAR

**Paso 1** **Paso 2** **Paso 3**

Figura 222. Mensaje de confirmación del ingreso al sistema del usuario.

Autenticacion Correcta. Bienvenido al Modo Usuario

Aceptar

*Nota:*

*En caso de existir algún error en los datos ingresados por el usuario, el sistema informará mediante un mensaje el motivo por el cual no se pudo dar acceso al sistema.*

#### **7.9.2.3.2 Pasos a seguir en el formulario procesos.**

A continuación se profundizan las funciones y el uso de cada uno de los submenús descritos anteriormente en el formulario de procesos según los parámetros correspondientes al usuario.

##### **7.9.2.3.2.1 Modificar Datos Personales.**

Este proceso se utiliza para modificar los datos del docente que se registra previamente, es decir, estos campos se actualizan solamente con los datos personales del usuario, de la misma forma que los otros procesos.

Para poder modificar los datos personales y de registro del docente se siguen los siguientes pasos:

1. Seleccionar el campo requerido y modificar la información. Estos campos tienen la siguiente información:
  - ✓ Primer Nombre Docente: Se debe modificar solamente el primer nombre del docente.
  - ✓ Segundo Nombre Docente: Se debe modificar solamente el segundo nombre del docente.
  - ✓ Primer Apellido Docente: Se debe modificar el apellido paterno del docente.
  - ✓ Segundo Apellido Docente: Se debe modificar el apellido materno del docente.



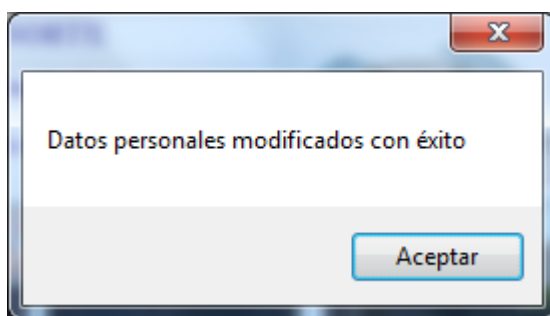
- ✓ Dirección: Se debe modificar la dirección del domicilio del docente.
- ✓ Teléfono Celular Personal: Se debe modificar el número de teléfono personal del docente con el cual se le pueda localizar.
- ✓ Correo: Se debe modificar el correo electrónico del docente proporcionado por la Universidad Técnica del Norte.
- ✓ Nombre Usuario: Se debe modificar un nombre con el cual se deberá identificar en el formulario “Autenticación” para ingresar al sistema.
- ✓ Contraseña: Se debe modificar la contraseña con la cual se deberá identificar en el formulario “Autenticación” para ingresar al sistema.

2. Una vez modificados los datos, se procede hacer click en el botón “OK” y esperar el mensaje de confirmación.

Figura 223. Pasos para modificar los datos personales del usuario en el sistema.

The screenshot shows a web browser window with the title 'USUARIO'. The main content area is titled 'MODIFICAR DATOS PERSONALES Y DE REGISTRO DEL SISTEMA'. The form is organized into two columns. The left column, labeled 'Datos Personales', contains the following fields: C.I.\_Docente (1003098215), Primer\_Nombre\_Docente (KATHERINE), Segundo\_Nombre\_Docente (JANETH), Primer\_Apellido\_Docente (BENAVIDES), Segundo\_Apellido\_Docente (CHAMORRO), Dirección (LA ADUANA), Teléfono\_Celular\_Principal (0987846542), and Correo (kbenavidesc@utn.edu.ec). The right column, labeled 'Identificación de Registro', contains: Nombre\_Usuario (Kat), Contraseña\_Usuario (kat57), and Rol\_Usuario (Usuario). Below these fields is an 'Ok' button. A yellow box highlights the 'Datos Personales' section, and another yellow box highlights the 'Identificación de Registro' section. A yellow arrow points from the 'Ok' button to the text 'Paso 2'. At the bottom of the window, there is a banner for 'Facultad de Ingeniería en Ciencias Aplicadas' and logos for 'CIERCOM' and 'UNIVERSIDAD TÉCNICA DEL NORTE'.

Figura 224. Confirmación de la modificación de los datos personales del usuario.



*Nota:*

*El campo “C.I. Docente” y “Rol usuario” de las secciones “Datos del Docente” y “Identificación de Registro” respectivamente no puede ser modificados en ningún momento por el usuario por lo tanto no se habilita.*

#### *7.9.2.3.2 Utilizar Aula Asignada o Reservada.*

Este proceso permite utilizar un aula asignada por el horario o reservada por el docente, el cual carga los datos provenientes del sistema verificando si el docente tiene clases a esa hora (en la que se registró) o no. Debido a esta eventualidad este proceso se presenta en tres etapas que son:

1. Cuando el docente tiene clases por medio de un horario.
2. Cuando el docente tiene clases por medio de una reserva.
3. Cuando el docente no tiene clases.

##### *7.9.2.3.2.1 Cuando el docente tiene clases por medio de un horario.*

El sistema calcula si en el horario de clases de ese periodo académico en curso el docente autenticado tiene clases a esa hora, el formulario se presenta de la siguiente forma.

Figura 225. Formulario de clases por medio de un horario.

USUARIO

SALIR

Modificar Datos Personales Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

### AULA ASIGNADA POR EL DISTRIBUTIVO DE AULAS

**Datos del Sistema**

Fecha 02/06/2015 Día MARTES Hora 09:51:37

**Datos del Docente**

Cédula 1003098215 Docente BENAVIDES CHAMORRO KATHERINE Rol Usuario

**Horario**

Materia ALGEBRA LINEAL Aula AULA\_201

Periodo\_Académico MARZO 2015 - AGOSTO 2015 Nivel 1ro Utilizar\_Aula OK

Carrera CIERCOM Paralelo B

7.9.2.3.2.2 Cuando el docente tiene clases por medio de una reserva.

El sistema calcula si el docente ha reservado un aula para esa fecha y hora en el periodo académico en curso, el formulario se presenta de la siguiente forma.

Figura 226. Formulario de clases por medio de una reserva.

USUARIO

SALIR

Modificar Datos Personales Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

### AULA RESERVADA POR EL DOCENTE

**Datos del Sistema**

Fecha 02/06/2015 Día MARTES Hora 14:57:40

**Datos del Docente**

Cédula 1003098215 Docente BENAVIDES CHAMORRO KATHERINE Rol Usuario

**Reserva**

Materia MATEMÁTICAS APLICADAS Detalle PRUEBAS DE TESIS USUARIO

Periodo\_Académico MARZO 2015 - AGOSTO 2015

Carrera CIERCOM

Fecha 02/06/2015 Nivel 4to

Aula AULA\_202 Paralelo B

Utilizar\_Aula OK

7.9.2.3.2.2.3 Cuando el docente no tiene clases.

En caso de no tener clases a esa hora el formulario se presenta de la siguiente forma.

Figura 227. Formulario de no tener clases.

USUARIO

SALIR

Modificar Datos Personales Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

**AULA NO ASIGNADA NI RESERVADA**

**Datos del Sistema**

Fecha 02/06/2015 Día MARTES Hora 17:58:37

**Datos del Docente**

Cédula 1003098215 Docente BENAVIDES CHAMORRO KATHERINE Rol Usuario

**USTED NO TIENE CLASES A ESTA HORA...!!!**



Los pasos a seguir en caso de tener clases, ya sea por medio del horario o la reserva son los siguientes:

1. Hacer click en el botón "OK" y esperar el mensaje de confirmación.

Una vez realizada esta acción la trama de datos es enviada hacia el hardware de este sistema, y guardados los datos de este proceso en el sistema para generar el reporte de la utilización del aula correspondiente. Posteriormente se visualiza el mensaje de confirmación de los datos guardados.

Figura 228. Pasos para utilizar el aula asignada o reservada para el docente.

USUARIO

SALIR

Modificar Datos Personales Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

### AULA ASIGNADA POR EL DISTRIBUTIVO DE AULAS

**Datos del Sistema**

Fecha 02/06/2015 Día MARTES Hora 10:00:12

**Datos del Docente**

Cédula 1003098215 Docente BENAVIDES CHAMORRO KATHERINE Rol Usuario

**Horario**

Materia ALGEBRA LINEAL Aula AULA\_201

Periodo\_Académico MARZO 2015 - AGOSTO 2015 Nivel 1ro Utilizar\_Aula OK

Carrera CIERCOM Paralelo B

**Paso 1**

Figura 229. Confirmación de los datos guardados en el reporte del sistema

DATOS CORRECTOS

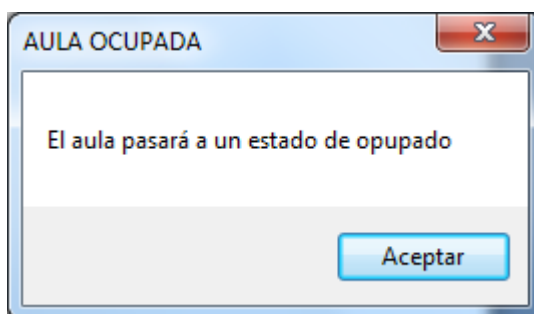
Reporte guardado correctamente

Aceptar

Una vez guardado el reporte y realizada la apertura de la puerta en el hardware, el sistema pone al aula correspondiente en un estado de “Ocupado” por lo que se genera un mensaje informando este evento.



Figura 230. Confirmación del estado del aula actual.



*Nota:*

*La información indicada en esta sección no se modifica debido a que los datos son extraídos de la base de datos del sistema.*

*En este proceso, debido a que se comunica el sistema con el hardware creado para las puertas de la facultad, se envía una trama de datos durante un periodo de tiempo específico, por lo cual puede tardar un poco la confirmación de la apertura de la puerta.*

#### *7.9.2.3.2.3 Reservar Aula Disponible.*

Este proceso le permite al docente realizar una reserva de un aula determinada en un tiempo establecido en un periodo diferente al creado en su horario de clases especificando el motivo de la misma.

Para utilizar este método se siguen los siguientes pasos:

1. Seleccionar el "Número de Aula" de la sección "*Datos del Aula*" la misma que contiene los siguientes campos:
  - ✓ Nombre del Aula: identifica el aula a realizar la reserva.
  - ✓ Ubicación: Identifica el lugar donde se encuentra físicamente esa aula.
  - ✓ Capacidad: informa el número máximo de estudiantes que pueden ingresar en esa aula seleccionada.

2. Escribir el motivo por el cual se está realizando la reserva de esa aula en el campo "Detalle".
3. Identificar la fecha en la cual se desea utilizar el aula seleccionada.
4. Identificar qué materia se va a impartir en esa aula.
5. Identificar la carrera a la que pertenece esa materia seleccionada.
6. Identificar el nivel al que pertenece esa materia seleccionada.
7. Seleccionar el paralelo a la cual se le va a impartir la materia seleccionada.
8. Especificar qué tiempo se va a utilizar el aula seleccionada.
9. Indicar la hora en la que se va a iniciar a impartir clases en esa aula seleccionada.
10. Finalmente, hacer click en el botón "OK" y esperar el mensaje de confirmación.

Figura 231. Pasos para realizar una reserva de un aula en el sistema.

**USUARIO**

SALIR

Modificar Datos Personales Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

**Paso 1 UTILIZAR AULA DISPONIBLE**

**Datos del Aula**

Número de Aula: AULA\_202

Ubicación: Segundo Piso

Capacidad de Estudiantes: 20

**Datos Docente**

C.I.\_Docente: 1003098215

Docente: BENAVIDES CHAMORRO KATHERIN

Rol\_Usuario: Usuario

**Detalle:**

PRUEBAS DE TESIS USUARIO

**Parámetros de Reserva**

Día: MARTES

Fecha de Reserva: Paso 3 02/06/2015

Periodo Académico: Paso 4 MARZO 2015 - AGOSTO 2015

Materia: Paso 5 MATEMÁTICAS APLICADAS

Carrera: Paso 6 CIERCOM Nivel: 4to Paralelo: B

**Tiempo**

Número de Horas: Paso 8 3

Hora Actual: 09:54:02

Hora Inicial: Paso 9 10:00:00

Hora Final: 13:00:00

**Controles**

LIMPIAR OK Paso 10

Paso 5 Paso 6 Paso 7



Figura 232.Sugerencia para reservar el aula.

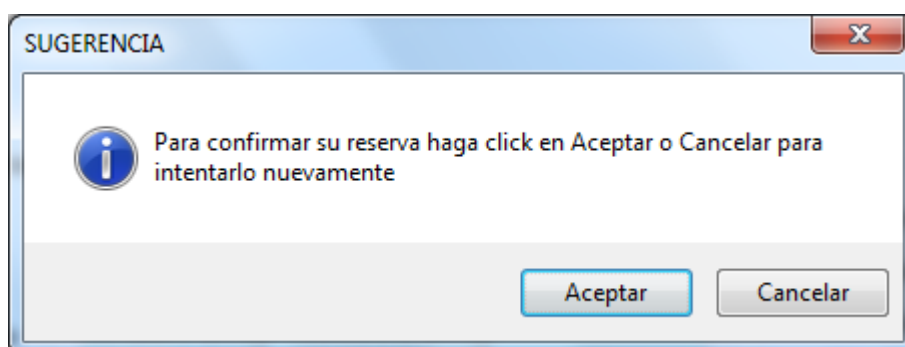
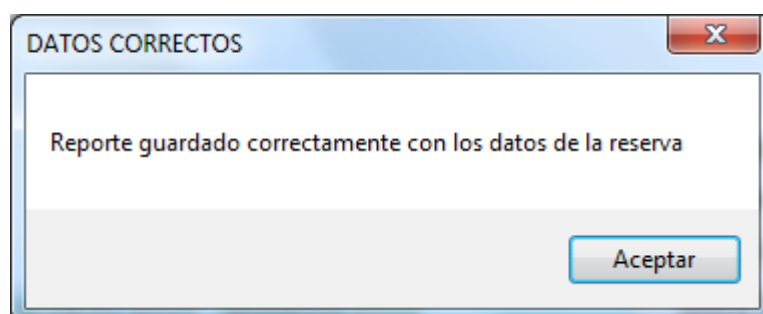


Figura 233. Confirmación de la reserva del aula realizada correctamente.



#### 7.9.2.3.2.4 Ver Aulas.

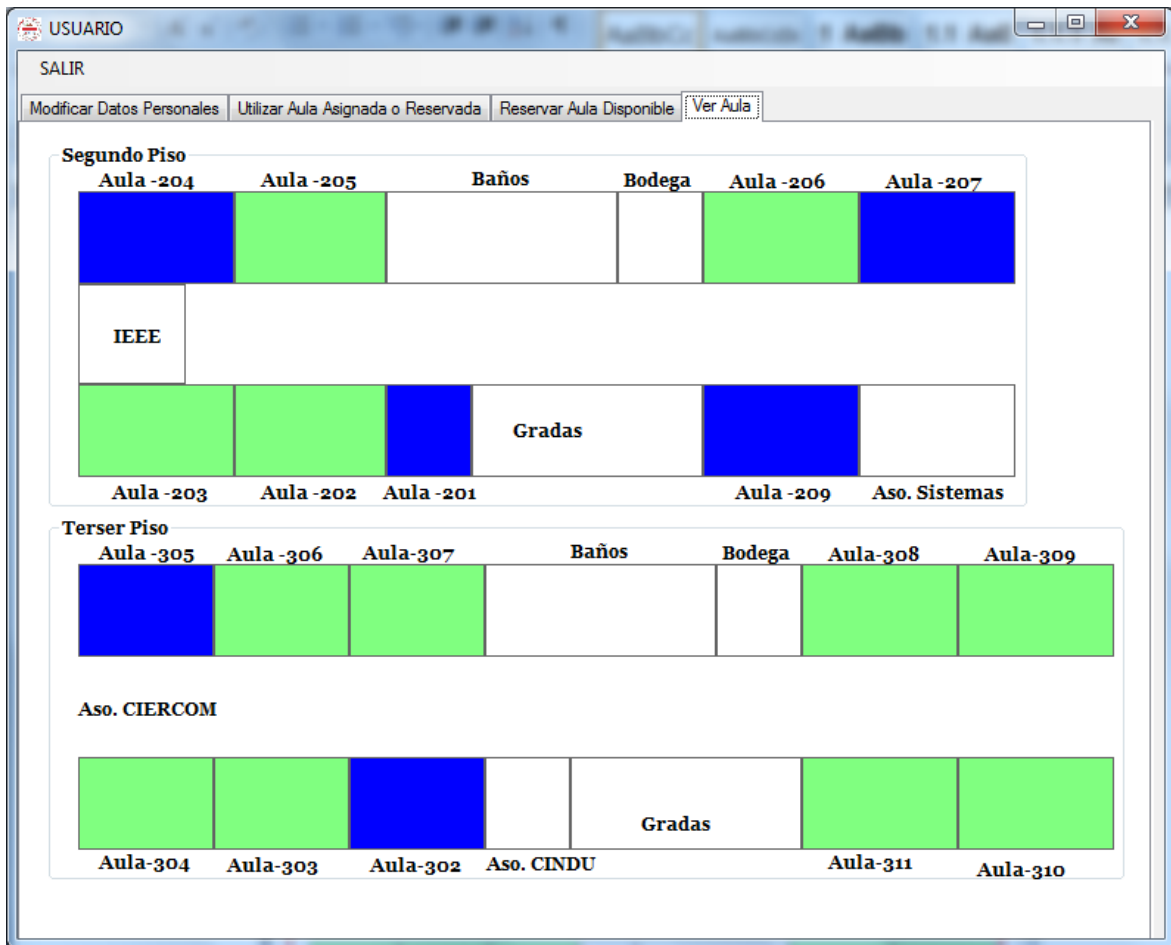
Este proceso permite visualizar el estado de las aulas de la facultad correspondiente a la disponibilidad de las mismas, es decir, es un parámetro que le permite al docente identificar de forma gráfica que aulas se encuentran disponibles u ocupadas.

Para identificar uno u otro estado se identifica con los siguientes colores:

- ✓ Color Azul: El aula se encuentra ocupada.
- ✓ Color Verde: El aula se encuentra disponible.

En la figura siguiente se visualiza el mapa de las aulas de la facultad unas disponibles y otras ocupadas.

Figura 234. Identificación del estado de las aulas de la facultad.



#### 7.9.2.4 Recomendaciones:

1. Leer detalladamente este manual para evitar cometer errores en la ejecución del sistema.
2. La información proporcionada como una “**Nota**” que se incluye en las diferentes secciones de este manual es de mucha importancia, hay que tener en cuenta, ya que de esto depende el buen funcionamiento del sistema creado.
3. En el caso de existir algún error al ingresar o seleccionar los datos, el sistema genera un mensaje de “**datos erróneos o sugerencia**” para evitar los posibles errores por lo tanto se debe tomar en cuenta dichos mensajes.
4. La utilización de este manual es precisamente para el usuario del sistema por ende la información proporcionada en él es de carácter reservado.

## **7.9.3 MANUAL DE USUARIO EVENTUAL**

### **7.9.3.1 INTRODUCCIÓN**

El sistema de control de acceso para la Facultad de Ingeniería en Ciencias Aplicadas es un software diseñado para mejorar el proceso de utilizar y controlar el ingreso a las diferentes aulas por parte de los docentes y funcionarios que laboran de dicha facultad.

En este manual se especifica el modo de ingreso al sistema denominado “*Eventual*” permitiendo dar a conocer los pasos a seguir en cada uno de los parámetros de forma general para la utilización de las aulas de la facultad, así como reservar un aula disponible.

### **7.9.3.2 FORMULARIOS DEL SISTEMA**

El sistema presenta tres formularios los cuales se dividen, dependiendo de la función que cumple, en:

- ✓ Formulario de autenticación.
- ✓ Formulario de procesos.
- ✓ Formulario de reportes.

#### ***7.9.3.2.1 Formulario de autenticación***

Permite verificar si el docente o funcionario que utilice el sistema se encuentra registrado como eventual, además de recopilar información como el día, fecha y hora actual del sistema, esta interfaz se presenta en la figura siguiente:

Figura 235. Formulario de autenticación del docente eventual.

The image shows a web-based authentication form titled "AUTENTICACIÓN". The form is for the "UNIVERSIDAD TÉCNICA DEL NORTE" and "FICA - CIERCOM" system, specifically for the "SISTEMA DE CONTROL DE ACCESO". It is divided into three main sections: "Modo de ingreso" (Login Mode), "Datos del Sistema" (System Data), and "Autenticación" (Authentication). In the "Modo de ingreso" section, there are three radio buttons: "Administrador", "Usuario", and "Eventual", with "Eventual" selected. The "Datos del Sistema" section displays system information: "Fecha" (Date) as 24/07/2015, "Dia" (Day) as VIERNES, and "Hora Actual" (Current Time) as 15:38:04. The "Autenticación" section has a label "Ingrese C.I." (Enter C.I.) next to a text input field, and a button labeled "INGRESAR" (Log In) below it.

La interfaz de autenticación del docente eventual presenta tres secciones que son:

- A. Modo de Ingreso: Permite seleccionar el rol de usuario con el cual se autenticará el docente para ingresar al sistema.
- B. Datos del Sistema: Visualiza los datos provenientes de la computadora con la que se encuentre trabajando el sistema.
- C. Autenticación: Permite ingresar la información al usuario con la que se registrará en el sistema. Esta sección depende de la sección "*Modo de Ingreso*" debido a que contiene campos que se habilitan o deshabilitan según el rol que se escoja.

Figura 236. Identificación de las secciones en el formulario de autenticación del docente eventual.

The image shows a web form titled 'AUTENTICACIÓN' for 'UNIVERSIDAD TÉCNICA DEL NORTE FICA - CIERCOM SISTEMA DE CONTROL DE ACCESO'. The form is divided into three sections marked with red boxes and arrows:

- Sección A:** 'Modo de ingreso' with radio buttons for 'Administrador', 'Usuario', and 'Eventual' (selected).
- Sección B:** 'Datos del Sistema' with a date field 'Fecha' (24/07/2015), a day field 'Dia' (VIERNES), and a time field 'Hora Actual' (15:38:04).
- Sección C:** 'Autenticación' with a text input field 'Ingrese C.I.' and an 'INGRESAR' button.

*Nota:*

*La Sección C tiene su propio control que realiza una tarea determinada, las mismas que se explican en el proceso de utilización más adelante.*

#### **7.9.3.2.2 Formulario de procesos**

En este formulario se presentan ciertos campos a seleccionar dependiendo de las prioridades del usuario, al cual le permite utilizar o reservar un aula al docente y visualizar las aulas de la facultad. Estas prioridades se encuentran especificadas más detalladamente en el trabajo de titulación de este proyecto. A continuación se presenta la interfaz de este formulario en la siguiente figura.

Figura 237. Formulario de procesos del docente eventual.

EVENTUAL

SALIR

Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

**AULA NO ASIGNADA NI RESERVADA**

**Datos del Sistema**

Fecha 24/07/2015 Día VIERNES Hora 15:58:26

**Datos del Docente**

Cédula 1003866843 Docente NEJER HARO XAVIER Rol Eventual

**USTED NO TIENE CLASES A ESTA HORA...!!!**

Además, esta interfaz presenta dos menús, uno principal y otro selectivo, los mismos que cuentan con los siguientes campos habilitados:

- ✓ **Menú Principal:** Permite abrir o cerrar otra interfaz dependiendo de la función, estas funciones pueden ser:
  - *Salir:* Cierra la interfaz actual del administrador y abre la interfaz de autenticación.
- ✓ **Menú Selectivo:** Permite moverse dentro de la misma interfaz, este campo tiene submenús los cuales realizan diferentes funciones tales como:
  - *Utilizar Aula Asignada o Reservada:* Este campo permite utilizar un aula que se encuentre reservada, con anticipación por el docente.

- *Reservar Aula Disponible*: Permite reservar una aula de clases que se encuentre disponible por un periodo determinado.
- *Ver Aulas*: Visualiza las aulas que se encuentran utilizadas y disponibles en la facultad.

Figura 238. Identificación de los menús en el formulario de procesos del docente eventual.



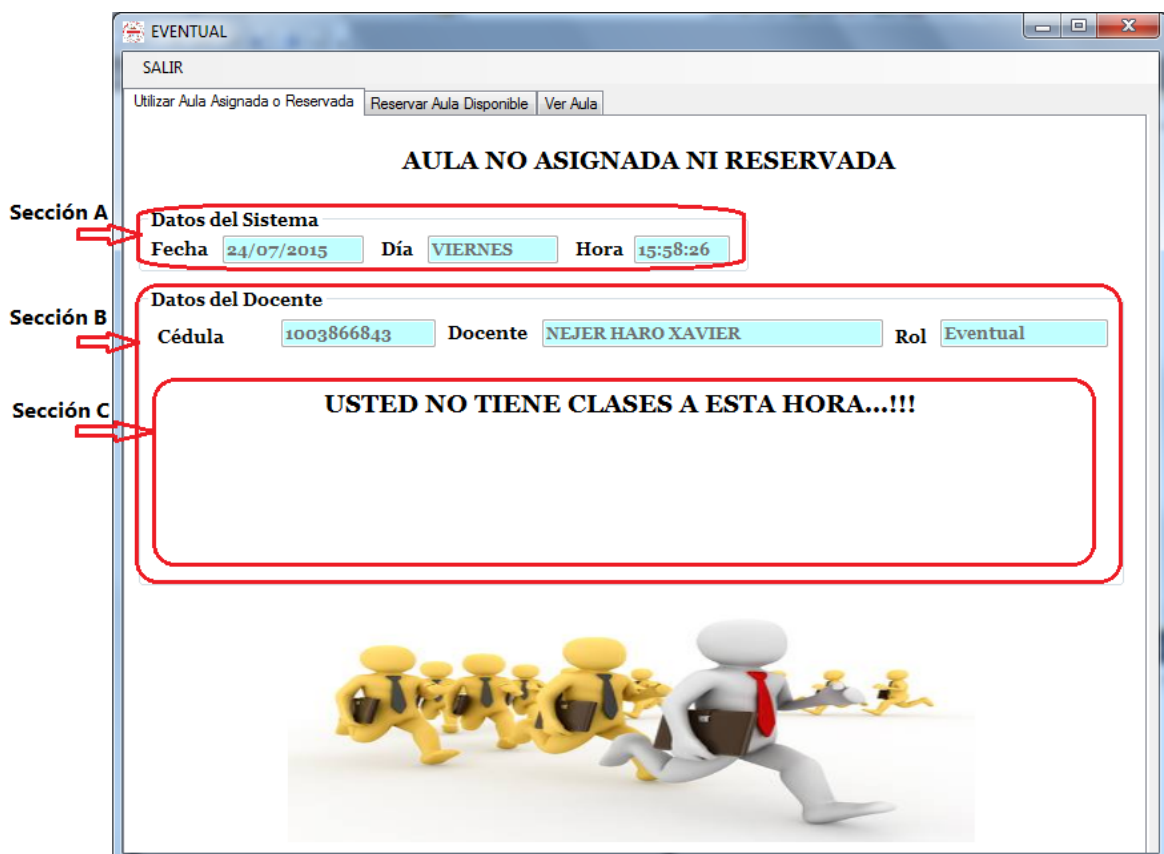
#### 7.9.3.2.2.1 Secciones del menú selectivo

En la interfaz de proceso, cada uno de los elementos del “menús selectivos”, presentan secciones diferentes que se acoplan a cada función que cumplen, las mismas que se detalla a continuación:

7.9.3.2.2.1.1 Utilizar Aula Asignada o Reservada.

- A. Datos del Sistema: Visualiza los datos provenientes de la computadora con la que se encuentre trabajando el sistema.
- B. Datos del Docente: Visualiza los datos del docente con los cuales se registró en el sistema.
- C. Reserva: Permite visualizar la información correspondiente a la reserva del aula a utilizar, por parte del docente en ese momento.
- D. Esta sección depende de la reserva realizada con anticipación para poder visualizar los datos del aula para utilizar, en caso de no existir estos parámetros se visualiza un mensaje informando al docente que no tiene clases a esa hora.

Figura 239. Identificación de las secciones en el formulario de procesos del docente eventual del submenú utilizar aula asignada o reservada.





*Nota:*

*En caso de no tener clases el docente en ese momento, no se cargan los datos y se visualiza un mensaje informándole que no tiene clases a esa hora.*

*La sección C tiene su propio control que realiza una tarea determinada, la misma que se explica en el proceso de utilización más adelante.*

*7.9.3.2.2.1.2 Reservar Aula Disponible.*

- A. Datos del Aula: Permite seleccionar un aula disponible a reservar por el docente.
- B. Detalle: Permite ingresar información del motivo por el cual el aula seleccionada en la sección A se va a utilizar.
- C. Parámetros de Reserva: Permite seleccionar parámetros de la materia que se va impartir en una fecha determinada.
- D. Datos Docente: Visualiza los datos del docente con los cuales se registró en el sistema.
- E. Tiempo: Permite seleccionar el tiempo en el cual el aula será utilizada por el docente.
- F. Controles: Permite guardar o borrar la información que se encuentre ingresando en este proceso.

Figura 240. Identificación de las secciones en el formulario de procesos del docente eventual del submenú reservar aula disponible.

The screenshot shows a web application window titled 'EVENTUAL' with a sub-menu 'Reservar Aula Disponible'. The main form is titled 'UTILIZAR AULA DISPONIBLE' and is divided into several sections:

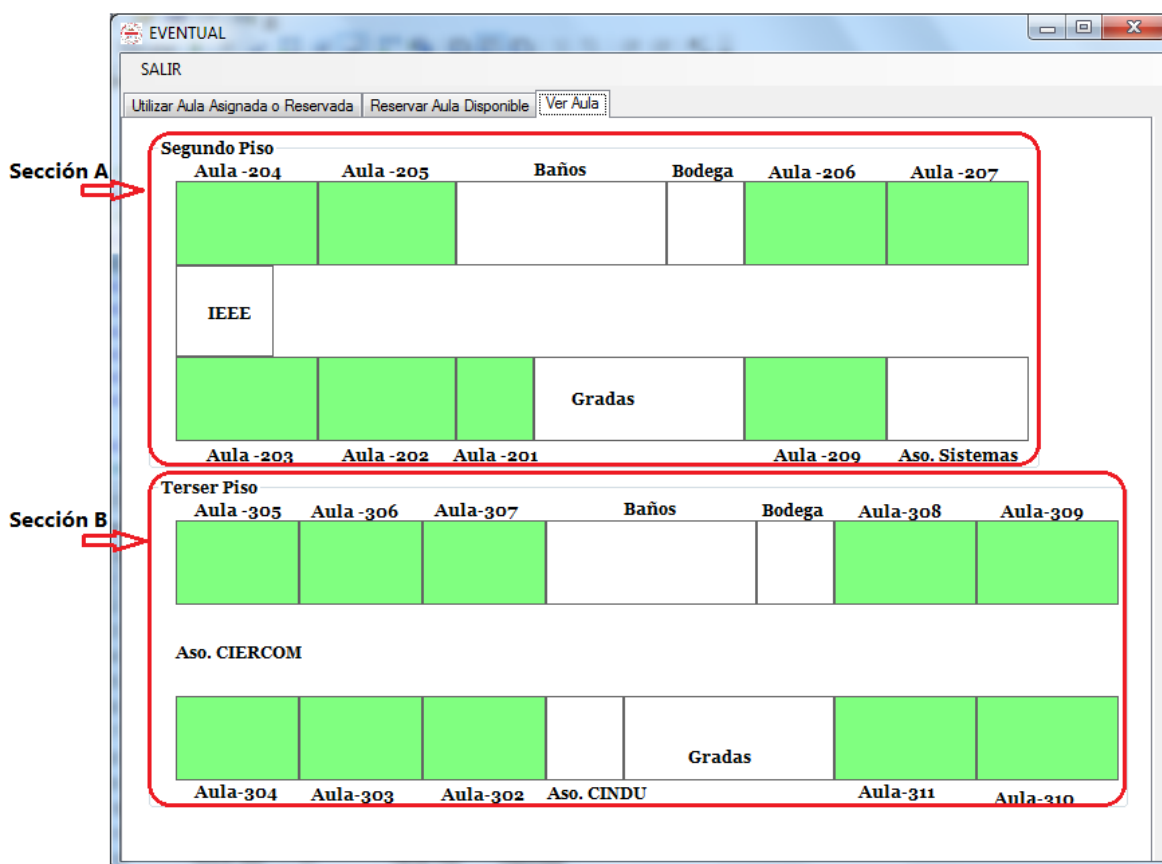
- Sección A:** 'Datos del Aula' containing 'Número de Aula' (dropdown), 'Ubicación' (text input), and 'Capacidad de Estudiantes' (text input).
- Sección D:** 'Datos Docente' containing 'C.I. Docente' (1003866843), 'Docente' (NEJER HARO XAVIER), and 'Rol\_Usuario' (Eventual).
- Sección B:** 'Detalle:' containing a large text area for 'MOTIVO DE LA RESERVA:'.
- Sección C:** 'Parámetros de Reserva' containing 'Día' (text input), 'Fecha de Reserva' (24/07/2015), 'Periodo Académico' (SEPTIEMBRE 2014 - FEBRERO 2015), 'Materia' (dropdown), and 'Carrera', 'Nivel', 'Paralelo' (dropdowns).
- Sección E:** 'Tiempo' containing 'Número de Horas' (dropdown), 'Hora Actual' (16:13:44), 'Hora Inicial' (dropdown), and 'Hora Final' (00:00:00).
- Sección F:** 'Controles' containing 'LIMPIAR' and 'OK' buttons.

At the bottom of the form, there are two small images showing classroom environments.

#### 7.9.3.2.2.1.3 Ver Aula.

- A. Segundo Piso: Identifica a todas las aulas de la facultad que se encuentran en el segundo piso de la facultad.
- B. Tercer Piso: Identifica a todas las aulas de la facultad que se encuentran en el tercer piso de la facultad.

Figura 241. Identificación de las secciones en el formulario de procesos del docente eventual del submenú ver aula.



Para identificar si un aula se encuentra ocupada o disponible se visualiza dependiendo de los siguientes colores:

- ✓ Color Azul: El aula se encuentra ocupada.
- ✓ Color Verde: El aula se encuentra disponible.

*Nota:*

En lo referente al menú selectivo, para el docente eventual, no todos los campos son habilitados, esto depende del modo de ingreso con el cual se registra el docente o funcionario ya que cada campo realiza una función determinada, por ende solamente se habilitan los campos en los cuales se tiene acceso a modificar o utilizar.

### **7.9.3.3 Proceso de utilización del sistema del usuario eventual.**

A continuación se especifican todos los pasos que debe seguir el docente que cumple las funciones de eventual en el sistema de control de acceso de las aulas de la facultad, de igual forma se presenta en tres etapas según los formularios que el sistema contiene.

#### **7.9.3.3.1 Pasos a seguir en el formulario de autenticación.**

Para poder ingresar al sistema mediante el modo eventual es necesario seguir los siguientes pasos:

1. Seleccionar en la sección "*Modo de ingreso*" la casilla que identifica este modo.
2. Una vez seleccionado el modo de ingreso se habilita el campo de texto en la sección "*Autenticación*" en el cual se debe proceder a ingresar la información que requiere con el respectivo dato del docente:

✓ *Ingrese CI:* Se debe escribir la cédula del docente con la cual está registrado en el sistema.

De igual forma se habilitan el botón con el cual se puede:

✓ *Ingresar:* En caso de que el dato se encuentren correctamente ingresado, el usuario puede acceder al sistema, en lo cual se abre una nueva ventana, caso contrario el sistema deniega el ingreso mostrando previamente el error suscitado.

3. Finalmente se hace click en el botón "Ingresar" y esperar el mensaje de confirmación.

Figura 242. Pasos a seguir en el formulario autenticación del docente eventual.

**AUTENTICACIÓN**

**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FICA - CIERCOM**  
**SISTEMA DE CONTROL DE ACCESO**

**Modo de ingreso**

Administrador

Usuario

**Eventual**

**Datos del Sistema**

Fecha 24/07/2015

Día VIERNES

Hora Actual 16:31:32

**Autenticación**

Ingreso C.I.

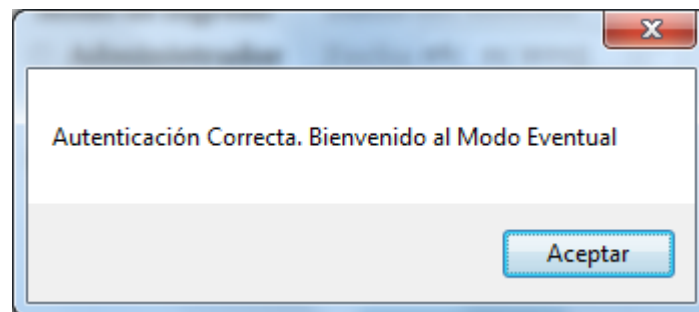
**INGRESAR**

**Paso 1**

**Paso 2**

**Paso 3**

Figura 243. Mensaje de confirmación del ingreso al sistema del docente eventual.



*Nota:*

*En caso de existir algún error en el dato ingresado por el usuario, el sistema informará mediante un mensaje el motivo por el cual no se pudo dar acceso al sistema.*

### **7.9.3.3.2 Pasos a seguir en el formulario procesos.**

A continuación se profundizan las funciones y el uso de cada uno de los submenús descritos anteriormente en el formulario de procesos según los parámetros correspondientes al usuario.

#### *7.9.3.3.2.1 Utilizar Aula Asignada o Reservada.*

Este proceso permite utilizar un aula reservada por el docente, el cual carga los datos provenientes del sistema, verificando si el docente tiene clases a esa hora (en la que se registró) o no. Debido a esta eventualidad este proceso se presenta en dos etapas que son:

1. Cuando el docente tiene clases por medio de una reserva.
2. Cuando el docente no tiene clases.

##### *7.9.3.3.2.1.1 Cuando el docente tiene clases por medio de una reserva.*

El sistema calcula si el docente ha reservado un aula para esa fecha y hora en el periodo académico en curso, el formulario se presenta de la siguiente forma.

Figura 244. Formulario de clases por medio de una reserva.

EVENTUAL

SALIR

Utilizar Aula Asignada o Reservada Reservar Aula Disponible Ver Aula

### AULA RESERVADA POR EL DOCENTE

**Datos del Sistema**

Fecha 25/07/2015 Día SÁBADO Hora 14:04:37

**Datos del Docente**

Cédula 1003866843 Docente NEJER HARO XAVIER Rol Eventual

**Reserva**

Materia INGLÉS I Detalle PRUEBAS DE TESIS DOCENTE EVENTUAL

Periodo\_Académico SEPTIEMBRE 2014 - FEBRERO

Carrera CAI

Fecha 25/07/2015 Nivel 1ro

Aula AULA\_201 Paralelo A

Utilizar\_Aula OK

7.9.3.3.2.1.2 Cuando el docente no tiene clases.

En caso de no tener clases a esa hora el formulario se presenta de la siguiente forma.

Figura 245. Formulario de no tener clases.

The screenshot shows a web application window with the title 'EVENTUAL'. At the top, there is a 'SALIR' button and three navigation buttons: 'Utilizar Aula Asignada o Reservada', 'Reservar Aula Disponible', and 'Ver Aula'. The main content area displays the message 'AULA NO ASIGNADA NI RESERVADA' in bold. Below this, there are two sections of data: 'Datos del Sistema' with fields for 'Fecha' (24/07/2015), 'Día' (VIERNES), and 'Hora' (15:58:26); and 'Datos del Docente' with fields for 'Cédula' (1003866843), 'Docente' (NEJER HARO XAVIER), and 'Rol' (Eventual). A large, bold message 'USTED NO TIENE CLASES A ESTA HORA...!!!' is centered below the data. At the bottom of the window, there is a 3D illustration of several yellow figures running, with a white figure in the foreground carrying a briefcase.

Los pasos a seguir en caso de tener clases, por medio de una la reserva son los siguientes:

1. Hacer click en el botón "OK" y esperar el mensaje de confirmación.

Una vez realizada esta acción la trama de datos es enviada hacia el hardware de este sistema, y guardados los datos de este proceso en el sistema para generar el reporte de la utilización del aula correspondiente. Posteriormente se visualiza el mensaje de confirmación de los datos guardados.



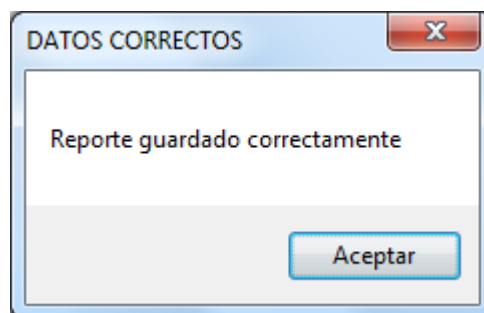
Figura 246. Pasos para utilizar el aula reservada para el docente eventual.

The screenshot shows a software window titled "EVENTUAL" with a menu bar containing "SALIR" and "Utilizar Aula Asignada o Reservada", "Reservar Aula Disponible", and "Ver Aula". The main content area is titled "AULA RESERVADA POR EL DOCENTE" and is divided into several sections:

- Datos del Sistema:** Fecha: 25/07/2015, Día: SÁBADO, Hora: 14:04:37.
- Datos del Docente:** Cédula: 1003866843, Docente: NEJER HARO XAVIER, Rol: Eventual.
- Reserva:** Materia: INGLES I, Detalle: PRUEBAS DE TESIS DOCENTE EVENTUAL, Periodo\_Académico: SEPTIEMBRE 2014 - FEBRERO, Carrera: CAI, Fecha: 25/07/2015, Nivel: 1ro, Aula: AULA\_201, Paralelo: A.

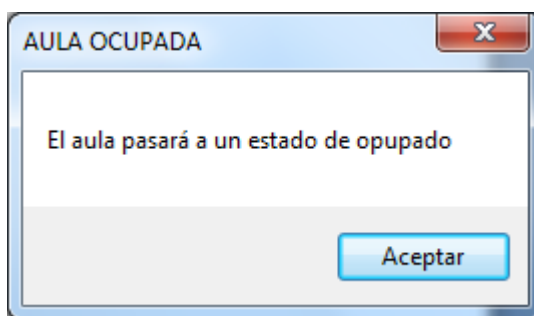
At the bottom right of the form, there is a button labeled "Utilizar\_Aula" with an "OK" button next to it. A yellow arrow points to the "OK" button, which is labeled "Paso 1". Below the form is a 3D illustration of a white businessman running towards a group of yellow businessmen.

Figura 247. Confirmación de los datos guardados en el reporte del sistema



Una vez guardado el reporte y realizada la apertura de la puerta en el hardware, el sistema pone al aula correspondiente en un estado de "Ocupado" por lo que se genera un mensaje informando este evento.

Figura 248. Confirmación del estado del aula actual.



*Nota:*

*La información indicada en esta sección no se modifica, debido a que los datos son extraídos de la base de datos del sistema.*

*En este proceso, debido a que se comunica el sistema con el hardware creado para las puertas de la facultad, se envía una trama de datos durante un periodo de tiempo específico, por lo cual puede tardar un poco la confirmación de la apertura de la puerta.*

#### *7.9.3.3.2 Reservar Aula Disponible.*

Este proceso le permite al docente realizar una reserva de un aula determinada en un tiempo establecido especificando el motivo de la misma.

Para utilizar este método se siguen los siguientes pasos:

1. Seleccionar el "Número de Aula" de la sección "*Datos del Aula*" la misma que contiene los siguientes campos:
  - ✓ Nombre del Aula: identifica el aula a realizar la reserva.
  - ✓ Ubicación: Identifica el lugar donde se encuentra físicamente esa aula.
  - ✓ Capacidad: informa el número máximo de estudiantes que pueden ingresar en esa aula seleccionada.

2. Escribir el motivo por el cual se está realizando la reserva de esa aula en el campo "Detalle".
3. Identificar la fecha en la cual se desea utilizar el aula seleccionada.
4. Identificar qué materia se va a impartir en esa aula.
5. Identificar la carrera a la que pertenece esa materia seleccionada.
6. Identificar el nivel al que pertenece esa materia seleccionada.
7. Seleccionar el paralelo a la cual se le va a impartir la materia seleccionada.
8. Especificar qué tiempo se va a utilizar el aula seleccionada.
9. Indicar la hora en la que se va a iniciar a impartir clases en esa aula seleccionada.
10. Finalmente, hacer click en el botón "OK" y esperar el mensaje de confirmación.

Figura 249. Pasos para realizar una reserva de un aula en el sistema.



Figura 250.Sugerencia para reservar el aula.

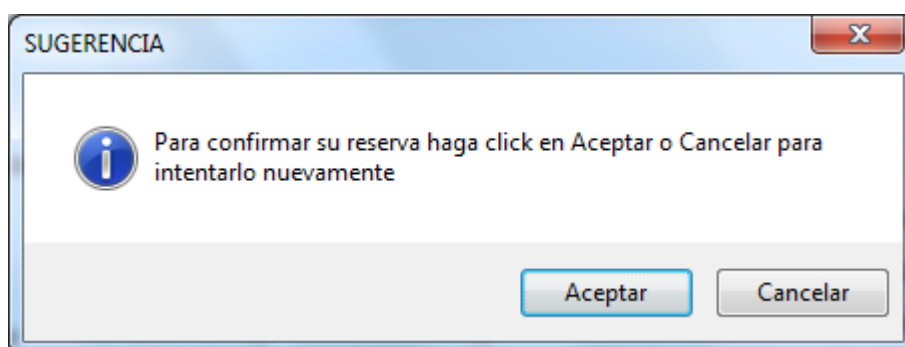
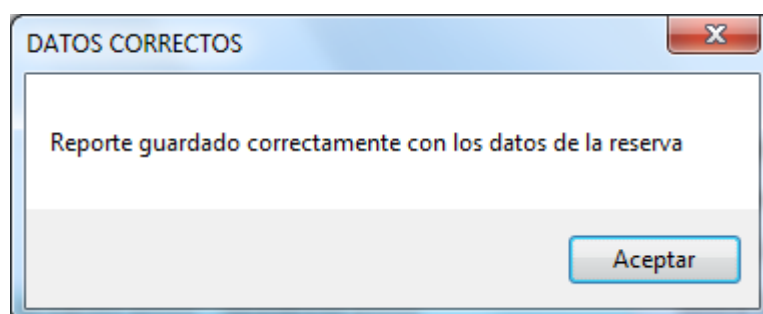


Figura 251. Confirmación de la reserva del aula realizada correctamente.



#### 7.9.3.3.2.3 Ver Aulas.

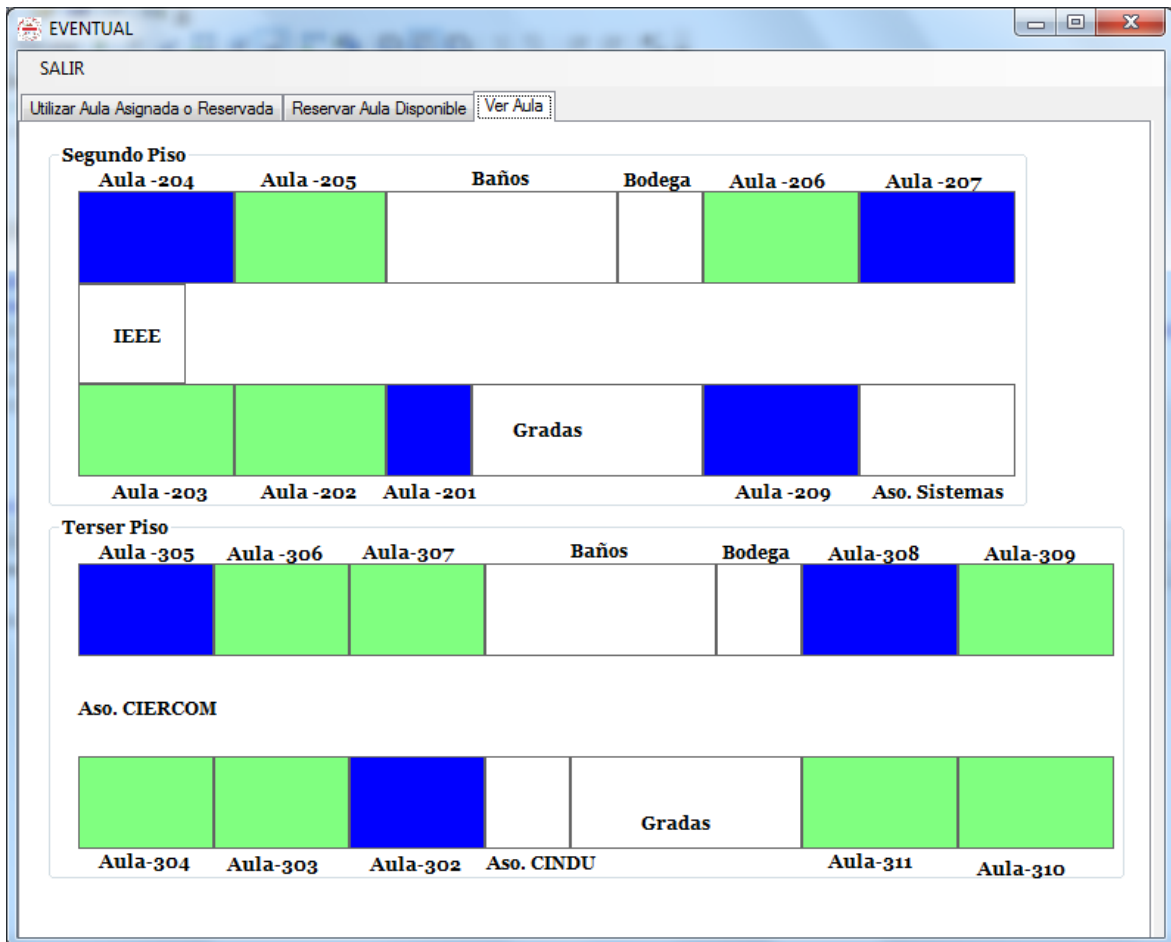
Este proceso permite visualizar el estado de las aulas de la facultad correspondiente a la disponibilidad de las mismas, es decir, es un parámetro que le permite al docente identificar de forma gráfica que aulas se encuentran disponibles u ocupadas.

Para identificar uno u otro estado se identifica con los siguientes colores:

- ✓ Color Azul: El aula se encuentra ocupada.
- ✓ Color Verde: El aula se encuentra disponible.

En la figura siguiente se visualiza el mapa de las aulas de la facultad, unas disponibles y otras ocupadas.

Figura 252. Identificación del estado de las aulas de la facultad.



#### 7.9.3.4 Recomendaciones:

1. Leer detalladamente este manual para evitar cometer errores en la ejecución del sistema.
2. La información proporcionada como una “**Nota**” que se incluye en las diferentes secciones de este manual es de mucha importancia, hay que tener en cuenta, ya que de esto depende el buen funcionamiento del sistema creado.
3. En el caso de existir algún error al ingresar o seleccionar los datos, el sistema genera un mensaje de “**datos erróneos o sugerencia**” para evitar

los posibles errores, por lo tanto se debe tomar en cuenta dichos mensajes.

4. La utilización de este manual es precisamente para el docente eventual del sistema, por ende la información proporcionada en él es de carácter reservado.

## **ANEXO F**

### **7.10 MANUALES DEL SISTEMA DE CONTROL DE ACCESO (HARDWARE) DE LAS AULAS DE LA FACULTAD PARA DOCENTES Y CONSERJES**

## **7.10.1 MANUAL DEL HARDWARE PARA DOCENTES**

### **7.10.1.1 Introducción**

El sistema de control (hardware) permite dar acceso al personal que labora en la facultad, en cada una de las puertas, por medio de la autenticación del docente o funcionario, cabe recalcar que para este proceso se ha separado en dos funciones, ya que el personal que puede ingresar a las aulas pueden ser:

- Los docentes, mismos que realizan la acción de impartir clases en un periodo determinado, según un horario de clases o reserva de un aula realizada con anticipación en el software del sistema.
- Los conserjes, los cuales se encargan de realizar la limpieza de las aulas a cualquier hora del día según corresponda.

. En este manual se identifican los elementos que contiene el hardware del sistema así como el proceso o pasos a seguir para la utilización del aula correspondiente a un docente.

### **7.10.1.2 Elementos del hardware del sistema**

El hardware cuenta con dos dispositivos, los cuales son el dispositivo transmisor y el dispositivo receptor, los mismos que se interconectan entre sí por medio de una red WPAN para brindar acceso a las aulas de la facultad, permitiendo tener un control total brindando seguridad a las mismas.

#### **7.10.1.2.1 *Dispositivo transmisor.***

Es el encargado de transmitir y recibir datos desde el software hacia el hardware del sistema de control de acceso respectivamente.



Figura 253. Dispositivo transmisor.



Frontal:

- Luz Verde (LV): Identifica el estado del dispositivo como disponible.

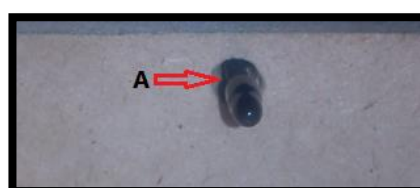
Figura 254. Parte frontal del dispositivo transmisor.



Lateral superior:

- Antena (A): Permite recibir y transmitir los datos el hardware.

Figura 255. Parte lateral superior del dispositivo transmisor.



Lateral inferior:

- Conector USB (USB): Permite conectar el dispositivo a un computador.

Figura 256. Parte lateral inferior del dispositivo transmisor.



En el caso del dispositivo transmisor, es necesario conectar a la computadora que contiene el software del sistema en un puerto USB y verificar que el indicador se encienda, confirmando así la correcta instalación del mismo, tal como se indica en la figura siguiente.

Figura 257. Conexión del dispositivo transmisor en el computador del sistema.



### 7.10.1.2.2 *Dispositivo receptor.*

Es el encargado de recibir y transmitir datos desde el hardware hacia el software del sistema de control de acceso respectivamente, y además verificar si los datos recibidos son correctos para la utilización de un aula en una hora determinada por un docente.

Figura 258. Dispositivo receptor.



Frontal:

- Pantalla (Información): Permite visualizar mensajes para el docente e información de un aula determinada.
- Luz Verde (LVD): Identifica el estado del aula como disponible.
- Luz Roja (LRO): Identifica el estado del aula como ocupada.
- Botón del conserje (BC): Permite realizar la limpieza del aula, es utilizado solamente por el conserje de la facultad.

- Botón de hora (BH): Permite terminar el proceso de utilización del aula.
- Teclado: Permite ingresar información para validar en el sistema, tiene dos etapas.
  - ✓ Numérico (TN): Permite ingresar información como la clave del personal.
  - ✓ Comandos (TC): Son los botones que permiten acceder a un menú determinado.
    - A: Selecciona el menú de ingrese clave.
    - B: Selecciona el menú de bloquear puerta.
    - C: Finaliza un proceso que se vaya a realizar en el teclado.
    - D: Borra los datos ingresados por el teclado.

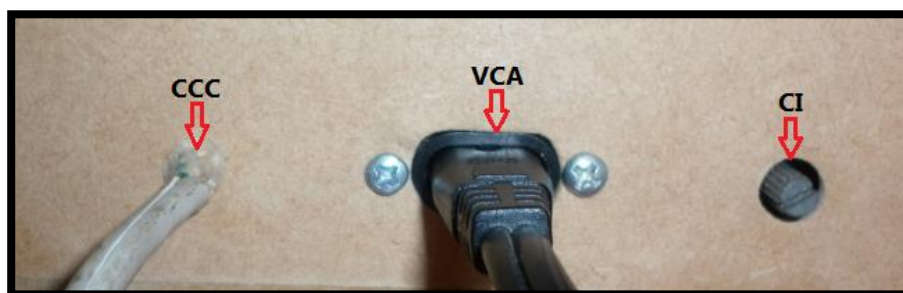
Figura 259. Parte frontal del dispositivo receptor.



Lateral Izquierdo:

1. Contraste de la información (CI). Permite aumentar o disminuir la intensidad de los mensajes en la pantalla.
2. Alimentación de energía (VCA): Permite mantener al circuito encendido.
3. Cable conector de la cerradura (CCC): Conecta la cerradura eléctrica con el hardware, este contiene dos botones:
  - Botón de apertura exterior (BAE): Permite abrir la puerta del aula desde el exterior.
  - Botón de apertura interior (BAI): Permite abrir la puerta del aula desde el interior de la misma.

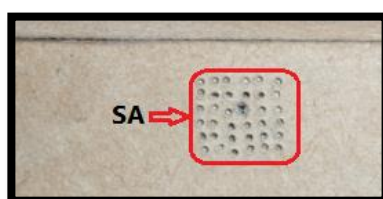
Figura 260. Parte lateral izquierda del dispositivo receptor.



Lateral Derecho:

4. Sonido de alarma (SA): Se encarga de avisar al docente, mediante un sonido, que se encuentra por concluir el periodo de clases.

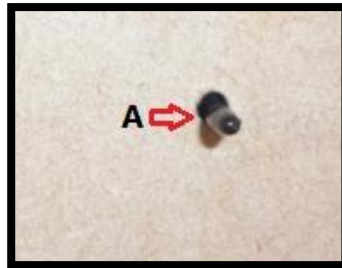
Figura 261. Parte lateral derecha del dispositivo receptor.



Lateral Inferior:

5. Antena (A): Permite recibir y transmitir los datos el hardware.

Figura 262. Parte lateral inferior del dispositivo receptor.



### 7.10.1.3 Pasos a seguir en el dispositivo receptor del sistema

Inicialmente el dispositivo receptor permanece en un estado de espera o disponible, manteniendo encendido el (LVD) e indicando un mensaje, este mensaje se lo puede visualizar en la figura siguiente.

Figura 263. Mensaje de bienvenida en el dispositivo receptor.



Una vez realizada la acción de utilización del aula en el sistema (software), los datos son transmitidos al aula que le corresponda, la misma que pasa a un menú, el cual indica la acción próxima a realizar, cambiando el estado del aula a ocupada, encendiendo el (LRO) . Tal como se indica en la figura siguiente.

Figura 264. Menú de inicio del dispositivo receptor.



Los pasos a seguir, una vez el docente se encuentre en la puerta del aula, son los siguientes:

1. Seleccionar la opción “Ingrese clave” digitando la tecla A del (TC), una vez presionada esa tecla se visualiza el siguiente mensaje.

Figura 265. Visualización del pedido de clave en el dispositivo receptor.



2. Digitar la clave correspondiente al docente en el (TN), esta clave se indica en la pantalla, como se puede apreciar en el siguiente mensaje

Figura 266. Proceso de digitalización de la clave en el dispositivo receptor.



Estos dos pasos se los realiza presionando las teclas como se mencionan anteriormente y se puede apreciar en la siguiente figura.



Figura 267. Identificación de los pasos a seguir en el teclado del dispositivo receptor.



Si la clave es correcta, el aula se habilita y en la pantalla se visualiza un mensaje de confirmación de clave y del estado actual de la puerta como se indica en las figuras siguientes.

Figura 268. Confirmación de la clave en el dispositivo receptor.



Figura 269. Apertura de la puerta en el dispositivo receptor.





Posteriormente se puede apreciar el número de horas que el docente debe permanecer en el aula y el tiempo transcurrido en la misma, como se indica en la figura siguiente.

Figura 270. Información del número de horas y el tiempo transcurrido en el dispositivo receptor.



Caso contrario si la clave es incorrecta, se visualiza un mensaje de error y el número de intentos que ha realizado para ingresar al aula, tal como se indica en la figura siguiente.

Figura 271. Mensaje de error y número de intentos en el dispositivo receptor.



Si este fuese el caso, el docente debe ingresar nuevamente la clave, realizando este paso nuevamente.

3. Finalmente, el docente puede ingresar al aula presionando el botón de apertura (BAE), el mismo que se encuentra instalado en la puerta del aula.

*Notas: Si en el paso 2 el número de intentos es superior a 3, el aula se bloquea denegando el acceso al docente, por lo que tendrá que realizar el proceso de utilización del aula en el software del sistema.*

*Si se presiona la tecla B del teclado de control (TN), el proceso es el mismo para validar datos con el software, de esta forma se bloqueará o no el aula dependiendo de la información que se ingrese.*

#### **7.10.1.4 Alerta de finalización**

En el caso de que el periodo de clases esté por finalizar, el dispositivo receptor procederá a informar al docente mediante un sonido de alarma y un mensaje en la pantalla acerca de este evento, cabe recalcar que este proceso se realiza 5 y 3 minutos antes de finalizar dicho periodo para que el docente tenga pendiente que su estadía en el aula ya culminó. El mensaje de información se puede apreciar en la siguiente figura.

Figura 272. Información del tiempo restante en el dispositivo receptor del aula.



#### **7.10.1.5 Finalización del tiempo en un aula**

El sistema permite interrumpir el tiempo que un docente puede permanecer en el aula, si así lo requiere, este proceso se puede realizar una vez que el

docente ya se haya registrado en el dispositivo y se encuentre utilizando un aula, tal como se puede apreciar en la siguiente figura.

Figura 273. Estado de utilización de un aula.



Los pasos para finalizar la utilización de un aula son:

1. Presionar el botón de la hora (BH), una vez presionado ese botón se visualiza el siguiente mensaje.

Figura 274. Visualización del pedido de clave de finalización en el dispositivo receptor.



2. Ingresar la clave del docente, esta clave se indica en la pantalla, como se puede apreciar en la siguiente figura.

Figura 275. Proceso de digitalización de la clave de finalización en el dispositivo receptor.



Si los datos ingresados son correctos, el dispositivo enviará una trama de confirmación al software del sistema para validar la clave del docente, durante este proceso este dispositivo esperará una confirmación de los datos mostrando el siguiente mensaje.

Figura 276. Información de espera de confirmación de datos en el dispositivo receptor.



Posteriormente pasará a finalizar el periodo de utilización del aula por parte del docente, mediante un mensaje de finalización y regresando al estado inicial el dispositivo.

Figura 277. Finalización del periodo de clases en el aula del dispositivo receptor.



Caso contrario si la clave ingresada es incorrecta, el dispositivo indicará, mediante un mensaje, que los datos son incorrectos, como se puede apreciar en la siguiente figura.

Figura 278. Mensaje de error de clave en el dispositivo receptor.



Siendo este el caso, el dispositivo continuará con la utilización del aula sin haber afectado el estado de la misma, continuando con el periodo de clases asignado.

*Nota: Si el docente desea realizar esta acción de finalización deberá volver a realizar los pasos indicados en esta sección.*

## **7.10.2 MANUAL DEL HARDWARE PARA CONSERJES**

### **7.10.2.1 Introducción**

El sistema de control (hardware) permite dar acceso al personal que labora en la facultad en cada una de las puertas por medio de la autenticación del docente o funcionario, cabe recalcar que para este proceso se ha separado en dos funciones, ya que el personal que puede ingresar a las aulas pueden ser:

- Los docentes, mismos que realizan la acción de impartir clases en un periodo determinado, según un horario de clases o reserva de un aula realizada con anticipación en el software del sistema.
- Los conserjes, los cuales se encargan de realizar la limpieza de las aulas a cualquier hora del día según corresponda.

En este manual se identifican los elementos que contiene el hardware del sistema, así como el proceso o pasos a seguir para la utilización del aula correspondiente a un conserje.

### **7.10.2.2 Elementos del hardware del sistema**

El hardware cuenta con dos dispositivos, los cuales son el dispositivo transmisor y el dispositivo receptor, los mismos que se interconectan entre sí por medio de una red WPAN para brindar acceso a las aulas de la facultad, permitiendo tener un control total brindando seguridad a las mismas.

#### **7.10.2.2.1 *Dispositivo transmisor.***

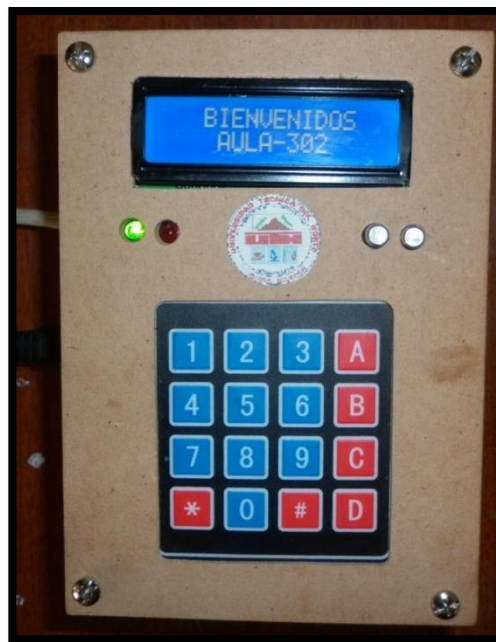
Es el encargado de transmitir y recibir datos desde el software hacia el hardware del sistema de control de acceso respectivamente.

*Nota: Este dispositivo es invisible para el conserje, por ello en caso de tener inconvenientes con el ingreso al aula, se debe poner en contacto con el administrador del sistema.*

#### **7.10.2.2.2 Dispositivo receptor.**

Es el encargado de recibir y transmitir datos desde el hardware hacia el software del sistema de control de acceso respectivamente y además verificar si los datos recibidos son correctos para la utilización de un aula en una hora determinada por un conserje.

Figura 279. Dispositivo receptor.



Frontal:

- Pantalla (Información): Permite visualizar mensajes para el docente e información de un aula determinada.
- Luz Verde (LVD): Identifica el estado del aula como disponible.
- Luz Roja (LRO): Identifica el estado del aula como ocupada.



- Botón del conserje (BC): Permite realizar la limpieza del aula, es utilizado solamente por el conserje de la facultad.
- Botón de hora (BH): Permite terminar el proceso de utilización del aula.
- Teclado: Permite ingresar información para validar en el sistema, tiene dos etapas.
  - ✓ Numérico (TN): Permite ingresar información como la clave del personal.
  - ✓ Comandos (TC): Son los botones que permiten acceder a un menú determinado.
    - A: Selecciona el menú de ingrese clave.
    - B: Selecciona el menú de bloquear puerta.
    - C: Finaliza un proceso que se vaya a realizar en el teclado.
    - D: Borra los datos ingresados por el teclado.

Figura 280. Parte frontal del dispositivo receptor.

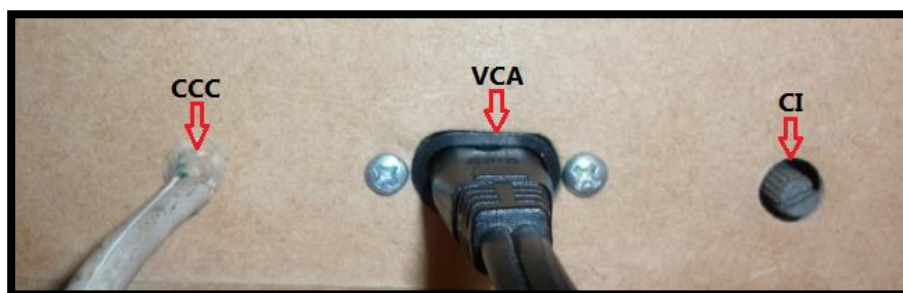




Lateral Izquierdo:

6. Contraste de la información (CI). Permite aumentar o disminuir la intensidad de los mensajes en la pantalla.
7. Alimentación de energía (VCA): Permite mantener al circuito encendido.
8. Cable conector de la cerradura (CCC): Conecta la cerradura eléctrica con el hardware, este contiene dos botones:
  - Botón de apertura exterior (BAE): Permite abrir la puerta del aula desde el exterior.
  - Botón de apertura interior (BAI): Permite abrir la puerta del aula desde el interior de la misma.

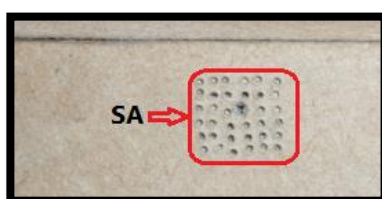
Figura 281. Parte lateral izquierda del dispositivo receptor.



Lateral Derecho:

9. Sonido de alarma (SA): Se encarga de avisar al docente, mediante un sonido, que se encuentra por concluir el periodo de clases.

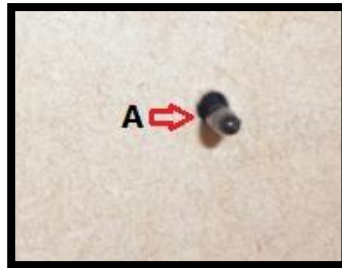
Figura 282. Parte lateral derecha del dispositivo receptor.



Lateral Inferior:

10. Antena (A): Permite recibir y transmitir los datos el hardware.

Figura 283. Parte lateral inferior del dispositivo receptor.



### 7.10.2.3 Pasos a seguir en el dispositivo receptor del sistema

Inicialmente el dispositivo receptor permanece en un estado de espera o disponible, manteniendo encendido el (LVD) e indicando un mensaje, este mensaje se lo puede visualizar en la figura siguiente.

Figura 284. Mensaje de bienvenida en el dispositivo receptor.



Los pasos a seguir, una vez el conserje se encuentre en la puerta del aula, son los siguientes:

1. Presionar el botón (BC), una vez presionado dicho botón, se visualiza el siguiente mensaje.

Figura 285. Visualización del pedido de clave en el dispositivo receptor.



2. Digitar la clave correspondiente al conserje en el (TN), esta clave se indica en la pantalla, como se puede apreciar en el siguiente mensaje.

Figura 286. Proceso de digitalización de la clave en el dispositivo receptor.



Estos dos pasos se los realiza presionando las teclas como se mencionan anteriormente y se puede apreciar en la siguiente figura.

Figura 287. Identificación de los pasos a seguir en el dispositivo receptor.



Una vez realizados estos pasos, el dispositivo enviará una trama de confirmación al software del sistema para validar la clave del conserje, durante este proceso, el dispositivo esperará una confirmación de los datos mostrando el siguiente mensaje.



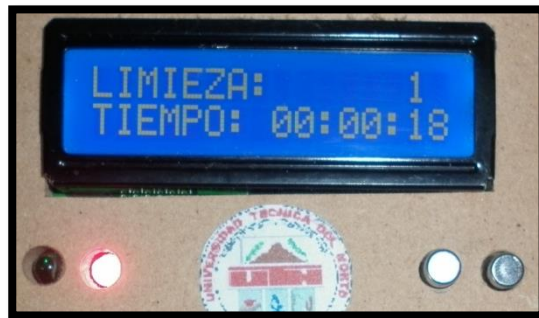
Si la clave es correcta, el aula se habilita y en la pantalla se visualiza un mensaje de confirmación de clave y del estado actual de la puerta, como se indica en las figuras siguientes.

Figura 288. Confirmación de la clave del conserje en el dispositivo receptor.



Posteriormente se puede apreciar el número de horas que el conserje debe permanecer en el aula y el tiempo transcurrido en la misma, como se indica en la figura siguiente.

Figura 289. Información del número de horas y el tiempo transcurrido del conserje en el dispositivo receptor.



Caso contrario si la clave es incorrecta, se visualiza un mensaje de error informando que los datos ingresados en el dispositivo son falsos, tal como se indica en la figura siguiente.

Figura 290. Datos ingresados falsos en el dispositivo receptor.



Si este fuese el caso, el docente debe ingresar nuevamente la clave realizando estos pasos nuevamente.

3. Finalmente, el docente puede ingresar al aula presionando el botón de apertura (BAE), el mismo que se encuentra instalado en la puerta del aula.

#### **7.10.2.4 Alerta de finalización**

En el caso de que el periodo de limpieza esté por finalizar, el dispositivo receptor procederá a informar al conserje mediante un sonido de alarma y un mensaje en la pantalla acerca de este evento, cabe recalcar que este proceso se

realiza 5 y 3 minutos antes de finalizar dicho periodo, para que el conserje tenga pendiente que su estadía en el aula ya culminó. El mensaje de información se puede apreciar en la siguiente figura.

Figura 291. Información del tiempo restante en el dispositivo receptor del aula.



#### **7.10.2.5 Finalización del tiempo en un aula**

El sistema permite interrumpir el tiempo que un conserje puede permanecer en el aula, si así lo requiere, este proceso se puede realizar una vez que el conserje ya se haya registrado en el dispositivo y se encuentre utilizando un aula tal como se puede apreciar en la siguiente figura.

Figura 292. Estado de utilización de un aula por un conserje.



Los pasos para finalizar la utilización de un aula son:

1. Presionar el botón de la hora (BH), una vez presionado ese botón se visualiza el siguiente mensaje.

Figura 293. Visualización del pedido de clave de finalización por el conserje.



2. Ingresar la clave del conserje, esta clave se indica en la pantalla, como se puede apreciar en la siguiente figura.



Figura 294. Proceso de digitalización de la clave del conserje de finalización.



Si los datos ingresados son correctos, el dispositivo enviará una trama de confirmación al software del sistema para validar la clave del conserje, durante este proceso, el dispositivo esperará una confirmación de los datos mostrando el siguiente mensaje.

Figura 295. Información de espera de confirmación de datos del conserje.



Posteriormente pasará a finalizar el periodo de utilización del aula por parte del docente, mediante un mensaje de finalización y regresando al estado inicial el dispositivo.

Figura 296. Finalización del periodo de limpieza en un aula.





Caso contrario si la clave ingresada es incorrecta, el dispositivo indicará, mediante un mensaje, que los datos son incorrectos como se puede apreciar en la siguiente figura.

Figura 297. Mensaje de error de la clave del conserje.



Siendo este el caso, el dispositivo continuará con la utilización del aula, sin haber afectado el estado de la misma, continuando con el periodo de clases asignado.

*Nota: Si el conserje desea realizar esta acción de finalización deberá volver a realizar estos pasos indicados en esta sección.*