

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:

“JUGUETE ELECTRÓNICO DIDÁCTICO, COMO ELEMENTO DE APOYO PARA LA
ENSEÑANZA DE PROGRAMACIÓN A NIÑOS Y NIÑAS DE 4 A 7 AÑOS”

AUTOR: EDGAR EDUARDO CARTAGENA OCAMPO

DIRECTOR: ING. DANIEL JARAMILLO

IBARRA - ECUADOR

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.

DATOS DEL CONTACTO	
Cédula de identidad	100382334-9
Apellidos y Nombres	Cartagena Ocampo Edgar Eduardo
Dirección	Salinas 15-14 y Rio Tumbes
E-mail	eecartagenao@utn.edu.ec
Teléfono móvil	0997931165
DATOS DE LA OBRA	
Título	JUGUETE ELECTRÓNICO DIDÁCTICO, COMO ELEMENTO DE APOYO PARA LA ENSEÑANZA DE PROGRAMACIÓN A NIÑOS Y NIÑAS DE 4 A 7 AÑOS
Autor	Edgar Eduardo Cartagena Ocampo
Fecha	Marzo del 2016
Programa	Pregrado
Título	Ingeniero en Electrónica y Redes de Comunicación
Director	Ing. Daniel Jaramillo

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Cartagena Ocampo Edgar Eduardo, con cedula de identidad Nro. 100382334-9, en calidad de autor y titular de los derechos patrimoniales dela obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma 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 de material y como apoyo a la educación, investigación y extensión, en concordancia con la ley de Educación Superior Artículo 144.



Firma:

Nombre: Cartagena Eduardo

Cedula: 100382334-9

Ibarra. Marzo 2016



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

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

Yo, Cartagena Ocampo Edgar Eduardo, con cedula de identidad Nro. 100382334-9, manifiesto ni 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 del trabajo de grado con el tema: JUGUETE ELECTRÓNICO DIDÁCTICO, COMO ELEMENTO DE APOYO PARA LA ENSEÑANZA DE PROGRAMACIÓN A NIÑOS Y NIÑAS DE 4 A 7 AÑOS. Que ha sido desarrollado con propósito de obtener el título de Ingeniero en Electrónica Redes de Comunicación de 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 en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Cartagena Ocampo Edgar Eduardo

100382334-9

Ibarra, Marzo 2016



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

INGENIERO DANIEL JARAMILLO, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA

Que, el presente trabajo de Titulación “JUGUETE ELECTRÓNICO DIDÁCTICO, COMO ELEMENTO DE APOYO PARA LA ENSEÑANZA DE PROGRAMACIÓN A NIÑOS Y NIÑAS DE 4 A 7 AÑOS” Ha sido desarrollado por el señor Cartagena Ocampo Edgar Eduardo bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.

.....
Ing. Daniel Jaramillo

100154514-2

DIRECTOR



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIAS

Yo, CARTAGENA OCAMPO EDGAR EDUARDO declaro bajo juramento que el trabajo aquí escrito es de mi autoría; y que este no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se presentan en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Universidad Técnica del Norte, según lo establecido por las leyes de propiedad intelectual, reglamentos y normatividad vigente de la Universidad Técnica del Norte.

En la ciudad de Ibarra, Febrero del 2016

EL AUTOR

Cartagena Ocampo Edgar Eduardo
CI: 1003823349-9



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

Con merito especial y con todo mi cariño dedico este trabajo de titulación a mis padres Marcelo Cartagena, Magdalena Ocampo y hermana Melanye Cartagena ya que son mi motor de vida y el principal motivo de mi inspiración, y les doy las gracias por haberme entregado toda su confianza y apoyo incondicional día a día durante el transcurso del desarrollo del mismo.

También dedico este trabajo de grado al resto de mi familia, profesores y amigos que de una u otra forma confiaron en mí y contribuyeron incondicionalmente en mi desarrollo profesional con sus palabras de aliento, consejos y enseñanzas.



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

Agradezco primeramente a Dios por haberme dado la salud y vida, así como la sabiduría, paciencia, y sobre todo la constancia de trabajo y esfuerzo que he puesto a diario para alcanzar mis objetivos. A la UNIVERSIDAD TÉCNICA DEL NORTE por darme la oportunidad de estudiar y ser un profesional.

También me gustaría agradecer a mis profesores durante toda mi carrera profesional porque todos han aportado con un granito de arena a mi formación, y en especial agradezco al Ingeniero Paul Rosero quien fue participe directo en el desarrollo de este proyecto con sus consejos, su enseñanza y más que todo por su amistad.

CONTENIDO

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.....	IV
CERTIFICACIÓN	V
CONSTANCIAS.....	VI
DEDICATORIA	VII
AGRADECIMIENTO	VIII
CONTENIDO	IX
ÍNDICE DE FIGURAS.....	XVII
ÍNDICE DE TABLAS	XXIII
ÍNDICE DE ECUACIONES	XXIV
RESUMEN	XXV
ABSTRACT.....	XXVI
PRESENTACIÓN.....	XXVII
CAPITULO I. ANTECEDENTES	1
1.1 TEMA.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA	1
1.3 OBJETIVOS	2
1.3.1 OBJETIVO GENERAL	2
1.3.2 OBJETIVOS ESPECÍFICOS	2

1.4 ALCANCE.....	3
1.5 JUSTIFICACIÓN	4
CAPÍTULO II. FUNDAMENTACIÓN TEÓRICA	7
2.1 TEORÍAS DEL APRENDIZAJE	7
2.1.1 TEORÍA CONDUCTISTA	8
2.1.2 TEORÍA COGNITIVISTA	9
2.1.2.1 Teoría del Desarrollo Cognitivo de Jean Piaget.....	9
2.1.2.1.1 Periodo Sensomotor	10
2.1.2.1.2 Periodo Pre-Operacional.....	11
2.1.2.1.3 Periodo de las Operaciones Concretas.....	12
2.1.2.1.4 Periodo de las Operaciones Formales.....	13
2.1.2.2 Teoría del Desarrollo Cognitivo de Lev Vygotsky	13
2.1.3 TEORÍA CONSTRUCTIVISTA.....	15
2.2 CARACTERÍSTICAS EVOLUTIVAS DE LOS NIÑOS DE 4 A 7 AÑOS	19
2.2.1 NIÑOS CON CAPACIDADES ESPECIALES	21
2.3 INCIDENCIA DE LOS JUGUETES DIDÁCTICOS EN LOS NIÑOS.....	22
2.3.1 INTRODUCCIÓN.....	22
2.3.2 IMPORTANCIA DE LOS JUGUETES DIDÁCTICOS	22
2.3.3 OBJETIVOS DE LOS JUGUETES DIDÁCTICOS	24
2.3.3.1 Propósito Didáctico.....	24
2.3.3.2 Acción Lúdica	25
2.3.3.3 Reglas del Juego.....	25
2.3.4 APRENDER JUGANDO	25
2.4 LOS NIÑOS Y LA TECNOLOGÍA	27
2.5 PROGRAMACIÓN PARA NIÑOS	29

2.5.1 INTRODUCCIÓN.....	29
2.5.2 CONCEPTOS BÁSICOS	30
2.5.2.1 Programa	30
2.5.2.1.1 Fases de creación de un programa	31
2.5.2.2 Lenguaje de programación.....	32
2.5.2.3 Algoritmo	32
2.5.2.4 Secuencia.....	32
2.5.2.5 Estructuras de Control.....	32
2.5.2.5.1 Estructuras de Control Condicional.....	33
2.5.2.5.2 Estructura de Control Bucles o Repeticiones	33
2.5.3 HERRAMIENTAS PARA ENSEÑAR A PROGRAMAR	34
2.5.3.1 CODE.ORG Aprende a Programar con los Angry Birds.....	34
2.5.3.2 SCRATCH	35
2.5.3.3 LOGO.....	36
2.5.3.4 LEGO MindStorms	36
2.5.4 VENTAJAS DE ENSEÑAR A PROGRAMAR A LOS NIÑOS	37
2.6 PLATAFORMA ARDUINO	38
2.6.1 VENTAJAS DE LA PLATAFORMA	39
2.6.2 MODELOS DE PLACAS ARDUINO.....	40
2.6.2.1 Arduino UNO.....	41
2.6.2.2 Arduino Zero.....	42
2.6.2.3 Arduino YUN.....	43
2.6.2.4 Arduino NANO.....	44
2.6.2.5 Arduino MEGA.....	45
2.6.2.6 Arduino MINI-PRO	46

2.6.2.7 Comparativa de Modelos Arduino	47
2.6.3 IDE (ENTORNO DE DESARROLLO INTEGRADO).....	49
2.6.4 FUNCIONALIDADES PRINCIPALES	50
2.6.4.1 Entradas/Salidas Digitales.....	50
2.6.4.2 Entradas Analógicas.....	52
2.6.4.3 Salidas PWM.....	52
2.6.4.4 Comunicación Serial	54
2.6.4.4.1 Tipos de comunicación serial	54
2.6.4.4.1.1 Comunicación Serial Síncrona.....	54
2.6.4.4.1.2 Comunicación Serial Asíncrona.....	55
2.6.4.4.2 Consideraciones de la comunicación serial de Arduino	56
2.6.4.4.3 Tecnologías Inalámbricas Compatibles	57
2.6.4.4.3.1 Wi-Fi.....	57
2.6.4.4.3.2 ZigBee.....	60
2.6.4.4.3.3 Bluetooth.....	62
2.6.4.4.4 Tabla Comparativa de Tecnologías Inalámbricas.....	65
2.7 SENSORES Y ACTUADORES	67
2.7.1 SENSORES	67
2.7.1.1 Clasificación de los sensores según la señal de salida	67
2.7.1.1.1 Sensores Analógicos.....	68
2.7.1.1.2 Sensores Digitales.....	68
2.7.1.2 Sensores Ultrasónicos de Distancia	69
2.7.1.2.1 Sensor Ultrasónico HC-SR04	69
2.7.1.3 Sensores Encoder	71
2.7.1.3.1 Tipos de Encoder	72

2.7.1.3.1.1 Encoder Incremental	72
2.7.1.3.1.2 Encoder Absolutos	73
2.7.1.3.2Modulo Sensor Encoder FZ0888.....	73
2.7.2 ACTUADORES	74
2.7.2.1 Motor de Corriente Directa	75
2.7.2.2 LED (Diodo Emisor de Luz).....	75
2.8 COMPONENTES ELECTRÓNICOS	76
2.8.1 RESISTENCIAS	77
2.8.2 REGULADOR LM7805	78
2.8.3 MULTIPLEXOR HEF4051B.....	79
2.8.4 BATERÍAS RECARGABLES.....	80
2.8.4.1 Batería de Níquel de Cadmio (Ni CD).....	81
2.8.4.2 Batería de Níquel Metal hidruro (Minh)	82
2.8.4.3 Batería de Ion de Litio (Li-ion).....	82
2.8.4.4 Batería de Polímero de Litio (Lio).....	83
2.8.4.5 Comparativa de los tipos de Baterías recargables.....	83
2.9 HERRAMIENTAS DE SOFTWARE.....	84
2.9.1 ISIS PROTEUS	84
2.9.2 EAGLE.....	84
2.10MATERIAL DE LOS JUGUETES.....	85
CAPÍTULO III. DESARROLLO Y PRUEBAS	87
3.1 DESCRIPCIÓN GENERAL DEL PROTOTIPO	87
3.1.1 ELECCIÓN DE LAS PLACAS ARDUINO.....	88
3.1.2 ELECCIÓN DE LA TECNOLOGÍA INALÁMBRICA.....	90
3.1.3 ELECCIÓN DEL MATERIAL PARA LAS CARCASAS DE JED-PRO	92

3.1.4 DIAGRAMA DE BLOQUES	93
3.2 DISEÑO DE LA LÓGICA DE PROGRAMACIÓN TANGIBLE PARA NIÑOS	96
3.2.1 PROGRAMACIÓN TANGIBLE.....	96
3.2.2 LÓGICA DE PROGRAMACIÓN	98
3.2.2.1 Especificaciones	98
3.2.2.2 Estructuras.....	99
3.2.2.2.1 Estructura Simple.....	99
3.2.2.2.2 Estructura de Control Repetitiva	100
3.2.2.2.3 Estructura de Control Condicional	101
3.3 PRIMER PROTOTIPO DE JED-PRO.....	102
3.4 DISEÑO DEL PROTOTIPO FINAL DE JED-PRO.....	105
3.4.1 INTERFAZ DE PROGRAMACIÓN	105
3.4.1.1 Diagrama De Flujo.....	106
3.4.1.2 Esquema de los circuitos electrónicos.....	107
3.4.1.3 Simulación	110
3.4.1.4 Diseño del circuito Impreso	111
3.4.1.5 Implementación del circuito.....	114
3.4.1.6 Configuración del Módulo Bluetooth como Maestro	115
3.4.1.6.1 Comandos AT para configurar como Maestro	116
3.4.1.6.2 Pasos para configurar como Maestro.....	117
3.4.1.7 Construcción de la Interfaz	119
3.4.2 ROBOT JED-PRO.....	121
3.4.2.1 Diagrama de flujo.....	122
3.4.2.2 Esquema del circuito electrónico	124
3.4.2.3 Simulación	125

3.4.2.4	Diseño del circuito Impreso	125
3.4.2.5	Implementación del circuito.....	128
3.4.2.6	Configuración del Módulo Bluetooth como esclavo	129
3.4.2.7	Construcción del Robot.....	130
3.4.3	BLOQUE DE INSTRUCCIONES	133
3.4.3.1	Elaboración de Fichas de Instrucciones	134
3.5	ANÁLISIS DEL PROTOTIPO JED-PRO	137
3.5.1	ANÁLISIS DE LA PROGRAMACIÓN TANGIBLE.....	138
3.5.2	ANÁLISIS DE LA COMUNICACIÓN BLUETOOTH.....	138
3.5.3	ANÁLISIS DEL ROBOT.....	139
3.5.4	MANUAL DE USUARIO	140
3.5.4.1	Introducción	140
3.5.4.2	Elementos de JED-Pro	141
3.5.4.2.1	Robot.....	141
3.5.4.2.2	Interfaz de Programación.....	142
3.5.4.2.3	Bloque de instrucciones	143
3.5.4.3	Empezar a jugar con JED-Pro	144
3.5.4.3.1	Vincular La Interfaz De Programación con El Robot	144
3.5.4.3.2	Generar una Secuencia de Instrucciones	145
3.5.4.3.3	Estructura de Control Repetitiva	146
3.5.4.3.4	Estructura de Control Condicional	147
3.5.4.4	Actividades Sugeridas	147
3.5.4.4.1	Actividad #1. Presentación del Juguete al niño	147
3.5.4.4.1.1	Presentación del Robot.....	148
3.5.4.4.1.2	Presentación de la Interfaz de Programación	148

3.5.4.4.1.3 Presentación del Bloque de Instrucciones.....	148
3.5.4.4.2 Actividad #2. Identificación de Instrucciones Básicas	148
3.5.4.4.3 Actividad #3. Secuencia de dos instrucciones	149
3.5.4.4.4 Actividad #4. Secuencia de Tres Instrucciones	150
3.5.4.4.5 Actividad #5. Presentación de las instrucciones de control repetitivas	151
3.5.4.4.6 Actividad #6. Aplicación de Estructuras de Control Repetitivas	152
3.5.4.4.7 Actividad #7. Estructura de Control Condicional.....	153
3.6 PRUEBAS Y RESULTADOS DEL PROTOTIPO JED-PRO.....	154
3.6.1 PRUEBAS CON NIÑOS DE 4 A 7 AÑOS	157
CAPÍTULO IV. ANÁLISIS ECONÓMICO	161
4.1 COSTO DEL HARDWARE.....	161
4.2 COSTO DEL SOFTWARE UTILIZADO	162
4.3 BENEFICIOS	162
4.4 MODELO DE NEGOCIO CANVAS	164
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....	166
5.1 CONCLUSIONES	166
5.2 RECOMENDACIONES.....	169
BIBLIOGRAFÍA	170
ANEXO 01 PROGRAMACIÓN DE LA INTERFAZ DE PROGRAMACIÓN	175
ANEXO 02. PROGRAMACIÓN DEL ROBOT.....	179

ÍNDICE DE FIGURAS

Figura 1. Jean Piaget.....	10
Figura 2. Niños en Periodo Sensomotor	11
Figura 3. Niños en Periodo Pre-operacional	12
Figura 4. Niños en Periodo de Operaciones concretas	13
Figura 5. Niños en Periodo de Operaciones formales.....	13
Figura 6. Lev Vygotsky Psicólogo Ruso	14
Figura 7. Sintaxis Estructura de Control "IF"	33
Figura 8. Sintaxis Estructura "FOR".....	34
Figura 9. Angry Bird de CODE.ORG.....	35
Figura 10. Interfaz gráfica de SCRATCH	35
Figura 11. Interfaz inicial de LOGO.....	36
Figura 12. Robot Programable Lego MindStorms.....	36
Figura 13. Placa Arduino UNO	41
Figura 14. Placa Arduino Zero.....	42
Figura 15. Placa Arduino YUN	43
Figura 16. Placa Arduino NANO.....	44
Figura 17. Placa Arduino MEGA	45
Figura 18. Placa Arduino Mini PRO.....	46
Figura 19. Interfaz gráfica del IDE de Arduino	49
Figura 20. Configuraciones PULL-UP y PULL-DOWN	51
Figura 21. Representación de valores analógicos	53
Figura 22. Comunicación serial síncrona.....	54
Figura 23. Comunicación serial Asíncrona.....	55
Figura 24. Logo Wi-Fi	58

Figura 25. Modulo WiFly	59
Figura 26. Logo ZigBee.....	60
Figura 27. Modulo XBee	61
Figura 28. Logo de Bluetooth	62
Figura 29. Espectro Ensanchado por Salto de Frecuencia.....	63
Figura 30. Clases de Bluetooth	63
Figura 31. Modulo Bluetooth HC-05.....	64
Figura 32. Sensor Ultrasónico HC-SR04.....	70
Figura 33. Funcionamiento del sensor HC-SR04	71
Figura 34. Sensor Encoder.....	72
Figura 35. Encoder Incrementales	72
Figura 36. Encoder Absoluto	73
Figura 37. Módulo Encoder FZ0888	73
Figura 38. Ejemplos de Actuadores Electrónicos	74
Figura 39. Elementos de un Motor de Corriente directa.....	75
Figura 40. Diodo Emisor de Luz.....	76
Figura 41. Componentes Electrónicos	76
Figura 42. Símbolo y aspecto común de la resistencia	77
Figura 43. Indicadores del Valor de una resistencia	77
Figura 44. Regulador LM7805	78
Figura 45. Diagrama de Pines LM7805	79
Figura 46. Funcionamiento Multiplexor	79
Figura 47. Chip HEF4051.....	80
Figura 48. Tabla de verdad 4051	80
Figura 49. Baterías Recargables	81

Figura 50. Ejemplos de baterías de NiCd	81
Figura 51. Ejemplos de Baterías de NiMH.....	82
Figura 52. Ejemplos de baterías de Ion de Litio	82
Figura 53. Ejemplo Batería LiPo	83
Figura 54. Interfaz de Inicio de Proteus.....	84
Figura 55. Interfaz de inicio de EAGLE.....	85
Figura 56. a) Arduino Mini PRO vista frontal. b) Arduino Mini PRO vista posterior.....	89
Figura 57. Distribución de Pines Arduino Mini PRO.....	89
Figura 58. Modulo Bluetooth HC-05 a) vista frontal. b) vista posterior	91
Figura 59. Tableros de MDF.....	93
Figura 60. Diagrama de Bloques del Prototipo JED-Pro.....	94
Figura 61. Ejemplo de Estructura Simple	100
Figura 62. Sintaxis Estructura de Control Repetitiva	100
Figura 63. Ejemplo de Estructura de Control Repetitiva.....	101
Figura 64. Sintaxis Estructura de Control Condicional	101
Figura 65 Ejemplo Estructura de Control Condicional.....	102
Figura 66. Primer Prototipo de JED-Pro.....	102
Figura 67. Interfaz de Programación del Primer prototipo de JED-Pro	103
Figura 68. Prototipo de pruebas del Robot JED-Pro.....	104
Figura 69. Diseño del Primer Prototipo de JED-Pro.....	105
Figura 70. Diagrama de Flujo Interfaz de Programación	106
Figura 71. Divisor de Voltaje Resistivo.....	107
Figura 72. Circuito Principal.....	108
Figura 73. Circuito de Entrada.....	109
Figura 74. Simulación en el Software Proteus.....	110

Figura 75. Terminal Virtual de Proteus	111
Figura 76. Circuito de Multiplexación.....	111
Figura 77. Esquema del circuito de la interfaz de programación de JED-Pro realizado en Eagle 6.5.....	112
Figura 78. a) Diseño del circuito impreso b) Ubicación de dispositivos y elementos electrónicos	113
Figura 79. Circuito Transferido a la Baquelita	113
Figura 80. Proceso de Quemado de la Baquelita 1	114
Figura 81. Baquelita de la Interfaz Finalizada	114
Figura 82. Perforación de la baquelita	115
Figura 83. Circuito en baquelita terminado	115
Figura 84. Diagrama de conexión para configurar el modulo bluetooth HC-05	116
Figura 85. Nueva Conexión	117
Figura 86. Selección del Puerto COM	118
Figura 87. Configuración del Puerto COM.....	118
Figura 88. Configuración de HyperTerminal.....	118
Figura 89. Verificación de la Configuración en HyperTerminal	119
Figura 90. Referencia para el tamaño de la interfaz de programación	120
Figura 91. Proceso de construcción de la interfaz de programación	120
Figura 92. Diseño Final del Prototipo de la Interfaz de programación.....	121
Figura 93. Diagrama de Flujo del Robot JED-Pro.....	122
Figura 94. Programación del Robot desarrollado en el IDE de Arduino.....	123
Figura 95. Circuito Electrónico Robot JED-Pro	124
Figura 96. Simulación del Robot JED Pro.....	125
Figura 97. Esquema para el circuito impreso en Eagle 6.5.....	126

Figura 98. a) Vista Posterior de la placa b) Ubicación de elementos	126
Figura 99. Proceso para la transferencia del circuito impreso a la baquelita.....	127
Figura 100. Circuito Impreso transferido a la baquelita	127
Figura 101. Proceso de quemado de la baquelita.....	127
Figura 102. Baquelita terminada.....	128
Figura 103. Circuito Principal del Robot implementado	128
Figura 104. Verificación de las configuraciones	130
Figura 105. Elementos del Chasis del robot	130
Figura 106. Armado del chasis	131
Figura 107. Interruptor y circuito regulador de voltaje.....	131
Figura 108. a) Incorporación del circuito principal. b) Protección de cables con material Termo- contraíble.....	132
Figura 109. Implementación de los elementos en el chasis del robot.....	133
Figura 110. Diseño final del Robot JED-Pro	133
Figura 111. Dimensiones de la base de la ficha.....	134
Figura 112 Proceso de elaboración de las fichas, alojar resistencia	135
Figura 113. Proceso de elaboración de las fichas, soldado de placas metálicas	135
Figura 114. Proceso de elaboración de las fichas, verificación del valor resistivo	135
Figura 115. Proceso de elaboración de las fichas, verificación del valor resistivo	136
Figura 116. Bloque de Instrucciones	136
Figura 117. Presentación de JED-Pro	140
Figura 118. Robot JED-Pro.....	141
Figura 119. Partes del Robot JED-Pro.....	142
Figura 120. Interfaz de Programación de JED-Pro	142
Figura 121. Partes de la Interfaz de Programación de JED-Pro	143

Figura 122. Bloque de Instrucciones	143
Figura 123. Función de las fichas de Instrucciones	144
Figura 124. Emparejamiento vía Bluetooth	144
Figura 125. Elaboración de una Secuencia de Instrucciones	145
Figura 126. Fichas de Ciclos.....	146
Figura 127. Ficha de Condición “Si hay obstáculo”	147
Figura 128. Ejemplo de la Actividad #2	149
Figura 129. Ejemplo de la Actividad #3	150
Figura 130. Ejemplo de la Actividad #4	150
Figura 131. a) Estructura del ciclo de 2 repeticiones. b) Estructura del ciclo de 3 repeticiones	151
Figura 132. Ejemplo de la Actividad #6	152
Figura 133. Estructura de la condición “Si hay Obstáculo”	153
Figura 134. Ejemplo de la Actividad #7	153
Figura 135. Pruebas con niños y niñas de 4 a 7 años.....	157
Figura 136. Modelo CANVAS de JED-Pro.....	165

ÍNDICE DE TABLAS

Tabla 1. Comparativa de las Teorías del Aprendizaje	17
Tabla 2. Características Arduino UNO	41
Tabla 3. Características Arduino Zero	42
Tabla 4. Características Arduino YUN	43
Tabla 5. Características Arduino NANO	44
Tabla 6. Características Arduino MEGA	45
Tabla 7. Características Arduino Mini PRO	46
Tabla 8. Comparativa #1 de Placas Arduino.....	47
Tabla 9. Comparativa #2 de Placas Arduino.....	48
Tabla 10. Opciones para expresar salidas lógicas en Arduino.....	51
Tabla 11. Parámetros para una comunicación Asíncrona	56
Tabla 12. Versiones del estándar 802.11	58
Tabla 13 Características Modulo WiFly	59
Tabla 14. Características del Módulo XBee	62
Tabla 15. Características Modulo HC-05	64
Tabla 16. Diferencias entre Tecnologías Inalámbricas	65
Tabla 17. Características Modulo HC-SR04.....	71
Tabla 18. Características del Módulo Encoder	74
Tabla 19. Código de Colores	78
Tabla 20. Comparativa de Tipos de Baterías Recargables.....	83
Tabla 21. Comparativa de Materiales	86
Tabla 22. Instrucciones y Estructuras de control	97
Tabla 23. Colores y valores resistivos de las fichas.....	98
Tabla 24. Comandos AT módulo HC-05	117
Tabla 25. Distribución de fichas y valores resistivos.....	134
Tabla 26. Valor De Conversión Análoga Digital.....	154

Tabla 27. Caracteres identificadores.....	155
Tabla 28. Activación de motores por instrucción	156
Tabla 29. Pruebas de funcionamiento del Prototipo	156
Tabla 30 Resultados generales de las pruebas con niños de 4 a 7 años	158
Tabla 31 Resultados y observaciones en las distintas edades	159
Tabla 32 Tiempos de perfeccionamiento del uso de JED-Pro	160
Tabla 33. Costos de Hardware	161
Tabla 34. Costo del Software.....	162

ÍNDICE DE ECUACIONES

Ecuación 1. Cálculo de la resolución de una Entrada Analógica	52
Ecuación 2. Cálculo de Distancia en metros.....	70
Ecuación 3. Cálculo de Distancia en centímetros.....	70
Ecuación 4. Calculo del voltaje de salida utilizando divisor resistivo.....	107

RESUMEN

El proyecto consiste en el desarrollo de un prototipo de Juguete Electrónico Didáctico que tiene como propósito la enseñanza de programación básica a niños y niñas comprendidos entre las edades de 4 a 7 años; el mismo que relaciona el aprendizaje con la diversión simplificando la programación de tal forma que los niños mediante el juego que es una actividad innata de ellos, puedan adoptar conocimientos de programación y a la vez los vaya involucrando con el mundo de la tecnología.

El Juguete Electrónico Didáctico diseñado en este trabajo involucra a los niños básicamente con tres conceptos de programación que son: instrucción, secuencia y estructuras de control (ciclos y condiciones); Los cuales son los conceptos primordiales al momento de aprender directamente la lógica de programación. El propósito del juego es programar secuencias de instrucciones en una interfaz de programación tangible para guiar un pequeño robot hacia un destino o meta propuesta a manera de reto, de tal forma que el niño desarrolle su creatividad, imaginación y a la vez se involucre en la programación de una forma lúdica.

Este prototipo está basado en la plataforma Arduino y contiene una lógica de programación tangible; la misma que permite a los niños programar el robot de forma inalámbrica mediante la tecnología Bluetooth y sin la necesidad de un computador o dispositivo inteligente, únicamente insertando pequeñas fichas de madera, lo que le da simplicidad al momento de la acción de jugar y aprender a programar.

ABSTRACT

The project consists of the creating of a prototype of Educational Electronic Toy which aim is the teaching of basic programming for children between ages from 4 to 7; It relates learning with fun simplifying the programming, so children through the game that is an innate activity, they could adopt programming knowledge and at the same time involving them with the world of technology.

Educational electronic toy designed in this work involves children basically with three concepts of programming which are: instruction, sequence and control structures (cycles and conditions); which are the key concepts when they learn directly the programming approach. The purpose of the game is to program sequences of instructions in a tangible programming interface to guide a robot toward a destination or a proposed goal by way of challenge, so that children develop their creativity and imagination, and at the same time they will be involved in programming in a playful way.

This prototype is based on the Arduino platform and it contains a tangible programming approach that allows children to program the robot wirelessly by Bluetooth technology without a computer or a smart device, only by inserting small pieces of wood, which gives simplicity at the time of the action of playing and learning how to program.

PRESENTACIÓN

En la actualidad con los grandes avances tecnológicos, la producción de sistemas electrónicos para uso diario de las personas y la internet; Los niños y jóvenes ecuatorianos deben tener una actitud positiva de cambio, donde no se limiten a ser consumidores, sino que sean los entes productores de nuevas tecnologías y retribuyan a los procesos educativos en los que están inmersos para la mejora de los mismos.

Las plataformas libres nos dan la oportunidad de liberar conocimiento y con las directrices adecuadas el Ecuador puede ser un potencial educativo e investigativo si se proporciona tecnologías a bajo costo a todo niño que desea aprender, sin olvidarse de la diversión y la creatividad que conlleva el ser niño. El juguete didáctico propuesto en este trabajo está diseñado con la intención de proporcionar herramientas de libre acceso para educar a niños del país y motivarlos por la innovación y el desarrollo tecnológico desde sus primeras etapas de vida.

El presente trabajo contiene una investigación de las teorías del aprendizaje y conceptos básicos de programación para niños; la misma que permite entender el comportamiento de los niños ante el aprendizaje y las ventajas de enseñar programación desde tempranas edades. Más adelante se detallan los elementos de software y hardware utilizados para el diseño del prototipo y cada uno de los pasos de su implementación. Después se realiza un análisis de los costos y beneficios que conlleva la elaboración del Juguete Didáctico y la enseñanza de programación a niños y finaliza con las conclusiones y recomendaciones obtenidas en la realización del proyecto.

CAPITULO I. ANTECEDENTES

1.1 TEMA

JUGUETE ELECTRÓNICO DIDÁCTICO, COMO ELEMENTO DE APOYO PARA LA ENSEÑANZA DE PROGRAMACIÓN A NIÑOS Y NIÑAS DE 4 A 7 AÑOS.

1.2 PLANTEAMIENTO DEL PROBLEMA

Los primeros años de vida estudiantil de los niños es una etapa de gran importancia ya que les permite ir adquiriendo habilidades e intereses propios tanto motrices como sociales, sin embargo se requieren metodologías y elementos adecuados para que los niños puedan explotar capacidades propias de su edad, y aunque se tiene una variedad de instrumentos enfocados a la educación no existe uno que se utilice directamente para inculcar las nuevas tendencias de innovación y desarrollo tecnológico desde tempranas edades.

Las instituciones educativas, padres de familia y principalmente los docentes no usan herramientas didácticas que mezclen el aprendizaje con la diversión y a la vez permitan a los niños fortalecer y aumentar las posibilidades de progreso investigativo desde sus inicios estudiantiles, por lo que lamentablemente se sigue utilizando materiales que no se enfocan en tendencias modernas de educación.

El no poseer elementos didácticos que incentiven a los niños por la innovación y el desarrollo tecnológico, aprovechando su época de curiosidad cognitiva y su interés entusiasta por adquirir conocimiento, disminuye considerablemente la posibilidad que cuando estén jóvenes se interesen por la creación de tecnología, lo que conlleva al país a un bajo nivel de desarrollo tecnológico y por ende a seguir siendo consumidores.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar un prototipo de Juguete Didáctico utilizando plataformas libres para la enseñanza de lógica de programación a niños y niñas de 4 a 7 años que provea estrategias de resolución de problemas y conceptos básicos de razonamiento lógico de programación.

1.3.2 OBJETIVOS ESPECÍFICOS

- Analizar acerca de la capacidad de aprendizaje de niños y niñas de 4 a 7 años para determinar la importancia y factibilidad de enseñar lógica de programación desde tempranas edades.
- Investigar la incidencia de los juguetes didácticos en los niños mediante diferentes técnicas de investigación que permitan identificar los beneficios de incorporarlos en la educación infantil.
- Diseñar una lógica de programación tangible mediante la utilización de software y hardware libre que permita familiarizar a los niños con una instrucción, una secuencia y un algoritmo de programación a través del juguete didáctico.
- Desarrollar guías y manuales de operación del juguete didáctico con retos que potencialicen el aprendizaje de lógica de programación en los niños.
- Construir el prototipo y ejecutar pruebas de funcionamiento con una muestra de niños comprendidos entre las edades de 4 a 7 años para documentar los resultados de aprendizaje obtenidos.

1.4 ALCANCE

El trabajo planteado consiste en el diseño y construcción de un Juguete didáctico que se basa en un tablero con una lógica de programación tangible que tiene como finalidad simplificar la programación de tal forma que los más pequeños del hogar vayan involucrándose en el mundo del desarrollo informático indirectamente sin necesidad de un computador o dispositivos inteligentes; El niño al insertar piezas con instrucciones simples y en un orden correcto podrá guiar y dar movimiento a un pequeño robot electrónico conectado inalámbricamente al tablero, lo que les permitirá de una forma divertida ir adquiriendo conocimientos básicos de lo que es la lógica de programación y dándole la capacidad de resolución de problemas en caso de que no funcione como ellos quieren.

El juguete consta de tres elementos principales, el primero es el tablero de comunicación que es la interfaz en la que va a programar el niño insertando las piezas de instrucciones, este tablero posee un microcontrolador que se encarga de procesar una secuencia de hasta 16 instrucciones consecutivas, leyendo el valor resistivo de las fichas para luego enviar los datos a través de una comunicación inalámbrica al segundo elemento, un pequeño robot representado por un carrito que consta de un segundo microcontrolador, que se encarga de recibir, interpretar y ejecutar las instrucciones, y el tercer elemento corresponde al bloque de instrucciones compuesto por un total de 27 fichas distribuidas de la siguiente manera: doce fichas de instrucciones básicas que son adelante, izquierda, derecha (cuatro de cada una), dos fichas de ciclo de 2 repeticiones, dos fichas de ciclo de 3 repeticiones, una ficha con la condición “SI Encuentra Obstáculo” y diez fichas de abrir y cerrar (indican donde inicia y donde termina una condición o ciclo).

Para el desarrollo de este trabajo primeramente se va a realizar un análisis acerca de las capacidad de aprendizaje, tanto de niños con desarrollo normal como de niños con capacidades especiales, comprendidos entre las edades de 4 a 7 años junto con una investigación de la incidencia de los juguetes didácticos en los niños y algunos conceptos básicos de programación que permita determinar la factibilidad de enseñar esta materia en estas edades y cual va ser la incidencia de su aprendizaje.

Luego se procederá con la construcción del prototipo haciendo uso de software y hardware libre. En lo que corresponde a la parte electrónica tanto del tablero como del robot será implementada en baquelita y las carcasas o parte externa del juguete se van a realizar en madera considerando no dejar circuitos o cables expuestos que puedan afectar la integridad de los niños.

Finalmente una vez culminada la construcción del prototipo se ejecutaran pruebas y se evaluará su funcionamiento con una muestra de niños comprendidos entre las edades de 4 a 7 años que permitan sacar conclusiones de factibilidad del producto y de cuál fue su incidencia en ellos.

1.5 JUSTIFICACIÓN

Las nuevas propuestas de aprendizaje impulsadas por el Gobierno Nacional están enfocadas en dotar a los estudiantes de habilidades que promuevan la investigación desde tempranas edades, es por esta razón y en busca con cumplir la misión de la Universidad Técnica del Norte, que fomenta e impulsa el desarrollo tecnológico, se ha planteado este proyecto de innovación tecnológica, desarrollando un juguete didáctico innovador que se enfoque en el ámbito de la enseñanza de programación a niños, que junto con la implementación de sistemas

electrónicos son pilares fundamentales para el progreso tecnológico en la actualidad. (PNBV, 2013-2017).

Según Code.org que es una institución sin fines de lucro que tiene como objetivo difundir la programación como parte de la educación básica de los jóvenes existe una gran demanda de programadores a nivel mundial y calcula que para el 2020 la demanda aumentaría, este proyecto impulsa la enseñanza de programación en niños de 4 a 7 años ya que son las edades en la que los niños tienen una capacidad asombrosa para adquirir conocimientos y el propósito de enseñarles programación no es únicamente despertar el interés por una de las profesiones del futuro sino que también la programación les permitirá tener la capacidades de resolución de problemas y mejorar su pensamiento lógico y abstracto. (Ibáñez, 2013) (Ministerio de Educación, 2014).

La metodología de este proyecto busca contribuir con el PNBV¹ que busca reducir la brecha tecnológica desarrollando un sistema electrónico con base a software y hardware libre para generar formas de enseñanza diferentes y realizar proyectos con base en ciencia y tecnológica al servicio de la comunidad enfocándose en los futuros profesionales del país que son los niños. (PNBV, 2013-2017).

En el aspecto personal este proyecto motiva la creación de nuevos sistemas electrónicos siempre enfocados en aportar con la sociedad y su progreso tecnológico ya que según un estudio realizado por el REACES² en Ecuador tenemos aún muy bajos niveles de innovación

¹ PNBV= Plan Nacional del Buen Vivir

² REACES= Red Ecuatoriana de Aseguramiento y Calidad de la Educación Superior

y desarrollo tecnológico y nos limitamos a ser simples consumidores de tecnología.
(Rodríguez, 2012).

CAPÍTULO II. FUNDAMENTACIÓN TEÓRICA

El presente capítulo de fundamentación teórica inicia con un análisis de las teorías del aprendizaje, el cual brinda pautas del comportamiento de los niños ante el aprendizaje y el desarrollo de la inteligencia en sus diferentes etapas; Este análisis también permite tomar acciones adecuadas en el momento de trabajar con niños ya que se pretende establecer fundamentos psicológicos que justifiquen la propuesta innovadora del desarrollo de un Juguete electrónico didáctico para la enseñanza de lógica de programación a niños y niñas de 4 a 7 años, que facilite y promueva la innovación tecnológica en los niños desde su etapa preescolar. Luego se revisa la incidencia de los juguetes didácticos en los niños y por último hace una recopilación de información de los elementos de software y hardware utilizados para la implementación del prototipo.

2.1 TEORÍAS DEL APRENDIZAJE

En la actualidad toda practica pedagógica o de enseñanza necesita tener una base de investigación teórica que presente bases o conceptos de la forma en que las personas o en este caso los niños aprenden. Los conceptos recaudados para conocer los resultados del comportamiento de los niños ante el aprendizaje pueden ser científicos o empíricos. (DONGO, 2008)

Desde los principios de la educación han existido algunas teorías del comportamiento humano con relación a su forma de aprender, los métodos de enseñanza han sido muy diversos con varios criterios muy válidos en relación a la era que se vivía, la enseñanza se basaba en el método del conductismo, donde el estudiante es un receptor de información y el docente se vuelve la fuente de la información, el espacio físico especifica que debe mantener un orden determinado y es un proceso muy metodista donde el estudiante especialmente usa su memoria

y no su razonamiento, por ende se entendía que la persona que era capaz de memorizar más información era considerado “inteligente”.

Los procesos educativos han ido mejorando a la par con el acceso a la información que se posee por medio de las tecnologías de la información, donde los estudiantes tienen mayor facilidad de poseer información sea valedera o no; estos estudios llevaron a una gran evolución de la educación donde dio resultado el constructivismo, ésta nueva metodología explica que los estudiantes deben satisfacer sus necesidades de conocimiento bajo experimentos o problemas propuestos en el área, el docente se vuelve un facilitador de información y el estudiante construye su propio conocimiento a base de la información recolectada y la experiencia adquirida en el proceso.

2.1.1 TEORÍA CONDUCTISTA

La teoría conductista inicia de un concepto empirista de la conducta, su mecanismo central del aprendizaje es el asociación de las cosas, se basa en los estudios del aprendizaje mediante condicionamiento (la secuencia básica es la de estímulo respuesta) y considera innecesario el estudio de los procesos mentales superiores para la comprensión de la conducta humana. (Santana, 2005)

Esta teoría defiende plenamente el cumplimiento de los procedimientos en el proceso de enseñanza, el conductismo se enfoca en la práctica y en las conductas observables descartando completamente todos los procesos del pensamiento del individuo, condicionándolo a que se aprenda lo que se demuestra o se exhibe.

En el proceso de enseñanza en la educación actual la concepción del aprendizaje como creación de asociaciones es poco utilizado, pero la práctica y la repetición como base del

aprendizaje de destrezas es un principio reconocido, por supuesto no se debe basar en conductismo toda la enseñanza porque se caería en un reduccionismo insostenible en el tiempo por no reconocer los procesos mentales de los niños.

Para esta teoría el aprendizaje es un cambio constante de la conducta que se logra con un proceso repetitivo y práctico, es decir según esta teoría el aprendizaje se fuerza sometiendo al individuo a una continua práctica y relacionándolo con el entorno permanentemente como un adiestramiento para reforzar su destrezas.

2.1.2 TEORÍA COGNITIVISTA

El cognitivismo se fundamenta en los procesos mentales y la representación del conocimiento, más allá del cambio de conducta observable de un individuo expuesto a un estímulo-respuesta, propuesto por el conductismo. Esta teoría se enfatiza en el conocimiento en lugar de la respuesta, en la estructura mental y ve al individuo como activo, constructivo, y como aquel capaz de resolver problemas y no como un recipiente pasivo que reacciona a la estimulación. (Arcega, 2009).

2.1.2.1 Teoría del Desarrollo Cognitivo de Jean Piaget

Jean Piaget fue un psicólogo nacido en Suiza que se dio a conocer gracias a sus grandes aportes relacionados con la psicología infantil y el desarrollo de la inteligencia. Piaget fue el principal exponente de la teoría del desarrollo cognitivo y uno de los fundadores del constructivismo, que se interesó por los cambios mentales de las personas desde que son niños hasta que alcanzan su etapa de madurez.

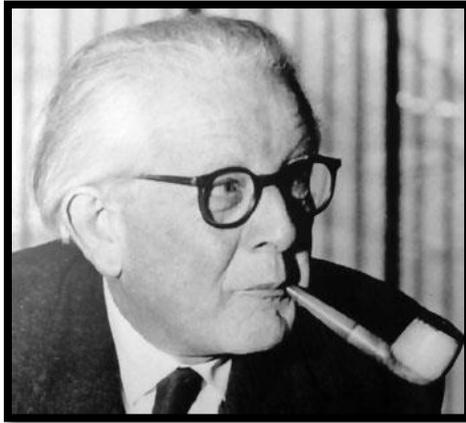


Figura 1. Jean Piaget

Fuente: <http://www.biografiasyvidas.com/biografia/p/piaget.html>

Jean Piaget en su teoría establece que el conocimiento debe ser estudiado desde una perspectiva biológica debido a que el desarrollo intelectual se forma partiendo del intelecto mismo y señala dos aspectos que son la Adaptación y la Acomodación. (Pérez, 2014).

Esta teoría establece algunos periodos del desarrollo cognitivo de los niños, planteando que los infantes captan el conocimiento por distintos canales como la observación, exploración, escucha y lectura. En el transcurso de su investigación Piaget se interesó principalmente en el hecho de que los niños no podían pensar lógicamente cuando son pequeños pero sin embargo podían resolver problemas con facilidad, lo que posteriormente lo llevo hacia la teoría constructivista donde relacionaba plenamente la capacidad cognitiva con la inteligencia.

2.1.2.1.1 Periodo Sensomotor

Este periodo de la vida infantil corresponde a la evolución del niño desde el nacimiento hasta los 2 primeros años de vida, Piaget llama a este periodo sensomotor, debido a que los infantes recién nacidos van conociendo el mundo poco a poco a través de sus sentidos y las tareas motrices de su cuerpo.

En los estudios realizados por Piaget los bebés en este periodo se caracterizan por sus reflejos innatos que cada vez se van haciendo más eficientes, después empiezan a delimitar su cuerpo y a despertar su curiosidad observando plenamente todo lo que les rodea, con el transcurso de las semanas él bebé empieza a reconocer objetos y a cogerlos con sus propias manos y a utilizar fórmulas para conseguir lo que desea junto con el desarrollo de la memoria que empieza desde este periodo, posteriormente entre las 18 y 24 meses se lleva a cabo cambios sensoriales a procesos mentales en el que puede deducir y recordar acciones.



Figura 2. Niños en Periodo Sensorial

Fuente: <http://www.psicodiagnos.es/areageneral/desarrollodelainteligenciasegunpiaget/>

2.1.2.1.2 Periodo Pre-Operacional

Se conoce como periodo pre-operacional a la etapa anterior a que un niño domine operaciones mentales lógicas, corresponde a las edades entre 2 y 7 años, este periodo a la vez se divide en dos etapas:

➤ *Etapa Pre-Conceptual*

En esta etapa se encuentran los niños entre las edades de 2 a 4 años, los niños en estas edades empiezan a desarrollar su representación simbólica manifestándola gráficamente o mediante imitaciones, son capaces de imitar palabras y entender su significado; sin embargo aún piensan que todas las cosas tienen vida y sienten.

También son muy egocéntricos, es decir, que piensan que todo es como ellos lo perciben y no entienden otros puntos de vista (DONGO, 2008).

➤ ***Etapa del Pensamiento Intuitivo***

Los niños que corresponden a esta etapa oscilan entre los 4 y 7 años, a estas edades ya pueden resolver problemas intuitivamente mediante juegos, actividades y experiencias que les permitan desarrollar su actividad mental, sin embargo aún presentan limitaciones por su falta de experiencia.



Figura 3. Niños en Periodo Pre-operacional

Fuente: <http://etapapreoperacional3.blogspot.com/2013/09/etapa-preoperacional.html>

2.1.2.1.3 Periodo de las Operaciones Concretas

El periodo de operaciones concretas corresponde a los niños de 7 a 12 años, en este periodo los niños manejan todas sus acciones de una forma más sistemática es decir, realiza sus actividades de forma ordenada y relacionándolas, para poder combinar diferentes posibilidades.(Angela Alvarez C, 1979).

También en esta etapa desarrollan un poco su pensamiento lógico, en el niño se empieza a notar claramente el desarrollo de su razonamiento siendo capaz de resolver clasificaciones de grupos y operaciones aritméticas.

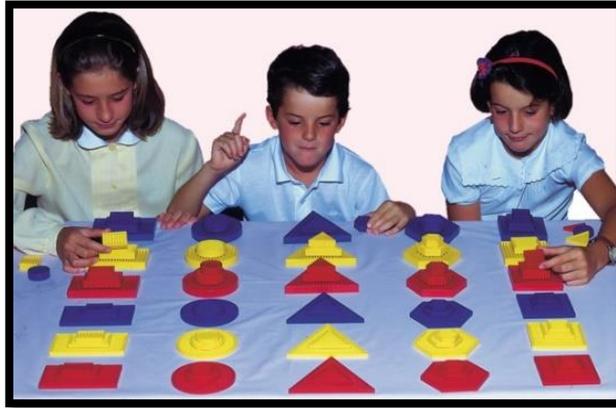


Figura 4. Niños en Periodo de Operaciones concretas

Fuente: <http://psicingrid-online7.blogspot.com/2012/07/etapas-de-operaciones-formales.html>

2.1.2.1.4 Periodo de las Operaciones Formales

El periodo de operaciones formales empieza a los 12 años, a partir de esta edad el niño ya razona, aumenta su comprensión lectora y la capacidad de expresión a través de la escritura. También los niños se vuelven más deductivos y son capaces de obtener sus propias conclusiones y pueden resolver problemas a través de su propia investigación.

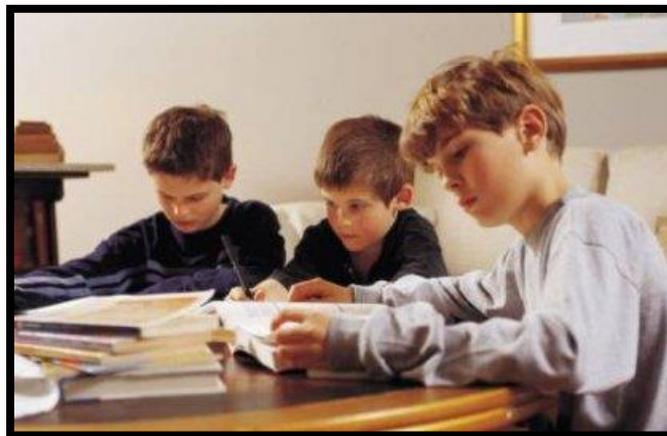


Figura 5. Niños en Periodo de Operaciones formales

Fuente: http://padres.facilísimo.com/reportajes/psicologia-infantil/el-paso-previo-a-la-adolescencia_750961.html

2.1.2.2 Teoría del Desarrollo Cognitivo de Lev Vygotsky

Lev Vygotsky fue un psicólogo ruso, en su teoría señaló las relaciones de los individuos con la sociedad y establece que no es posible entender el desarrollo del niño si no se conoce la cultura donde se cría.



Figura 6. Lev Vygotsky Psicólogo Ruso

Fuente: http://www.paidopsiquiatria.cat/files/teorias_desarrollo_cognitivo.pdf

La Teoría cognitivista de Vygotsky o también conocida como teoría sociocultural relaciona el individuo con la sociedad, en la cual dice que para comprender el desarrollo de los niños es necesario conocer la cultura y la sociedad en que ellos crecen.

Vygotsky expresó que el conocimiento no se desarrolla de modo individual como estaba propuesto en la teoría de Piaget sino que el conocimiento se desarrolla en una interacción con compañeros y adultos más conocedores, los cuales constituyen el medio principal de desarrollo intelectual y motivacional de los niños. También dijo que el niño debe tener una relación con su ambiente o su cultura de forma asertiva, activa y sobretodo muy curiosa. (Orengo, 2013).

Entre las herramientas que resalta Vygotsky como las más importantes para lograr el pensamiento de los niños están el lenguaje, los signos, símbolos, dibujos, mapas y la escritura; Estas herramientas junto con buenas metodologías de enseñanza, dotarían a los niños de habilidades para la resolución de problemas y que logren una capacidad de pensamiento más elevado (Andrade, 2010).

En si lo que expresa la teoría sociocultural de Vygotsky es que el aprendizaje del niño se relaciona plenamente con la cultura y la sociedad en que se desarrolla, participando de una forma activa en su desarrollo cognitivo y en base a su curiosidad y la combinación de ideas con otras personas el niño es capaz de aprender con mayor facilidad.

2.1.3 TEORÍA CONSTRUCTIVISTA

Los autores Jean Piaget y Lev Vygotsky nombrados anteriormente en la teoría cognitivista también son los principales entes de la teoría constructivista, por lo que estas dos teorías se asemejan y a la vez pueden confundirse ya que las dos postulan la existencia de los factores mentales, el constructivismo básicamente establece la idea que el individuo es capaz de construir su propio conocimiento a partir de conceptos básicos aprendidos.

Los niños en el modelo constructivista deben ser partícipes del conocimiento formando parte de la actividad de aprendizaje y no permanecer de manera pasiva únicamente observando lo que se les explica. Básicamente los niños aprenden con mayor facilidad cuando pueden controlar un objeto o una actividad, sin necesidad de una alfabetización rigurosa únicamente con las pautas básicas el niño va construyendo su propio conocimiento.

Las personas cuando están recibiendo alguna enseñanza no son capaces de captar o entender completamente la información, en especial si son argumentos teóricos, ni tampoco van a utilizar inmediatamente estos conocimientos en el momento en que se les enseña, por lo que se crea la necesidad individual de captar únicamente las pautas importantes para luego construir su propio conocimiento.

Para Jean Piaget los individuos van construyendo su aprendizaje a medida que se relaciona directamente con él, es decir cuando un niño recibe alguna enseñanza él no es capaz de entender completamente ya que poseen ideas nulas de la realidad funcional de ese aprendizaje, pero en el momento en que se interrelacione directamente con la actividad aprendida el niño será capaz de formar una estructura entre el conocimiento actual y el conocimiento aprendido anteriormente e interrelacionarlo y así construir su conocimiento.

El constructivismo puede considerarse una rama de la teoría cognitivista debido a que las dos trabajan con los procesos mentales de los individuos, pero estas teorías se diferencian en la forma en que toman los procesos de la mente ya que en el cognitvismo se considera a la mente como una herramienta de referencia para el mundo real y en el constructivismo se considera que el trabajo de la mente es filtrar lo que llega del mundo para producir su propia y única realidad (Newby, 1993).

Por otra parte Vygotsky aporta con el constructivismo sin dejar de lado su postura sociocultural analizando las relaciones entre el individuo y el entorno mediante cuatro niveles: el nivel ontogénico en el que se analiza las transformaciones del pensamiento y la evolución de la conducta personal, el nivel de desarrollo filogenético analizando lo relativo a la herencia y la genética de las personas, el nivel sociocultural refiriéndose a la evolución de la cultura de los individuos y por último el nivel micro genético. (Santana, 2005).

En resumen el constructivismo consiste en la necesidad de proveer a los estudiantes las herramientas adecuadas y los conceptos básicos que únicamente los guíen hacia la construcción de su propio conocimiento y les permita desarrollar sus propias ideas y procedimientos para la resolución de problemas, de esta forma los estudiantes serán capaces de corregir sus ideas y

seguir aprendiendo. Según esta teoría el individuo es capaz de reconstruir su aprendizaje en base a experiencias adquiridas y no se le debe forzar a copiar la realidad.

A continuación se muestra una tabla con las características más importantes de las teorías del aprendizaje:

Tabla 1. Comparativa de las Teorías del Aprendizaje

CARACTERÍSTICAS	CONDUCTISMO	COGNITIVISMO	CONSTRUCTIVISMO
Fundamento Teórico	Modelo Estimulo – Respuesta y reflejos condicionados	Modelos mentales y procesamiento de la información	Teoría constructivista del conocimiento
Conocimiento	Respuesta pasiva y automática a estímulos externos	Representación simbólica en la mente de los individuos	Construcción individual por interacción entre sujeto y objeto
Aprendizaje	Transmisión	Asociación	Reconstrucción
Contenidos del aprendizaje	Pre-especificados	Pre-especificados	Rechaza la pre-especificación
Contexto del aprendizaje	Aprendizaje controlado	Aprendizaje por instrucción	Aprendizaje por experiencia
Estrategias de aprendizaje	Controladas por el ambiente	Unas son específicas y otras controladas	Individuales y personales, los individuos controlan su propia instrucción
Aprendizaje activo y colaborativo	Aprendizaje pasivo y no negociado	Aprendizaje activo y no necesariamente negociado	Aprendizaje activo y negociado
Metodologías de estudio	Observación y experimentación	Análisis de tareas	Investigación, reconstrucción, análisis y experimentación
Evaluación	En función de lo enseñado	En función a lo analizado	En función a la resolución de un problema
Individuo	Pasivo	Activo	Dinámico
Interpretación personal	Otros deciden lo que el alumno debe saber	El estudiante ya trabaja su mente pero sigue recibiendo instrucciones	Los estudiantes puede construir su propio conocimiento y tendrá una interpretación personal de la resolución de problemas

Fuente: Adaptado de Santana, M. S. (2005). LA ENSEÑANZA DE LAS MATEMÁTICAS Y LAS NTIC. UNA ESTRATEGIA DE FORMACIÓN PERMANENTE. Tarragona.

El afán de realizar un análisis de las teorías del aprendizaje es conocer la reacción de los niños ante el proceso de aprendizaje y la capacidad de asimilación de conocimientos en sus diferentes etapas; y de esta forma poder brindarles una nueva herramienta que impulse la programación y el desarrollo tecnológico desde tempranas edades, con una forma diferente de proporcionarles conocimientos.

Las diferentes teorías del aprendizaje brindan pautas importantes que permitirán entender claramente la forma en que los niños pueden aprender. Una teoría puede ser más validera que otra sin embargo cada una aporta con información importante.

Los niños de acuerdo a las teorías de aprendizaje se encuentran en un proceso de continuo aprendizaje y captan con facilidad cualquier información, por lo que esta etapa debe ser aprovechada para brindarles toda la información posible, siempre tomando en cuenta que son niños y que se debe buscar las herramientas y metodologías adecuadas. También hay que tener claro que no se puede generalizar a todos los niños, ya que no todos tienen las mismas capacidades; unos pueden desarrollar capacidades intelectuales con mayor facilidad que otros.

La teoría constructivista señala que los niños pueden ser capaces de construir su conocimiento en base a experiencias, y pueden aprender y desarrollarse en cualquier edad, lo que afirma que los niños tienen la capacidad de aprender programación e involucrarse con la tecnología desde muy pequeños.

Según Vygotsky y Piaget si existe una estimulación se va a tener una respuesta, es por eso que en este trabajo se quiere incorporar un juguete didáctico que estimule y llame la atención de los niños para poder tener como respuesta el interés de los infantes por la

programación y sobre todo que adquieran conceptos básicos que les facilite su futuro aprendizaje y así irlos involucrando de a poco con la innovación y el desarrollo tecnológico.

2.2 CARACTERÍSTICAS EVOLUTIVAS DE LOS NIÑOS DE 4 A 7 AÑOS

Los niños y niñas comprendidos entre las edades de 4 a 7 años según la teoría cognitivista de Piaget se encuentran en la etapa de pensamiento intuitivo, en esta etapa desarrollan una coordinación motriz que les permite manipular herramientas y accesorios de una forma más exacta y precisa, en estas edades los niños se interesan mucho por adquirir conocimientos y se tornan muy curiosos, por lo que se interesan por actividades que les permitan descubrir, reflexionar y crear; es por esto que se deben desarrollar estrategias que aprovechen esta etapa tomando en cuenta que los niños deben desarrollar actividades acordes a sus posibilidades ya que si es muy fácil o de lo contrario es muy difícil los niños podrían perder interés.

Constantemente el pensamiento de los niños se va haciendo más analítico y lógico, siendo capaces de entender situaciones más complicadas, sacar conclusiones y realizar clasificaciones. Al mismo tiempo van evolucionando su madurez emocional, lo que les permite ser más sociables y tener la capacidad de trabajar en grupo con mayor facilidad, siendo participativos y colaboradores, también desarrollan su curiosidad cognitiva y tratan de adquirir conocimiento de todo lo que para ellos es desconocido. (Carmen, 2007).

A partir de los cuatro años los niños comienzan a interesarse por las cosas que suceden a su alrededor, les gusta explorar, comparar y sobretodo poner a prueba sus conocimientos, también a partir de esta edad se empiezan a interesar mucho por la tecnología por lo que se

debe aprovechar mediante las herramientas adecuadas para su edad poder introducirlos de a poco en este campo sin forzarlos ni fastidiarlos.

De los cuatro a los siete años se puede observar en los infantes que tienen movimientos más precisos y también son más conscientes de sus capacidades ya que se les desarrolla la memoria, la atención, la imaginación, el razonamiento, la creatividad y se inician en el pensamiento lógico matemático; aspectos que se les puede desarrollar con mayor facilidad de acuerdo al tipo de estímulo y motivación que reciben los niños de parte de sus profesores y padres de familia, también depende de los métodos y herramientas que se utilicen para enseñarles. (Carmen, 2007).

Los infantes son muy capaces de aprender con facilidad sin embargo es muy importante su estado emocional para que el niño sea capaz de aprender, es fundamental que el niño crezca en un ambiente tranquilo y con el aprecio de sus padres, ya que un ambiente de intranquilidad, lleno de gritos, peleas y agresiones puede perjudicar considerablemente su rendimiento.

Es primordial en el proceso de enseñanza y aprendizaje que el niño tenga interés por aprender, no se le debe obligar ni forzar a realizar actividades en las que no se siente cómodo o no le gustan ya que esto podría perjudicar psicológicamente al niño y en vez de interesarlo por la información se podría provocar que el niño aborrezca lo que se le quiere enseñar.

Aunque las capacidades de los niños en la etapa de pensamiento intuitivo son muchas aun los infantes poseen debilidades en relación a niños de mayor edad, entre esas debilidades están; El realismo, los niños aun no ven la realidad tal y como es; El Animismo, le dan vida a

todos los objetos que les rodean; El Artificialismo que se caracteriza por que los niños no son capaces de entender aun el porqué de los fenómenos naturales. (Muñoz, 1976).

El crecimiento es un proceso biológico y psicológico en donde los niños van adquiriendo conocimientos a partir de la sociedad en la que se desarrolla, es por esta razón que la intervención de los padres en el proceso de aprendizaje es de vital importancia ya que los infantes poseen demasiadas dudas sobre el entorno que los rodea y estas preguntas deben ser contestadas de la manera adecuada sin misterios o respuestas complicadas que puedan confundirlos.

2.2.1 NIÑOS CON CAPACIDADES ESPECIALES

Los niños con capacidades especiales son aquellos que poseen una discapacidad; La misma que puede ser mental, auditiva, visual, emocional, ortopédica, autismo, lesión cerebral traumática u otro impedimento que requiera de una educación personalizada y especial.

Los niños con algún tipo de discapacidad suelen tener problemas para aprender conceptos y adquirir capacidades en el ambiente de un aula tradicional, por lo que requieren de una educación personalizada que no desmerezca sus capacidades intelectuales, ya que con estrategias apropiadas y acordes al tipo de discapacidad, ellos pueden fortalecer sus discapacidades y superar los obstáculos, y pueden llegar a obtener los mismos conocimientos de un niño con capacidad normal.

Resulta complicado analizar a los niños con capacidades especiales de una forma general y principalmente en su capacidad de aprendizaje, ya que esto depende de muchos factores que son variables de acuerdo a la discapacidad que el niño posea; Sin embargo al igual

que los niños con capacidades normales su capacidad de aprendizaje depende mucho del ambiente social en el que el niño desarrolle y adquiere conocimientos.

Aunque los niños discapacitados requieran un trato personalizado y especial, tienen las mismas características evolutivas de cualquier niño con capacidades normales. Por lo que pueden ser incluidos en cualquier actividad de aprendizaje; Esta claro que se debe tomar algunas precauciones referentes a su discapacidad, pero también se pueden explotar ciertas capacidades propias del ser niño haciendo uso de herramientas y técnicas adecuadas.

2.3 INCIDENCIA DE LOS JUGUETES DIDÁCTICOS EN LOS NIÑOS

2.3.1 INTRODUCCIÓN

Los Juguetes Didácticos son herramientas educativas que se pueden utilizar como estrategia de enseñanza en cualquier edad, estos juguetes en el proceso de aprendizaje de los niños son muy importantes ya que con una metodología adecuada pueden permitir a los infantes desarrollar y fortalecer sus aptitudes, expectativas, conceptualización y socialización; Es por esta razón que en este trabajo se cree importante resaltar algunas razones de la importancia de incluir los juguetes en el proceso de aprendizaje de los menores.

2.3.2 IMPORTANCIA DE LOS JUGUETES DIDÁCTICOS

Utilizar juguetes como herramienta de enseñanza es una técnica muy antigua, desde épocas primitivas ya se utilizaban juguetes para desarrollar las habilidades de los niños y jóvenes para que puedan asimilar más fácilmente los procedimientos enseñados.

Los Juguetes Didácticos son elaborados con el afán de contribuir en el desarrollo de niños y niñas en aspectos relacionados con el pensamiento, el lenguaje oral y escrito, la

imaginación, la socialización, el mejor conocimiento de sí mismo y de los demás. Estos han ido creciendo su importancia debido a que los procesos de enseñanza han cambiado, evitando memorizaciones forzadas y amenazas físicas por métodos de estimulación de los sentidos y la imaginación. (Gomez & Coronel, 2011).

El juego es una actividad innata, placentera y fundamental, mediante el cual el niño empieza a aprender y explorar su entorno; Los niños jugando puede desarrollar su imaginación, su creatividad, sus experiencias, sus habilidades y lo más importante a través del juego también puede ir formando su personalidad (Montes, 2012).

Desde hace muchos años los juguetes desempeñan un papel muy significativo en el desarrollo de los niños, y aunque los tiempos han cambiado y las características de los juguetes han mejorado mucho, el propósito sigue siendo el mismo dar alegría, placer y crear oportunidades para incrementar el aprendizaje y el desarrollo infantil. (Benitez, 2010).

Los juguetes siempre tiene un propósito que les permite a los niños aprender, pero lo que les hace especiales es que le añaden a la actividad de aprendizaje un poco de fantasía, entretenimiento y emoción. (Benitez, 2010).

Es importante que los padres de familia sepan escoger un juguete adecuado para sus hijos que no subestime las capacidades, es por eso que existen juguetes con diferentes propósitos y para diferentes edades, un buen juguete debe estimular la imaginación y el aprendizaje.

Para los niños los juguetes son importantes ya que para ellos el juego es algo natural, es decir nacen con la necesidad de jugar, y el papel que cumplen es ayudar a despertar la creatividad, curiosidad, habilidad y fomentar el aprendizaje; Un juguete bien estructurado y con objetivos bien planteados sería la estrategia más adecuada para inculcar conocimiento en los niños.

2.3.3 OBJETIVOS DE LOS JUGUETES DIDÁCTICOS

Los juguetes didácticos deben contar con una serie de objetivos que permitan conocer los propósitos del juego y las metas que se desean lograr. Entre los objetivos principales de un juego o juguete didáctico está primeramente el establecimiento de un problema, el cual debe resolverse con un nivel de análisis y dificultad adecuada para despertar el interés y la comprensión mental del niño. (Chacón, 2008)

Estos juguetes deben dar la posibilidad de aumentar la capacidad de asimilación de conocimientos, es decir deben aportar con una enseñanza o tener un propósito académico. Desarrollar habilidades y hábitos en los niños, brindar un ambiente de motivación que llame la atención e impulse a los infantes por aprender mediante el juego son algunos de los objetivos. (Chacón, 2008).

Existen tres elementos que deben poseer todos los juguetes didácticos:

2.3.3.1 Propósito Didáctico

Este elemento se refiere a la intención educativa que tiene el juguete, por ejemplo en este caso es un juguete electrónico didáctico que tiene como propósito didáctico la enseñanza de lógica de programación a los niños.

2.3.3.2 Acción Lúdica

La Acción Lúdica se refiere a la actividad de jugar, ya que si no existe juego no se trata de un juguete didáctico sino se trataría de una herramienta didáctica que es algo diferente. La acción de jugar estimula al niño y despierta su interés lo que provoca que el niño voluntariamente vea la necesidad de aprender.

2.3.3.3 Reglas del Juego

Constituyen la parte organizativa del juego, las reglas son las que establecen como se debe jugar, cuales acciones están permitidas y cuales no; Las reglas del juego son muy importantes ya que van a determinar acciones justas en el juego.

2.3.4 APRENDER JUGANDO

La educación actual se ha caracterizado por su seriedad y rigidez en el momento de aplicar los conceptos de enseñanza a los alumnos, esta propuesta es muy valiosa y no se puede cuestionar, sin embargo las metodologías impartidas en la educación deben evaluarse y analizar que no es lo mismo la enseñanza a un joven o adulto que la enseñanza a un niño, el trabajo con niños debe ser más participativo, dinámico y menos agresivo. (Pers, 2009).

El Juego puede ser utilizado en propuestas educativas siempre y cuando este tenga como propósito el impartir conocimientos; Cuando se imparte conocimientos a partir de juegos estamos creando satisfacción por aprender, eliminando tensiones y facilitando el trabajo tanto del educando como del educador.

Para los niños el aprender jugando les ayuda a romper las tenciones que existen cuando se involucran por primera vez hacia un mundo desconocido, se sienten relajados y cómodos, además hacen del aprendizaje algo placentero.

En la actualizad la enseñanza mediante juegos y juguetes es un método muy valedero tanto que se han desarrollado infinidad de herramientas didácticas para la enseñanza de diferentes disciplinas como: lenguaje, ciencias sociales, matemáticas, química, física, historia, geografía, memoria, inteligencia emocional, etcétera. (Pers, 2009). Todas estas materias son muy importantes sin embargo los tiempos han cambiado y el auge de la actualidad es la tecnología, por esta razón es necesario plantear nuevas propuestas para poder enseñar a los niños temas más modernos y de interés futuro.

En conclusión la incorporación de juguetes en el proceso de enseñanza y a aprendizaje de niños es una técnica muy viable, durante muchos años han podido comprobar que el juego es algo innato de ellos por lo tanto les resulta verdaderamente fácil adquirir conocimientos lúdicamente y además les permite desarrollar otras capacidades tanto físicas como mentales.

Es importante recalcar que no todos los niños serán capaces de perfeccionar todos los juegos, esto es debido a que existen diferentes gustos, tendencias y capacidades, así que hay que tomar en cuenta que no se debe obligar a un niño a realizar actividades que no desee, la actividad del juego debe realizarse de forma voluntaria y sin presión alguna para que realmente sea beneficioso.

2.4 LOS NIÑOS Y LA TECNOLOGÍA

Los niños últimamente están creciendo rodeados de tecnología, al hablar de tecnología se refiere a teléfonos inteligentes, computadoras, tabletas electrónicas, televisores inteligentes entre otros; estos dispositivos podrían resultar una ventaja viéndolos desde un punto de vista de comodidad y entretenimiento, sin embargo estos dispositivos también pueden causar una dependencia y un consumismo irreversible.

Recientemente se ha podido ver a niños de tres años o menores localizar un programa en un teléfono o computador de sus padres, entretenerse con un video juego interactivo o encender un aparato tecnológico en el hogar, esto se debe a que los niños al estar continuamente expuestos a la tecnología, suele resultarles muy fácil utilizarla. Los niños así como tienen la necesidad de aprender a leer, a escribir, sumar, restar y calcular para que les vaya bien en la escuela y luego en el trabajo, los niños también tendrán que saber manejar computadoras y otros dispositivos tecnológicos. (Kotch, 2012).

Los padres de familia son los responsables de decidir si desean involucrar a sus hijos en la tecnología o no, en el caso de que deseen hacerlo deben tomar en cuenta todos los aspectos del desarrollo saludable del niño, además deben ser conscientes de los riesgos de salud que pueden presentar los dispositivos tecnológicos. Entre los aspectos que hay que tomar en cuenta es el tiempo que sus hijos están expuestos a pantallas ya que no se recomienda que niños pequeños pasen demasiado tiempo junto a una pantalla debido a que podría afectar el desarrollo de los ojos, algunos dispositivos tecnológicos también contiene plomo o toxinas que pueden afectar su salud. (Kotch, 2012).

Es necesario identificar los dispositivos y programas apropiados para involucrar a los niños con la tecnología, estos dispositivos no deben reemplazar ni perturbar el tiempo que los infantes pasan en actividades de comunicación, interacción y participando con otros niños y adultos. (Kotch, 2012).

Los niños en general siguen un proceso común cuando empiezan a interactuar con un dispositivo tecnológico, primero exploran el dispositivo, explorar se refiere a conocer para que sirve, lo tocan, lo golpean, lo agitan, preguntan a sus padres, satisfacen todas sus curiosidades y luego si proceden a dominarlo. Un niño es muy capaz de dominar la tecnología sin embargo es necesario alfabetizarlos de alguna forma para que conozcan que la tecnología es una ayuda para las actividades diarias pero no es una necesidad vital.

Las ciencias de la computación es una disciplina tecnológica en la cual intervienen múltiples habilidades y competencias que sirven para toda la vida y son aplicables a otras áreas del conocimiento. El enseñar esta disciplina a los niños aporta nuevas formas de razonar y solucionar problemas, es decir les ayuda a comprender el mundo que les rodea. (Elía, 2014).

Es importante resaltar las diferencias que existe entre enseñar a utilizar una computadora y enseñar computación; el saber utilizar una computadora corresponde a saber manejar editores de texto, crear presentaciones, utilizar el internet, enviar correos electrónicos, editar videos, realizar impresiones, crear archivos de texto y audio entre otros; saber utilizar estos programas no es suficiente para lograr los conocimientos, competencias y habilidades fundamentales que necesita la sociedad actual y futura. De lo contrario el saber computación implica conocer cómo funcionan los sistemas computacionales, como están diseñados y como estos se programan. (Elía, 2014).

El saber computación incluye conocer algoritmos y programación, como se realizan los procesos de almacenamiento de la información, como es la estructura física y lógica de las computadoras y cómo funcionan las redes de computadoras; esta idea nos lleva a pensar que en la educación actual principalmente en la infantil únicamente se enseña a manejar una computadora y no se enseña realmente computación.

La educación del último siglo se rige por los avances tecnológicos, por lo tanto se crea la necesidad de que existan más personas que puedan pensar en forma creativa y sean capaces de inventar soluciones innovadoras a problemas nuevos. El propósito de la educación debería ser incentivar a los estudiantes por la creación de tecnología potenciando la creatividad y la lógica de los estudiantes para evitar seguir siendo únicamente consumidores.

En varios países ya se están desarrollando proyectos que tienen como objetivo enseñar a programar a los niños. La idea es involucrarlos en la tecnología desde muy pequeños, para tener mayor probabilidad de tener mejores programadores y desarrolladores de tecnología en el futuro.

2.5 PROGRAMACIÓN PARA NIÑOS

2.5.1 INTRODUCCIÓN

Un número creciente de países ha comenzado a enseñar a sus alumnos a escribir código con el objetivo de formar a creadores digitales, y no solo a meros consumidores de contenido, potenciando así su creatividad y su mente lógica. Programar es una palabra que puede sonar aburrida, pero la cosa cambia cuando se le pregunta a un niño si le gustaría aprender a crear sus propios juguetes, aplicaciones y videojuegos. (Rojo, 2013).

Los niños navegan constantemente por Internet sin conocimientos sobre cómo desarrollar una aplicación y dedican un gran número de horas delante de las consolas sin tener ni idea de cómo se crean sus videojuegos preferidos, por esta razón es necesario alfabetizar a los infantes para que de alguna u otra forma ellos tengan una perspectiva de cómo funcionan las aplicaciones que están utilizando y así interesarlos a la creación y no solo al consumismo (Rojo, 2013).

Programar es dar una serie de instrucciones y poder ver el efecto en tiempo real. Es decir, lo que se indica mediante la instrucción se traslada a una acción y ésta acción trae consecuencias. El programador es el responsable de la acción que se llevó a cabo a partir de la instrucción que dio. Programar implica estructurar el pensamiento, las ideas, convertirlas en un proyecto de construcción para generar algo nuevo, no solamente para manipular lo que ya existe. (Elía, 2014).

En otras palabras programar es indicar paso a paso a un computador o sistema embebido el proceso que debe realizar ante alguna acción. A un conjunto de instrucciones bien estructuradas se le conoce como programa o software. Los programas son quienes permiten que estos dispositivos sepan cómo trabajar.

2.5.2 CONCEPTOS BÁSICOS

2.5.2.1 Programa

Un programa es un conjunto de instrucciones que guían a la computadora para realizar alguna actividad o resolver algún problema; en el programa se ejecutan diferentes acciones de acuerdo a los datos que se están procesando. (Román, 2011).

2.5.2.1.1 Fases de creación de un programa

➤ *Analizar el problema*

Consiste en identificar perfectamente de que se trata el problema a solucionar y cuáles son las soluciones y resultados que se esperan obtener con el programa a realizar.

➤ *Planificación*

Para resolver un problema se debe realizar una secuencia de los pasos a realizar con la finalidad de simplificar el proceso, la planificación permite conocer de antemano todo lo que se desea elaborar y en el orden que se van a ir ejecutando.

➤ *Edición*

Corresponde a la escritura del programa utilizando un lenguaje de programación.

➤ *Compilación y ejecución*

La compilación corresponde en transformar o convertir el programa escrito en algún lenguaje de programación a un lenguaje de máquina y la ejecución corresponde a poder probar el programa y visualizar su funcionamiento.

➤ *Corrección de errores*

Por lo general los programas en un principio no funcionan realmente como desea el programador por eso después de comprobar la ejecución del programa se pueden corregir las instrucciones del programa repitiendo las fases de edición, compilación y ejecución.

➤ *Documentación*

Después de realizar todas las pruebas de funcionamiento del programa y estas sean satisfactorias es muy recomendable documentar y comentar el programa con breves explicaciones que permitan conocer los argumentos de entrada y salida, y las funciones del programa.

2.5.2.2 Lenguaje de programación

Un lenguaje de programación es el medio a través del cual le comunicamos a la computadora o al controlador la secuencia de instrucciones que debe ejecutar para llevar a cabo actividades, tareas o solución de problemas. (Román, 2011). En si es el idioma mediante el cual podemos tener comunicación con el dispositivo programable.

2.5.2.3 Algoritmo

El algoritmo es una secuencia ordenada y cronológica de pasos que llevan a la solución de un problema o a la ejecución de una tarea. Los pasos del algoritmo deben tener las siguientes características: ser simples, claros, precisos, exactos, tener un orden lógico y tener un principio y un final. (Román, 2011).

2.5.2.4 Secuencia

Es un conjunto de instrucciones que se van ejecutando en el orden en que fueron escritas una a continuación de la otra.

2.5.2.5 Estructuras de Control

Son parte fundamental de cualquier lenguaje de programación. Sin ellas las instrucciones de un programa solo podrían ejecutarse en el orden en que fueron escritas.

Las estructuras de control permiten modificar este orden. (Fernando, 2009).

2.5.2.5.1 Estructuras de Control Condicional

Las estructuras de control condicionales permiten ejecutar diferentes secuencias en función a una determinada condición, es decir si se cumple la condición solicitada se ejecutan el conjunto de instrucciones.

➤ **Estructura condicional “IF”**

Este es el modelo más sencillo de estructura condicional y el más primordial y utilizado en la programación, la condición “IF” espera que se cumpla una condición, es decir si se cumple la condición especificada en esta estructura se van ejecutar las instrucciones solicitadas, de lo contrario no se ejecutan y saltan las instrucciones de esta condición.

SINTAXIS:

```
If (condición)
{
INSTRUCCIÓN1;
INSTRUCCIÓN2;
}
```

Figura 7. Sintaxis Estructura de Control "IF"
Fuente: El Autor

2.5.2.5.2 Estructura de Control Bucles o Repeticiones

Estas estructuras permiten que se ejecuten repetidamente un conjunto de instrucciones, la cantidad de repeticiones se las puede pre-definir o se puede realizar que las repeticiones se ejecuten mientras se cumple una condición. (Fernando, 2009).

➤ Estructura de Repetición “FOR”

Este modelo de estructura de repetición también es muy utilizado, permite realizar la repetición de un conjunto de instrucciones por un número pre-determinado de veces.

SINTAXIS:

```
for (int i=0; i<=x; i++)  
{  
  INSTRUCCIÓN1;  
  INSTRUCCIÓN2;  
}
```

Figura 8. Sintaxis Estructura "FOR"
Fuente: El Autor

2.5.3 HERRAMIENTAS PARA ENSEÑAR A PROGRAMAR

A nivel mundial la enseñanza de programación a niños ya es una idea muy validada en el afán de mejorar la educación y el uso de la tecnología. En los últimos años se ha creado un debate que tiene como tema central si se debe incorporar la asignatura de programación en el currículo de la educación primaria o no. Actualmente algunas ciudades europeas ya han incorporado estas asignaturas sin embargo más allá de que se incorpore o no se han creado varias herramientas para ayudar a los pequeños a aprender a programar. (Roberto, 2014).

2.5.3.1 CODE.ORG Aprende a Programar con los Angry Birds

Code.org es una organización de los Estados Unidos que cuenta con el apoyo de grandes empresas como Google, Microsoft, Facebook y Twitter; esta organización está orientada a la enseñanza de programación a todas las personas de diferentes edades en particular a los niños, su última propuesta trata de enseñar a los niños a programar para crear su propio videojuego llamado Angry Birds esta herramienta tiene el propósito de guiar un pajarito hacia un objetivo mediante programación por bloques. La programación por bloques consiste en unir o enganchar bloques que representan instrucciones, la idea de esta programación es ocultar el código de

programación el cual puede resultar inentendible para los niños y utilizar un solo bloque represente varias líneas de código de forma transparente. (code.org).

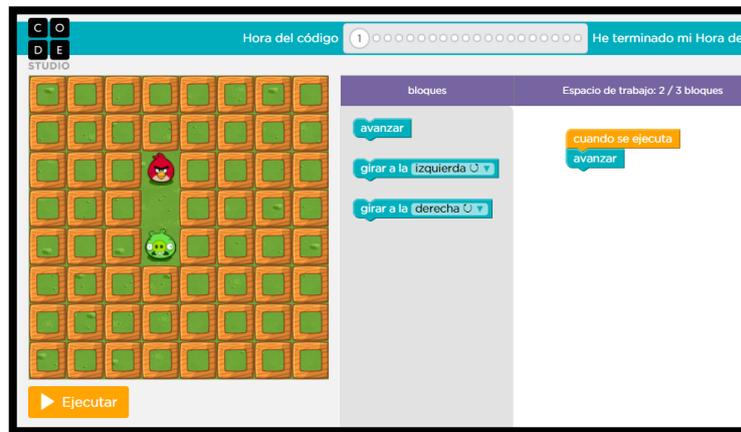


Figura 9. Angry Bird de CODE.ORG
Fuente: <https://studio.code.org/hoc/1>

2.5.3.2 SCRATCH

Scratch es una herramienta muy difundida a nivel mundial creada por el Instituto Tecnológico de Massachusetts, esta es una herramienta enfocada en los niños y jóvenes entre 8 y 16 años para que entiendan cómo funciona el código utilizando una programación a través de gráficos con una interfaz muy limpia. En Scratch los niños pueden programar sus propias historias interactivas, videojuegos y animaciones y así aprender a pensar creativamente, razonar sistemáticamente y trabajar colaborativamente. (scratch.mit.edu).

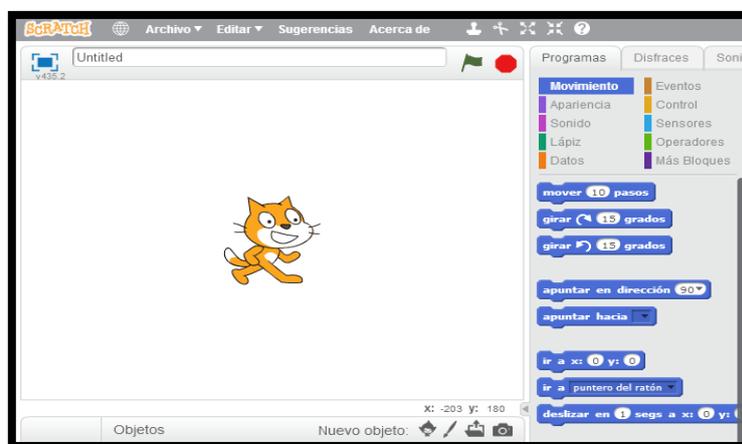


Figura 10. Interfaz gráfica de SCRATCH
Fuente: <https://scratch.mit.edu/about/>

2.5.3.3 LOGO

Logo es un lenguaje de programación diseñado con fines didácticos para aprender conceptos básicos como manejo de listas, archivos, entradas/salidas entre otros. Dispone de un intérprete de comandos que admite órdenes en español y además dispone de recursos para empezar desde cero en la programación. (Roberto, 2014)



Figura 11. Interfaz inicial de LOGO

Fuente: <http://www.bloglenovo.es/la-programacion-es-un-juego-de-ninos-o-eso-intentan-estas-catorce-propuestas/>

2.5.3.4 LEGO MindStorms

LEGO MindStorms es una línea de juguetes de robótica para niños fabricado por la empresa LEGO, que posee elementos básicos de las teorías de robótica como la unión de piezas y la programación de acciones de forma interactiva. LEGO MindStorms puede ser usado para construir un modelo de sistema integrado con partes electromecánicas controladas por computador. Prácticamente todo puede ser representado con las piezas tal como en la vida real, como un elevador o robots industriales. (Monsalve, 2015).



Figura 12. Robot Programable Lego MindStorms

Fuente: <http://www.amazon.com/LEGO-4544091-Mindstorms-NXT-2-0/dp/B001V7RF9U>

2.5.4 VENTAJAS DE ENSEÑAR A PROGRAMAR A LOS NIÑOS

En su estudio Marina Bers (2008) concluye que:

“Para un niño, ser él quien decide y dirige una acción, y que como por arte de magia una iniciativa suya deviene realidad, es doblemente satisfactorio. Le da autonomía y lo hace sentir responsable e importante. Ver y no imaginar, comprobar que dio una orden y que como consecuencia algo sucedió, demuestra su capacidad y su éxito. La motivación nace de su propio logro: él es causa y efecto de su propia decisión. Y sin embargo, no todo es éxito. Aun así resulta alentador, porque suma experiencia. Si uno se equivoca en la orden, en lo que programa, el pedido no funciona. Ahí entra la magia del tiempo real. No hay que esperar. Se aprende instantáneamente por prueba y error” (Portal EDUC.AR).

La programación es uno de los principales pilares para el desarrollo tecnológico, por lo que es importante que todas las personas sepan programar, no solo para mejorar la habilidad para entender y usar la tecnología sino que también permite ser parte de ella y ser capaz de crearla.

“Programar te ayuda a pensar” fue la frase de Steve Jobs fundador de la empresa Apple y uno de las personas más influyentes en el mundo de la tecnología. Las personas están cada vez más dependientes de la tecnología y el aprender a programar nos ayudaría a entender de mejor manera el mundo que nos rodea ya que en casi todas las actividades que se realizan diariamente se tiene contacto con varios dispositivos programados, entonces al enseñar a programar a los niños se está aumentando la cultura tecnológica y la competitividad. (Zamora, 2015).

El aprender a programar tempranamente no solo te involucra con la tecnología sino que también favorece a gran medida a desarrollar la creatividad y aumentar la capacidad de

resolución de problemas, esto es asegurado por varios estudios que demuestran científicamente que aprender a programar mejora la capacidad de hablar y pensar de los niños sobre problemáticas más complejas. (Zamora, 2015).

En Madrid para el año académico 2014-2015 se implementó de manera obligatoria la materia de Programación y Comunicaciones en las escuelas destacando algunas ventajas que han tenido los alumnos en lo que va del año:

- ✓ Mejoran la capacidad de resolver problemas y aumentan su atención y entendimiento de otras materias como Matemática.
- ✓ Los niños que aprenden programación amplían su capacidad no solo de interactuar con la tecnología sino que pueden crearla y expresarse a través de ella de distintas formas
- ✓ El mundo cada vez requiere de más programadores por lo que enseñarles programación no solo les va ayudar en el presente sino que irán preparados al futuro.

Otra de las ventajas de la programación de los niños es que se puede despertar su atención y el interés al indicarles que la programación no solo les permitirá utilizar la tecnología sino también crearla, para ellos les va resultar sumamente fácil aprender ya que en la niñez los conocimientos se afianzan en el cerebro de una manera natural. (Zamora, 2015).

2.6 PLATAFORMA ARDUINO

Arduino es una plataforma electrónica de software y hardware libre, constituida por una placa con un microcontrolador y un entorno de desarrollo y diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. (Arduino, 2015). “Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede

afectar aquello que le rodea controlando luces, motores y otros actuadores” (Letrán, 2011), lo que ha permitido motivar a varios estudiantes y desarrolladores a utilizar esta plataforma.

“Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador” (Herrador, 2009), esto permite que se pueda combinar las diferentes placas Arduino con otros sistemas o programas para mejorar los proyectos y sus utilidades.

Arduino es una placa electrónica que comunica un microcontrolador con puertos de entrada y/o salida, tiene un lenguaje de programación propio y una memoria EEPROM³ que actúa como un pequeño disco duro, aquí se almacenan los programas que se van a ejecutar. Esta memoria es no volátil, es decir que así se apague la placa Arduino los datos permanecen ahí, también se debe indicar que soporta interfaces de comunicación como: Wireless, Bluetooth, Ethernet, entre otras (Cevallos, 2013).

2.6.1 VENTAJAS DE LA PLATAFORMA

En la actualidad existen algunas herramientas que facilitan el trabajo con microcontroladores, sin embargo Arduino presenta algunas características que lo ponen en ventaja sobre otros sistemas como las siguientes:

- Arduino es de Código Abierto, la cual es una de sus principales ventajas ya que permite acceder a su estructura física y lógica, y poder modificarla, además existe una gran comunidad de usuarios que se colaboran entre sí compartiendo

³ EEPROM= Memoria ROM Eléctricamente Programable y Borrable

información que pueda ser útil para la resolución de problemas e implementación de proyectos.

- Es un sistema multiplataforma, “el software de Arduino se ejecuta en sistemas operativos Windows⁴, Macintosh OSx⁵ y GNU/Linux⁶” (Herrador, 2009).
- No es necesario tener conocimientos avanzados en electrónica e informática para manejar Arduino esto es gracias a que Arduino cuenta con un lenguaje propio de programación basado en Processing⁷. “Arduino te ofrece un entorno de desarrollo integrado (IDE) con funciones preestablecidas que reducen la lógica a lectura de entradas, control de tiempos y salidas de una manera semántica e intuitiva” (Saburido, 2014).

2.6.2 MODELOS DE PLACAS ARDUINO

Existen diferentes placas de Arduino, las cuales dependen mucho del tamaño del proyecto, de la cantidad de entradas y/o salidas requeridas, si la alimentación y programación se va a realizar con la propia placa, si va a interactuar con su misma circuitería o si va a comunicarse con dispositivos externos móviles como: PDA's, receptores, celulares, entre otros; así, se tienen entre las placas Arduino más conocidas: Arduino Uno, Arduino Mega, Arduino Zero, Arduino Pro, Arduino Nano, entre otras. (Cevallos, 2013).

⁴ Windows= Sistema operativo desarrollado por Microsoft.

⁵ Macintosh OSx= Sistema Operativo basado en Unix desarrollado y comercializado por Apple.

⁶ GNU/Linux= Sistemas Operativos bajo software libre, su código puede ser utilizado, modificado y redistribuido gratuitamente.

⁷ Processing= es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java

2.6.2.1 Arduino UNO

Es la placa estándar y posiblemente la más conocida y documentada. “Es una buena opción para empezar a usar Arduino y familiarizarse con el entorno”, (Urgiles & Colcha, 2015).

En el siguiente grafico se detallan las características de este modelo.



Figura 13. Placa Arduino UNO

Fuente: http://arduino.cc/en/uploads/Main/ArduinoUno_r2_front450px.jpg

Tabla 2. Características Arduino UNO

MICROCONTROLADOR	ATMEGA 328 Extraíble
PINES DIGITALES	14 Entrada y Salida (6 Salidas PWM)
PINES ANALÓGICOS	6 Entradas Analógicas (8 bits de resolución)
VOLTAJE DE FUNCIONAMIENTO	5 Voltios
CORRIENTE POR PIN	40 Miliamperios
MEMORIA FLASH	32 Kb
MEMORIA SRAM	2 Kb
MEMORIA EEPROM	1 Kb
FRECUENCIA	16 MHz
DIMENSIONES	70 x 55 mm

Nota: Adaptado de la Página Oficial de Arduino. Fuente: <http://www.arduino.cc/en/Main/ArduinoBoardUno>

2.6.2.2 Arduino Zero

El Arduino Zero es una extensión de 32 bits simple y poderosa de la plataforma establecida por Arduino UNO. La junta Zero tiene como objetivo proporcionar a las personas con potencial creativo mayores recursos para la creación de sistemas innovadores. Esta placa se puede utilizar en la elaboración de dispositivos inteligentes, tecnología portátil, alta tecnología de automatización, robótica y otros proyectos. Está basada en un microcontrolador ATMEL⁸ SAMD21, que cuenta con un núcleo de 32 bits. (Arduino, 2015).

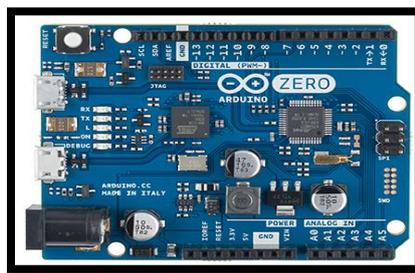


Figura 14. Placa Arduino Zero

Fuente: http://arduino.cc/en/uploads/Main/Arduino_Zero_front450.png

Tabla 3. Características Arduino Zero

MICROCONTROLADOR	ATMEL SAMD21G18
PINES DIGITALES	14 Entrada y Salida (12 Salidas PWM)
PINES ANALÓGICOS	6 Entradas Analógicas (12 bits de resolución y 1 DAC ⁹)
VOLTAJE DE FUNCIONAMIENTO	3.3 - 5 Voltios
CORRIENTE POR PIN	7 Miliamperios
MEMORIA FLASH	256 Kb
MEMORIA SRAM	32 Kb
MEMORIA EEPROM	16 Kb
FRECUENCIA	48 MHz
DIMENSIONES	70 x 55 mm

Nota: Adaptado de la Página Oficial de Arduino. Fuente: <http://www.arduino.cc/en/Main/ArduinoBoardZero>

⁸ Atmel= compañía de semiconductores

⁹ DAC= Conversión Digital Analógica

2.6.2.3 Arduino YUN

El Arduino YUN es una placa electrónica basada en el ATmega32u4 y el Atheros¹⁰ AR9331. La junta ha incorporado interfaces Ethernet¹¹ y Wi-Fi¹², un puerto USB¹³, ranura para tarjeta micro-SD¹⁴, 20 pines digitales de entrada y salida (de los cuales 7 se pueden utilizar como salidas PWM¹⁵ y 12 como entradas analógicas), un cristal oscilador de 16 MHz y 3 botones de reposición. (Arduino, 2015)

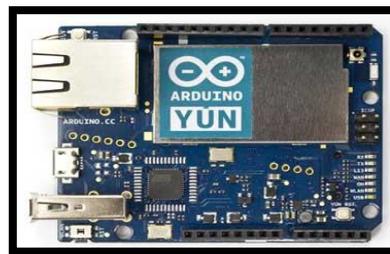


Figura 15. Placa Arduino YUN

Fuente: http://arduino.cc/en/uploads/Main/ArduinoYunFront_2_450px.jpg

Tabla 4. Características Arduino YUN

MICROCONTROLADOR	ATMEGA 32u4 Y CHIP ATHEROS AR9331
PINES DIGITALES	20 de Entrada y Salida (7 Salidas PWM)
PINES ANALÓGICOS	12 Entradas Analógicas (8 Bits de resolución)
VOLTAJE DE FUNCIONAMIENTO	5 voltios
CORRIENTE POR PIN	40 miliamperios
MEMORIA FLASH	32 Kb
MEMORIA SRAM	2.5Kb
MEMORIA EEPROM	1Kb
FRECUENCIA	16 MHz
DIMENSIONES	73x53 mm
OTRAS CARACTERÍSTICAS	<ul style="list-style-type: none"> - Interfaz Ethernet IEEE 802.3 10/100 Mb/s - Wi-Fi IEEE 802.11 b/g/n - 2 Puertos USB - RAM 64 MB - Lector de tarjetas Micro SD

Nota: Adaptado de la Página Oficial de Arduino.

Fuente: <http://www.arduino.cc/en/Main/ArduinoBoardYun?from=Main.ArduinoYUN>

¹⁰ Atheros= Desarrollador de semiconductores para redes y comunicaciones, particularmente Chipsets inalámbricos

¹¹ Ethernet= Estándar de redes de área local para computadores

¹² Wi-Fi= Nombre comercial referente a comunicación inalámbrica

¹³ USB= Bus Universal en Serie.

¹⁴ Micro SD= Formato de tarjeta de memoria flash desarrollada por SanDisk

¹⁵ PWM= Modulación por Ancho de Pulso

2.6.2.4 Arduino NANO

El Arduino Nano es una pequeña placa, completa y amistosa basada en el ATmega328. Tiene más o menos la misma funcionalidad de la Arduino UNO, pero en un diseño diferente. Le falta sólo un conector de alimentación de Corriente Continua y funciona con un cable USB Mini-B en lugar de una normal. El Nano fue diseñado y está siendo producido por Gravitech¹⁶. (Arduino, 2015).

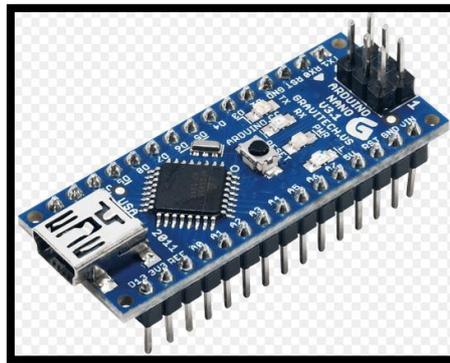


Figura 16. Placa Arduino NANO

Fuente: http://cdn-reichelt.de/bilder/web/xxl_ws/A300/ARDUINO_NANO_03.png

Tabla 5. Características Arduino NANO

MICROCONTROLADOR	ATMEGA 328
PINES DIGITALES	14 Entrada y Salida (6 Salidas PWM)
PINES ANALÓGICOS	6 Entradas Analógicas (8 bits de resolución)
VOLTAJE DE FUNCIONAMIENTO	5 - 12 Voltios
CORRIENTE POR PIN	40 Miliamperios
MEMORIA FLASH	2 Kb
MEMORIA SRAM	1 Kb
MEMORIA EEPROM	512 Kb
FRECUENCIA	16 MHz
DIMENSIONES	45 x 18 mm

Nota: Adaptado de la Página Oficial de Arduino. Fuente: <http://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardNano>

¹⁶ Gravitech= Tienda de elementos electrónicos.

2.6.2.5 Arduino MEGA

El Arduino Mega es una placa electrónica basada en el ATmega1280. Cuenta con 54 pines digitales de entrada y salida (de los cuales 14 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 Puertos seriales, un cristal oscilador de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP¹⁷, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o la batería para empezar. La Placa Mega es compatible con la mayoría de los modelos de Arduino. (Arduino, 2015).

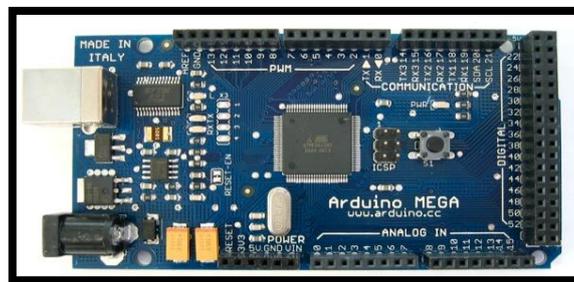


Figura 17. Placa Arduino MEGA

Fuente: <http://arduino.cc/en/uploads/Main/ArduinoMega.jpg>

Tabla 6. Características Arduino MEGA

MICROCONTROLADOR	ATMEGA 1280
PINES DIGITALES	54 Entrada y Salida (15 Salidas PWM)
PINES ANALÓGICOS	16 Entradas Analógicas (8 bits de resolución)
VOLTAJE DE FUNCIONAMIENTO	5 - 12 Voltios
CORRIENTE POR PIN	40 Miliamperios
MEMORIA FLASH	128 Kb
MEMORIA EEPROM	4Kb
FRECUENCIA	16 MHz
DIMENSIONES	101 x 53 mm

Nota: Adaptado de la Página Oficial de Arduino. Fuente: <http://www.arduino.cc/en/pmwiki.php?n=Main/arduinoBoardMega>

¹⁷ ICMP= Programación Serial en Circuito

2.6.2.6 Arduino MINI-PRO

El Arduino Mini Pro es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 pines digitales de entrada y salida, 6 entradas analógicas, un botón de reinicio y los agujeros para el montaje de pines. La placa Arduino Mini Pro está diseñada para la instalación permanente o semipermanente en objetos o exposiciones. El tablero viene sin zócalos pre-instalados, permitiendo el uso de varios tipos de conectores o soldadura directa de cables. (Arduino, 2015).

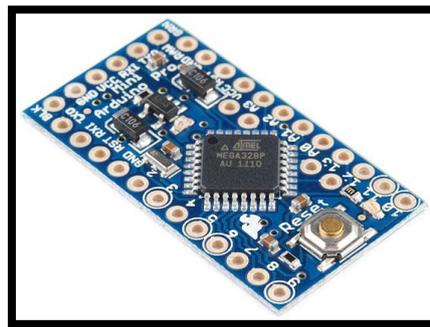


Figura 18. Placa Arduino Mini PRO

Fuente: <https://cdn.sparkfun.com/assets/f/4/e/2/7/51eeb8f9ce395f0778000000.png>

Tabla 7. Características Arduino Mini PRO

MICROCONTROLADOR	ATMEGA 328
PINES DIGITALES	14 Entrada y Salida (6 PWM)
PINES ANALÓGICOS	6 Entradas Analógicas (8 bits de resolución)
VOLTAJE DE FUNCIONAMIENTO	5 - 12 Voltios
CORRIENTE POR PIN	40 Miliamperios
MEMORIA FLASH	32 Kb
MEMORIA SRAM	8 Kb
MEMORIA EEPROM	1Kb
FRECUENCIA	16 MHz
DIMENSIONES	33 x 18 mm

Nota: Adaptado de la Página Oficial de Arduino. Fuente: <http://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardProMini>

2.6.2.7 Comparativa de Modelos Arduino

Tabla 8. Comparativa #1 de Placas Arduino

CARACTERÍSTICAS	ARDUINO UNO	ARDUINO MEGA2560	ARDUINO NANO	ARDUINO YUN	ARDUINO MINI-PRO
Microcontrolador	Atmega328	Atmega2560	Atmega2560	Atmega 32u4	Atmega 328
Tensión de funcionamiento	5 voltios	5 voltios	5 voltios	5 voltios	5 voltios
Voltaje de entrada recomendando	7-12 voltios	7-12 voltios	7-12 voltios	5 voltios	7-12 voltios
Voltaje de entrada límite	6-20 voltios	6-20 voltios	6-20 voltios	5 voltios	5-12voltios
Pines de entrada/salida digitales	14 los cuales 6 proporcionan salida PWM	54 los cuales 15 proporcionan salida PWM	54 los cuales 15 proporcionan salida PWM	20 los cuales 7 proporcionan salida PWM	14 los cuales 6 proporcionan salida PWM
Pines de entrada analógicos	6	16	16	12	6
Corriente DC por pin de entrada/salida	40Ma	40mA	40mA	40mA	40mA
Corriente DC del pin de 3.3 Voltios	50Ma	50mA	50mA	50mA	50mA
SRAM	2 KB de memoria flash	128 KB de memoria flash	256 KB de memoria flash	2,5 KB	2KB
EEPROM	1 KB	4 KB	4 KB	1 KB	1 KB
Velocidad de reloj	16 MHz	16MHz	16MHz	16 MHz	16 MHz
Precio aproximado	28 dólares	35 dólares	20 dólares	110 dólares	9 dólares

Nota: Los precios encontrados en esta tabla son referenciales de fábrica. Fuente: Adaptado de la Página Oficial de Arduino: <http://www.arduino.cc/en/Products.Compare>

Tabla 9. Comparativa #2 de Placas Arduino

CARACTERÍSTICAS	ARDUINO MICRO	ARDUINO ETHERNET	ARDUINO FIO	ARDUINO ROBOT Control - Motor		ARDUINO EXPLORA	ARDUINO LILYPAD
Microcontrolador	Atmega32u4	Atmega328	Atmega328P	Atmega 32u4	Atmega 32u4	Atmega 32u4	Atmega 128V-328V
Tensión de funcionamiento	5 voltios	5 voltios	3.3 voltios	5 voltios	5 voltios	5 voltios	2.7 a 5.5 voltios
Voltaje de entrada recomendando	7-12 voltios	7-12 voltios	3,7-7 voltios	5 voltios	9 voltios	5 voltios	2.7 a 5.5 voltios
Voltaje de entrada límite	6-20 voltios	6-20 voltios	3,35-12 voltios	5 voltios	9 voltios	5 voltios	2.7 a 5.5 voltios
Pines de entrada/salida digitales	20 y 7 proporcionan salida PWM	14 y 4 proporcionan salida PWM	14 y 6 proporciona salida PWM	5 los cuales 6 proporcionan salida PWM	4 y 1 proporciona salida PWM	--	14 los cuales 6 son salida PWM
Pines de entrada analógicos	12	6	16	4 de los pines de E/S	4	--	6
Corriente DC por pin de entrada/salida	40Ma	40mA	40mA	40mA	40mA	--	40mA
Corriente DC del pin de 3.3 Voltios	50mA	50Ma	--	--	--	--	--
Memoria Flash	32 KB	32 KB	32 KB (2KB arranque)	32 KB (4Kb arranque)	32 KB (4Kb arranque)	32 KB (4 KB arranque)	16 KB (2 KB arranque)
SRAM	2,5 KB	2 KB	2 KB	2,5 KB	2,5 KB	2,5 KB	1 KB
EEPROM	1 KB	4 KB	1 KB	1 KB	1 KB	1 KB	512 bytes
Velocidad de reloj	16 MHz	16 MHz	8 MHz	16 MHz	16 MHz	16 MHz	8 MHz
Precio aproximado	28 dólares	20 dólares	25 dólares	110 dólares		60 dólares	16 dólares

Nota: Los precios encontrados en esta tabla son referenciales de fábrica. Fuente: Adaptado de la Página Oficial de Arduino: <http://www.arduino.cc/en/Products.Compare>

2.6.3 IDE (ENTORNO DE DESARROLLO INTEGRADO)

Arduino dispone de un entorno propio de desarrollo de programación conocido como IDE que se puede obtener para diferentes sistemas operativos como: Windows, Mac y Linux, el cual se transfiere mediante cable USB. Esta interfaz permite introducir el programa a ejecutar en la placa Arduino y es donde se define que hacer tanto con las entradas como con las salidas que dispone la placa. (Cevallos, 2013).

El IDE incluye un editor de código fuente que es un editor de texto con algunas características útiles para la programación y un compilador que convierte el código que ha sido escrito en las instrucciones legibles por código máquina.

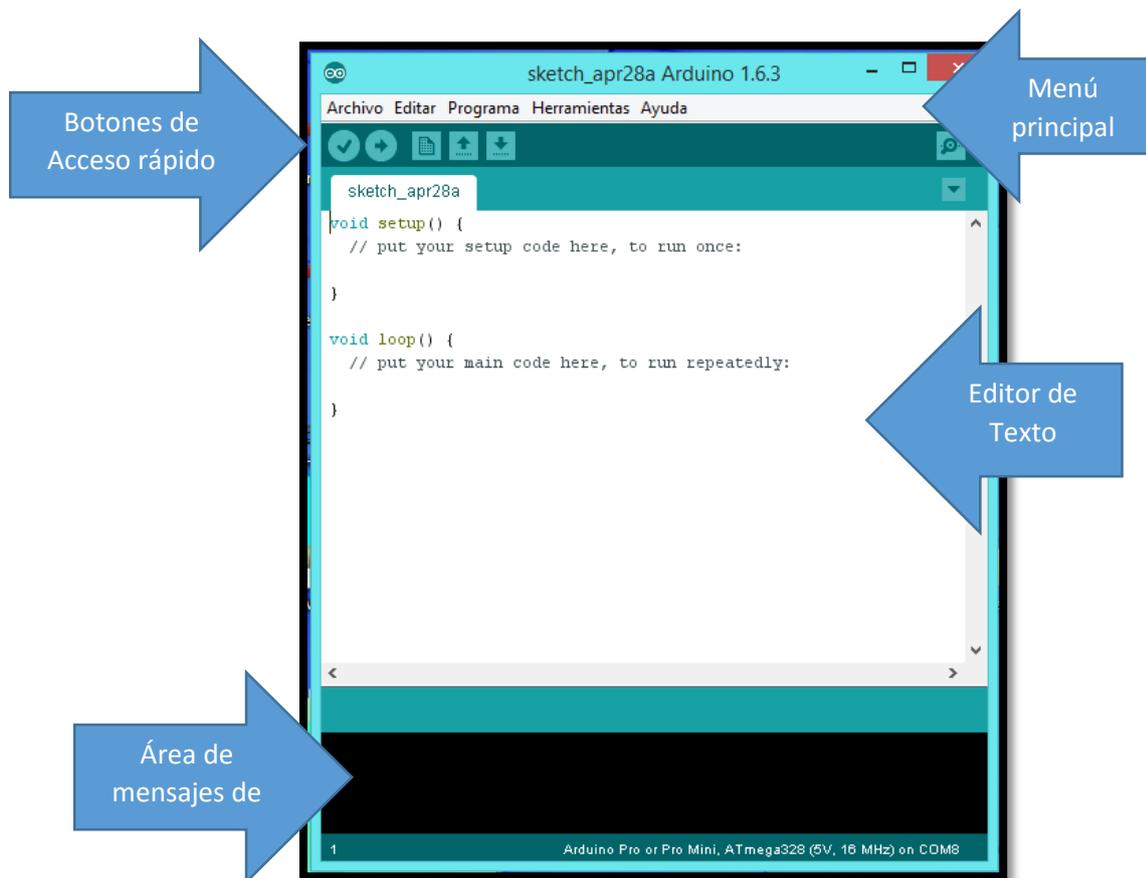


Figura 19. Interfaz gráfica del IDE de Arduino
Fuente: El Autor

El entorno de desarrollo integrado de Arduino está implementado en Java y basado en Wiring¹⁸ que descende a su vez de Processing¹⁹. Cuando se escribe un programa Arduino, se está haciendo implícitamente el uso de la biblioteca de Wiring, que se incluye con el Arduino IDE. Esto le permite hacer programas ejecutables mediante el uso de sólo dos funciones: setup²⁰ () y loop²¹ ().(Arduino, 2015)

2.6.4 FUNCIONALIDADES PRINCIPALES

2.6.4.1 Entradas/Salidas Digitales

Arduino está formado básicamente por pines de entrada, un microcontrolador en el medio y pines de salida. Los pines de entrada sirven para escuchar y capturar información del exterior, por ejemplo de pulsadores, sensores, lectores, entre otros; El microcontrolador sirve para procesar el programa cargado y finalmente los pines de salida sirven para enviar información desde la tarjeta Arduino hacia el exterior. (Benitez R. , 2012).

Estas entradas y salidas son consideradas digitales debido a que únicamente pueden manejar valores lógicos es decir 1L y 0L. En Arduino los pines digitales vienen declarados por defecto como entradas, para cambiar esta configuración únicamente se utiliza la función pinMode() la cual permite indicarle al microcontrolador como va a trabajar el pin.

Los pines digitales configurados como entrada se encuentran con un nivel alto de impedancia lo que permite recibir un estado lógico 1 o 0 con facilidad, según la lógica de Arduino para leer una entrada digital se utiliza la función digitalRead(). (Arduino, 2015).

¹⁸ Wiring= Plataforma de programación de código abierto

¹⁹ Processing= lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java

²⁰ Setup()= Parte del programa que se ejecuta una sola vez

²¹ Loop()= Parte del programa que se ejecuta en forma de ciclo continuamente

Para que la placa Arduino reciba una señal digital comúnmente se utiliza una configuración PULL UP o PULL DOWN esto para mantener siempre un estado lógico ya sea alto o bajo, con una configuración PULL UP en la entrada de Arduino siempre tendremos un estado lógico alto y cuando se presione el botón arrojará un nivel bajo, en cambio con una configuración PULL DOWN se tendrá siempre un estado lógico bajo y cuando se presione el botón un nivel alto.

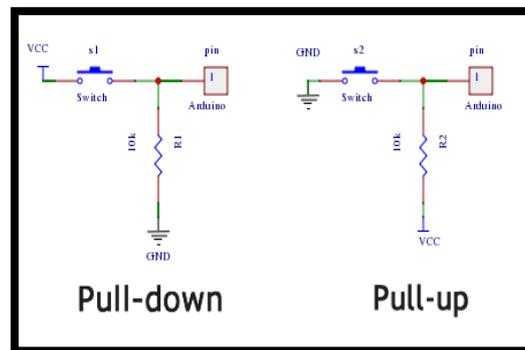


Figura 20. Configuraciones PULL-UP y PULL-DOWN

Fuente: <http://blog.make-a-tronik.com/wp-content/gallery/tutorialarduino/pullups.png>

Los pines configurados como salida se encuentran en un nivel de baja impedancia lo que permite proporcionar hasta 40mA de corriente a otros circuitos o dispositivos (depende del modelo de Arduino), esta cantidad de corriente permite encender un LED o activar circuitos pero no es suficiente para activar directamente relés o motores. (Arduino, 2015). Para indicar el valor lógico de la salida en Arduino se utiliza la función `digitalWrite()` y se posee diferentes opciones para señalar si la salida será 1L (5 voltios) o 0L (0 voltios) los cuales se indican en la siguiente tabla:

Tabla 10. Opciones para expresar salidas lógicas en Arduino

Estado Digital	5 voltios	0 voltios
Opción 1	1	0
Opción 2	HIGH	LOW
Opción 3	TRUE	FALSE

Fuente: El Autor. Adaptado de: <https://botscience.wordpress.com/2012/06/20/tutorial-02-entradas-y-salidas-digitales-y-uso-de-temporizadores/>

2.6.4.2 Entradas Analógicas

Las placas Arduino disponen de entradas analógicas que pueden recibir voltajes entre 0 y 5 voltios. Aunque la electrónica de los microcontroladores solo pueden trabajar con datos digitales, incorporan un circuito que convierte el valor analógico recibido a un valor digital lo más aproximado posible. (Artero, 2013).

La mayoría de modelos cuentan con un conversor análogo digital de 6 entradas, cada entrada dispone de 10 bits de resolución para guardar el valor del voltaje convertido digitalmente, estos bits de resolución son los que dan el grado de precisión a la transformación del dato analógico al digital.

Las tarjetas Arduino al poseer un conversor de 10 bits permitirá obtener valores entre 0 y 1023 ($2^{10}-1$), basados en un voltaje de referencia que por lo general es 5 voltios, en otras palabras Arduino realiza un mapeo del valor analógico recibido y da como resultado un valor binario utilizando la siguiente formula:

$$\text{Resolución} = \frac{\text{Voltaje de refernecia}}{2^n - 1}$$

Ecuación 1. Cálculo de la resolución de una Entrada Analógica
Fuente: <http://playground.arduino.cc/ArduinoNotebookTraduccion/Appendix6>

Donde:

n= bits de resolución

2.6.4.3 Salidas PWM

Modulación de ancho de pulso o PWM, es una técnica para obtener resultados análogos a partir de medios digitales. Esta técnica se utiliza ante la necesidad de enviar señales

analógicas al entorno, por ejemplo para variar la velocidad de un motor, la frecuencia de un sonido emitido por un zumbador o cambiar la intensidad de un LED. (Artero, 2013).

Arduino no dispone de una salida propiamente analógica es por eso que se utiliza la técnica de modulación por ancho de pulso para simular un comportamiento analógico a través de una salida digital.

Lo que hace esta técnica es enviar una señal cuadrada formada por pulsos de frecuencia constante, al variar la duración de estos pulsos se está variando proporcionalmente el voltaje promedio resultante, es decir entre más cortos sean los pulsos menor será el voltaje de salida resultante y entre más largo sea el pulso mayor será dicho voltaje. (Artero, 2013).

En la *Figura 21* se muestran algunas señales con diferente tamaño de pulso que simulan valores analógicos. Las líneas verdes representan el periodo de la señal y cómo se puede ver si se cambia el porcentaje del pulso con relación al periodo se puede obtener valores analógicos entre 0 y 5 voltios.

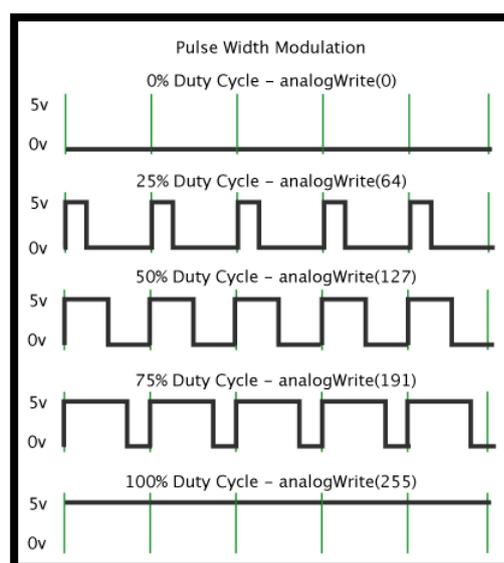


Figura 21. Representación de valores analógicos
Fuente: <http://www.arduino.cc/en/pmwiki.php?n=Tutorial/PWM>

2.6.4.4 Comunicación Serial

La comunicación serial se utiliza para el intercambio de información entre la placa Arduino y un ordenador u otros dispositivos. Todas las placas Arduino tienen al menos un puerto serial que permite la comunicación de una forma directa o también da la posibilidad de conectar módulos de comunicación inalámbrica como Bluetooth, ZigBee, WiFly entre otros. (Arduino, 2015).

“El puerto serial de las computadoras es conocido como puerto RS-232, la ventaja de este puerto es que todas las computadoras traen al menos un puerto serial, este permite la comunicación con otros dispositivos tales como otra computadora, el mouse, la impresora y en este caso también con los microcontroladores”. (Pérez I. E.). El hecho de que varios dispositivos utilicen la comunicación serial bajo un estándar permite la convergencia de diferentes dispositivos sin inconvenientes.

2.6.4.4.1 Tipos de comunicación serial

2.6.4.4.1.1 Comunicación Serial Síncrona

La comunicación síncrona es aquella que se somete a una rígida temporización de reloj que permite que el dispositivo receptor sea capaz de conocer en que instante la señal que le llega tiene plena validez. (Casares, 2012)

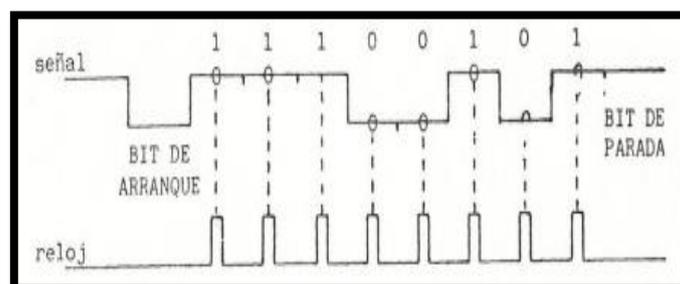


Figura 22. Comunicación serial síncrona

Fuente: http://www.uco.es/electrotecnia-etsiam/pdf-master/05-Comunicacion_Serie-paralelo.pdf

➤ *Características de la comunicación serial Síncrona:*

- La transmisión de datos se la realiza carácter a carácter.
- El control de tiempo es rígido es decir a cada carácter le corresponde determinado tiempo para trasmitirse.
- Los caracteres se trasmiten por separado y el tiempo que separa cada carácter puede ser cualquiera.
- Los datos del carácter se propagan bit a bit sincrónicamente a golpe de reloj

2.6.4.4.1.2 Comunicación Serial Asíncrona

La comunicación asíncrona no está sujeta a ninguna temporización, en este tipo de comunicación los datos en caracteres ASCII pueden ser transmitidos en cualquier instante. El reloj se sincroniza al principio de cada carácter recibido. (Casares, 2012).

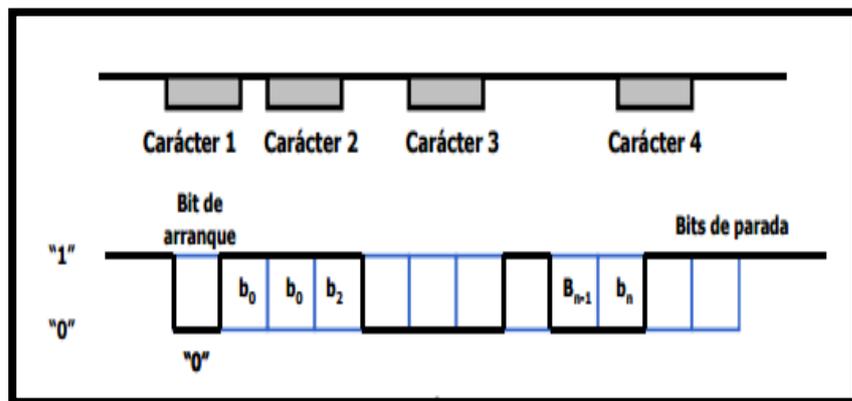


Figura 23. Comunicación serial Asíncrona

Fuente: http://www.uco.es/electrotecnia-etsiam/pdf-master/05-Comunicacion_Serie-paralelo.pdf

➤ *Características de la comunicación serial Asíncrona:*

- En la comunicación asíncrona cada carácter va precedido por un bit de arranque.

- Cuando no se están transmitiendo datos la línea de transmisión se encuentra en un nivel lógico alto.
- Cuando termina de transmitir el carácter envía un carácter de parada.
- Primero transmite el bit de más peso para luego enviar los de mayor peso.

En la comunicación asíncrona es necesario definir algunos parámetros como para mejorar la transmisión:

Tabla 11. Parámetros para una comunicación Asíncrona

Parámetros	Opciones
Velocidad de transmisión en baudios	1200,2400,4800,9600,19200,38400,57600, 115600
Tiempo por bit	1, 1.5, 2
Numero de bits de datos	5, 6, 7, 8
Bit de parada	Par, impar, ninguno

Fuente: El Autor. Adaptado de: http://www.uco.es/electrotecnia-etsiam/pdf-master/05-Comunicacion_Serie-paralelo.pdf

2.6.4.4.2 Consideraciones de la comunicación serial de Arduino

La mayoría de plataformas Arduino están basadas en microcontroladores ATMEGA, estos microcontroladores soportan nativamente comunicación serie mediante una UART²² implementada dentro del chip. La ventaja de soportar nativamente una UART permite que el microcontrolador haga otras tareas mientras recibe datos serie que puede ir guardando en un buffer interno del ATMEGA. (Arduino, 2015).

Arduino para comunicarse con el ordenador puede utilizar las siguientes velocidades seriales: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, o 115200 que

²² UART= Transmisor-Receptor Asíncrono Universal

además soporta el monitor serial. (Arduino, 2015). Esto permite la comunicación con varios dispositivos o módulos que trabajen a diferente velocidad.

La conexión USB con el ordenador la realiza el chip externo FTDI²³-USB²⁴, este recibe datos en formato TTL²⁵ serial y el chip los envía vía USB a través de un driver de software que ve al Arduino como un puerto serial. (Arduino, 2015).

2.6.4.4.3 Tecnologías Inalámbricas Compatibles

La comunicación serial permite incorporar distintas tecnologías inalámbricas dependiendo de las exigencias del proyecto a implementar. Entre los aspectos que se debe tomar en cuenta para elegir el tipo de tecnología está la velocidad de transmisión de datos, distancia del enlace, seguridad requerida, consumo energético, frecuencias, funcionalidades y recursos; todo esto tratando de no desperdiciar las capacidades de la tecnología elegida.

Las tecnologías inalámbricas más utilizadas en el desarrollo de sistemas embebidos son Wi-Fi, ZigBee y Bluetooth.

2.6.4.4.3.1 Wi-Fi

“WI-FI es una de las tecnologías de comunicación inalámbrica mediante ondas más utilizada en la actualidad, también llamada WLAN (Wireless LAN, red inalámbrica) o estándar IEEE 802.11. WI-FI no es una abreviatura de Wireless Fidelity, simplemente es un nombre comercial” (Moreta, 2009).

²³ FTDI= Dispositivos Futuros de Tecnología Internacional

²⁴ USB= Bus Serial Universal

²⁵ TTL= Lógica Transistor-Transistor



Figura 24. Logo Wi-Fi

Fuente: <http://images.gizmag.com/hero/2012-wifi-standard.jpg>

Esta tecnología se usa principalmente en la implementación de redes de área local y permite alcances de hasta 100 metros en ambientes abiertos y hasta 30 metros en ambientes cerrados, además existen diferentes versiones del estándar IEEE 802.11.

Tabla 12. Versiones del estándar 802.11

NOMBRE DEL ESTÁNDAR	DESCRIPCIÓN
IEEE 802.11a	Permite velocidades de transmisión de datos de hasta 54 Mbps aunque en la práctica es de 30 Mbps. Trabaja en la banda de frecuencia ISM de 5Ghz.
IEEE 802.11b	Es el más utilizado en la actualidad, ofrece 11Mbps aunque en la práctica es de 6 Mbps. Tiene un alcance de 300m en un espacio abierto. Usa el rango de frecuencia de 2.4GHz
IEEE 802.11g	Hace uso de la banda de 2,4GHz pero opera a una velocidad teórica máxima de 54 Mbps, que en promedio es de 22 Mbps.
IEEE 802.11n	Usa simultáneamente las bandas 2.4GHz y 5.4GHz. Suministra velocidades superiores a 100 Mbps lo cual duplica la velocidad de 802.11g y 802.11a

Fuente: El Autor basado en: <http://repositorio.espe.edu.ec/bitstream/21000/6395/1/T-ESPEL-CDT-0991.pdf>

➤ MODULO WI-FLY

Wi-Fly es un dispositivo de comunicación inalámbrica basado en el estándar 802.11 b/g que permite conectar los sistemas embebidos a la red y es una buena

solución debido a su pequeño tamaño y consumo de energía extremadamente bajo, es útil para aplicaciones embebidas móviles inalámbricas, tales como sistemas de monitoreo, seguimiento GPS, entre otras. Posee manejo de la pila TCP - IP a bordo, además de programas de aplicación en redes tales como TELNET y FTP. El hardware requiere solamente cuatro conexiones (PWR, TX, RX, GND) para crear una conexión básica. (Cabrera & Nato, 2012).



Figura 25. Módulo WiFly

Fuente: <http://i01.i.aliimg.com/wspphoto/v0/960252893/-font-b-Wifly-b-font-rn-xv-wireless-wifi-font-b-module-b-font-band.jpg>

Tabla 13 Características Modulo WiFly

ESTÁNDAR	IEEE 802.11 b/g
POTENCIA DE TRANSMISIÓN	-82 a 93 dBm
POTENCIA DE RECEPCIÓN	15 dBm
FRECUENCIA	2.4 Ghz
TIPO DE ANTENA	Dipolo
CONSUMO DE CORRIENTE	40 miliamperios
VOLTAJE DE ALIMENTACIÓN	3.3V
PROTOCOLOS SOPORTADOS	TCP/IP, ICMP, Telnet, TFTP, DHCP, FTP
SEGURIDAD	Wep128, wpa,wpa2

Fuente: El Autor. Adaptado de: <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/WiFi/WiFly-RN-XV-DS.pdf>

2.6.4.4.3.2 ZigBee

ZigBee es un estándar abierto para radiocomunicaciones de baja potencia conocido bajo el estándar IEEE 802.15.4 de redes de área personal inalámbricas. Incluye un robusto y confiable protocolo de red, de bajo consumo y coste, con servicios de seguridad y capa de aplicación que garantiza la interoperabilidad entre dispositivos. ZigBee trabaja a una velocidad de datos de 250 Kbps sobre la banda libre de 2,4 GHz, aunque también soporta las bandas de 868 y 900 MHz. (Montesinos, 2013).



Figura 26. Logo ZigBee

Fuente: <http://buildyoursmarthome.co/home-automation/protocols/zigbee/>

Una de las características más destacables de la tecnología ZigBee es su soporte de redes malladas. Las redes malladas permiten interconectar varios nodos entre sí. Las conexiones entre nodos se actualizan dinámicamente y se optimizan mediante sofisticadas tablas de encaminamiento integradas. Las redes malladas son descentralizadas; cada nodo es capaz de descubrir la red por sí mismo. Además, cuando un nodo abandona la red, la topología en malla permite a los demás nodos reconfigurar las rutas en función de la nueva estructura de la red. Las posibilidades de una topología mallada y del enrutamiento ad-hoc proporcionan una mayor estabilidad y adaptación a condiciones impredecibles o caída de enlaces y nodos. (Montesinos, 2013).

CARACTERÍSTICAS PRINCIPALES DEL ESTÁNDAR

- Permite trabajar en diferentes bandas de frecuencia como: 2.4Ghz, 915Mhz y 868 MHz.
- Direccionamiento de red de 16 bits
- Permite tanto topologías malladas punto multipunto como redes punto a punto.
- Utiliza el método de acceso al canal CSMA-CA²⁶
- Bajo consumo energético
- Gran densidad de nodos por red
- Alta seguridad ya que proporciona integridad de datos y autenticación haciendo uso de cifrado AES-128.
- Baja latencia

➤ MODULO XBEE

Uno de los módulos compatibles con Arduino es el XBee Pro Serie Uno creado por la empresa Maxstream, el cual se utiliza en aplicaciones que requieran interconexión de varios nodos y que requieran grandes distancias con un nivel alto de seguridad en especial proyectos de domótica.

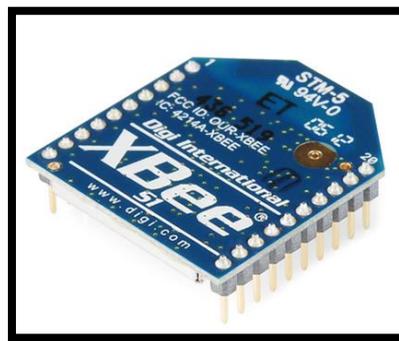


Figura 27. Módulo XBee

Fuente: <http://5hertz.com/images/10401.jpg>

²⁶ CSMA-CA= Acceso Múltiple con Escucha de Portadora y Evasión de Colisiones

Tabla 14. Características del Módulo XBee

ALCANCE	30 metros en Interiores y 90 metros en Exteriores
POTENCIA DE TRANSMISIÓN	1mW
SENSIBILIDAD DE RECEPCIÓN	0 dB
SEGURIDAD	Encriptación 128 Bits
VOLTAJE DE FUNCIONAMIENTO	2.8 a 3.3 Voltios
CONSUMO DE CORRIENTE TRANSMISIÓN	45mA
CONSUMO DE CORRIENTE RECEPCIÓN	50mA
FRECUENCIA	ISM 2.4 GHz

Fuente: El Autor. Adaptado de: <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>

2.6.4.4.3 Bluetooth

Bluetooth es una tecnología que define el estándar IEEE 802.15.1 para la transmisión de voz y de datos entre diferentes dispositivos mediante un enlace por radiofrecuencia omnidireccional. Está diseñado especialmente para dispositivos de bajo consumo con un rango de cobertura bajo especialmente para redes personales. (Fernández, 2011).

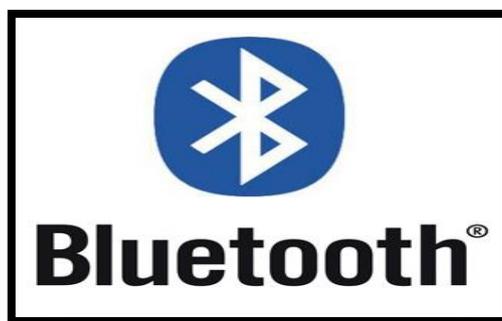


Figura 28. Logo de Bluetooth

Fuente: <http://f.tqn.com/y/tabletas/1/S/k/7/-/-/Bluetooth-logo.jpg>

Esta tecnología tiene como propósito primeramente facilitar las comunicaciones entre dispositivos móviles y fijos, también busca eliminar los cables y conectores; y ofrecer la posibilidad de crear pequeñas redes inalámbricas con facilidad de sincronización de datos entre diferentes dispositivos. (Fernández, 2011).

Bluetooth opera en la banda de frecuencias ISM²⁷ de 2.4 GHz empleando la técnica de salto de frecuencia FHSS²⁸ que consiste en dividir su banda de frecuencia en varios canales de longitud de 1Mhz y realizar 1600 saltos por segundo de una manera pseudoaleatoria. (Fernández, 2011).

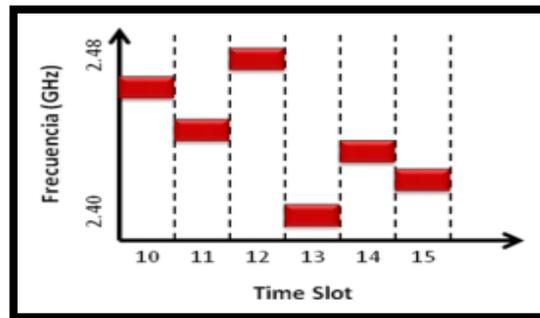


Figura 29. Espectro Ensanchado por Salto de Frecuencia
Fuente: <http://www.joefernandez.me/docs/MemoriaBlueWay.pdf>

Existen diferentes clases de la tecnología bluetooth que se utilizan de acuerdo a las necesidades de potencia y distancias de los enlaces, estas clases son totalmente compatibles.

Clase	Potencia máxima permitida	Rango
Clase 1	100 mW (20 dBm)	Aprox. 100 metros
Clase 2	2,5 mW (4 dBm)	Aprox. 10 metros
Clase 3	1 mW (0 dBm)	Aprox. 1 metro

Figura 30. Clases de Bluetooth
Fuente: <http://www.joefernandez.me/docs/MemoriaBlueWay.pdf>

²⁷ ISM= Industrial, Scientific and Medical (Industrial, Científicas y Médicas).

²⁸ FHSS= Frequency Hopping Spread Spectrum (Espectro Ensanchado por Salto de Frecuencia).

➤ MODULO BLUETOOTH HC-05

El módulo HC-05 es un dispositivo que tiene como función enviar y recibir datos inalámbricamente, la principal característica de este elemento es que puede configurarse como esclavo o como maestro, lo que permite aparte de poder conectarse directamente con un dispositivo móvil celular o tableta electrónica también permite conectarse con otros módulos bluetooth, además es ideal para interconexiones de corta distancia y es compatible con todos los microcontroladores y Arduino. (Guzhñay & Calle, 2015).

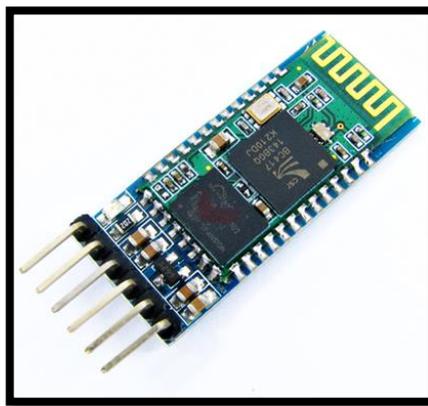


Figura 31. Modulo Bluetooth HC-05

Fuente: Diseño Propio basado en: <http://dspace.ups.edu.ec/bitstream/123456789/8843/1/UPS-CT005042.pdf>

Tabla 15. Características Modulo HC-05

MODO DE TRABAJO	Maestro y Esclavo
FRECUENCIA	ISM 2.4 GHz
CLASE	2
VELOCIDAD DE TRANSMISIÓN	2.1 Mbps Asíncrona
FUENTE DE ALIMENTACIÓN	3.3 a 5 V
DIMENSIÓN	26 x 13mm
SEGURIDAD	Encriptación y Autenticación
PINES	VCC, GND, TX, RX, KEY

Fuente: El Autor. Adaptado de: <http://dspace.ups.edu.ec/bitstream/123456789/8843/1/UPS-CT005042.pdf>

2.6.4.4.4 Tabla Comparativa de Tecnologías Inalámbricas

Tabla 16. Diferencias entre Tecnologías Inalámbricas

ATRIBUTOS	BLUETOOTH (HC-05)	ZIGBEE (XBee)	WI- FI (Wi-Fly RN- XV)
ESTÁNDAR	IEEE 802.15.1	IEEE 802.15.4	802.11
POTENCIA	≤ 4dbm (2.5mW)	0dB (1mW)	- Potencia de transmisión -82 a -93 dBm - Potencia de Recepción 15 dBm
APLICACIONES	- Enlaces de corta distancia - Transmisión de voz y datos - Conexiones con dispositivos móviles como celulares y tabletas electrónicas - Audífonos inalámbricos - Sistemas embebidos - Juguetes	- Control y monitoreo industrial - Red de sensores - Control de automatización del hogar (Domótica) - Redes de seguridad	- Redes WLAN - Internet de las cosas - Sistemas que requieren acceso a internet
VELOCIDAD DE TRANSMISIÓN	- 2.1 Mbps Asincrónica - 1 Mbps Sincrónica	250 Kbps (2.4ghZ)	11 & 54 Mbps
COBERTURA	10 metros	- 30 metros interiores - 60 metros en exteriores	50-100 metros
TIPO DE RED	- Ad-hoc - Punto a Punto	- Malla - Punto – Multipunto	- Punto – Multipunto - Ad-hoc

			- Punto a Punto
FRECUENCIA DE OPERACIÓN	2.4 GHz	2.4 GHz	2.4 GHz y 5 GHz
Seguridad	Autenticación y encriptación de 64 y 128 bits	Encriptación de 128 bits AES y permite aumentar la seguridad mediante software	WEP ,WPA, SSID
Consumo de corriente	40mA transmitiendo y 0.2mA cuando está en reposo	30mA transmitiendo 3uA en reposo	40mA transmitiendo y 20mA en reposo
Número máximo de dispositivos interconectados	7	32	Hasta 65000 nodos por red
Tiempo de establecimiento de conexión	10 segundos	30 ms	De 3 a 5 segundos
Costo	15 dólares aproximadamente (Mercado Libre Ecuador)	45 dólares aproximadamente (Mercado Libre Ecuador)	70 dólares aproximadamente (Mercado Libre Ecuador)

Fuente: El Autor

2.7 SENSORES Y ACTUADORES

En la actualidad la mayoría de sistemas electrónicos requieren medir, controlar y monitorear información de su entorno físico, para esto se necesita de elementos de entrada y salida llamados Transductores. Los Transductores son elementos que son capaces de transformar o convertir cualquier tipo de energía a señales eléctricas para que puedan ser procesadas o controladas.

Los Transductores de entrada son conocidos como sensores, estos se encargan de recibir información de una magnitud del exterior y convertirla en otra magnitud que por lo general es eléctrica para que pueda ser procesar, cuantificar y manipular. Los Transductores de salida son los que generan una respuesta a los sensores como una señal lumínica, sonora o generar movimiento.

2.7.1 SENSORES

Los sensores son dispositivos que se encargan de capturar y convertir un tipo de energía física, química o biológica a otro tipo de energía que por lo general es eléctrica. Estos dispositivos posibilitan la comunicación entre el mundo físico y los sistemas de medición o de control, tanto eléctricos como electrónicos, utilizándose extensivamente en todo tipo de procesos industriales y no industriales para propósitos de monitoreo, medición, control y procesamiento. (Roman, 2013).

2.7.1.1 Clasificación de los sensores según la señal de salida

Hoy en día se pueden realizar diferentes clasificaciones de los sensores, sin embargo para los sistemas electrónicos embebidos la forma más general de clasificarlos es atendiendo a la forma de codificar la magnitud medida podemos establecer una clasificación en:

2.7.1.1.1 Sensores Analógicos

Los sensores Analógicos son aquellos que dan como salida un valor de tensión o corriente variable en forma continua en el tiempo dentro del campo de medida. Normalmente en electrónica este tipo de sensores proporcionan salidas analógicas entre 0 y 5 voltios. (Cortes, 2015).

Ejemplos de Sensores Analógicos:

- Sensor de temperatura
- Sensor de presión
- Sensor de Caudal
- Sensor de proximidad
- Sensor de velocidad
- Sensor de aceleración
- Sensor de Fuerza

2.7.1.1.2 Sensores Digitales

Los sensores digitales son aquellos que dan como resultado una salida codificada en forma de pulsos o en forma de una palabra digital codificada en binario, BCD²⁹ u otro sistema cualquiera. Es decir de un sensor digital únicamente se puede obtener dos valores lógicos 1 o 0, estos sensores se basan en un umbral o límite para arrojar cualquiera de los dos resultados dependiendo de la función y configuración del sensor. En electrónica generalmente se conoce a 5 voltios como un 1 lógico y a 0 voltios como un 0 lógico (Cortes, 2015).

Ejemplos de Sensores Digitales:

- Sensor de presencia

²⁹ BCD= Decimal codificado en binario

- Sensor contador de pulsos
- Sensores codificadores de posición (Encoder).
- Sensor Hall
- Sensor de nivel

2.7.1.2 Sensores Ultrasónicos de Distancia

Los sensores ultrasónicos o de ultrasonido como también son conocidos son detectores de proximidad que perciben objetos a distancias de hasta 8 metros sin necesidad de un contacto físico directo. El sensor emite una señal de sonido que al llegar a un objeto se refleja y regresa hacia el sensor, el tiempo que transcurre desde el momento en que se emite la señal hasta que se recibe la señal reflejada permite obtener la distancia a la que está situado el objeto detectado.

Este tipo de sensor puede detectar cualquier tipo de objeto siempre y cuando este tenga un índice mínimo de reflexión que permita obtener una señal acústica reflejada hacia el sensor. Los materiales de los objetos pueden ser sólidos, líquidos o polvorientos de cualquier forma, color o material. (Almeida & Ochoa, 2013).

2.7.1.2.1 Sensor Ultrasónico HC-SR04

El Sensor HC-SR04 es un módulo compatible con la plataforma Arduino y la mayoría de los microcontroladores. HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más recientes.

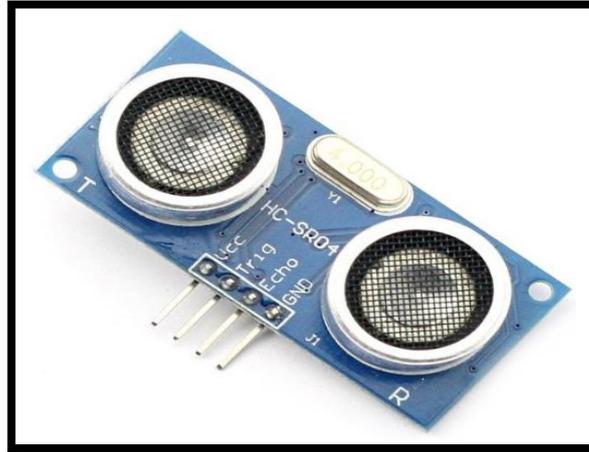


Figura 32. Sensor Ultrasónico HC-SR04

Fuente: <http://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/>

➤ **Funcionamiento**

El sensor HC-SR04 consta de un emisor de sonido de alta frecuencia y un receptor tipo micrófono. Para inicial la medición, se envía un pulso de 5 voltios durante 10 microsegundos en la entrada del pin TRIG del módulo. Esto hace que el modulo inicie la transmisión de 8 pulsos de un sonido a 40Khz a través del emisor, el pin ECHO se coloca en 5 voltios y mantiene este estado por un tiempo proporcional a la distancia. (Cabrera & Delgado, 2014). La distancia puede calcularse en metros o en centímetros con las siguientes formulas:

$$distancia[metros] = (tiempo ECHO[segundos]) * \frac{velocidad\ del\ sonido[\frac{m}{s}]}{2}$$

Ecuación 2. Cálculo de Distancia en metros

Fuente: <http://dspace.uazuay.edu.ec/bitstream/datos/3610/1/10292.pdf>

$$distancia[cm] = (tiempo ECHO[micro\ segundos]) * \frac{velocidad\ del\ sonido[\frac{cm}{us}]}{2}$$

Ecuación 3. Cálculo de Distancia en centímetros

Fuente: <http://dspace.uazuay.edu.ec/bitstream/datos/3610/1/10292.pdf>

Donde la velocidad del sonido en el Aire es 340 m/s o 0.034 cm/us

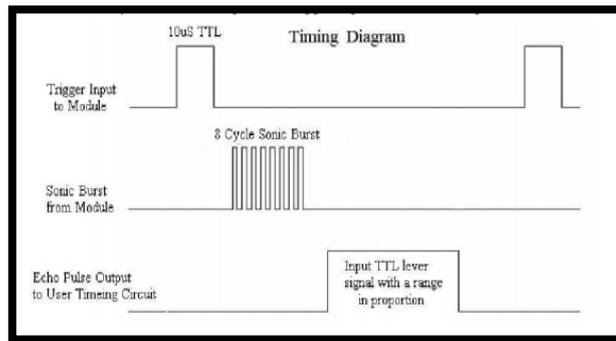


Figura 33. Funcionamiento del sensor HC-SR04
 Fuente: <http://www.micropik.com/PDF/HCSR04.pdf>

Tabla 17. Características Modulo HC-SR04

Voltaje de trabajo	DC 5V
Consumo de Corriente	15mA
Frecuencia de trabajo	40Hz
Distancia Máxima	4 metros
Distancia mínima	2 centímetros
Angulo de medición	15 grados
Señal de entrada TRIG	Pulso de 10 microsegundos
Dimensión	45*20*15 mm

Fuente: El Autor. Adaptada de: <http://www.micropik.com/PDF/HCSR04.pdf>

2.7.1.3 Sensores Encoder

Los encoder son sensores rotativos que transforman un movimiento angular en impulsos digitales. Estos impulsos generados pueden ser utilizados para controlar los desplazamientos de tipo angular o de tipo lineal mediante microcontroladores, contadores lógicos programables, sistemas de control entre otros. Las aplicaciones principales de estos sensores están en los sistemas de motores de máquinas, herramientas, robots y en aparatos de control y medición. (TABERNER, 2015)

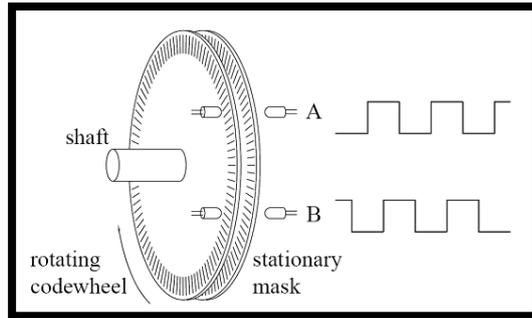


Figura 34. Sensor Encoder

Fuente: http://hades.mech.northwestern.edu/index.php/Rotary_Encoder

2.7.1.3.1 Tipos de Encoder

Existen dos tipos de sensores encoder según el tipo de tarea que se quiera llevar a cabo, el tipo de encoder absoluto nos permite obtener diferentes salidas en cualquier momento es decir no es necesario que este en rotación a diferencia del encoder incremental que brinda información del estado del eje cuando se encuentre en movimiento. (TABERNER, 2015).

2.7.1.3.1.1 Encoder Incremental

Los sensores encoder incrementales se caracterizan porque determinan la posición contando el número de impulsos que se generan cuando un rayo de luz es atravesado por los agujeros o marcas opacas en una superficie de un disco unido, este tipo de encoder genera pulsos digitales los cuales pueden ser procesados por el sistemas de control y poder determinar el ángulo de rotación. (TABERNER, 2015).

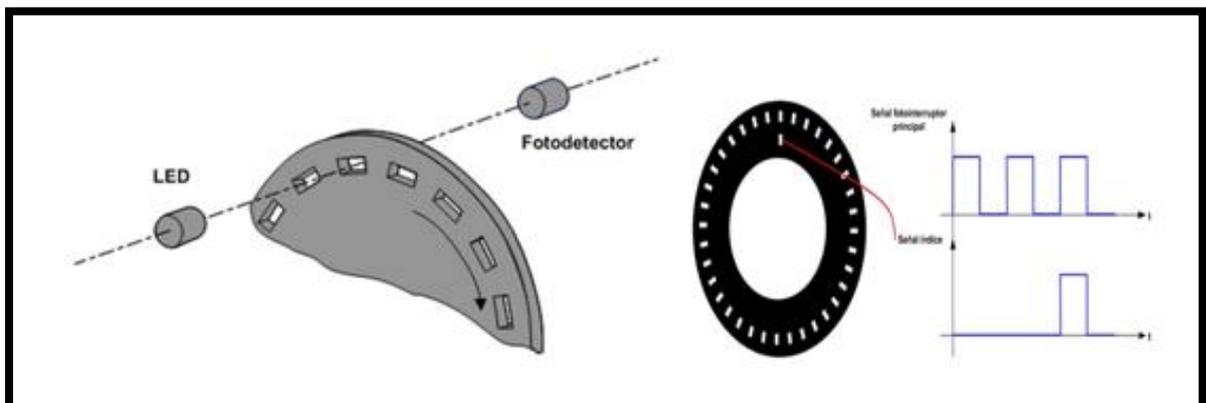


Figura 35. Encoder Incrementales

Fuente: <http://ira.unileon.es/es/book/export/html/320>

2.7.1.3.1.2 Encoder Absolutos

Los encoder absolutos son sensores más específicos y complejos que cuentan con un disco especial programado con una codificación absoluta, es decir que no cambia en el tiempo de ahí viene su nombre, los pulsos generados por este tipo de encoder representan una codificación que puede ser interpretada por el sistema de control, entre las principales aplicaciones en las que se destaca este tipo de encoder están la robótica avanzada, movimiento d cámaras de seguridad, elevadores, dispositivos de entrada en las computadoras, controles industriales, radares giratorios entre otros. (TABERNER, 2015).

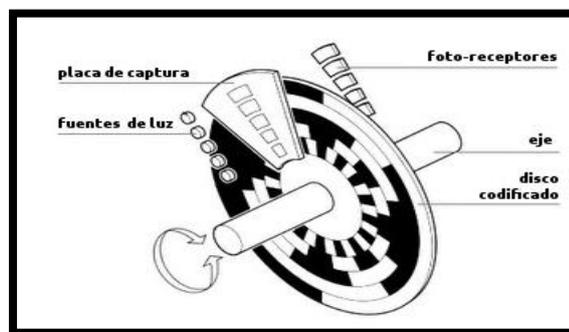


Figura 36. Encoder Absoluto

Fuente: <http://www.lbaindustrial.com.mx/que-es-un-encoder-absoluto/>

2.7.1.3.2 Módulo Sensor Encoder FZ0888

El Módulo lector encoder FZ0888 es un sensor de tipo incremental que ayuda en la creación de prototipos de robótica y automatización, el cual permite en base a pulsos digitales conocer la velocidad de un motor o la posición exacta del mismo. Este módulo es compatible con cualquier microcontrolador en especial con Arduino.

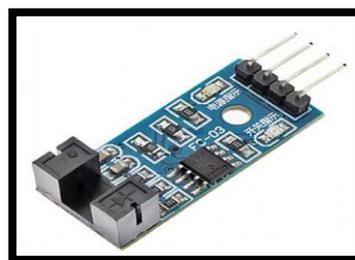


Figura 37. Módulo Encoder FZ0888

Fuente: <http://www.arduconce.cl/image/cache/data/sensor%20velocidad-01-500x500.jpeg>

Este sensor al ser un encoder incremental funciona detectando pequeñas muescas en un disco ranurado acoplado al motor, al estar el disco ranurado entre el receptor y el transmisor del sensor infrarrojo provoca una serie de interrupciones entre el haz de luz que viaja de un extremo a otro, la medir la cantidad de interrupciones de la luz infrarroja detectada se saber no solo hasta cuanto ha recorrido sino también la velocidad con la que lo ha hecho.

Tabla 18. Características del Módulo Encoder

Aplicaciones	Detección de velocidad del motor, conteo de pulsos, límites de posición
Voltaje de trabajo	3.3 V – 5 V
Consumo de Corriente	15 ma
Ancho de ranura	5mm
Salida	Digital

Fuente: El Autor. Adaptado de: http://es.aliexpress.com/store/product/Speed-Sensor-Module-Motor-Test-Module-Groove-Coupler-Module-FZ0888/406986_1981225359.html

2.7.2 ACTUADORES

Los actuadores en sistemas electrónicos son los elementos que generalmente ejercen de interfaces para convertir magnitudes físicas normalmente de carácter eléctrico a otro tipo de magnitud que permita actuar sobre el medio o proceso a controlar, por ejemplo un motor convierte los pulsos eléctricos en movimiento, un LED convierte los pulsos eléctricos en luz y así podemos tener otros ejemplos; en otras palabras los actuadores permiten interactuar o ejercer acciones ante las señales eléctricas.



Figura 38. Ejemplos de Actuadores Electrónicos
Fuente: El Autor

2.7.2.1 Motor de Corriente Directa

Los motores de corriente continua son dispositivos rotatorios que transforman una energía eléctrica en energía mecánica en base a la acción de un campo eléctrico que permite girar en los dos sentidos (horario o anti horario). Están compuestos por dos elementos principales que son el estator y el rotor. El estator es la parte mecánica del motor donde están los polos del imán. El rotor es la parte móvil del motor con devanado y un núcleo, al que llega la corriente a través de las escobillas. (DIY Makers, 2013).

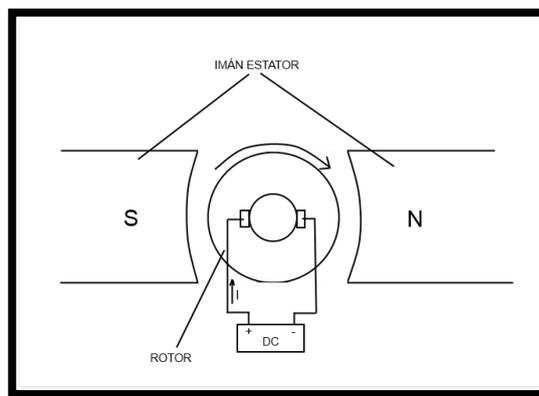


Figura 39. Elementos de un Motor de Corriente directa
Fuente: <http://diymakers.es/control-velocidad-y-sentido-de-motor-dc/>

Cuando la corriente eléctrica circula por el devanado del rotor, se crea un campo electromagnético. Este interactúa con el campo magnético del imán del estator. Esto deriva en un rechazo entre los polos del imán del estator y del rotor creando un par de fuerza donde el rotor gira en un sentido de forma permanente. (DIY Makers, 2013).

2.7.2.2 LED (Diodo Emisor de Luz)

El LED o diodo emisor de luz es un tipo especial de semiconductor, cuya característica principal es convertir en luz la corriente eléctrica de bajo voltaje que atraviesa su chip. Desde el punto de vista físico un LED común se presenta como un bulbo miniaturizado, carente de filamento o de cualquier otro tipo de elemento o material peligroso, con la ventaja sobre otras tecnologías que no contamina el medio ambiente. (García, 2015).

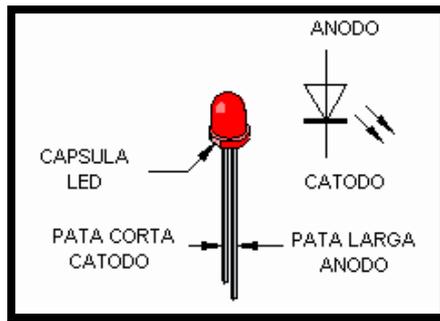


Figura 40. Diodo Emisor de Luz

Fuente: <http://logica-digital.blogspot.com/2007/11/las-tres-funciones-logicas-bsicas.html>

De acuerdo con las características de funcionamiento de cada LED, el chip puede operar con voltajes de polarización directa entre 1 y 4 voltios aproximadamente, con una corriente entre 10 y 40 miliamperios, que puede variar de acuerdo con el tamaño y color de luz que emite cada uno en particular. La aplicación de un voltaje o corriente de trabajo por encima del valor establecido por el fabricante para cada diodo LED en particular puede llegar a acortar su vida útil, o también destruirlo. (García, 2015).

2.8 COMPONENTES ELECTRÓNICOS

Los componentes electrónicos son los dispositivos o elementos que conforman un circuito electrónico. Estos elementos suelen estar encapsulados generalmente de material cerámico, metálico o plástico. Están diseñados para ser conectados entre ellos normalmente por soldadura. (Prada, 2015).

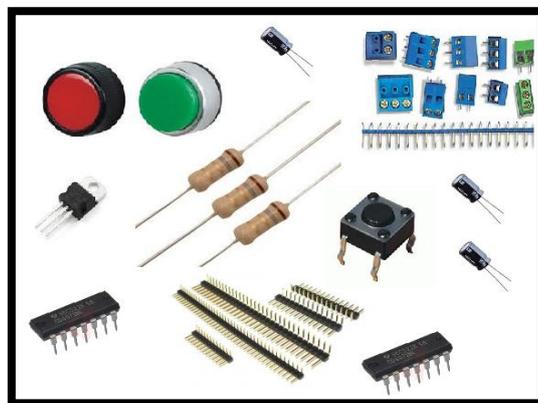


Figura 41. Componentes Electrónicos

Fuente: El Autor

2.8.1 RESISTENCIAS

Las resistencias son unos elementos eléctricos cuya misión es dificultar el paso de la corriente eléctrica a través de ellas. Su característica principal es su resistencia óhmica aunque tienen otra no menos importante que es la potencia máxima que pueden disipar. Ésta última depende principalmente de la construcción física del elemento. (Rubio, 2000).

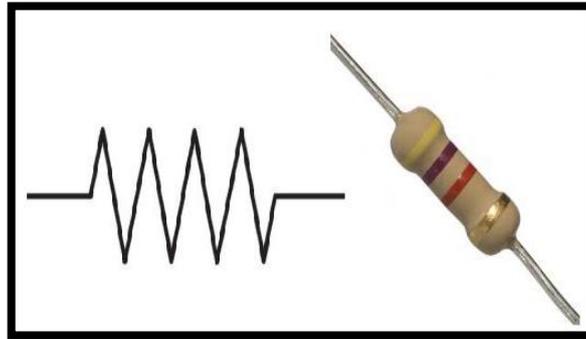


Figura 42. Símbolo y aspecto común de la resistencia
Fuente: El Autor

El valor de una resistencia viene determinado según la tabla del código de colores, en donde la primera y segunda banda representan las dos primeras cifras, la tercera banda señala por cuanto se debe multiplicar las dos primeras cifras o el número de ceros que hay que añadir a los dos primeros valores obtenidos; mientras que la última banda indica la tolerancia que es el porcentaje más o menos que podrá tener esa resistencia. (Rubio, 2000).

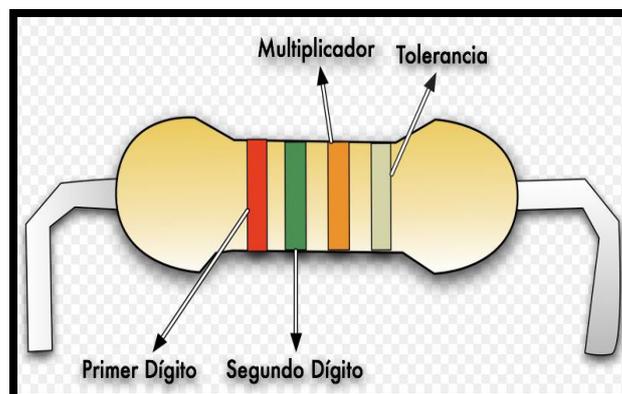


Figura 43. Indicadores del Valor de una resistencia
Fuente: <http://componenteselectronicosanaydig.blogspot.com/2015/02/resistencias.html>

Tabla 19. Código de Colores

COLOR	PRIMERA BANDA	SEGUNDA BANDA	TERCERA BANDA	TOLERANCIA
Negro	0	0	x 1	
Café	1	1	x 10	2%
Rojo	2	2	x 100	
Naranja	3	3	x 1000	
Amarillo	4	4	x 10000	
Verde	5	5	x 100000	
Azul	6	6	x 1000000	
Violeta	7	7	x 10000000	
Gris	8	8	x 100000000	
Blanco	9	9	x 1000000000	
Dorado				5%
Plata				10

Fuente: Obtenida de: <http://componenteselectronicosanaydig.blogspot.com/2015/02/resistencias.html>

2.8.2 REGULADOR LM7805

El LM7805 es un regulador de voltaje que entrega 5 voltios y una corriente máxima de un Amperios a partir de un voltaje de entrada entre 7 y 25 voltios. Pertenece a la serie de los reguladores LM78XX y se utiliza principalmente para alimentar microcontroladores, circuitos integrados y otros elementos electrónicos activos. (Arroyo, 2014).

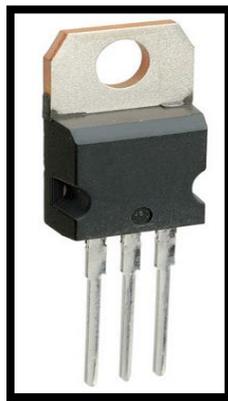


Figura 44. Regulador LM7805

Fuente: <http://static.rapidonline.com/catalogueimages/Module/M077611P01WL.jpg>

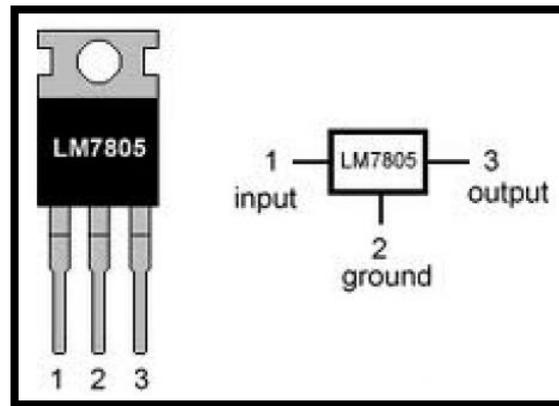


Figura 45. Diagrama de Pines LM7805

Fuente: http://potentiallabs.com/cart/image/cache/data/7805_1-500x416.jpg

2.8.3 MULTIPLEXOR HEF4051B

Los multiplexores son circuitos combinacional con varias entradas y una salida de datos, y están dotados de entradas de control capaces de seleccionar una, y sólo una, de las entradas de datos para permitir su transmisión desde la entrada seleccionada a la salida que es única. (Angelfire, 2015).

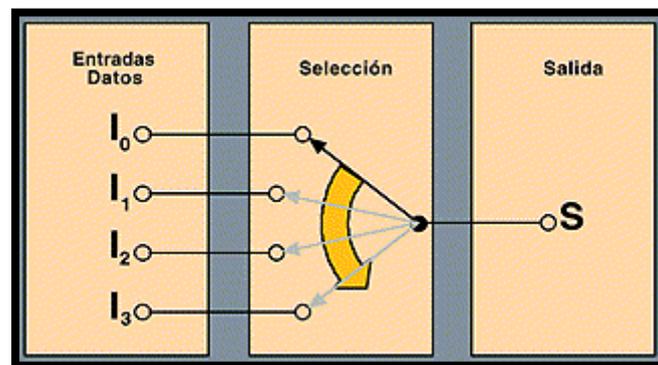


Figura 46. Funcionamiento Multiplexor

Fuente: <http://www.angelfire.com/al2/Comunicaciones/Laboratorio/multiple.html>

La entrada seleccionada viene determinada por la combinación de ceros (0) y unos (1) lógicos en las entradas de control. La cantidad de entradas de control será dada en función de las salidas que disponga el multiplexor. (Angelfire, 2015).

El circuito integrado HEF4051B es un Multiplexor analógico de 8 canales. La selección de cada una de las 8 entradas se realiza con las líneas A, B y C. El pin 3 es común a las 8 entradas. También cuenta con un pin de inhibición el cual cuando está conectado a un nivel alto pone todas las entradas en un nivel alto y cuando está en un nivel alto habilita el circuito.

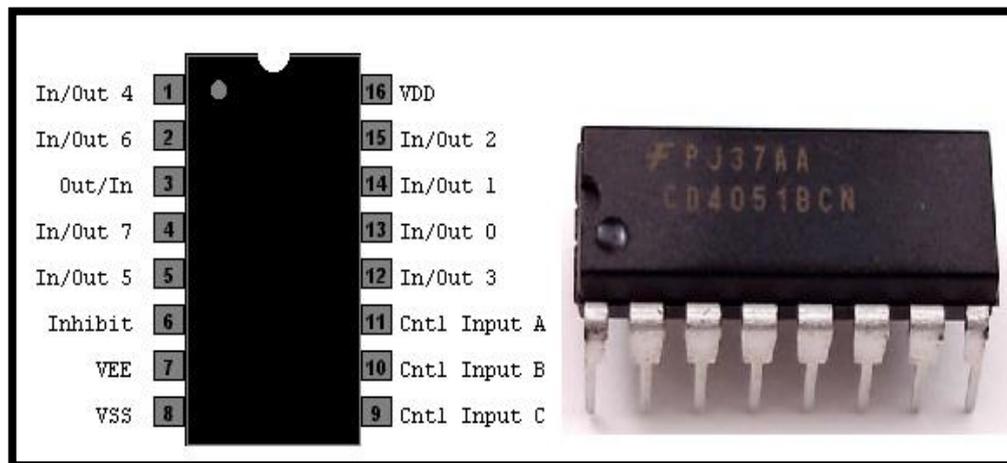


Figura 47. Chip HEF4051
Fuente: El Autor

Input				Channel ON
E	S3	S2	S1	
L	L	L	L	Y0 to Z
L	L	L	H	Y1 to Z
L	L	H	L	Y2 to Z
L	L	H	H	Y3 to Z
L	H	L	L	Y4 to Z
L	H	L	H	Y5 to Z
L	H	H	L	Y6 to Z
L	H	H	H	Y7 to Z
H	X	X	X	switches off

[1] H = HIGH voltage level; L = LOW voltage level; X = don't care.

Figura 48. Tabla de verdad 4051
Fuente: http://www.nxp.com/documents/data_sheet/HEF4051B.pdf

2.8.4 BATERÍAS RECARGABLES

En la actualidad existe una gran cantidad de dispositivos electrónicos que hacen uso de diferentes tipos de baterías por lo que existen numerosos tipos que se pueden diferenciar por

su tamaño, capacidad, peso, duración. Básicamente una batería es un recipiente de almacenamiento de la electricidad que necesita una herramienta determinada, esta electricidad se almacena de forma química para posteriormente se convierta en energía eléctrica. (Briconatur, 2014).



Figura 49. Baterías Recargables
Fuente: El Autor

2.8.4.1 Batería de Níquel de Cadmio (Ni CD)

Este tipo de batería funciona a partir de un ánodo de cadmio y un cátodo compuesto por hidróxido de níquel. Por su parte, el electrolito se conforma de hidróxido de potasio. Pueden ser recargadas una vez gastadas. Tienen efecto memoria (importante no cargar hasta que están totalmente vacías) y el proceso de descarga no es lineal (van perdiendo carga poco a poco). (Briconatur, 2014).



Figura 50. Ejemplos de baterías de Ni CD

Fuente: http://upload.wikimedia.org/wikipedia/commons/thumb/4/41/NiCd_various.jpg/225px-NiCd_various.jpg

2.8.4.2 Batería de Níquel Metal hidruro (Minh)

Emplean un ánodo de hidróxido de níquel y un cátodo compuesto por una aleación de hidruro metálico. Son muy ligeras, mejoran la capacidad de almacenamiento en un 40% por lo tanto, para las mismas características, pesan menos (portátiles, móviles, herramienta profesional). No presentan efecto memoria pero son sensibles a la temperatura (cuanto más frío menos rendimiento). (Briconatur, 2014).



Figura 51. Ejemplos de Baterías de Minh

Fuente: <http://www.tacbattery.com/photo/pl286238-baterias-recargables-1800mah-del-ni-quel-e-hidruro-metalico-de-alta-temperatura-del-ni-mh.jpg>

2.8.4.3 Batería de Ion de Litio (Li-ion)

Este tipo de baterías son las más utilizadas en la actualidad por los dispositivos como celulares, tabletas electrónicas y computadoras, tiene un ciclo de vida alto (500-600 cargas/descargas) pero son mucho más estables y fiables que los anteriores tipos. La curva de descarga es lineal. Funciona prácticamente hasta quedarse sin carga, no tienen efecto memoria, mayor densidad de carga (menor tamaño y menor peso) y mayor voltaje por célula. (Briconatur, 2014).



Figura 52. Ejemplos de baterías de Ion de Litio

Fuente: <https://sites.google.com/site/foxserverhost/home/imagenes/bateria-litio-300x300.jpg>

2.8.4.4 Batería de Polímero de Litio (Lio)

La batería de polímeros de litio o Lio es muy similar a las baterías de iones de litio con la ventaja de que posee un tamaño muy reducido y un tiempo de vida estimado mayor que la de ion de litio, la desventaja es su costo económico. (Briconatur, 2014).



Figura 53. Ejemplo Batería Lio

Fuente: <http://blog.briconatur.com/baterias-recargables-tipos-y-caracteristicas/>

2.8.4.5 Comparativa de los tipos de Baterías recargables

Tabla 20. Comparativa de Tipos de Baterías Recargables

TIPO	Níquel-Cadmio (Ni-Cd)	Níquel-Hidruro (Ni-MHZ)	Ion de Litio (Li-ion)	Polímeros de Litio (Lio)
Voltaje por célula	1.2V	1.2V	3.7V	3.7V
Amperaje hora	0.5-1A	0.5-10Ah	Varios tipos	Varios tipos
Memoria	Muy alto	Bajo	Inexistente	Inexistente
Sobrecarga	Soportado	No recomendable	Soportado	Soportado
Numero de Recargas (Aproximadamente)	500	1000	4000	5000
Tiempo de descarga/mes	30%	20%	6%	6%
Tiempo de carga	10-14horas	2-4 horas	2-4 horas	1-1.5horas

Fuente: El Autor. Adaptado de: <http://blog.briconatur.com/baterias-recargables-tipos-y-caracteristicas/>

2.9 HERRAMIENTAS DE SOFTWARE

2.9.1 ISIS PROTEUS

ISIS Proteus es un programa de diseño electrónico creado por Labcenter Electronics que permite diseñar circuitos con una variedad de componentes que van desde simples resistencias hasta sofisticados microcontroladores, incluyendo fuentes de alimentación, generadores de señales y muchas otras prestaciones. Los circuitos electrónicos diseñados en este software pueden ser simulados en tiempo real gracias a que permite introducir numerosas familias de microcontroladores y añadir programas compilados que se quieren llevar a cabo. Esta herramienta además de permitir simular microcontroladores también permite combinar diferentes dispositivos como por ejemplo pantallas LCD³⁰ o GLCD³¹, Motores, compuertas lógicas, circuitos integrados, displays, entre otros. (Lecaro, Gonzalez, & Valdivieso, 2011).



Figura 54. Interfaz de Inicio de Proteus

Fuente: <http://2.bp.blogspot.com/-hvcT88fNBT8/U36hTs5oCjI/AAAAAAAAAC8/kfePiKV9afM/s1600/proteus.png>

2.9.2 EAGLE

EAGLE es un software de diseño de diagramas y circuitos impresos con autor enrutador. Este software es muy conocido a nivel mundial debido a que posee muchas versiones gratuitas y con una gran cantidad de bibliotecas que permiten diseñar placas para infinidad de

³⁰ LCD= Pantalla de Cristal Liquido

³¹ GLCD= Pantalla Grafica de Cristal Liquido

dispositivos electrónicos. EAGLE contiene un editor de diagramas electrónicos en el que los componentes pueden ser colocados con un solo clic y fácilmente enrutarlos con los otros componentes tanto de forma manual como automática. (Hernández, 2013).



Figura 55. Interfaz de inicio de EAGLE

Fuente: https://blog.adafruit.com/wp-content/uploads/2013/12/1619_LRG.jpg

2.10 MATERIAL DE LOS JUGUETES

Para la elaboración de juguetes se puede emplear distintos materiales que pueden clasificarse según su compuesto o elaboración en materiales naturales o sintéticos. Los materiales naturales son aquellos que se encuentran en la naturaleza como la madera, piedras, algodón, lana entre otros y los materiales sintéticos son aquellos que han sido creados artificialmente y cuya existencia no ha sido observada en la naturaleza.

El tipo de material utilizado en la construcción de juguetes es importante para que su elaboración y consumo no tenga impacto sobre el entorno y la salud de las personas en especial de los niños que son los principales consumidores, pero la elección del material no solo va acorde con las personas sino que también con el medio ambiente y al costo económico de producción. La explotación de madera, papel y plásticos provoca deforestación de bosques, contaminación del aire, gran consumo de agua y alto costo de procesamiento por lo que se debe

tratar de elegir el material más acorde al propósito del juguete y a la vez amigable con el ambiente.

Tabla 21. Comparativa de Materiales

CARACTERÍSTICAS	MADERA	PLÁSTICO	METAL
Componente	Natural	Químico sintético	Químico natural
Conductividad eléctrica	Baja (aislante)	Baja (aislante)	Alta (conductor)
Potencial de interferencia	Bajo	Bajo	Alto
Resistencia	Alta	Media	Alta
Flexibilidad	Media (depende del grosor)	Alta (depende del compuesto)	Alta (depende del compuesto)
Durabilidad	Alta	Media	Alta
Elasticidad	Baja	Media	Baja
Costo de procesamiento	Bajo	Medio	Alto

Fuente: El Autor. Basada en: https://prezi.com/heeln_9hegha/propiedades-de-la-madera-metales-plasticos-ceramicos-petres/

CAPÍTULO III. DESARROLLO Y PRUEBAS

En este capítulo se indica el proceso de diseño y construcción de cada uno de los elementos del Jugete Electrónico Didáctico para la enseñanza de programación a niños y niñas de 4 a 7 años, en el proceso se tomara en cuenta la elección del material acorde con el juguete y los diferentes elementos o dispositivos electrónicos que mejor se adapten a la propuesta para que permitan tener un buen funcionamiento del dispositivo.

3.1 DESCRIPCIÓN GENERAL DEL PROTOTIPO

En el presente prototipo el cual será denominado JED-Pro (Jugete Electrónico Didáctico Programable) tiene como característica principal la lógica de programación tangible para niños. Esta lógica de programación está basada en pequeñas piezas de instrucciones con diferente valor resistivo, las cuales son insertadas en un tablero electrónico en forma de secuencia para poder ser leídas a través de los pines analógicos del microcontrolador que en este caso va ser la plataforma Arduino.

La lógica de programación tangible diseñada en este proyecto consta de 8 bloques de instrucciones diferentes, esto conlleva a la necesidad de utilizar 8 valores resistivos que permitan diferenciar cada una de las funciones o instrucciones de las fichas insertadas sin ningún problema, por lo que estos valores deben ser bastante distantes entre sí.

Las fichas de instrucciones son insertadas en la interfaz de programación, la cual consta de 16 entradas y un circuito de control basado en la plataforma Arduino, Este circuito de control se encarga de leer el valor resistivo de cada una las instrucciones insertadas y va relacionando el valor leído con el valor de las instrucciones previamente asignado. Después de

identificar va almacenando las instrucciones temporalmente en un vector alojado en la memoria flash mientras termina de leer todas las instrucciones. Una vez finalizada la lectura de instrucciones procede a enviar en forma de cadena una por una las instrucciones a través del puerto serial conectado a un módulo inalámbrico hacia el robot representado por un carrito que procede a ejecutar las instrucciones una después de otra hasta culminar la secuencia. Todo este proceso se realiza en milisegundos una vez que el niño pulse el botón de inicio igualmente ubicado el tablero.

El robot recibe todas las instrucciones y las almacena en un vector en el orden en que fueron recibidas para posteriormente ejecutarlas, entre las instrucciones que puede ejecutar el robot están: Avance Adelante, Giro a la Derecha y Giro a la Izquierda; Estas son las instrucciones básicas las cuales se pueden acompañar con estructuras de control repetitivas o condicionales como IF y FOR.

3.1.1 ELECCIÓN DE LAS PLACAS ARDUINO

Como se puede ver en el Capítulo II Arduino cuenta con varios modelos de placas acordes a las necesidades y funcionalidades del proyecto. Una de las características que más resalta en este proyecto es la necesidad de que la placa Arduino a utilizarse disponga de 16 entradas analógicas, la cual según la *Tabla 8* la única placa que cuenta con esta cantidad de entradas es la Arduino Mega 2560. La placa Arduino Mega puede cumplir con la necesidad más sobresaliente de nuestro dispositivo sin embargo se tiene la desventaja de que es una placa relativamente grande y también cuenta con más recursos de los requeridos, la idea es elegir la placa que cumpla los requerimientos del sistema sin desperdiciar recursos y optimizar el coste económico. En base a esta idea se optó por elegir la placa Arduino Mini PRO para los dos elementos del prototipo ya que cumple con los recursos y funcionalidades requeridas tanto para

la interfaz programable como para el robot. La placa Arduino Mini PRO es destinada para proyectos permanentes y semipermanentes por lo que cuenta con las dimensiones apropiadas y un coste relativamente económico en comparación a las demás placas que poseen características similares. El único inconveniente de la placa Mini PRO es que cuenta únicamente con 6 entradas analógicas, a esto se le dio solución mediante la incorporación de un circuito multiplexor que permite obtener las 16 entradas a partir de la utilización de 2 entradas analógicas de la placa.

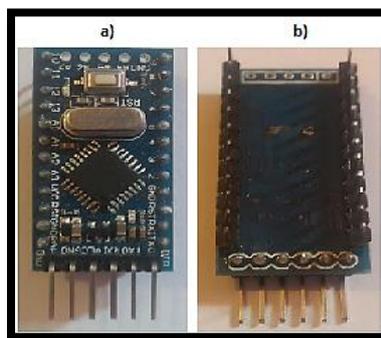


Figura 56. a) Arduino Mini PRO vista frontal. b) Arduino Mini PRO vista posterior
Fuente: El Autor

Arduino Mini Pro es la placa más pequeña existente en el mercado por lo que su tamaño se adapta perfectamente a las dimensiones del juguete electrónico a implementar, además cuenta con los requerimientos funcionales como lo es la comunicación serial, lectura de valores analógicos y la cantidad de salidas digitales requeridas tanto para el robot como para la interfaz de programación tangible.

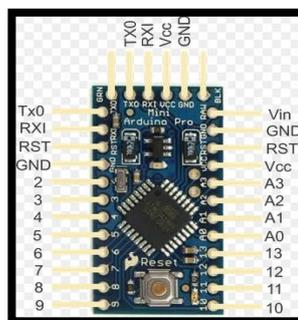


Figura 57. Distribución de Pines Arduino Mini PRO

Fuente:

http://www.engineersgarage.com/sites/default/files/imagecache/Original/wysiwyg_imageupload/4214/Arduino%20Pro%20Mini%20copy.png

En lo que corresponde a alimentación de voltaje una de las grandes ventajas de este modelo es que cuenta con un regulador interno que permite polarizar la placa con una fuente externa de hasta 12 voltios, lo que es de gran ayuda ya que en el circuito del robot se requiere una alimentación de 7.4 voltios debido al uso de motores y el sensor ultrasónico que son adaptados al mismo circuito.

3.1.2 ELECCIÓN DE LA TECNOLOGÍA INALÁMBRICA

En el Capítulo 2 se muestra algunas características de las tecnologías inalámbricas que son compatibles con la plataforma Arduino, entre las cuales se tiene la tecnológica Wi-Fi, ZigBee y Bluetooth. De estas tecnologías se optó por elegir la tecnología bluetooth ya que es la que mejor se adapta a las necesidades de comunicación requeridas para la comunicación entre el tablero de programación y el robot.

Se descartó las tecnologías Wi-Fi y ZigBee debido a la complejidad de su implementación y escases de módulos en el mercado nacional lo que hace a estas tecnologías sean mucho más costosas e inaccesibles, además estas dos tecnologías sobreestiman las necesidades del juguete ya que únicamente se requiere un enlace punto a punto a corta distancia. Otro de los inconvenientes de estas tecnologías es que sus módulos trabajan a 3.3 voltios lo que crea la necesidad de implementar circuitos extras para obtener la tensión necesaria.

Uno de los criterios más importantes para elegir la tecnología bluetooth fue la accesibilidad a los módulos compatibles con Arduino que se tiene en el país al ser una tecnología muy difundida y utilizada; Esto resulta ser una ventaja ya que en el caso de que el modulo tenga algún inconveniente se podría adquirir otro con facilidad para poder cambiarlo.

El modulo seleccionado es el HC-05, este cumple sin inconvenientes con el requerimiento de distancia del juguete que es de 1 a 5 metros en ambientes cerrados, se puede configurar como maestro o esclavo lo que permite tener comunicación entre dos módulos; trabaja a una tensión de 5 voltios que es similar a la tensión de la placa Arduino por lo que no necesita un circuito adicional para polarizarlo y por último es relativamente económico en comparación a los módulos de las otras tecnologías lo que resulta bastante beneficioso ya que al tratarse de un juguete se debe también optimizar su precio.

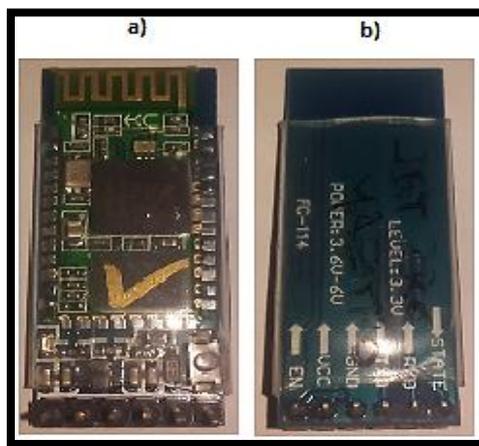


Figura 58. Modulo Bluetooth HC-05 a) vista frontal. b) vista posterior
Fuente: El Autor

El módulo HC-05 cuenta básicamente de 5 pines fundamentales, contiene un pin Rx (Receptor) mediante el cual recibe los datos de la placa Arduino para enviarlos inalámbricamente, un pin Tx (transmisor) a través del que envía los datos recibidos hacia la placa, un pin VCC que va conectado a la fuente de voltaje de 5 voltios, un pin GND que va conectado a tierra y por ultimo un pin ENABLE el cual se debe activar cuando se quiere realizar las configuraciones del módulo.

3.1.3 ELECCIÓN DEL MATERIAL PARA LAS CARCASAS DE JED-PRO

La elección del material a utilizarse en la carcasa de los juguetes puede resultar un poco ambiguo ya que depende de los gustos y criterios de las personas, en la actualidad existen juguetes de plástico, metal o madera; Algunos de estos materiales son más acogidos que otros, todo depende del propósito del juguete, funcionalidades y de las razones de los usuarios. Sin embargo para la implementación del prototipo del juguete didáctico propuesto en este trabajo se optó por elegir la madera como material principal tanto para la carcasa del robot como para la carcasa del tablero de comunicación.

La madera es un producto natural, renovable y amigable con el ambiente que se adapta adecuadamente a los propósitos del juguete propuesto ya que según pedagogos y padres de familia la madera logra estimular de forma natural los sentidos de los niños debido a su tacto, olor y sonido agradable que produce, además es un material muy resistente lo que hace que el juguete sea más duradero y que las bacterias no puedan reproducirse fácilmente lo que también es bueno para cuidar la salud de los niños.

Otro de los criterios importantes que se tomó en cuenta es el bajo índice de interferencia que posee este material ya que al poseer una comunicación inalámbrica es importante tener la menor pérdida posible de la potencia de la señal, también es un material aislante lo que asegura que los niños estarán totalmente protegidos de la parte electrónica.

El derivado de madera seleccionado para este trabajo es MDF³² de 5 mm de espesor el cual es muy utilizado en juguetes debido a que es muy ligero, no se hincha con la humedad, no

³² MDF= Fibra de Densidad Media

se apostilla como otros tipos de madera, permite mejores acabados ya que es fácil de moldear y pintar.

Todos los elementos correspondientes a la carcasa del juguete con este tipo de madera pueden ser elaborados a mano lo que resulta beneficioso para el ambiente al no necesitar de maquinaria eléctrica que consuma varias horas de energía y contaminación para su procesamiento.



Figura 59. Tableros de MDF

Fuente: http://i01.i.aliimg.com/img/pb/222/909/212/1237873281114_hz_mylibaba_web5_3154.jpg

3.1.4 DIAGRAMA DE BLOQUES

Mediante un diagrama de bloques se explica gráficamente como va ser la interconexión de los diferentes dispositivos que conforman el prototipo del juguete didáctico propuesto en este trabajo.

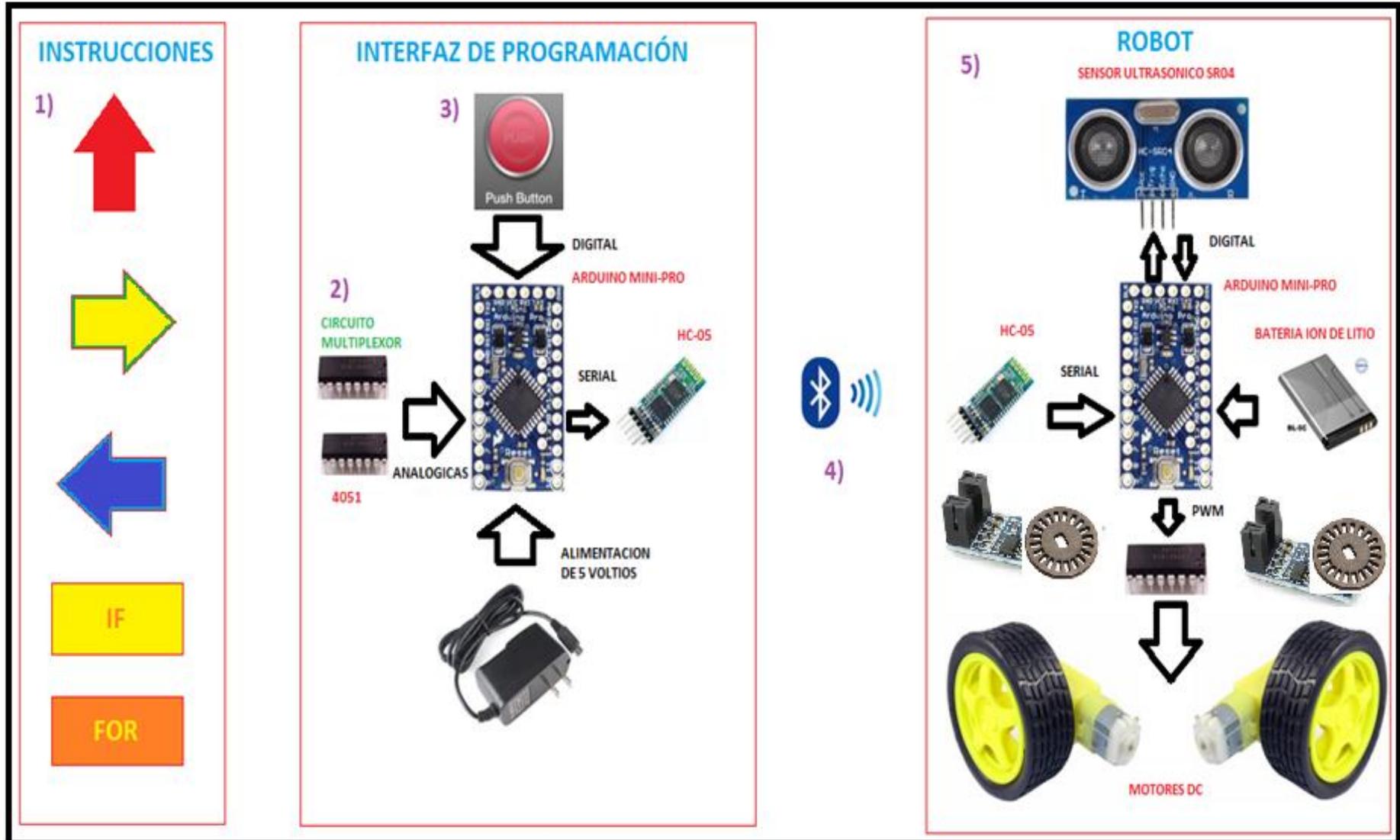


Figura 60. Diagrama de Bloques del Prototipo JED-Pro
Fuente: El Autor

En la *Figura 60* se puede observar un diagrama de bloques de los elementos que conforman el prototipo JED-Pro y como estos se encuentran interconectados, este diagrama está dividido en cinco partes las cuales se describen a continuación:

- 1) El bloque de instrucciones está compuesto por un total de 27 fichas divididas de la siguiente manera: 4 de Avance Adelante, 4 de Giro a la Derecha, 4 de Giro a la Izquierda, 4 de estructuras de control repetitivas (2 de dos repeticiones y 2 fichas de tres repeticiones), una de estructura control IF y 10 de inicio y fin que permiten indicar cuando una estructura de control inicia y cuando termina. Estas fichas de instrucciones son insertadas en forma de secuencia en la interfaz de acuerdo a la programación que se le quiera dar al robot.
- 2) El circuito multiplexor está basado en 2 circuitos integrados HEF4051B, el cual permite obtener las 16 entradas analógicas a partir de la utilización de 2 entradas en la placa Arduino Mini-Pro.
- 3) La interfaz de programación está compuesta básicamente por una placa Arduino Mini-Pro, un módulo bluetooth HC-05, un botón de inicio y la fuente de alimentación de 5 voltios. Después de insertar las fichas de instrucciones se procede a presionar el botón el cual dará inicio a la lectura y procesamiento de las instrucciones para luego enviarlas al robot.
- 4) La comunicación inalámbrica se la realizara mediante la tecnología bluetooth utilizando los módulos HC-05 los cuales tienen un alcance aproximado de 10 metros en ambientes cerrados y sin interferencia.
- 5) El robot está compuesto por una placa Arduino Mini-Pro, un sensor ultrasónico, un módulo HC-05 mediante el cual recibe los datos, un driver que permite controlar la

corriente, un par de motores de corriente directa y una batería de ion de litio para la alimentación del circuito.

3.2 DISEÑO DE LA LÓGICA DE PROGRAMACIÓN TANGIBLE PARA NIÑOS

La programación consiste en escribir instrucciones para que un computador o sistema embebido ejecute una tarea específica; pero como se explica en el capítulo 2, para poder indicarle al computador o controlador que acciones debe realizar para resolver un problema se necesita de un lenguaje de programación, el cual nos permite comunicarnos con el sistema y que este pueda entendernos. En la actualidad existen varios lenguajes de programación los cuales pueden ser unos más fáciles que otros, todo depende del sistema a programar y del propósito del mismo. Hoy en día se quiere involucrar a los niños en este mundo de la programación sin embargo los lenguaje de programación de alto nivel son en su mayoría textuales por lo que están siempre sujetos a errores de escritura y pueden resultar muy complejos para ellos.

Es por esta razón que en este proyecto se propone en base a un lenguaje de programación Arduino crear una lógica de programación tangible la cual le permita a los niños de una forma fácil y sin uso de dispositivos inteligentes o computadoras poder aprender programación de una forma lúdica es decir a través del juego.

3.2.1 PROGRAMACIÓN TANGIBLE

La programación tangible está orientada al aprendizaje y consiste en hacer que las instrucciones se vuelvan físicas es decir que se puedan tocar o manipular directamente con nuestras manos. El propósito de la programación tangible es ocultar las varias líneas de código que pueden asustar o intimidar a los usuarios bajo fichas físicas que tengan funciones

determinadas para facilitar la asignación de instrucciones, en este caso se va utilizar la programación tangible para guiar un pequeño robot que cumple con las siguientes instrucciones: avance adelante, giro a la derecha y giro a la izquierda.

En la *Tabla 22* se va especificar las instrucciones y estructuras que va a contener la programación tangible de JED-Pro.

Tabla 22. Instrucciones y Estructuras de control

TIPO	FUNCIÓN	DETALLE
Instrucción	Avance Adelante	Recorre hacia adelante aproximadamente 20 cm
Instrucción	Giro a la Derecha	Gira en su propio eje aproximadamente 90 grados
Instrucción	Giro a la Izquierda	Gira hacia la izquierda en su propio eje aproximadamente 90 grados
Estructura de control	FOR X2	Repite 2 veces el conjunto de instrucciones especificadas
Estructura de control	FOR X3	Repite 3 veces el conjunto de instrucciones especificadas
Estructura de control	OBSTÁCULO	Si hay Obstáculo ejecuta las instrucciones especificadas
Control	Inicio	Indica donde inicia una estructura de control
Control	Fin	Indica donde termina una estructura de control

Fuente: El Autor

Las instrucciones y estructuras de control especificadas en la *Tabla 22* van a estar representadas cada una por una ficha, estas fichas contienen una resistencia de diferente valor el cual al ser leído por la interfaz de programación esta podrá identificar la fusión para enviarla al robot.

Tabla 23. Colores y valores resistivos de las fichas

FICHA	COLOR	VALOR DE RESISTENCIA
Avance hacia Adelante	Rojo	100k Ω
Giro a la Derecha	Amarillo	47k Ω
Giro a la Izquierda	Azul	10k Ω
FOR X2	Azul	5.6k Ω
FOR X3	Naranja	4.7k Ω
OBSTÁCULO	Amarillo	3.3k Ω
INICIO	Verde	1k Ω
FIN	Rojo	330 Ω

Nota: Los valores resistivos especificados en esta tabla fueron elegidos en base a pruebas para obtener valores analógicos bastante distantes con la finalidad de que no se confundan las instrucciones. Fuente: El Autor

3.2.2 LÓGICA DE PROGRAMACIÓN

La Lógica de Programación se refiere a las reglas y estructuras que se deben cumplir para que el dispositivo a programar interprete las instrucciones de forma correcta. La lógica de programación realizada en este trabajo está dirigida a niños de 4 a 7 años por lo que debe ser lo más amigable y fácil posible para que puedan entenderla.

3.2.2.1 Especificaciones

- Las instrucciones deben ser insertadas en la interfaz de forma ordenada es decir una instrucción después de otra, esto es para que el niño vaya entendiendo que para que algo funcione como él quiere debe cumplir un orden específico.
- La interfaz de programación posee 16 entradas las cuales pueden ser utilizadas tanto para las instrucciones básicas como para las estructuras de control.

- Para utilizar estructuras de control debe existir por lo menos una instrucción básica
- Cuando se utiliza estructuras de control se debe indicar mediante las fichas de control donde inicia y donde termina cada estructura.
- Cuando se realiza una secuencia únicamente con instrucciones básicas no es necesario insertar las fichas de inicio y fin.
- Las estructuras serán ejecutadas en el orden que fueron insertadas una después de otra, en caso de existir una estructura de control repetitiva primero termina de ejecutarse el ciclo completamente y después sigue con las siguientes instrucciones en el caso de existir.
- El niño después de insertar su secuencia de instrucciones presiona el botón de inicio para que el robot ejecute las instrucciones, en caso de que no funcione como él quiere puede verificar su secuencia una vez el robot termina la secuencia anterior.

3.2.2.2 Estructuras

La estructura es el orden y los parámetros que se deben cumplir para ejecutar una secuencia. A continuación se va especificar las estructuras para programar instrucciones simples, estructuras de control repetitivas y estructura de control condicional.

3.2.2.2.1 Estructura Simple

Estructura Simple es aquella secuencia que está conformada únicamente de instrucciones básicas. Este tipo de estructura no necesita ningún parámetro de control únicamente que sean insertadas en orden para que puedan ser leídas, compiladas y ejecutadas por el Robot.

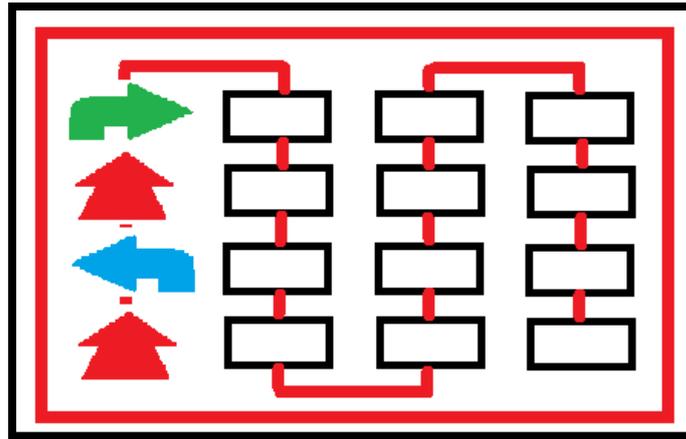


Figura 61. Ejemplo de Estructura Simple
Fuente: El Autor

En la *Figura 61* se puede observar un ejemplo de estructura simple que cumple la siguiente secuencia, Avance adelante, Giro a la izquierda, Avance adelante y Giro a la derecha. Cabe recalcar que las fichas están insertadas en orden y que no es necesario que llenar todas las entradas con fichas.

3.2.2.2 Estructura de Control Repetitiva

La estructura de control repetitiva es aquella que permite repetir una misma secuencia por determinado número de veces (2 o 3 repeticiones). Esta estructura al igual que la estructura simple debe tener las instrucciones en orden y además incorpora las fichas de control para indicar en donde inicia y donde termina la secuencia a ejecutarse.

SINTAXIS

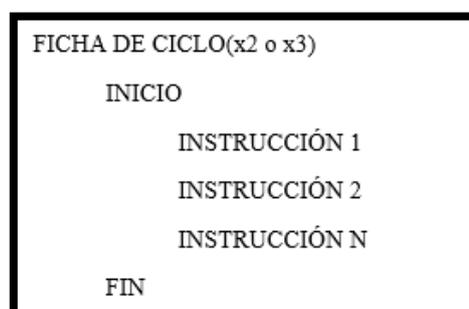


Figura 62. Sintaxis Estructura de Control Repetitiva
Fuente: El Autor

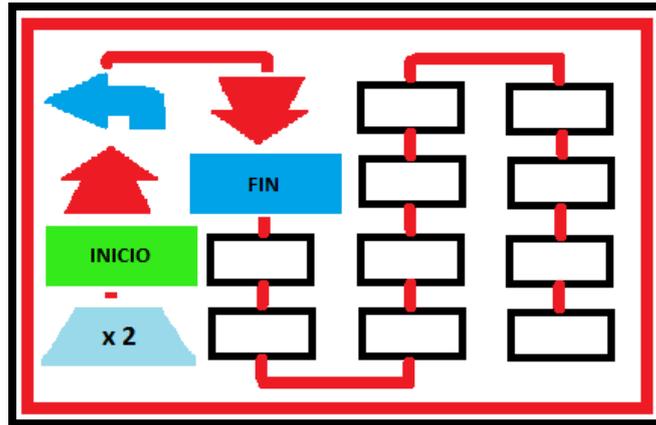


Figura 63. Ejemplo de Estructura de Control Repetitiva
Fuente: El Autor

En la *Figura 63* se puede observar un ejemplo de estructura de control repetitiva la cual ejecuta las instrucciones siguientes: Avance hacia Adelante, Giro a la izquierda, Avance hacia Adelante, este conjunto de instrucciones lo ejecuta 2 veces.

3.2.2.2.3 Estructura de Control Condicional

La estructura de control condicional se ejecuta cuando ocurre algún evento que cumpla con la condición antes no. Este tipo de estructura al igual que las estructuras de control necesita que se le indique donde inicia y donde termina la estructura.

SINTAXIS

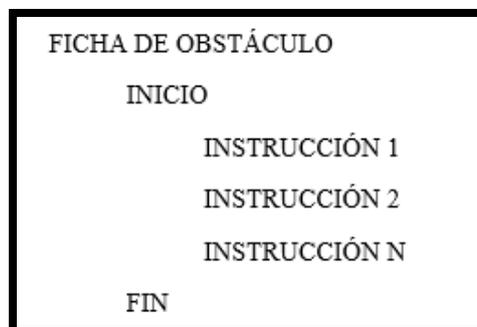


Figura 64. Sintaxis Estructura de Control Condicional
Fuente: El Autor

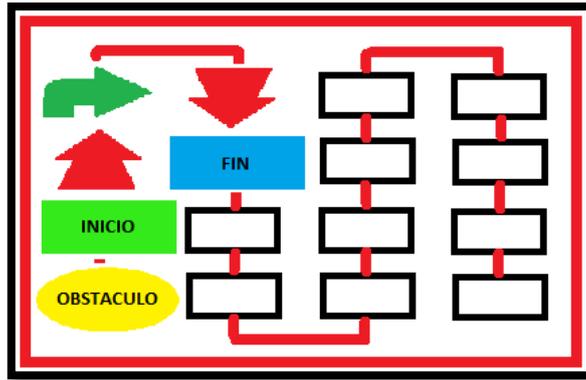


Figura 65 Ejemplo Estructura de Control Condicional
Fuente: El Autor

En el ejemplo de la *Figura 65* se indica un ejemplo de la Estructura de Control Condicional, esta estructura como se explicaba anteriormente únicamente se ejecuta cuando ocurre algún evento especificado, en este caso el evento es encontrar un obstáculo, el robot va avanzar hacia adelante si encuentra el obstáculo ejecuta las instrucciones de lo contrario va seguir avanzando hacia adelante.

3.3 PRIMER PROTOTIPO DE JED-PRO

El primer prototipo de JED-Pro se construyó como un elemento para la realización de pruebas de funcionamiento y poder tener claro los requerimientos finales del sistema tanto de hardware como de software. Este primer prototipo se elaboró utilizando las placas Arduino UNO y Arduino MEGA para el Robot y la Interfaz de Programación respectivamente.



Figura 66. Primer Prototipo de JED-Pro
Fuente: El Autor

En primer instante la interfaz de programación tangible utilizaba porta-fusibles, los cuales permitían sujetar las fichas de instrucciones a la interfaz ya que las fichas estaban elaboradas de fusibles de vidrio modificados, esta técnica funcionó correctamente y resulto muy valedera sin embargo los porta-fusibles tenían puntas metálicas que podrían resultar peligrosas a la integridad de los niños por lo que en el diseño final de JED-Pro los fusibles y porta fusibles fueron remplazados.

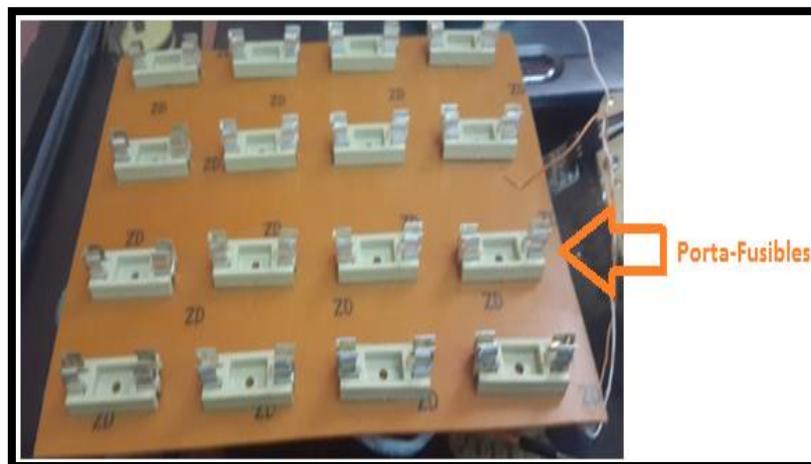


Figura 67. Interfaz de Programación del Primer prototipo de JED-Pro
Fuente: El Autor

Como se puede observar en la *Figura 67*, la primera interfaz de programación de JED-Pro fue elaborada en una baquelita con 16 porta-fusibles que funcionaban como entrada de instrucciones. Esta primera interfaz permitió realizar las primeras pruebas de lectura del valor analógico de las fichas de instrucciones.

La primera prueba se ejecutó realizando la lectura de una ficha de instrucción ubicándola en diferentes entradas de la interfaz, esta prueba fue de gran importancia ya que se pudo notar que en cada entrada de la interfaz existía un desfase del valor analógico debido a las pérdidas que pueden tener los cables conductores o las variaciones de voltaje. Este pequeño inconveniente se corrigió mediante software asignándole un rango de variación a cada intrusión el cual dio buenos resultados.

El primer robot realizado para JED-Pro estaba basado en la placa Arduino UNO, la cual da muchas facilidades para añadir o quitar elementos del dispositivo, este primer prototipo permitió realizar las pruebas de ejecución de instrucciones, cálculo de los giros, velocidad de los motores, distancias de recorrido entre otras.

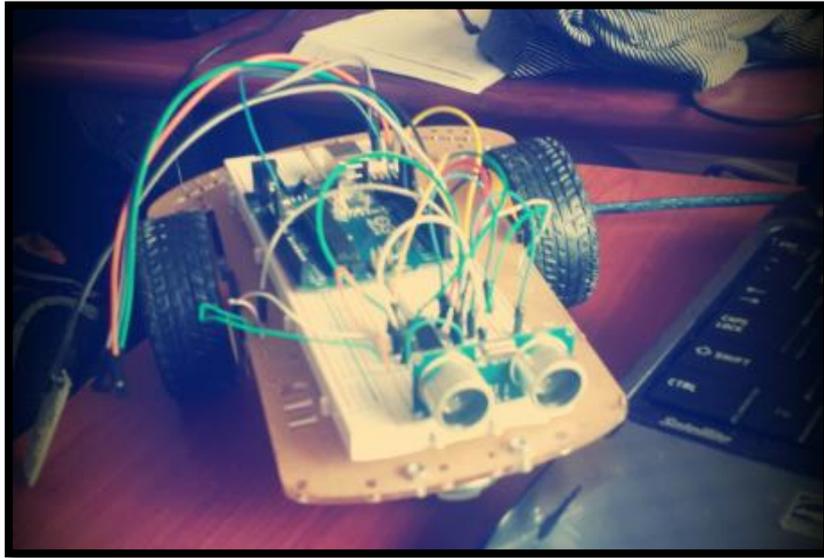


Figura 68. Prototipo de pruebas del Robot JED-Pro
Fuente: El Autor

La *Figura 68* contiene el prototipo de pruebas del robot, el cual se implementó en una placa de pruebas o también llamada Protoboard para facilitar la conexión de los elementos electrónicos y el sensor. El armado en Protoboard permitió comprobar la correcta conexión y funcionamiento de los elementos para luego diseñar el circuito final del robot.

Otro de los elementos de JED-Pro es el bloque de instrucciones. Las fichas de instrucciones fueron elaboradas manualmente mediante la modificación de fusibles de vidrio, la modificación consistió en remplazar el alambre fusible por una resistencia para que permita identificar la instrucción que representa la ficha.

Una vez realizada las pruebas de funcionamiento tanto del software como del hardware se procedió a diseñar el aspecto visual del juguete mediante la elaboración de cubiertas o

carcasas tanto para el robot como la interfaz. Como material principal de las carcasas se utilizó Fomix, este material a parte de darle un aspecto atractivo también resulta ser un material aislante que ayuda a proteger los dispositivos electrónicos.



Figura 69. Diseño del Primer Prototipo de JED-Pro
Fuente: El Autor

3.4 DISEÑO DEL PROTOTIPO FINAL DE JED-PRO

3.4.1 INTERFAZ DE PROGRAMACIÓN

La interfaz de programación consiste en un tablero de forma rectangular en el cual los niños insertan las fichas de instrucciones permitiendo programar una secuencia de hasta 16 instrucciones. La interfaz únicamente debe ser alimentada con una fuente de poder de 5 voltios y va estar lista para que niños y niñas se diviertan programando.

El circuito de control está compuesto por dos circuitos integrados que permiten multiplexar las entradas para obtener las 16 entradas analógicas que requiere el sistema, en lo se refiere a la programación del microcontrolador de la placa Arduino Mini-Pro se utilizan las funciones de conversión analógica digital para la lectura de los valores analógico que arrojen las fichas de instrucciones, también se va a utilizar un pin digital configurado como entrada para el botón de inicio y un pin digital como salida para un led indicador de encendido, además se inicializara la función de comunicación serial a través de la cual se va enviar los datos hacia el modulo bluetooth HC-05 el que estará configurado como Maestro para que el modulo que contiene el robot se pueda vincular y recibir los datos.

A continuación se va indicar los procesos básicos que realiza la interfaz mediante un diagrama de flujo.

3.4.1.1 Diagrama De Flujo

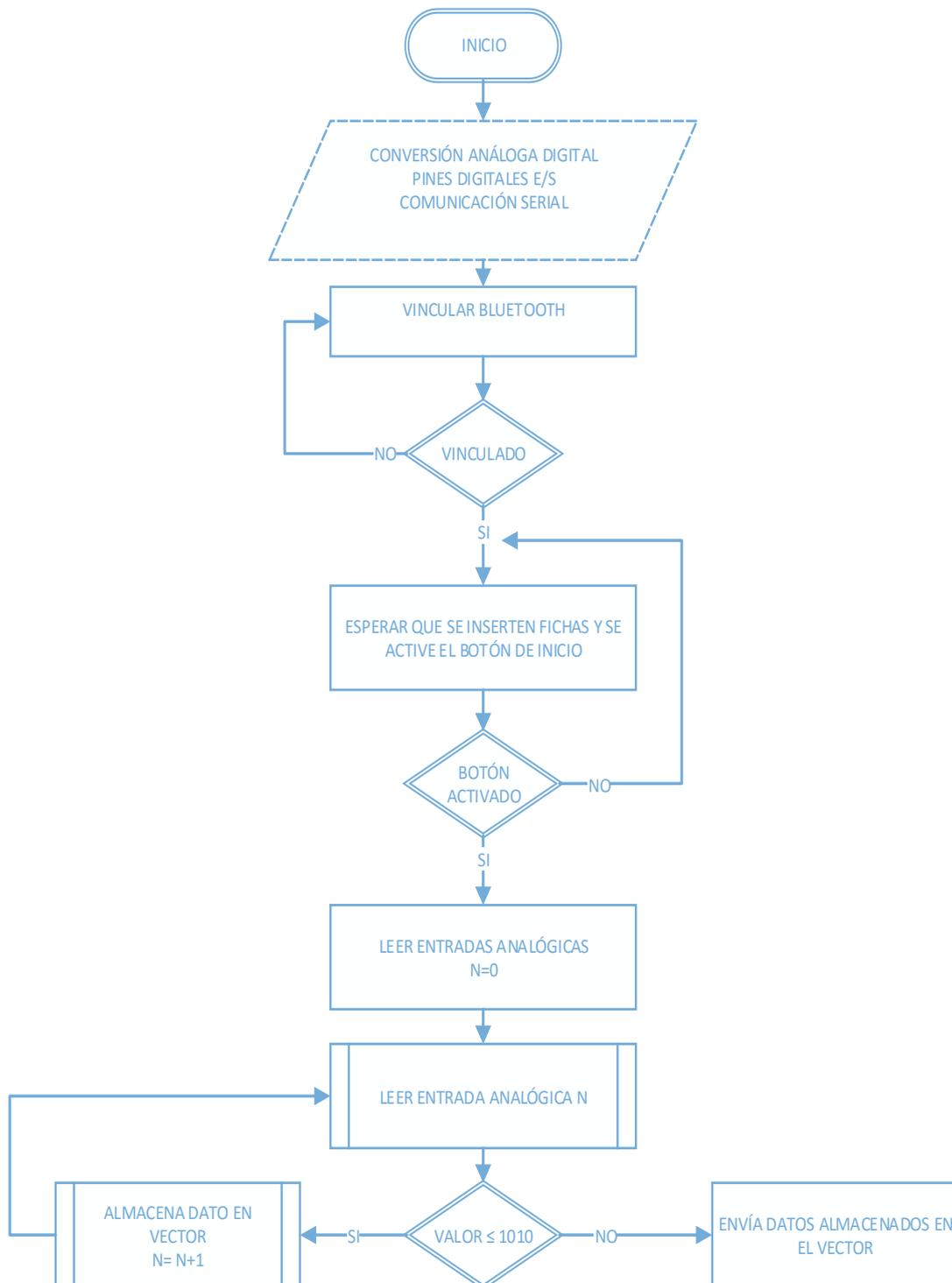


Figura 70. Diagrama de Flujo Interfaz de Programación
Fuente: El Autor

En base a este diagrama se realizó el código de programación utilizando el IDE de Arduino para que la interfaz lea, identifique y envíe los datos al robot. El programa completo se muestra en el *ANEXO 1* el cual está realizado para que sea compatible únicamente con Arduino Mini –Pro, en caso de requerir utilizarlo con otra plataforma o con otro modelo existe la posibilidad de que requiera algunos cambios.

3.4.1.2 Esquema de los circuitos electrónicos

La *Figura 72* muestra el esquema de cómo se debe conectar los diferentes elementos de la interfaz de programación, el cual está compuesto por la placa Arduino Mini-Pro, dos circuitos integrados 74HC4051 que realizan la multiplexación y el módulo Arduino.

En la *Figura 73* se indica un circuito basado en divisores resistivos los cuales facilitan la lectura analógica ya que para cualquier microcontrolador es más fácil leer voltaje que leer la resistencia. Lo que hace esta pequeña configuración de resistencias es repartir la tensión de la fuente de voltaje entre una o más impedancias conectadas.

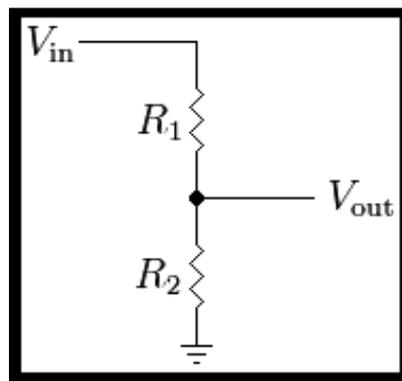


Figura 71. Divisor de Voltaje Resistivo
Fuente: <http://5hertz.com/tutoriales/?p=91>

$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2}$$

Ecuación 4. Calculo del voltaje de salida utilizando divisor resistivo
Fuente: <http://5hertz.com/tutoriales/?p=91>

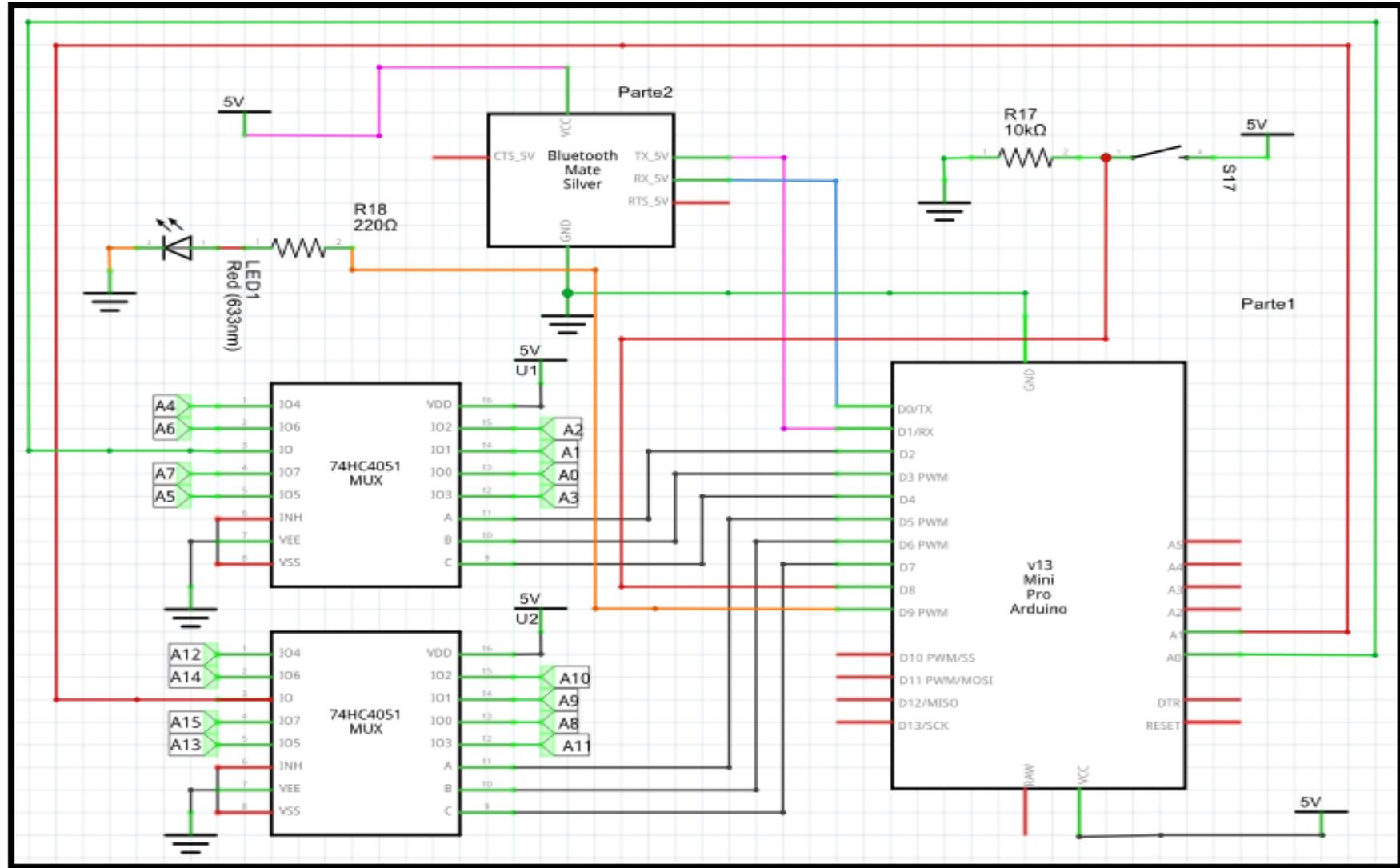


Figura 72. Circuito Principal
Fuente: El Autor

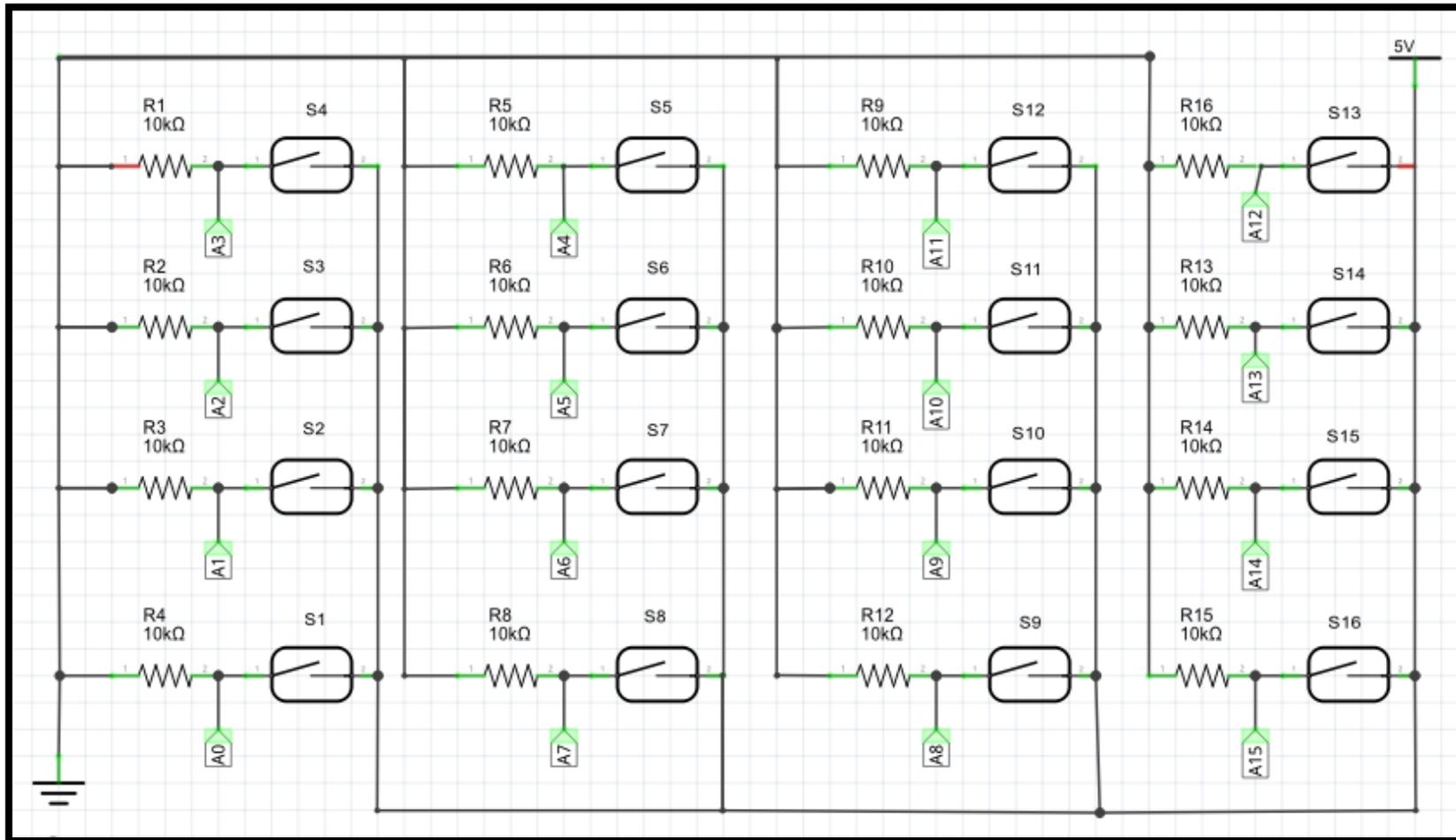


Figura 73. Circuito de Entrada
Fuente: El Auto

3.4.1.3 Simulación

La simulación fue realizada en el software Proteus el cual tiene un porcentaje de error muy bajo en comparación a otros sistemas y tiene una buena aproximación a la realidad. Mediante esta simulación se pudo comprobar el funcionamiento del programa y además permitió realizar la corrección de errores directamente, esto es una gran ventaja de los programas de simulación ya que se ahorra tiempo y recursos que se requiere para armar físicamente el circuito para realizar pruebas.

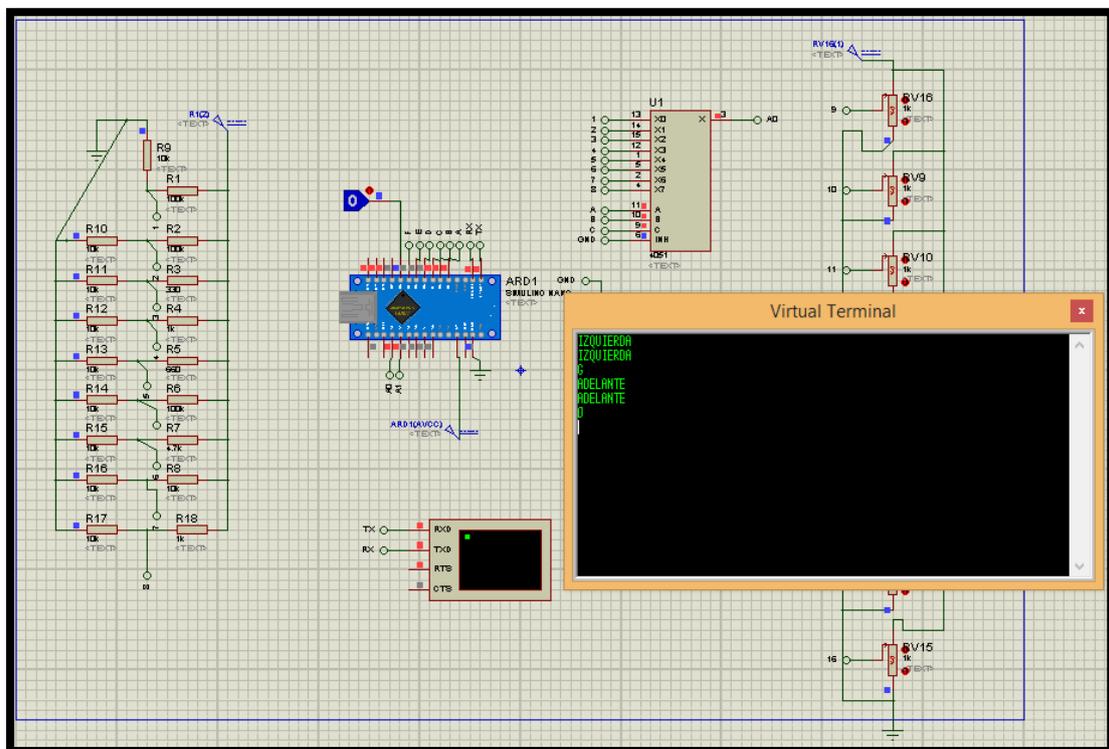


Figura 74. Simulación en el Software Proteus
Fuente: El Autor

En el esquema de la *Figura 74* se comprobó que el circuito de multiplexación funciona correctamente, además permitió analizar los valores de las resistencias óptimos para el sistema mediante el cambio de los valores y por último se comprobó que los datos transmitidos correctamente mediante un terminal virtual de visualización.

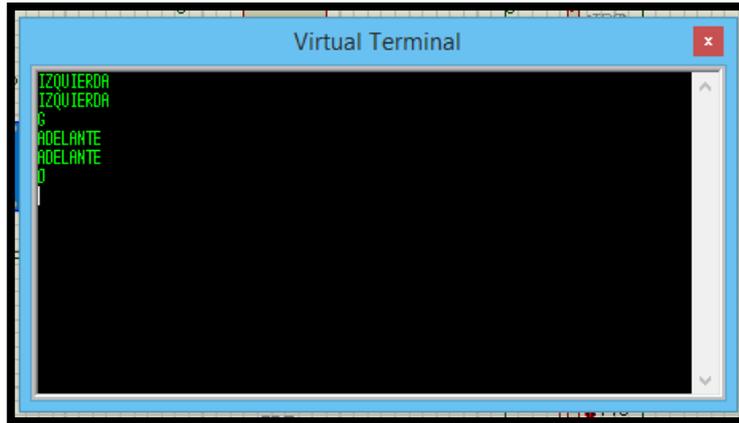


Figura 75. Terminal Virtual de Proteus
Fuente: El autor

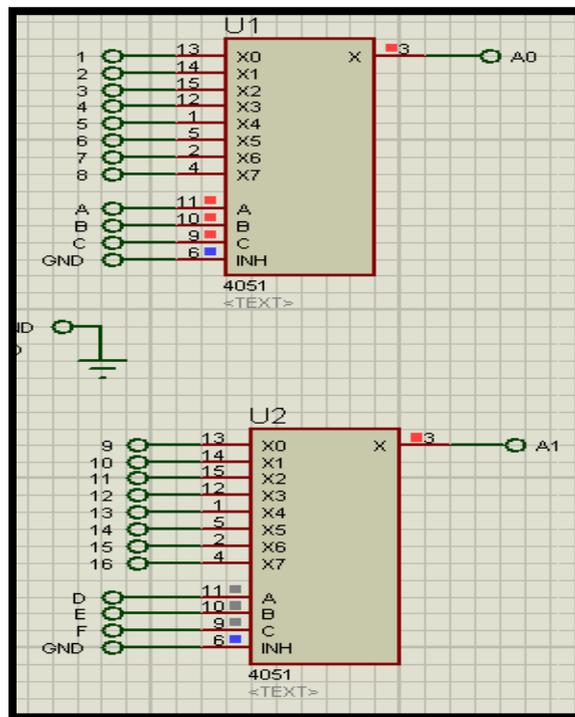


Figura 76. Circuito de Multiplexación
Fuente: El Autor

3.4.1.4 Diseño del circuito Impreso

Una vez comprobado el circuito y el funcionamiento del programa mediante la simulación se procede a crear el esquema para el circuito impreso con la ayuda del programa EAGLE en su versión 6.5.0. Lo primero que se debe realizar es descargar las librerías de la placa Arduino Mini-Pro, estas se pueden obtener directamente de la página oficial de Arduino en el siguiente link: <http://www.arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardProMini>

Una vez descargadas se deben colocar en la carpeta de librerías del programa Eagle y estamos listo para empezar a realizar el circuito.

Para empezar se debe seleccionar los elementos del circuito e interconectarlos como se muestra en la *Figura 77*. Es importante verificar que los dispositivos estén correctamente conectados en este esquema para no tener errores en el funcionamiento de la baquelita ya que de este depende la creación de pistas.

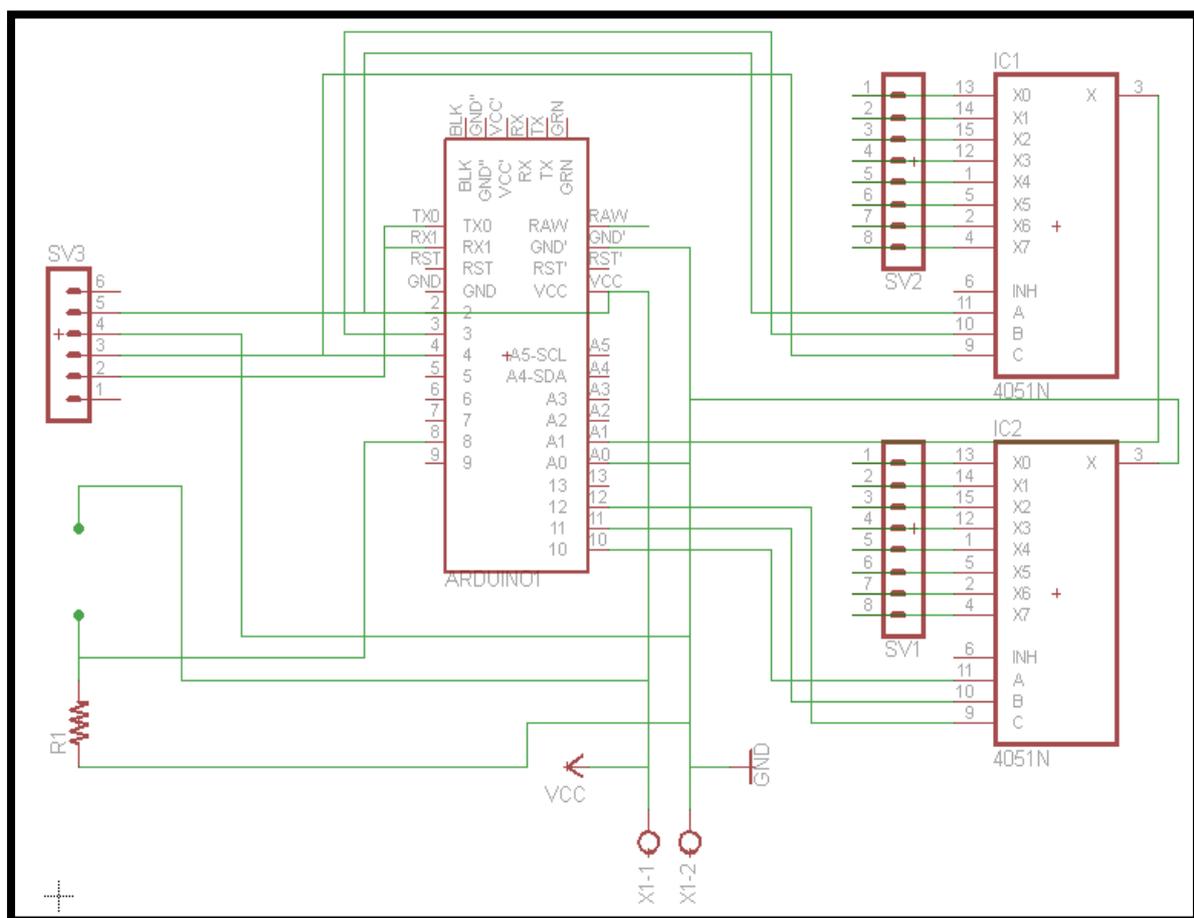


Figura 77. Esquema del circuito de la interfaz de programación de JED-Pro realizado en Eagle 6.5
Fuente: El Autor

Después de elaborar el esquema y de verificar que estén correctamente conectados los dispositivos se procede a generar el esquema real del circuito, el cual se debe ordenar de la mejor manera tratando de optimizar el tamaño de la placa y tener cuidado que las pistas no se junten entre sí para no tener problemas de cortos o incorrecto funcionamiento del circuito.

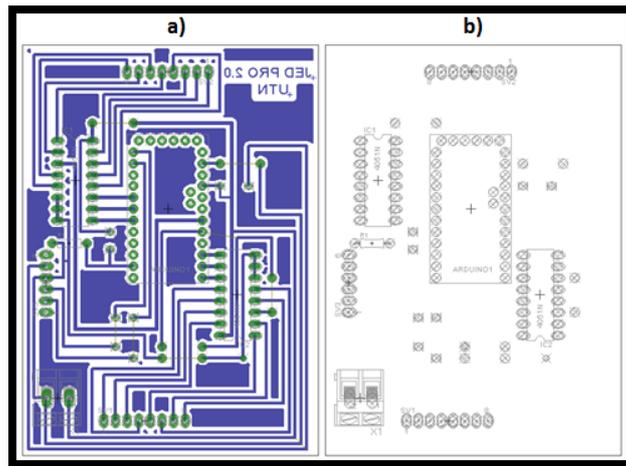


Figura 78. a) Diseño del circuito impreso b) Ubicación de dispositivos y elementos electrónicos
Fuente: El Autor

Una vez se obtuvo el diseño final del circuito se procedió a imprimirlo en una hoja de papel termo-transferible la cual nos permite transferir el circuito a la baquelita mediante la inducción de calor proporcionada por una plancha eléctrica. Es importante mencionar que la impresión debe realizarse a través de una impresora láser para que tanto el proceso de transferencia del circuito como el quemado de la placa hagan efecto.

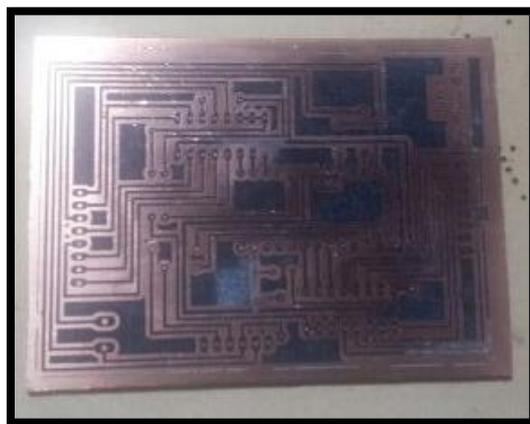


Figura 79. Circuito Transferido a la Baquelita
Fuente: El Autor

Después de realizar la transferencia del circuito impreso a la baquelita se procede a quitar el cobre que no nos sirve, este proceso se le conoce como quemado de la baquelita para el cual se requiere ácido cloruro férrico, este ácido va a retirar todo el cobre que no esté recubierto por la tinta para poder obtener únicamente las pistas que se requieren.

Para el proceso de quemado se debe colocar el ácido en un recipiente plástico o de vidrio, luego se vierte agua tibia con mucho cuidado hasta cubrir completamente la placa, en este momento se va observar como el ácido hace efecto sobre el cobre de la baquelita pero si se quiere acelerar un poco el proceso se puede agitar un poco el recipiente para que se quemara con mayor rapidez.

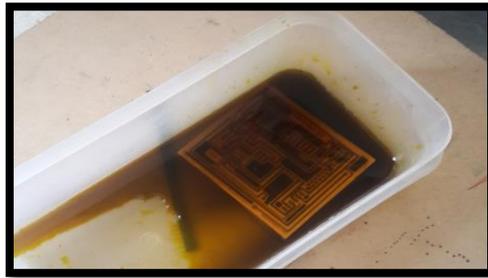


Figura 80. Proceso de Quemado de la Baquelita 1
Fuente: El Autor

Cuando se observe que el cobre ha sido retirado completamente, sacamos la placa y limpiamos cuidadosamente con abundante agua y la ayuda de un lustre fino la tinta que nos queda en la placa y se verifica que no exista cobre uniendo las pistas y tenemos la placa terminada.

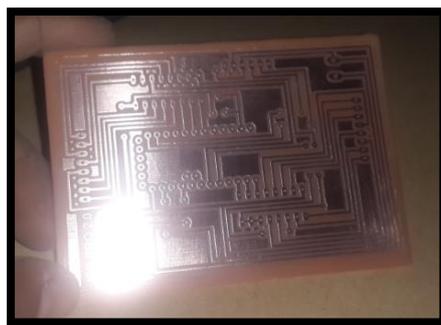


Figura 81. Baquelita de la Interfaz Finalizada
Fuente: El Autor

3.4.1.5 Implementación del circuito

Una vez finalizada la elaboración de la baquelita se procede a implementar el circuito ubicando los elementos electrónicos requeridos en la baquelita, para lo cual se requiere primeramente perforar la baquelita cuidadosamente con un taladro que acepte brocas pequeñas.

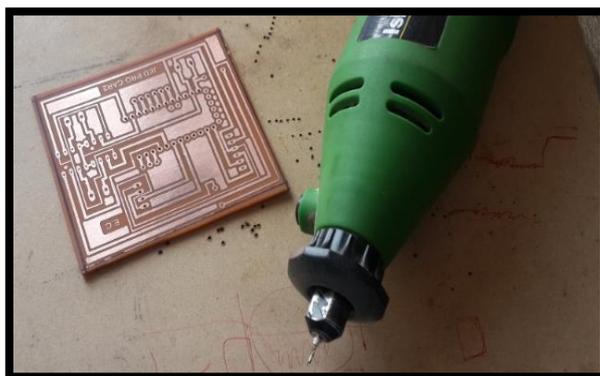


Figura 82. Perforación de la baquelita
Fuente: El Autor

Terminada la perforación se debe ubicar cada uno de los elementos donde le corresponde y se realiza el soldado tomando las precauciones para no hacerse daño y para que el circuito funcione correctamente, la ubicación de los elementos está basado en la *Figura 78*, cabe mencionar que hay que tener cuidado con la polaridad de los elementos electrónicos ya que una equivocación podría dañar el circuito y los elementos en sí.

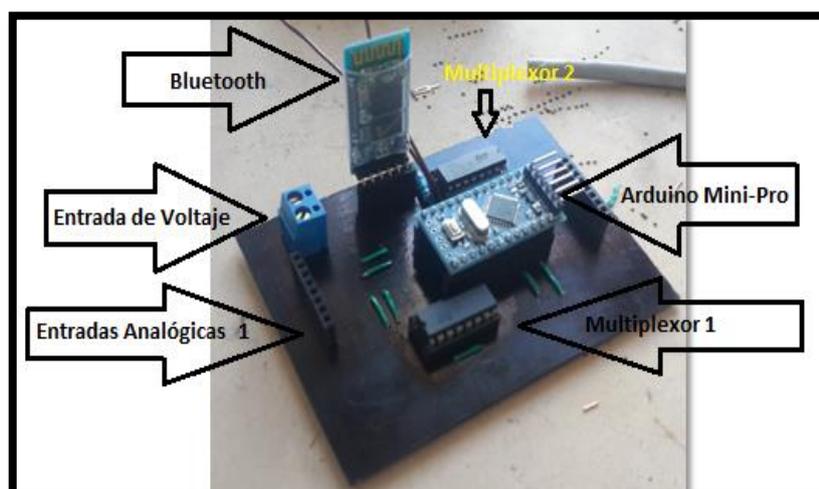


Figura 83. Circuito en baquelita terminado
Fuente: El Autor

3.4.1.6 Configuración del Módulo Bluetooth como Maestro

El modulo bluetooth HC-05 tiene la propiedad de que se puede configurar como maestro o esclavo, en este caso para la interfaz de programación de JED-Pro se va a configurar el modulo como maestro ya que esta podrá conectarse al módulo del Robot que va estar configurado como esclavo.

Para realizar la configuración del módulo bluetooth se necesita tener instalado un terminal virtual y un cable convertidor de TTL a USB que permita comunicar el módulo con nuestro computador.

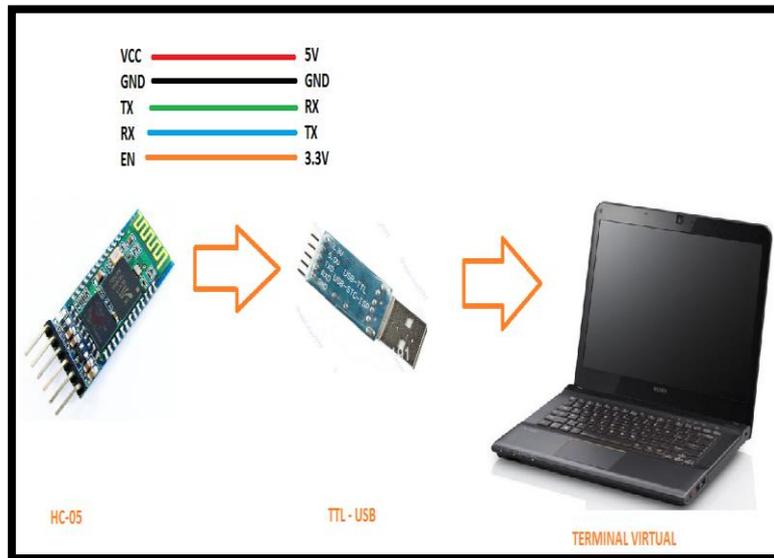


Figura 84. Diagrama de conexión para configurar el modulo bluetooth HC-05
Fuente: El Autor

La *Figura 84* indica cómo se debe conectar el modulo al convertidor TTL- USB para poder tener comunicación y que se pueda configurar. El modulo bluetooth HC-05 se debe configurar mediante comandos AT, estos comandos son una especie de lenguaje de programación mediante el cual se asignan las configuraciones y pueden encontrarse en la hoja de datos del módulo.

3.4.1.6.1 Comandos AT para configurar como Maestro

La *Tabla 24* indica los comandos necesarios para configurar el modulo como maestro, cabe mencionar que también se configuro el modulo para que se conecte a un dispositivo esclavo especifico mediante la dirección MAC, esto es para evitar que se conecte con algunos dispositivos que pueden encontrarse dentro del radio de alcance equivocadamente.

Tabla 24. Comandos AT módulo HC-05

COMANDO	ACCIÓN	EJEMPLO
AT	Verifica conectividad. Si responde "OK" es porque si tenemos conexión	AT OK
AT+NAME= "Nombre del Dispositivo"	Permite asignar un nombre al modulo	AT+NAME=Interfaz-JEDPro
AT+PSWD= "clave de 4 dígitos"	Asigna contraseña	AT+NAME=1111
AT+ROLE= "1"	Configura como Maestro	AT+ROLE=1
AT+CMODE= "0"	Indica que el modulo va conectarse a una dirección especifica	AT+CMODE= 0
AT+BIND="Dirección MAC"	Indica la dirección especifica del esclavo, el formato es: AAAA,AA,AAAAAA	AT+BIND=3014,11,172448
AT+UART= "Velocidad de TX"	Configura la velocidad de transmisión en baudios	AT+UART= 9600,0,0

Nota: Adaptada de la hoja de datos del módulo HC-05 Fuente: El Autor

3.4.1.6.2 Pasos para configurar como Maestro

- 1) Conectar el modulo bluetooth al computador como se especifica en la *Figura 84*. Y Verificar en que puerto está conectado el modulo, para eso se ingresa al administrador de dispositivos.

- 2) Ingresar al Terminal virtual. En este caso se utilizó HyperTerminal y se crea una nueva conexión.



Figura 85. Nueva Conexión
Fuente: El Autor

- 3) Se selecciona el puerto COM al que está conectado el modulo, en este caso es el COM 8 y se da clic en aceptar.



Figura 86. Selección del Puerto COM
Fuente: El Autor

- 4) Configurar las propiedades como se muestra en la *Figura 87* y dar clic en aceptar

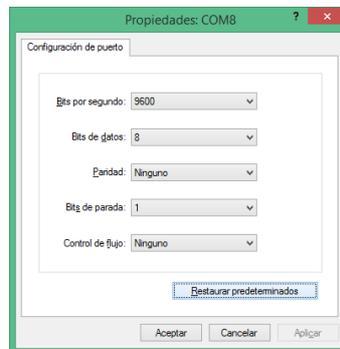


Figura 87. Configuración del Puerto COM
Fuente: El Autor

- 5) Después se procede a habilitar algunas características del HyperTerminal para no tener problemas en el momento de ingresar los comandos de configuración, para esto se accede a Archivo/Propiedades/Configuración/ Configuración ASCII y se habilita las casillas como indica la *Figura 88*.

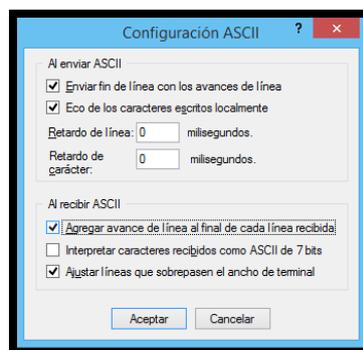


Figura 88. Configuración de HyperTerminal
Fuente: El Autor

- 6) Luego de configurar el Hyperterminal, se procede a configurar el modulo bluetooth haciendo uso de los comando indicados en la *Tabla 24*.

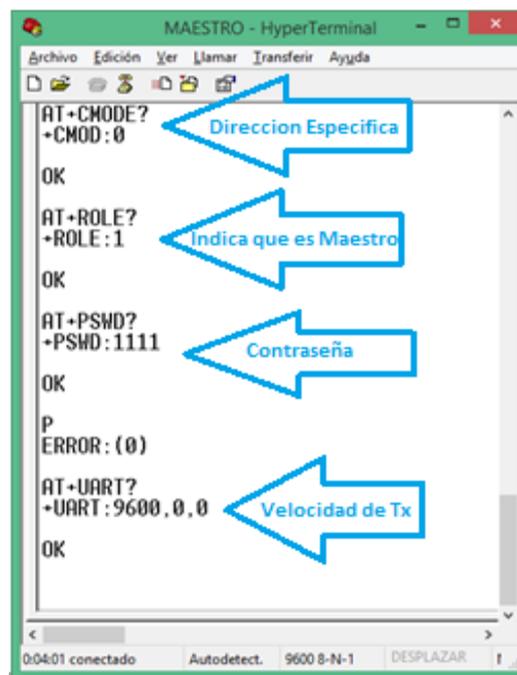


Figura 89. Verificación de la Configuración en HyperTerminal
Fuente: El Autor

3.4.1.7 Construcción de la Interfaz

Después de realizar las pruebas de funcionamiento de la interfaz y de haber obtenido una aceptación por parte de los niños del diseño de las carcasas vistas en el primer prototipo de JED-Pro, en el segundo prototipo se mejora la resistencia y durabilidad de la interfaz eliminando la carcasa de fomix y creando una carcasa de madera la cual es mucho más resistente, otro de los cambios realizados fue la incorporación de imanes para ingresar las fichas de instrucciones, esto se hizo para eliminar los portafusiles los cuales tenían bordes que podían lastimar a los niños y de esta forma poder mejorar la seguridad.

Primeramente se construyó el circuito de la *Figura 73* el cual sirve para poder leer el valor de la instrucción ingresada. Ahora en el segundo prototipo con el afán de eliminar los porta-fusibles que podían causar daño a los niños se mejoró el circuito haciendo uso de imanes

para que mediante el principio de magnetismo las fichas se mantengan fijas al tablero y así poder leer su valor.

Después de obtener el circuito de entrada se elaboró el tablero, el cual tiene un tamaño aproximado de 200mm x 150mm, para elegir este tamaño se tomó como referencia el tamaño de las manos de un niño de 5 años con el propósito de obtener un tamaño acorde a los usuarios que van a ser precisamente niños.



Figura 90. Referencia para el tamaño de la interfaz de programación
Fuente: El Autor

En la *Figura 91* se indica el proceso de construcción del tablero que va ser la interfaz de programación. Toda su elaboración fue realizada a mano, tratando de obtener el mejor resultado posible y cumpla con todas las características del prototipo.

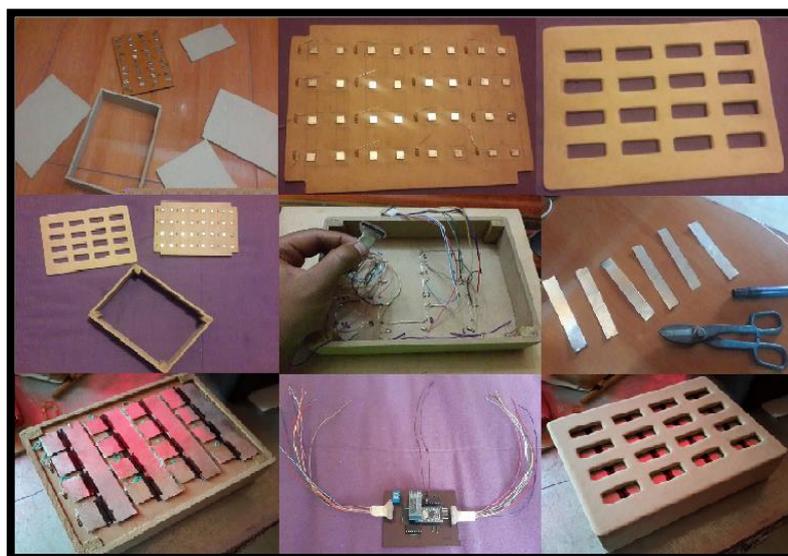


Figura 91. Proceso de construcción de la interfaz de programación
Fuente: El Autor



Figura 92. Diseño Final del Prototipo de la Interfaz de programación
Fuente: El Autor

3.4.2 ROBOT JED-PRO

El robot de JED-Pro es el elemento que ejecuta las instrucciones programadas por los niños en tiempo real. Este robot está compuesto básicamente por un kit chasis, el cual está conformado por dos motores DC, una base plástica y una rueda omnidireccional; un sensor ultrasónico y el circuito de control basado en Arduino.

Para el diseño del robot se acogieron las características del primer prototipo debido a la gran aceptación que tuvo su apariencia; en lo único que cambia en relación al primer prototipo es en el circuito de control, el cual utiliza una plataforma Arduino Mini-Pro en lugar del Arduino uno, este cambio se realizó únicamente con el afán de optimizar el costo del prototipo ya que las dos placas cuentan con características similares.

El circuito de control del robot está basado en la placa Arduino Mini-Pro como se explicaba anteriormente y cuenta con los elementos electrónicos necesarios para realizar la conexión de los motores, el modulo bluetooth mediante el cual recibe las instrucciones y el sensor ultrasónico el que es utilizado como detector de obstáculos.

3.4.2.1 Diagrama de flujo

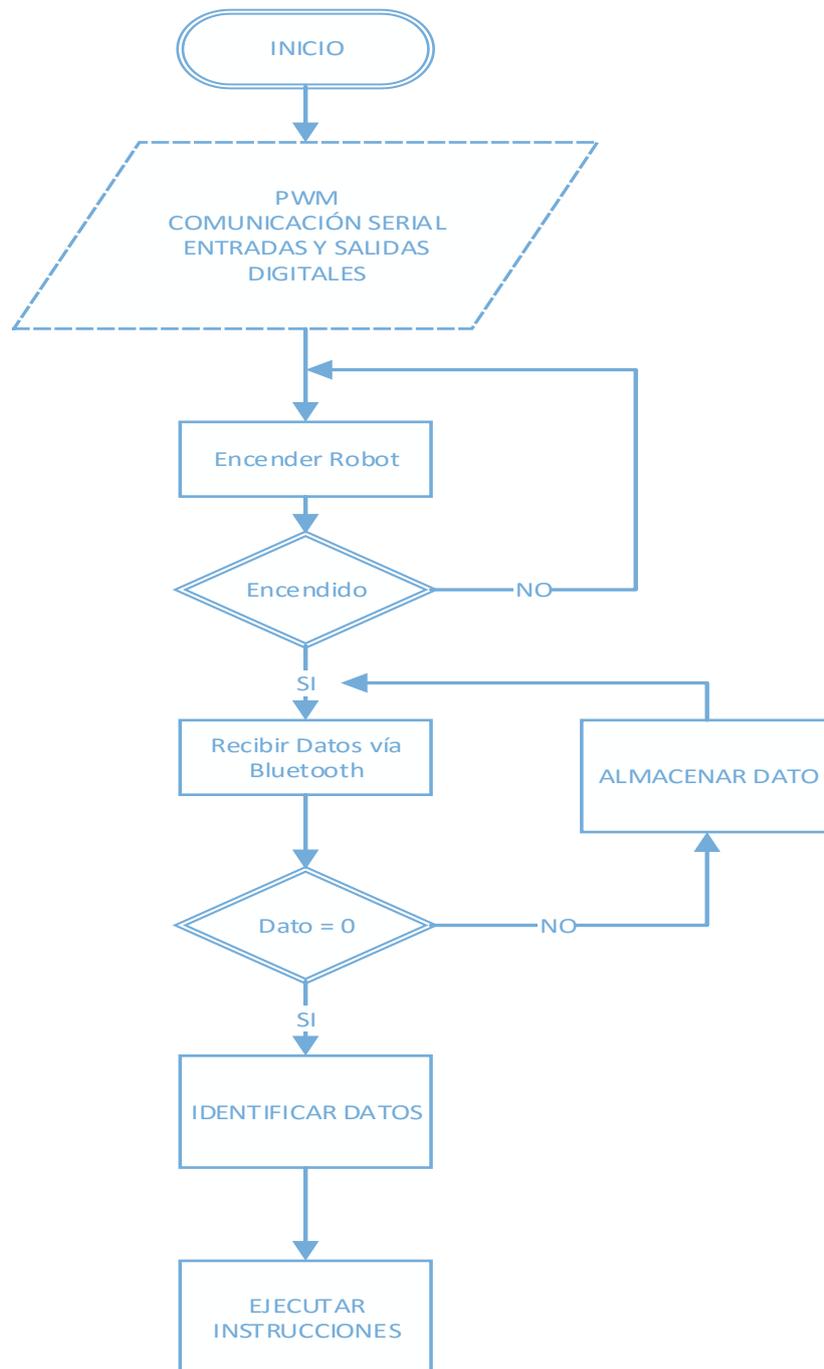
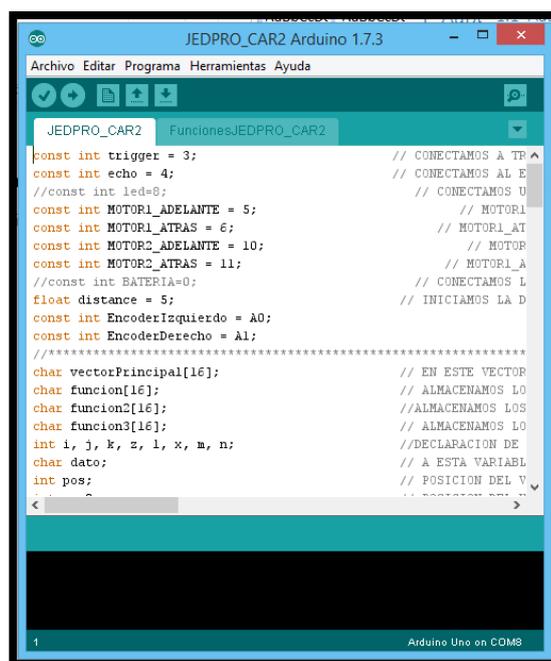


Figura 93. Diagrama de Flujo del Robot JED-Pro
Fuente: El Autor

De igual forma que en el diseño de la interfaz de programación tangible aquí también se elaboró un diagrama de flujo que sirve como guía para el desarrollo del programa utilizado por el robot.

El programa del robot es muy importante en el funcionamiento del JED-Pro ya que se encarga primeramente de recibir los datos enviados desde la interfaz, luego identifica los datos recibidos y los va clasificando en diferentes vectores, este proceso de clasificación de los datos se realiza para identificar cuáles son instrucciones simples y cuales son instrucciones de control. Cuando el programa recibe un dato identificador de fin envía las señales a través de sus puertos de salida para que el robot ejecute las instrucciones.

Al igual que en la interfaz el programa del robot fue realizado con el IDE de Arduino y está dividido en dos partes, la primera parte del programa corresponde a los procesos de identificación y clasificación de instrucciones y la segunda parte corresponde a los procesos de ejecución. El programa completo se muestra en el ANEXO 2.



```
const int trigger = 3; // CONECTAMOS A TR
const int echo = 4; // CONECTAMOS AL E
//const int led=8; // CONECTAMOS U
const int MOTOR1_ADELANTE = 5; // MOTOR1
const int MOTOR1_ATRAS = 6; // MOTOR1_AT
const int MOTOR2_ADELANTE = 10; // MOTOR
const int MOTOR2_ATRAS = 11; // MOTOR_A
//const int BATERIA=0; // CONECTAMOS L
float distance = 5; // INICIAMOS LA D
const int EncoderIzquierdo = A0;
const int EncoderDerecho = A1;
//*****
char vectorPrincipal[16]; // EN ESTE VECTOR
char funcion[16]; // ALMACENAMOS LO
char funcion2[16]; //ALMACENAMOS LOS
char funcion3[16]; // ALMACENAMOS LO
int i, j, k, z, l, x, m, n; //DECLARACION DE
char dato; // A ESTA VARIABL
int pos; // POSICION DEL V
```

Figura 94. Programación del Robot desarrollado en el IDE de Arduino
Fuente: El Autor

3.4.2.2 Esquema del circuito electrónico

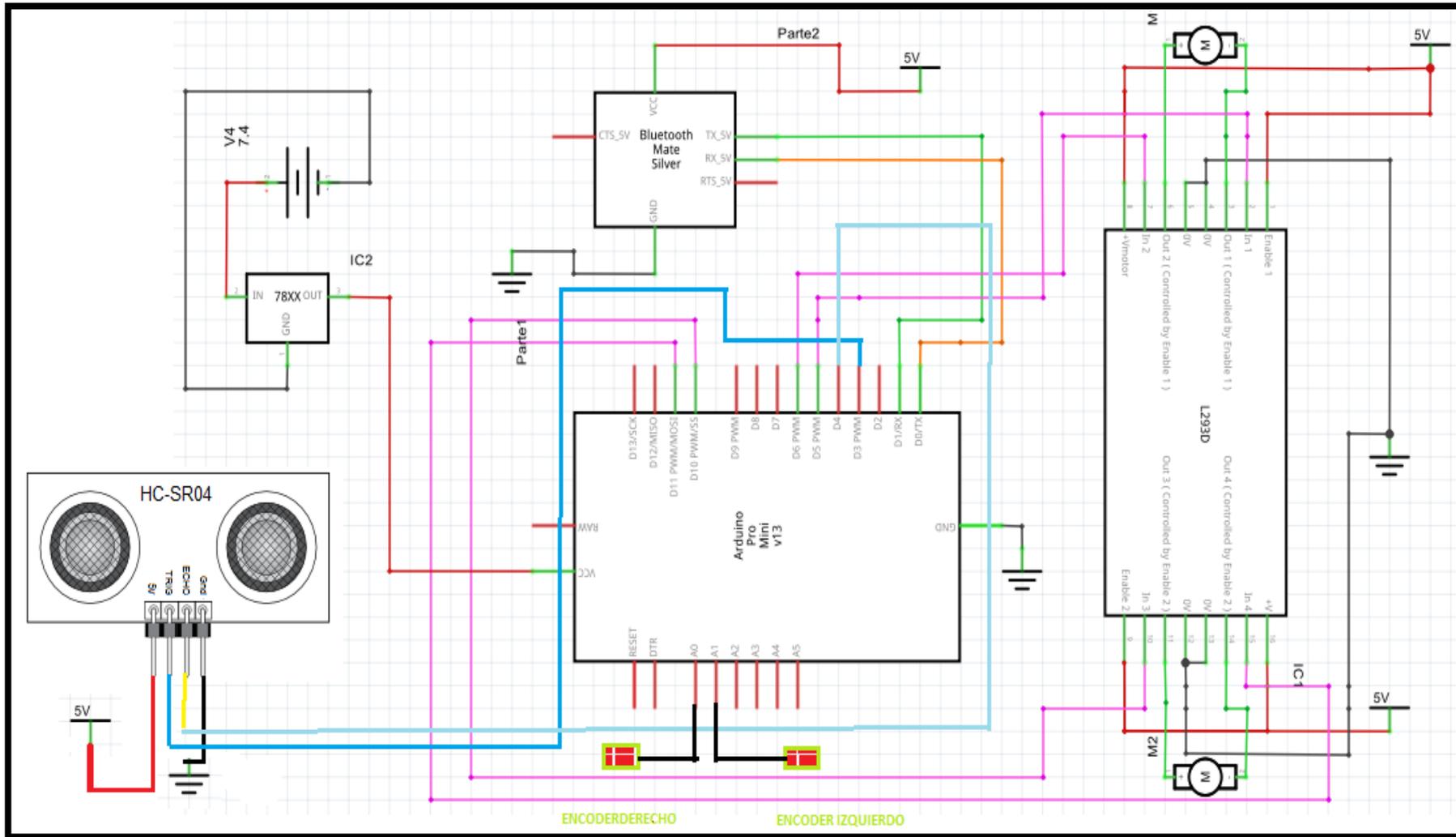


Figura 95. Circuito Electrónico Robot JED-Pro
Fuente: El Autor

La *Figura 95* muestra el circuito electrónico del robot, el cual va ser alimentado con una batería de Ion-Litio de 7.4v, pero debido a que los elementos electrónicos trabajan a 5v se incorporó un circuito integrado 7805 el cual permite regular y obtener el voltaje necesario.

3.4.2.3 Simulación

La simulación permitió comprobar tanto el funcionamiento del circuito como la correcta ejecución del programa. Además permitió realizar la corrección de errores de ejecución al dejar utilizar un terminal virtual para enviar datos hacia la placa Arduino para verificar su almacenamiento e identificación de las instrucciones.

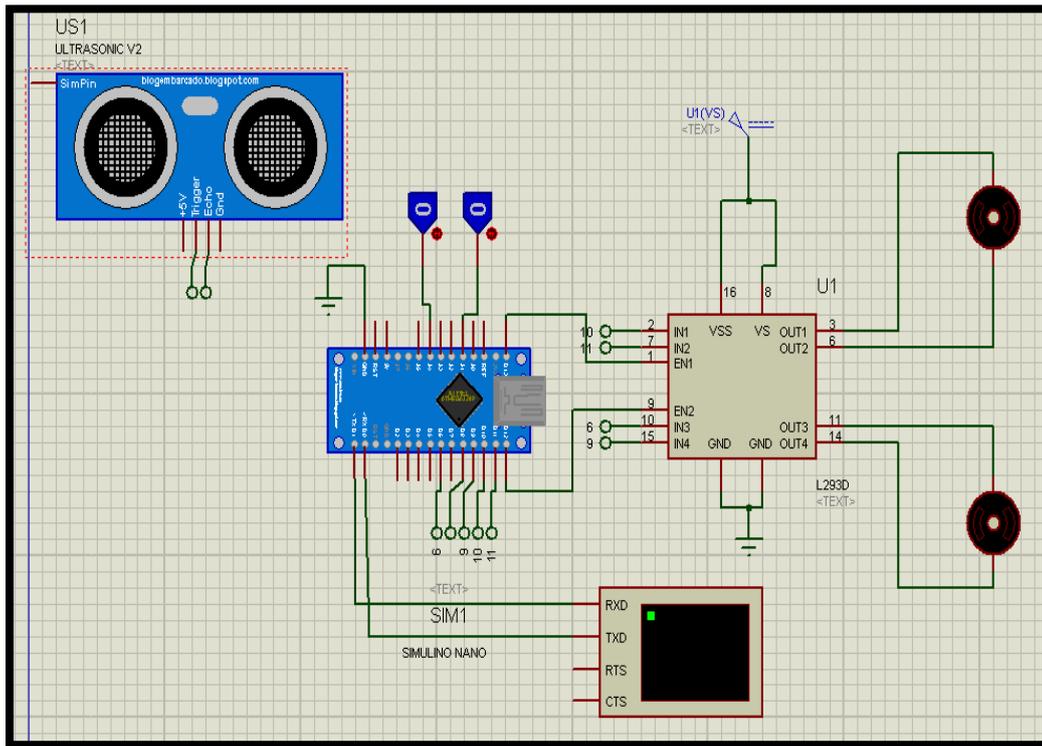


Figura 96. Simulación del Robot JED Pro
Fuente: El Autor

3.4.2.4 Diseño del circuito Impreso

Una vez comprobado el funcionamiento del circuito y del programa mediante la simulación se procedió a diseñar el esquema para el circuito impreso siguiendo el mismo procedimiento efectuado en el diseño de la interfaz de programación tangible.

Primeramente se implementó el esquema del circuito, el cual se muestra en la *Figura 97*. Este esquema se debe realizar tomando muy en cuenta la conexión de los dispositivos y su polarización ya que un error en la conexión podría causar el mal funcionamiento del robot.

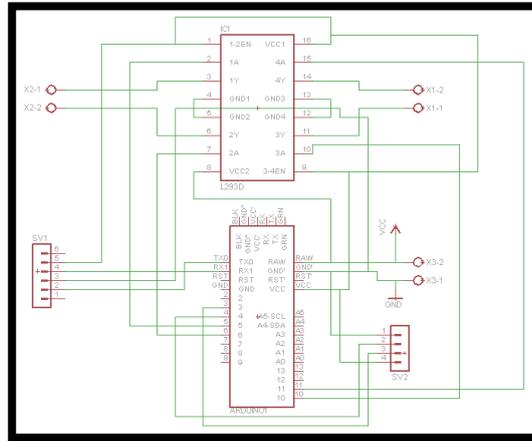


Figura 97. Esquema para el circuito impreso en Eagle 6.5
Fuente: El Autor

Una vez creado el esquema y después de verificar que las conexiones sean las correctas se procedió a generar el esquema real de la placa. La cual se realizó manualmente para un mejor enrutado de las pistas.

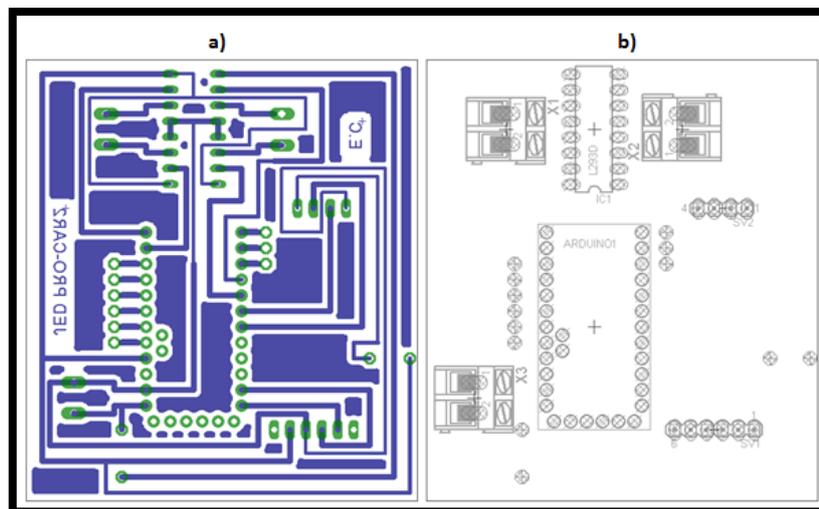


Figura 98. a) Vista Posterior de la placa b) Ubicación de elementos
Fuente: El Autor

Una vez que se obtuvo el diseño final del circuito se ejecutó el mismo proceso realizado anteriormente en la placa de la interfaz, el cual primeramente es la impresión en el papel termo-transferible para mediante el planchado trasladar el circuito a la baquelita.

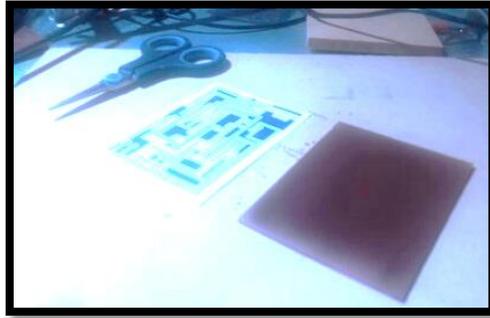


Figura 99. Proceso para la transferencia del circuito impreso a la baquelita
Fuente: El Autor

Después de imprimir se procede a recortar el circuito como se puede ver en la *Figura 99* y realizamos el proceso de transferencia del circuito.

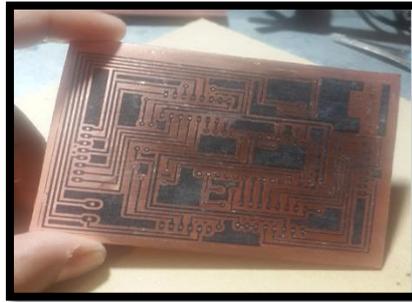


Figura 100. Circuito Impreso transferido a la baquelita
Fuente: El Autor

Luego se procedió con el quemado de la baquelita haciendo uso de un recipiente plástico tomando en cuenta todos los cuidados que conlleva el trabajo con ácido.

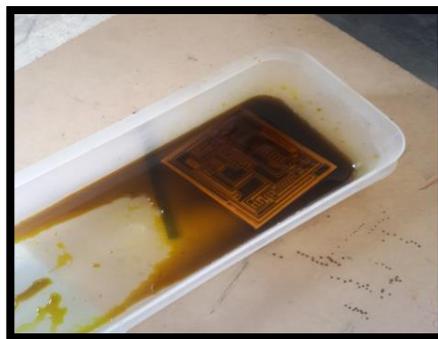


Figura 101. Proceso de quemado de la baquelita
Fuente: El Autor

Después de algunos minutos de someter a la baquelita al ácido se pudo obtener las pistas necesarias para el circuito

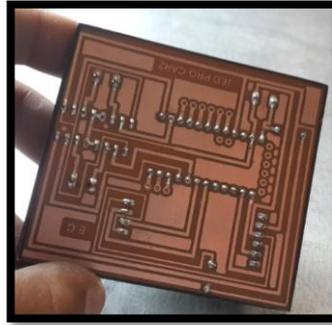


Figura 102. Baquelita terminada
Fuente: El Autor

3.4.2.5 Implementación del circuito

Para la implementación del circuito primeramente se realizó la perforación de la baquelita y luego se procedió al soldado de cada uno de los elementos. La *Figura 98* indica la ubicación en la baquelita de los elementos, pero se debe tomar muy en cuenta la polaridad de estos antes de realizar las soldaduras.

El Robot cuenta con los sensores encoder y el circuito regulador de voltaje que son externos al circuito principal, por lo que se necesitó cables conductores flexibles para unir los circuitos, además se utilizó un material termo-contraíble, el que es un aislante que se coloca alrededor de los cables y luego se somete a calor para que este se encoja, esto se realiza para darle mayor robustez a los cables y además permite proteger los cables sueltos y evitar problemas como corto circuitos o que los cables se rompan.

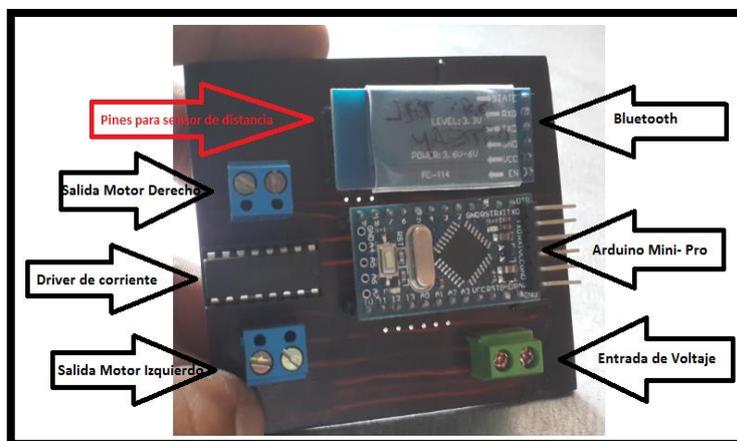


Figura 103. Circuito Principal del Robot implementado
Fuente: El Autor

3.4.2.6 Configuración del Módulo Bluetooth como esclavo

En el caso del Robot se configuró el módulo bluetooth como esclavo para que pueda conectarse a la interfaz que como se explicó anteriormente está configurado como maestro. El proceso de conexión del módulo es el mismo que el explicado anteriormente en la configuración del módulo de la interfaz, únicamente se realizó cambios en la configuración.

Es importante recordar que el modulo esclavo se debe configurar con la misma contraseña utilizada por el modulo maestro y además deben tener la misma velocidad de transmisión de datos para que estos puedan vincularse e intercambiar datos sin problemas. A continuación se detalla los pasos de la configuración del módulo esclavo.

1. Configuración del nombre del dispositivos, en este caso se le puso el nombre

“ROBOT-JEDPRO”

Comando AT: AT+NAME=ROBOT-JEDPRO

2. Configuración de la contraseña

Comando AT: AT+PSWD=1111

3. Configuración de la velocidad de transmisión

Comando AT: AT+UART=9600,0,0

4. Configuración como esclavo

Comando AT: AT+ROLE=0

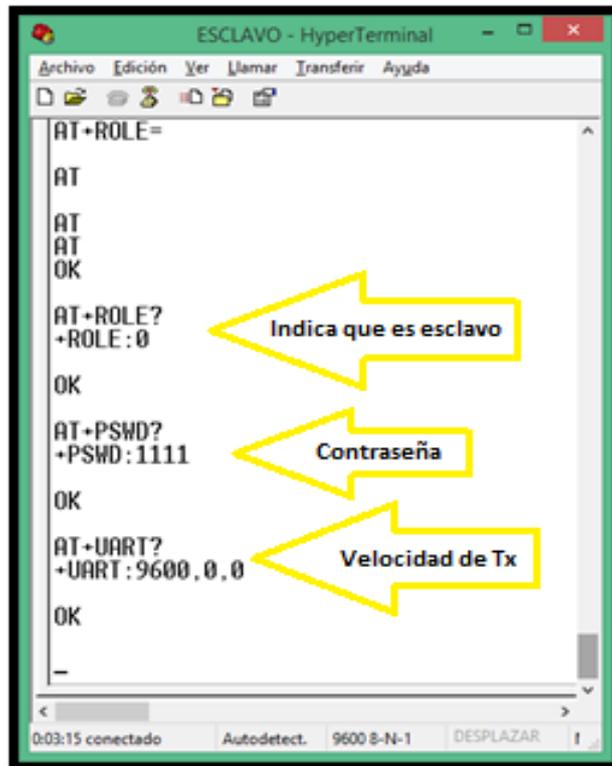


Figura 104. Verificación de las configuraciones
Fuente: El Autor

3.4.2.7 Construcción del Robot

El primer procedimiento a realizar para la construcción del prototipo final del robot fue armar el chasis, el cual puede armarse de acuerdo a las necesidades del Robot. Los elementos que conforman el chasis se detallan a continuación en la *Figura 105*.

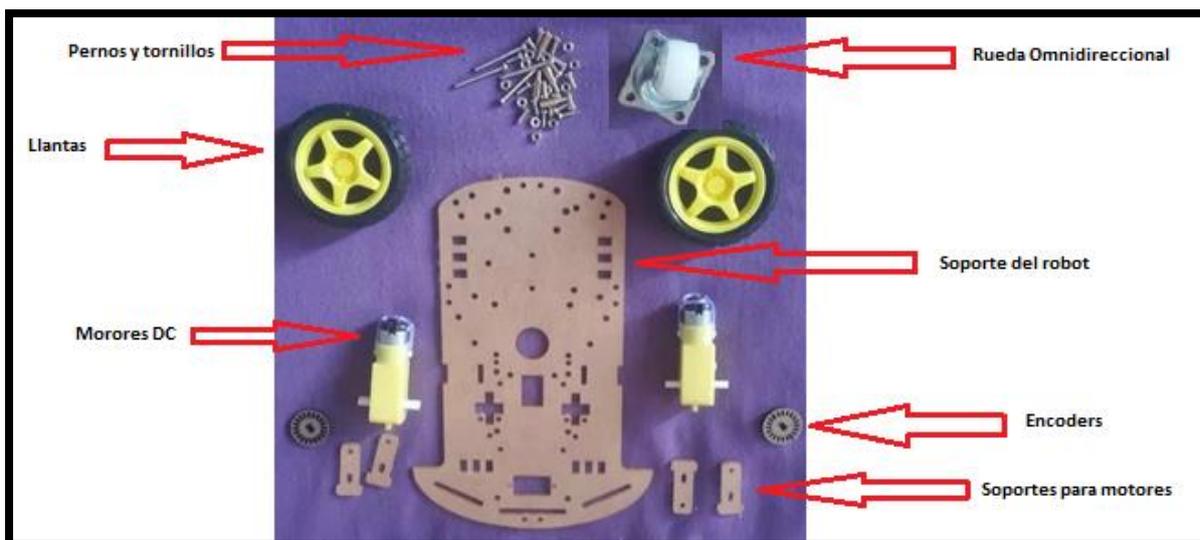


Figura 105. Elementos del Chasis del robot
Fuente: El Autor

Una vez se tienen todos los elementos que conforman el chasis se procede a realizar el armado, el cual se realizó todo manualmente tratando de que la piezas encajen perfectamente y que queden correctamente ajustadas.

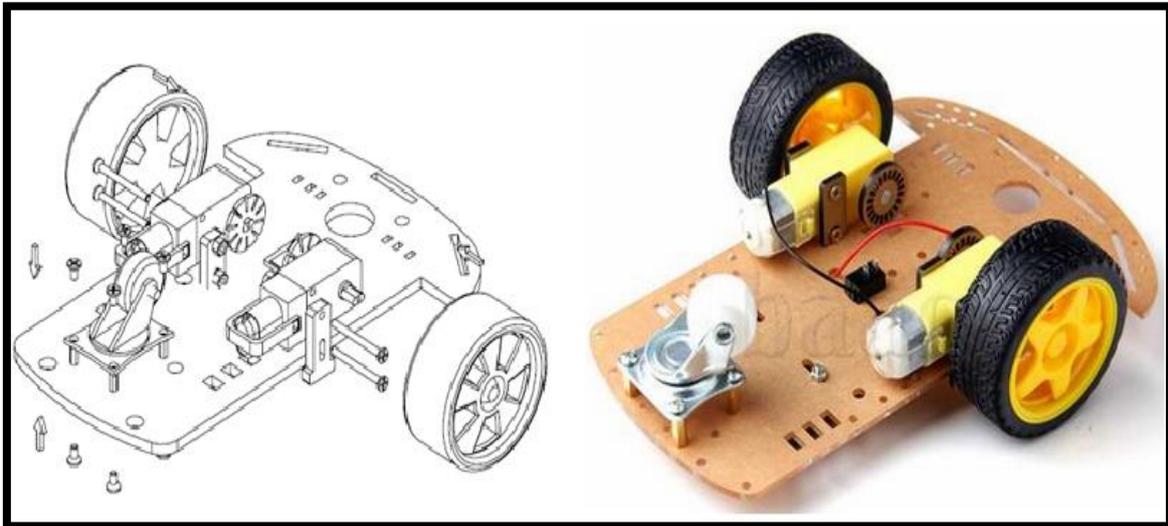


Figura 106. Armado del chasis
Fuente: El Autor

Luego del armado del chasis se procedió a incorporar los circuitos del robot. Como se indicaba anteriormente se tiene un circuito regulador de voltaje el cual fue el primero en ser incorporado junto con un interruptor para poder encender o apagar el robot.

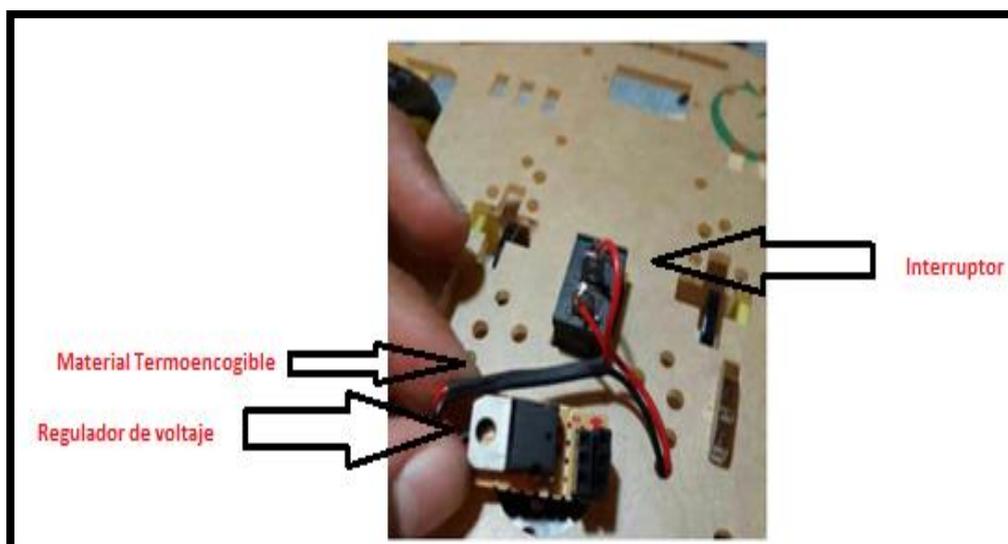


Figura 107. Interruptor y circuito regulador de voltaje
Fuente: El Autor

La *Figura 107* muestra el interruptor y el circuito regulador de voltaje el que permite obtener un voltaje de 5 voltios a partir de la batería que tiene un voltaje aproximado de 7 voltios. También señala la protección de los cables con la cinta termo-contráible.

Luego se procedió a incorporar el circuito principal y los sensores encoder. Se intentó proteger los cables y ubicarlos de la mejor manera para evitar que estos se rompan o que se junten provocando cortos o daños de los elementos.

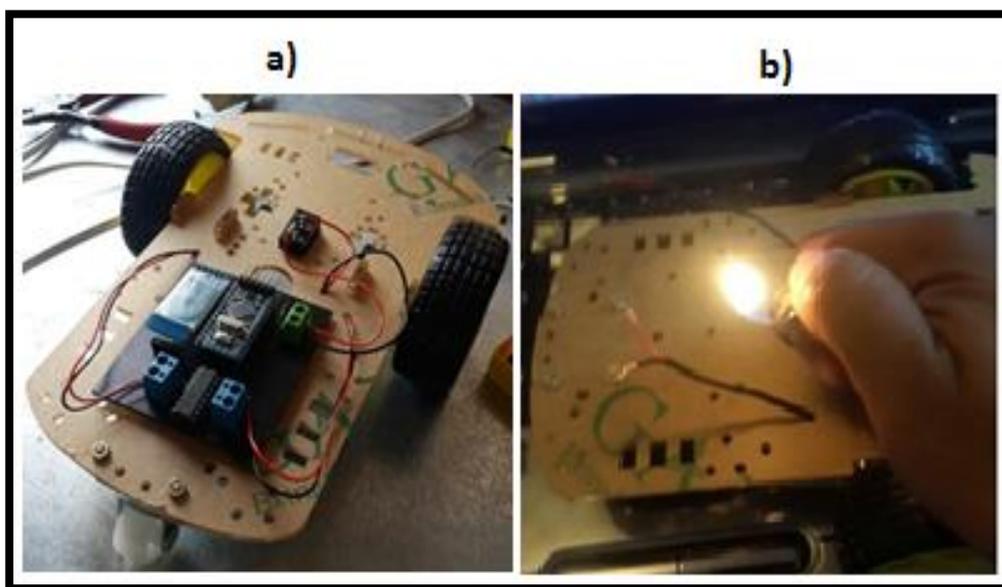


Figura 108. a) Incorporación del circuito principal. b) Protección de cables con material Termo-contráible
Fuente: El Autor

La *Figura 109* indica la ubicación de los elementos dentro del chasis, los cuales fueron cuidadosamente fijados con silicona caliente para evitar que se muevan o se caigan del chasis. También se protegió todos los cables conductores con el material termo-contráible para evitar que estos se rompan.

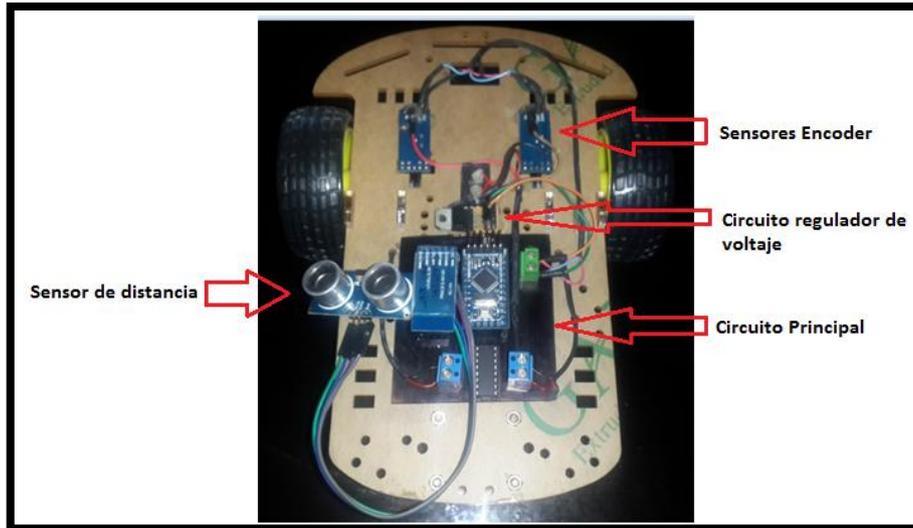


Figura 109. Implementación de los elementos en el chasis del robot
Fuente: El Autor

Culminada la construcción de la parte electrónica y del chasis se procedió a implementar una carcasa que llame la atención de los niños siguiendo las características del primer prototipo obteniendo como diseño final del prototipo del Robot el que se muestra en la *Figura 110*.

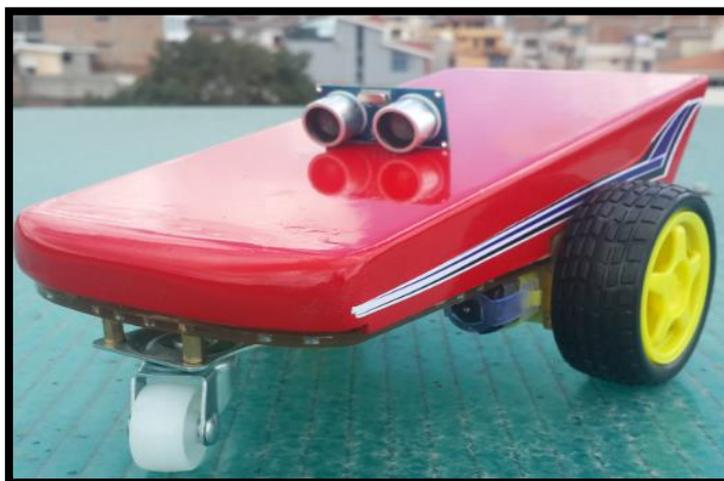


Figura 110. Diseño final del Robot JED-Pro
Fuente: El Autor

3.4.3 BLOQUE DE INSTRUCCIONES

El Bloque de Instrucciones está compuesto por un total de 27 de fichas. Cada ficha representa una instrucción o una estructura de control de programación y para que la interfaz pueda identificar la función de la ficha insertada cada una posee un valor resistivo diferente.

La distribución de fichas y el valor resistivo de cada una se detalla a continuación en la siguiente tabla.

Tabla 25. Distribución de fichas y valores resistivos

FUNCIÓN	CANTIDAD	VALOR DE RESISTENCIA
Avance hacia Adelante	4	100k Ω
Giro a la Derecha	4	47k Ω
Giro a la Izquierda	4	10k Ω
FOR X2	2	5.6k Ω
FOR X3	2	4.7k Ω
OBSTÁCULO	1	3.3k Ω
INICIO	10	1k Ω
FIN	10	330 Ω

Fuente: El Autor

3.4.3.1 Elaboración de Fichas de Instrucciones

El proceso de elaboración de cada ficha inicia cortando un trozo de madera de (30x12x7) mm el cual sirve como base de la ficha.

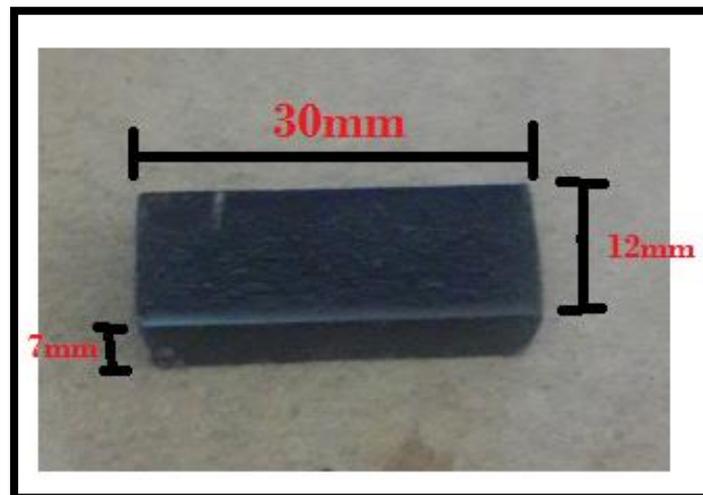


Figura 111. Dimensiones de la base de la ficha

Fuente: El Auto

La *Figura 111* indica la base de la ficha, de las cuales se necesitan 27, una para cada ficha. Luego se procedió a realizar un orificio en el centro de la base como se muestra en la *Figura 112* para poder alojar la resistencia.



Figura 112 Proceso de elaboración de las fichas, alojar resistencia
Fuente: El Autor

A continuación se procedió a soldar dos placas metálicas que sirvan como conductores y a la vez sean atraídas por los imanes de la interfaz. Esto se realizó con el afán de tener una mejor lectura del valor resistivo de la ficha.



Figura 113. Proceso de elaboración de las fichas, soldado de placas metálicas
Fuente: El Autor

Una vez soldadas las placas se procede a realizar una prueba para verificar que la soldadura fue correcta mediante la utilización de un multímetro verificando el valor de la resistencia.

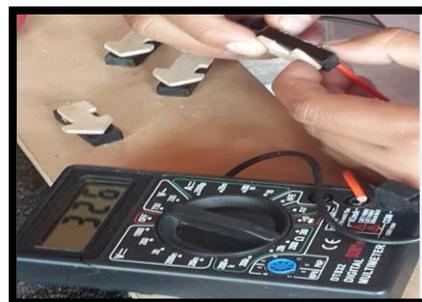


Figura 114. Proceso de elaboración de las fichas, verificación del valor resistivo
Fuente: El Autor

Luego de verificar la ficha mediante el multímetro se procedió a cortar y pegar la figura correspondiente a la ficha con silicón caliente.



Figura 115. Proceso de elaboración de las fichas, verificación del valor resistivo
Fuente: El Autor

Este proceso explicado se realizó con cada una de las fichas de instrucciones utilizando las resistencias especificadas en la *Tabla 25*. La *Figura 116* muestra todas las fichas antes de realizar el proceso de pintado.

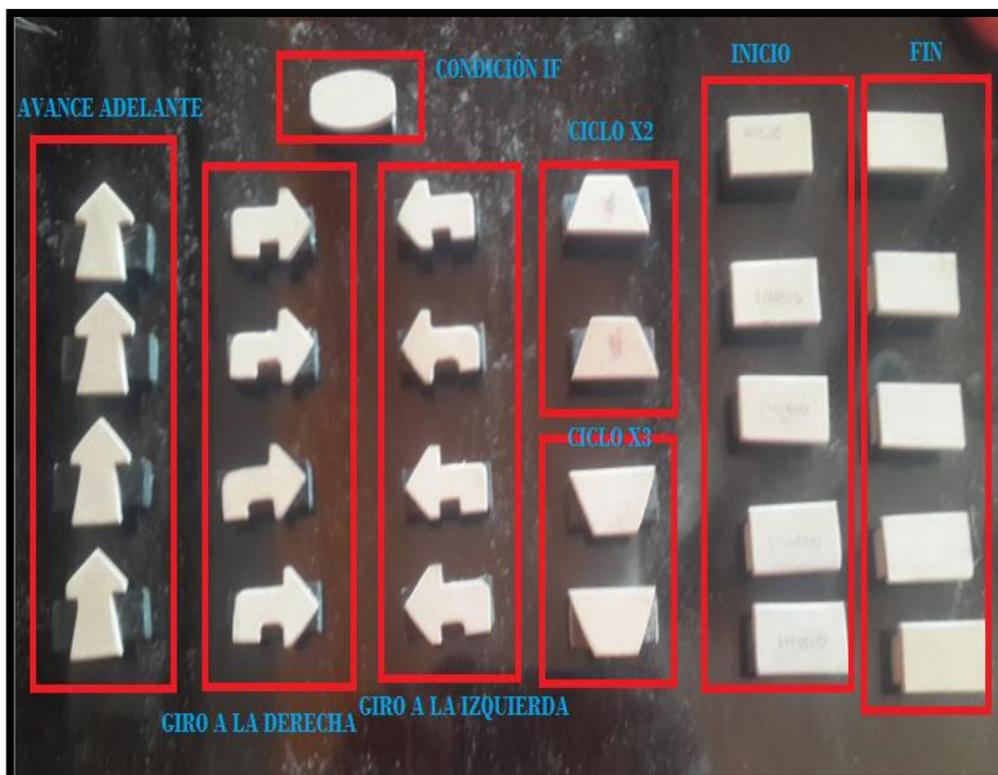


Figura 116. Bloque de Instrucciones
Fuente: El Autor

3.5 ANÁLISIS DEL PROTOTIPO JED-PRO

JED-Pro es un juguete didáctico basado en software y hardware que sirve como elemento de apoyo para enseñar lógica de programación y fomentar la cultura de creación de tecnología desde tempranas edades de forma lúdica. El objetivo del juego es programar un robot mediante programación tangible para guiarlo a través de un trayecto o ruta propuesta.

En el diseño y construcción del prototipo de JED-Pro se tomaron en cuenta los siguientes aspectos:

- Elaborar el diseño de las carcasas del juguete con colores y formas que llamen la atención de los niños y a la vez proteja los circuitos electrónicos.
- Desarrollar un programa en la plataforma Arduino que permita a los niños programar el robot de forma tangible.
- Utilizar la placa Arduino Mini-Pro para optimizar el costo económico del juguete.
- Elegir los elementos y dispositivos electrónicos acordes al funcionamiento del prototipo.
- Garantizar la vinculación entre módulos bluetooth mediante la autenticación de la dirección MAC.
- Utilizar una batería de Ion de Litio para alimentar el circuito electrónico del Robot para que funcione de forma totalmente inalámbrica.

Después de cumplir con los aspectos planteados se realiza un análisis de la programación tangible, la comunicación inalámbrica vía bluetooth y la ejecución de instrucciones del robot.

3.5.1 ANÁLISIS DE LA PROGRAMACIÓN TANGIBLE

Por lo general los lenguajes de programación existentes en la actualidad son textuales y propensos a errores de escritura y sintaxis, por lo que para su manejo se necesitan conocimientos avanzados que pueden resultar complejos para los niños. Debido a que la edad objetivo de este proyecto es de 4 a 7 años, la lógica de programación tangible diseñada busca facilitar de tal forma la programación que los niños puedan ir adquiriendo conocimientos sin necesidad de ningún conocimiento o alfabetización previa relacionada con los lenguajes de programación de alto nivel.

Las fichas de instrucciones contienen una resistencia con un valor óhmico diferente cada una, el cual resulta ser el valor identificador de la instrucción, la interfaz de programación después de ser insertada la secuencia de instrucciones realiza la lectura de los valores insertados y genera una secuencia de datos en un vector, estos datos posteriormente son enviados uno a uno vía bluetooth al robot para su ejecución.

La programación tangible aparte de ser una técnica validada para el aprendizaje de la lógica de programación para niños también apoya con su desarrollo cognitivo y social, esto se debe a que los niños cuando programan pueden visualizar sus ensayos, errores y variaciones de sus programas realizados, lo que despierta su interés, creatividad e interacción con la tecnología y a la vez le permite plasmar sus ideas y ver la causa y efecto de su creación.

3.5.2 ANÁLISIS DE LA COMUNICACIÓN BLUETOOTH

La comunicación entre la interfaz de programación y el robot es realizada de forma inalámbrica mediante la tecnología bluetooth. Esta comunicación se realiza mediante la

incorporación de dos módulos HC-05 los cuales están configurados como maestro el modulo correspondiente a la interfaz de programación y como esclavo el modulo perteneciente al robot. Para garantizar la vinculación entre el modulo maestro y el modulo esclavo se configuro una autenticación MAC en el dispositivo maestro para que solo pueda conectarse con el robot, esto es para evitar que el dispositivo maestro se vincule a un dispositivo equivocado.

En el momento en que los dos módulos se polaricen, el modulo maestro genera una solicitud de conexión al módulo esclavo, para que esta conexión se efectúe sin inconvenientes es necesario que la contraseña de ambos módulos sean las mismas, caso contrario los dispositivos rechazarían la vinculación.

Una vez establecida la comunicación entre el robot y la interfaz de programación los dispositivos están listos para el intercambio de datos. Es importante mencionar que la interfaz envía los datos de uno en uno para evitar saturar el canal de transmisión y a la vez garantizar que el dispositivo receptor en este caso el robot almacene los datos sin inconvenientes.

3.5.3 ANÁLISIS DEL ROBOT

El Robot de JED-Pro tiene la función de ejecutar las instrucciones programadas por los niños cumpliendo las instrucciones de avance adelante, giro a la derecha y giro a la izquierda. Además el robot de JED-Pro contiene un sensor ultrasónico que permite a los niños programar secuencias en función de un obstáculo encontrado.

La alimentación del circuito electrónico del robot es realizada mediante una batería de Ion de Litio de 7.4 voltios la cual según pruebas realizadas tiene una duración aproximada de

2 horas en continuo funcionamiento, este tipo de batería es la más utilizada en la alimentación de dispositivos electrónicos como celulares y tabletas electrónicas por su gran durabilidad y además es recargable lo que es una gran ventaja.

El robot no requiere ninguna configuración previa, únicamente se debe activarlo mediante el interruptor de encendido y apagado ubicado en la parte inferior del mismo y está listo para que el niño pueda jugar. Una vez encendido los niños programan la secuencia de instrucciones y presionan el botón de inicio, en el momento en que presionan el botón la interfaz realiza la lectura de los datos y envía uno a uno los datos. El robot recibe cada uno de los datos y los va almacenando en un vector hasta recibir el dato de fin de secuencia, en el momento que recibe este dato procede a interpretar las instrucciones almacenadas y a ejecutarlas.

3.5.4 MANUAL DE USUARIO

3.5.4.1 Introducción

JED-Pro es un juguete electrónico que tiene como propósito didáctico la enseñanza de lógica de programación a niños y niñas de 4 a 7 años, para introducirlos desde sus primeras etapas de vida en el pensamiento computacional que es uno de los pilares fundamentales para el desarrollo tecnológico.



Figura 117. Presentación de JED-Pro
Fuente: El Autor

JED-Pro contiene una lógica de programación tangible que permite guiar un pequeño Robot usando fichas de colores como instrucciones, evitando así el uso de dispositivos inteligentes. Además permite conocer los principales conceptos de la programación como los son instrucción, algoritmo, secuencia, ciclo y condición; Todo esto de una forma lúdica sin necesidad una alfabetización previa sino a través del juego.

3.5.4.2 Elementos de JED-Pro

3.5.4.2.1 Robot



Figura 118. Robot JED-Pro
Fuente: El Autor

Un robot con un diseño amigable que ejecuta las instrucciones programadas por los niños en tiempo real. Tiene 2 llantas principales que están conectadas a los motores que producen el movimiento, una rueda omnidireccional en la parte delantera que sirve como soporte y ayuda en los giros, un sensor ultrasónico que funciona como detector de obstáculos, un interruptor de encendido y apagado y un led indicador de encendido. El suministro de energía del robot se realiza con una batería de ion de litio de 7.4 Voltios.

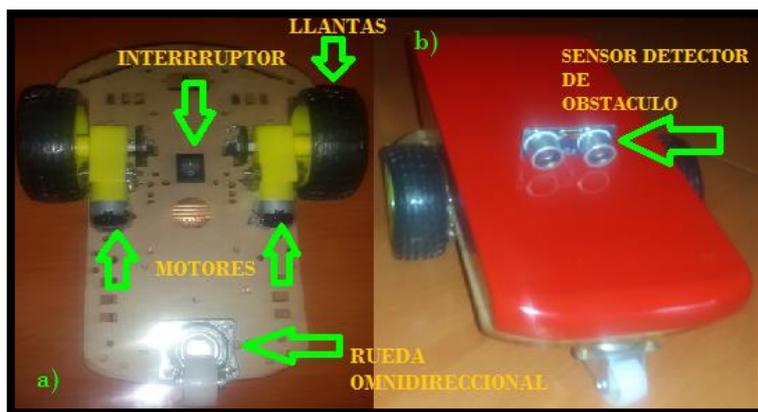


Figura 119. Partes del Robot JED-Pro
Fuente: El Autor

3.5.4.2.2 Interfaz de Programación



Figura 120. Interfaz de Programación de JED-Pro
Fuente: El Autor

Una interfaz de programación mediante la cual los niños controlan el robot insertando las fichas de instrucciones en las 16 ranuras disponibles. Las ranuras están interconectadas entre sí en forma de “culebra” con la finalidad de no interrumpir la cola de instrucciones. Posee un conector USB para la alimentación de circuito, un botón inicio el cual envía las instrucciones programadas al robot y un led indicador de encendido. La alimentación se puede realizar conectando la interfaz a un ordenador a través de un cable USB o también se podría alimentar directamente con una fuente de poder de 5 voltios a 1A.

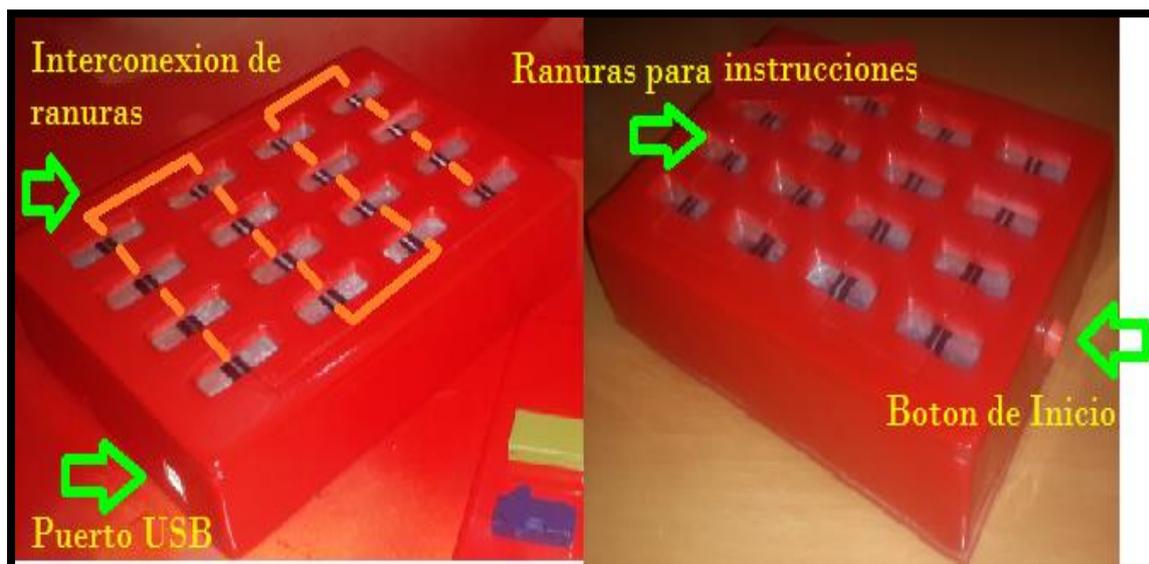


Figura 121. Partes de la Interfaz de Programación de JED-Pro
Fuente: El Autor

3.5.4.2.3 *Bloque de instrucciones*

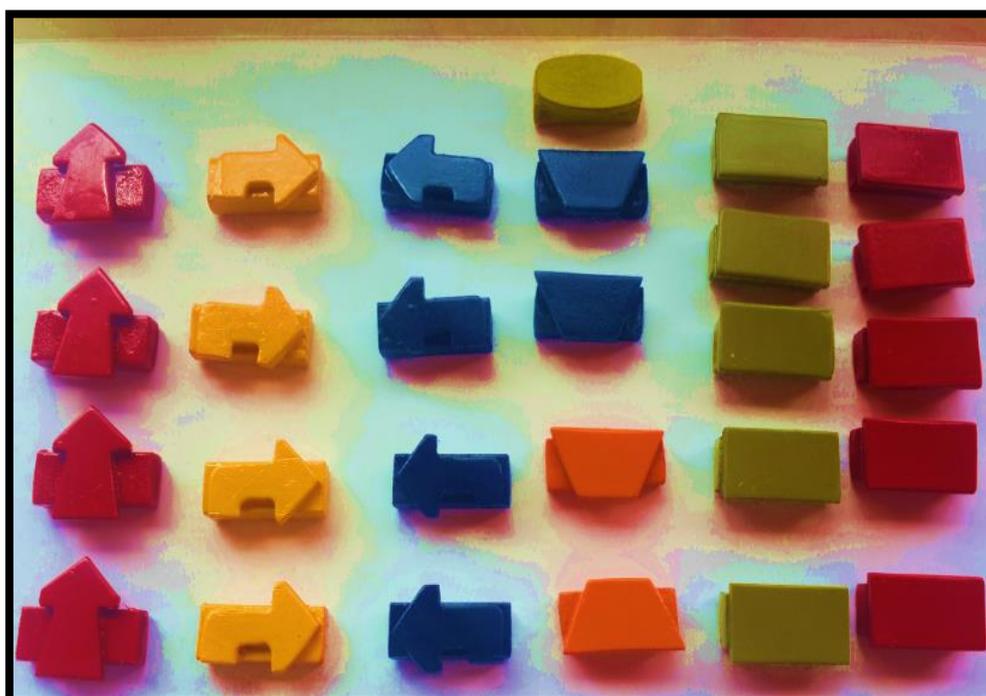


Figura 122. Bloque de Instrucciones
Fuente: El Autor

Un conjunto de fichas de colores que representan las instrucciones de programación y se diferencian por su color y forma. El bloque de instrucciones está dividido en 12 fichas de instrucciones básicas (4 de Avance Adelante, 4 de Giro a la derecha y 4 de Giro a la izquierda), 4 fichas de estructuras de control repetitivas (2 de ciclo de 2 repeticiones y 2 de ciclo de 3

repeticiones), una ficha de estructura condicional “si hay obstáculo” y 10 fichas de inicio y fin que indican donde inicia y donde finaliza una estructura de control (5 de inicio y 5 de fin).



Figura 123. Función de las fichas de Instrucciones
Fuente: El Autor

3.5.4.3 Empezar a jugar con JED-Pro

3.5.4.3.1 Vincular La Interfaz De Programación con El Robot

La interfaz de programación y el robot se comunican de forma inalámbrica por lo tanto es necesario que estos estén vinculados antes de empezar a programar.



Figura 124. Emparejamiento vía Bluetooth
Fuente: El Autor

El proceso de vinculación es transparente a los usuarios es decir, no se requiere que el usuario realice ninguna configuración únicamente debe alimentar la interfaz a través del puerto

USB y habilitar el interruptor del Robot en encendido, luego esperamos unos segundos y el robot y la interfaz estarán vinculados.

Para comprobar la vinculación inserte una ficha de instrucción básica (Avance Adelante, Giro a la derecha o Giro a la izquierda) en la primera ranura de la interfaz, luego presione el botón de inicio y el robot debería ejecutar la instrucción.

En caso de no existir comunicación una de las causas puede ser que la batería este descargada. Para solucionar remplace o cargue la batería e intente de nuevo el proceso de vinculación.

3.5.4.3.2 *Generar una Secuencia de Instrucciones*

Una vez comprobado que existe comunicación entre la interfaz y el robot estamos listos para empezar a programar. La programación consiste en dar un conjunto de instrucciones que el robot debe ejecutar, este conjunto de instrucciones resulta ser una secuencia.

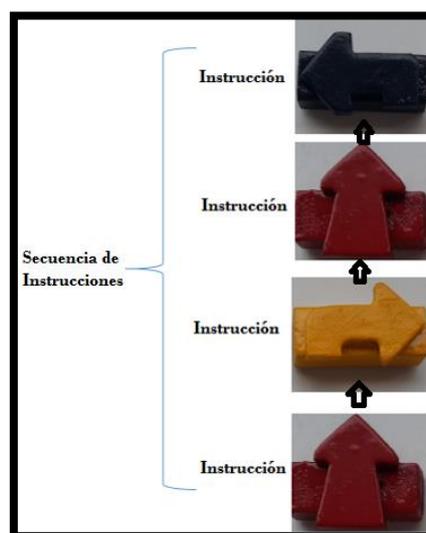


Figura 125. Elaboración de una Secuencia de Instrucciones
Fuente: El Autor

Para generar una secuencia inserte un conjunto de fichas en forma ordenada en las ranuras de la interfaz. Al combinar las fichas de instrucciones secuencialmente está escribiendo un programa para que el robot lo ejecute. Use el botón de inicio para compilar y enviar el programa al robot. El robot recibe el programa y lo ejecuta. Para realizar algún cambio en el programa ejecutado, espere que el robot termine de ejecutar la secuencia y luego realice los cambios.

Las instrucciones deben insertarse una después de otra iniciando desde la ranura inferior izquierda siguiendo la línea dibujada en la superficie de la interfaz. En caso de insertar una ficha saltándose una ranura, la instrucción no será leída por lo tanto no será ejecutada.

3.5.4.3 Estructura de Control Repetitiva

Las fichas de Ciclo x2 y Ciclo x3 son fichas de control que permiten repetir un conjunto de instrucciones por dos o tres veces respectivamente. Lo que permite realizar programas más complejos y resolución de problemas más avanzados.



Figura 126. Fichas de Ciclos
Fuente: El Autor

3.5.4.3.4 Estructura de Control Condicional

La Ficha “Si hay obstáculo” es una instrucción condicional que permite ejecutar un conjunto de instrucciones si el robot encuentra un obstáculo, es decir el robot al entrar en esta estructura de control va avanzar hacia adelante hasta encontrar un obstáculo, en el momento que detecta el obstáculo ejecuta el conjunto de instrucciones programadas dentro de la estructura.



Figura 127. Ficha de Condición “Si hay obstáculo”
Fuente: El Autor

3.5.4.4 Actividades Sugeridas

El propósito de JED-Pro es permitir que los niños generen programas para que sean ejecutados por el robot, desarrollando su imaginación y creatividad. Cuanto más tiempo el niño dedique a JED-Pro obtendrá mayor capacidad de pensamiento lógico de programación. El desarrollo del aprendizaje de los niños puede ser medido mediante la observación ya que se podrá notar que el niño cada vez tendrá mayor capacidad de resolución de problemas con la realización de secuencias más complejas.

3.5.4.4.1 Actividad #1. Presentación del Jugete al niño

Según la investigación realizada la capacidad de aprendizaje de los niños varia de uno a otro, es decir unos niños puede aprender con mayor facilidad que otros, sin embargo siempre

es necesario introducir a los niños al juguete desde lo más básico sin importar que tan rápido ellos puedan aprender.

3.5.4.4.1.1 *Presentación del Robot*

Presentar al Robot como un elemento que ellos pueden programar. Los niños deben tener en claro que el Robot no puede pensar ni moverse por sí mismo y que solo va ejecutar las instrucciones que el niño le indique.

3.5.4.4.1.2 *Presentación de la Interfaz de Programación*

La interfaz debe ser presentada como una especie de control remoto mediante el cual los niños podrán controlar o dirigir los movimientos del robot. Los niños tienen que tener en claro que sin la interfaz no hay forma de enviar instrucciones al Robot.

3.5.4.4.1.3 *Presentación del Bloque de Instrucciones*

Las fichas de instrucciones deben ser presentadas como los elementos para guiar al robot y las direcciones que sigue cuando son insertadas en la interfaz y el botón de inicio es presionado. Hay que indicarle la función de cada una de las fichas.

3.5.4.4.2 *Actividad #2. Identificación de Instrucciones Básicas*

La segunda actividad y la más importante es la identificación de las instrucciones, para empezar primero inserte la ficha roja (Avance Adelante) en la primera ranura de la interfaz y presione el botón de inicio. El robot ejecutara una sola instrucción moviéndose hacia adelante. Debe asegurarse que el niño asocie claramente la forma y el color de la ficha con la acción

realizada, si es necesario repita el proceso algunas veces hasta que el niño tenga claro que la ficha roja con la flecha hacia adelante mueve el robot hacia adelante.



Figura 128. Ejemplo de la Actividad #2
Fuente: El Autor

Repita este procedimiento con las otras instrucciones básicas (Giro a la derecha y Giro a la izquierda) hasta que el niño tenga claro estas instrucciones.

3.5.4.4.3 Actividad #3. Secuencia de dos instrucciones

Coloque un objetivo o meta directamente frente al Robot, a continuación pida al niño que cree un programa con dos fichas que lleve al robot hacia la meta aplicando los conocimientos adquiridos anteriormente en la identificación de las instrucciones. Deje que el niño razone y decida que fichas de instrucción insertar e indicarle que después de insertar las instrucciones debe presionar el botón para enviar las instrucciones. Si el niño elige la ficha o instrucción equivocada no se preocupe simplemente restablezca la posición inicial del robot y motívelo a razonar su elección y probar nuevamente.

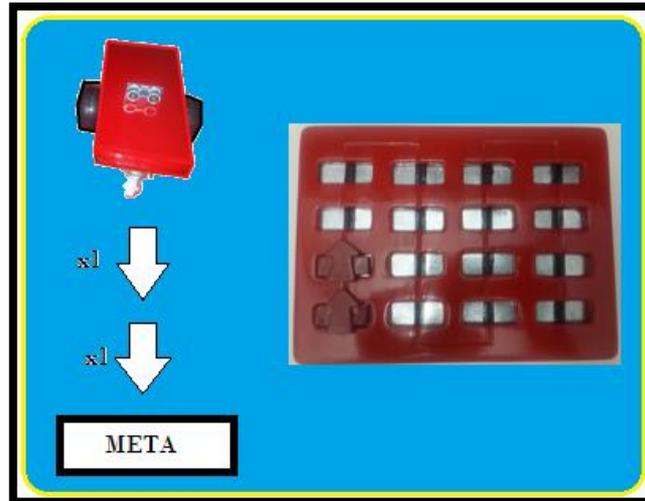


Figura 129. Ejemplo de la Actividad #3
Fuente: El Autor

3.5.4.4.4 Actividad #4. Secuencia de Tres Instrucciones

Ubique el objetivo o meta a la izquierda del robot de tal manera que el niño tenga que insertar tres fichas para llegar al objetivo. Indíquelo al niño que debe generar un programa que conduzca al robot a la meta insertando tres fichas en las tres primeras ranuras de la interfaz y luego presionar el botón de inicio. En caso de seleccionar las instrucciones erróneas deje que el robot ejecute la instrucciones y que el niño razone y determine cuál fue su error, restablezca la posición del robot e intente de nuevo. Este proceso de que el niño identifique el error le dará capacidad de resolución de problemas.

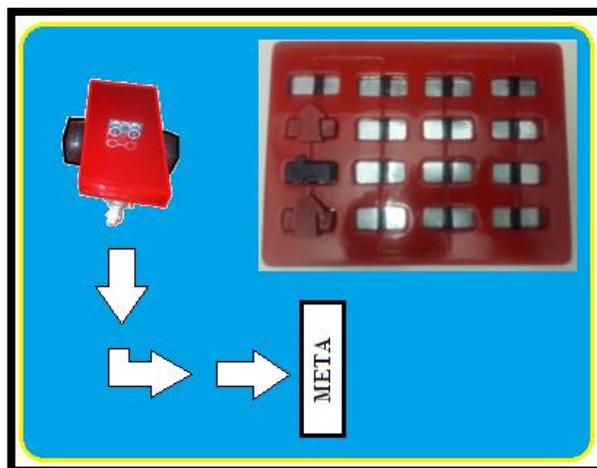


Figura 130. Ejemplo de la Actividad #4
Fuente: El Autor

Esta actividad debe realizarse ubicando la meta en diferentes posiciones agregándole un poco de complejidad a la par con la que va superando los retos, es importante indicarle al niño que únicamente puede utilizar tres fichas de instrucciones para llegar al objetivo en esta actividad.

Luego de que el niño haya superado todos los retos propuestos genere actividades un poco más complejas sin limitar la cantidad de instrucciones, deje que el niño razone e intente resolver los retos por sí mismo, no importa las veces que falle ya que los niños no se van a quedar tranquilos hasta encontrar la solución, esta curiosidad cognitiva es propia de su edad.

3.5.4.4.5 Actividad #5. Presentación de las instrucciones de control repetitivas

Antes de pasar a esta actividad el niño debe tener claro la programación con instrucciones básicas propuestas en las actividades anteriores. Para esta actividad coloque un objeto delante del robot y genere la siguiente secuencia de instrucciones: Ficha de control X2 + Ficha de inicio+ Ficha Adelante + Ficha de fin. Indicarle al niño que con ese conjunto de instrucciones podrá repetir dos veces la instrucción adelante e indicarle la función de cada uno de las fichas.



Figura 131. a) Estructura del ciclo de 2 repeticiones. b) Estructura del ciclo de 3 repeticiones
 Fuente: El Autor

Después cambie la ficha de control X2 por la ficha X3 y presione el botón de inicio el robot deberá repetir la acción de avance hacia adelante 3 veces, deje que el niño razone e identifique lo que sucedió y cuál es la función de la nueva ficha. Luego cambie la instrucción de avance adelante por la ficha de avance a la derecha y compruebe que el robot va a girar tres veces a la derecha, realice este proceso con las demás instrucciones las veces que sean necesarias para que le niño reconozca las instrucciones de control repetitivas.

3.5.4.4.6 *Actividad #6. Aplicación de Estructuras de Control Repetitivas*

Después de haber familiarizado al niño con las estructuras de control colocar un objetivo o meta delante del robot y decirle que debe realizar un programa utilizando una estructura de control repetitiva para llegar a la meta. De igual forma que en las actividades anteriores tratar de que el niño razone e intente por sí mismo generar el programa, no tratar de corregir los errores únicamente ayudarlo a entender cuál es la idea pero dejar que el mismo resuelva el reto planteado. En un principio va a resultar un poco complicado pero con la practica va entender la utilización delas estructuras.

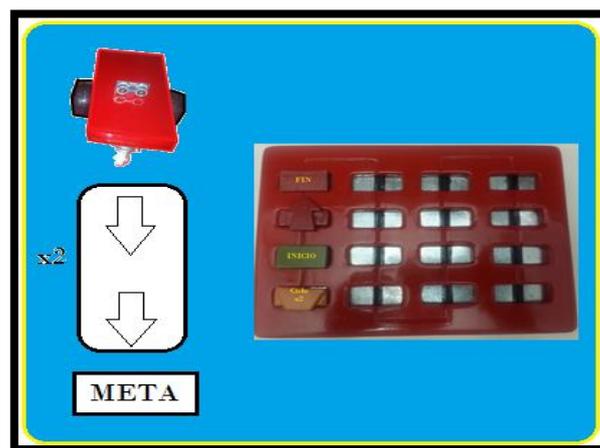


Figura 132. Ejemplo de la Actividad #6
Fuente: El Autor

Repita la utilización de las estructuras repetitivas ubicando el objetivo en diferentes sitios, después puede añadir más instrucciones como las del ejemplo.

3.5.4.4.7 Actividad #7. Estructura de Control Condicional

Después de perfeccionar el uso de las instrucciones básicas y las estructuras repetitivas se puede dar a conocer a los niños una funcionalidad muy utilizada en la programación que es la estructura de control condicional. Para presentar esta estructura al niño inserte las siguientes instrucciones en la interfaz de programación: Ficha de “Si hay Obstáculo” + Ficha de Inicio + Ficha Giro a la derecha + Ficha de fin. Luego presionar la ficha de inicio, se podrá observar que el Robot va avanzar hacia adelante y si encuentra un obstáculo va a girar hacia la derecha y se va a detener.



Figura 133. Estructura de la condición “Si hay Obstáculo”
Fuente: El Autor

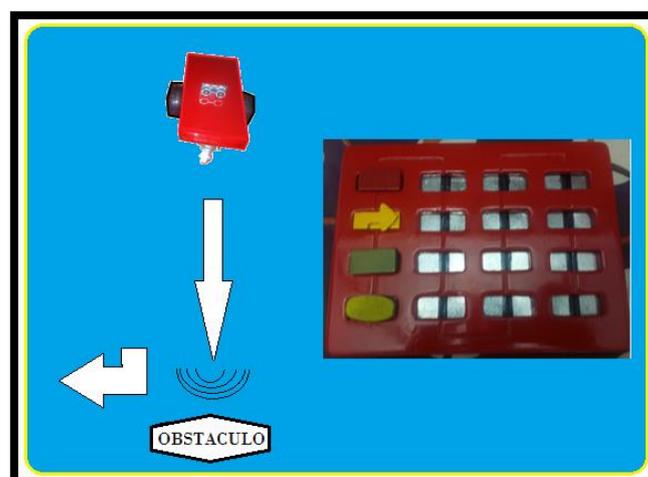


Figura 134. Ejemplo de la Actividad #7
Fuente: El Autor

3.6 PRUEBAS Y RESULTADOS DEL PROTOTIPO JED-PRO

Para la recolección de datos se ejecutó individualmente cada una de las funciones de la programación tangible de JED-Pro; Esta programación tangible se diseñó tomando en cuenta que sea lo más parecida posible a un lenguaje de programación tradicional, pero adaptándolo a un medio físico para no hacer uso de ningún computador o dispositivo inteligente.

Cada Ficha del bloque de instrucciones posee un valor resistivo diferente, el mismo que permite identificar la función que cumple la ficha o la instrucción a la que representa. Las fichas al ser insertadas en la interfaz de programación se someten a una conversión análoga digital, la cual arroja un valor numérico comprendido entre 0 y 1023, este valor número es el que permite identificar qué tipo de instrucción es la que corresponde. A continuación la *Tabla 26* muestra el valor numérico que se arroja como resultado de la conversión análoga digital.

Tabla 26. Valor De Conversión Análoga Digital

FICHA	VALOR CONVERSIÓN ANÁLOGA
AVANCE ADELANTE	802
GIRO A LA DERECHA	930
GIRO A LA IZQUIERDA	991
CICLO DE 2 REPETICIONES	510
CICLO DE 2 REPETICIONES	88
CONDICIÓN “SI HAY OBSTÁCULO”	837
ABRIR ESTRUCTURA	32
CERRAR ESTRUCTURA	720

Fuente: El Autor

Basándose en la *Tabla 26*, la programación de la Interfaz al recibir el pulso de inicio realiza un proceso de lectura de las instrucciones, en el mismo que realiza la conversión análoga digital y obtiene el valor numérico de cada instrucción. Después de obtener los valores realiza una comparativa en la cual dependiendo del valor leído lo transforma en un dato tipo Char (Carácter), el cual es posteriormente es enviado a el Robot través de la comunicación serial. La *Tabla 27* muestra el carácter identificador de cada ficha.

Tabla 27. Caracteres identificadores

FICHA	CARÁCTER ENVIADO
AVANCE ADELANTE	A
GIRO A LA DERECHA	B
GIRO A LA IZQUIERDA	C
CICLO DE 2 REPETICIONES	D
CICLO DE 3 REPETICIONES	E
CONDICIÓN “SI HAY OBSTÁCULO”	F
ABRIR ESTRUCTURA	{
CERRAR ESTRUCTURA	}

Fuente: El Autor

El Robot va recibiendo uno a uno todos los caracteres de la secuencia programada y los almacena en un vector temporalmente hasta terminar de recibir toda la secuencia. Una vez recibidos todos los caracteres los clasifica en tres tipos que son instrucciones simples, ciclo de dos repeticiones, ciclos de tres repeticiones y condición “Si hay obstáculo”; Una vez realizada la clasificación procede con la ejecución de las instrucciones activando los motores dependiendo de la instrucción. La activación de los motores se detalla en la *Tabla 28*.

Tabla 28. Activación de motores por instrucción

INSTRUCCIÓN	MOTOR DERECHO		MOTOR IZQUIERDO	
	ADELANTE	ATRÁS	ADELANTE	ATRÁS
AVANCE ADELANTE	HIGH	LOW	HIGH	LOW
GIRO A LA DERECHA	LOW	HIGH	HIGH	LOW
GIRO A LA IZQUIERDA	HIGH	LOW	LOW	HIGH

Fuente: El Autor

La *Tabla 29* se indica cada una las pruebas a las que se sometió el Prototipo final de JED-Pro, las mismas que fueron realizadas mediante experimentación para comprobar el correcto funcionamiento del prototipo.

Tabla 29. Pruebas de funcionamiento del Prototipo

PRUEBA	CUMPLE
Lectura de fichas de instrucciones mediante la conversión análoga digital	SI
Conversión del valor numérico al carácter identificador de cada una de las instrucciones	SI
Emparejamiento de los módulos bluetooth a una distancia máxima de 5m	SI
Envío y Recepción de datos a través de la tecnología bluetooth	SI
Ejecución de la instrucción "Avance Adelante"	SI
Ejecución de la instrucción "Giro a la Derecha"	SI
Ejecución de la instrucción "Giro a la Izquierda"	SI
Comprobación del funcionamiento de la Estructura del Ciclo de 2 repeticiones	SI
Comprobación del funcionamiento de la Estructura del Ciclo de 3 repeticiones	SI
Comprobación del funcionamiento de la Estructura de la condición "Si hay Obstáculo"	SI

Fuente: El Autor.

3.6.1 PRUEBAS CON NIÑOS DE 4 A 7 AÑOS

Las pruebas fueron realizadas con dieciséis niños y niñas de la ciudad de Ibarra comprendidos entre las edades de 4 a 7 años que son las edades acordes a la propuesta del presente proyecto.



Figura 135. Pruebas con niños y niñas de 4 a 7 años
Fuente: El Autor

Las pruebas tenían como propósito determinar el grado de respuesta y aceptación que JED-Pro es capaz de generar en los niños a partir de sus características físicas y funcionales. Las pruebas básicamente consistían en evaluar mediante observación directa mientras los niños llevan a cabo las actividades especificadas en el manual de usuario mediante el cual se obtuvo los resultados especificados en la *Tabla 30*.

Tabla 30 Resultados generales de las pruebas con niños de 4 a 7 años

ACTIVIDAD	INDICACIONES	RESULTADOS
Presentación de los elementos de JED-Pro	Los tres elementos de JED-Pro fueron presentados por separado dándoles las indicaciones de funcionamiento de cada uno.	Desde el primer momento en que observaron el prototipo lo reconocieron como un juguete y acataron las funcionalidades de cada uno de los elementos con facilidad.
Puesta en marcha del juguete	Se le explico a los niños que deben conectar la interfaz de programación al toma corriente o a la computadora, activar el interruptor de encendido del robot y esperar unos segundo que se encienda la luces de color azul del robot, lo que indica que está listo para ser programado	Acataron con facilidad el proceso de puesta en marcha y se pudo notar que les llamaba mucho la atención el momento en que el robot encendía las luces.
Reconocimiento de instrucciones básicas (Avance hacia adelante, Giro a la derecha y Giro a la Izquierda)	Se indicó la ejecución de cada instrucción básica por separado y repitiendo varias veces la función de cada instrucción.	En un principio tuvieron dificultad principalmente en identificar en base a la ubicación del robot cual es la derecha e izquierda sin embargo conforme se familiarizaban con las instrucciones y en base a los colores de las fichas se les fue facilitando el proceso.
Generación de secuencias con instrucciones básicas	En base a la colocación de retos con obstáculos se les pedía a los niños que inserten las fichas necesarias para llegar a la meta propuesta.	Acataron las indicaciones con facilidad y se notaba el interés por resolver los problemas o retos propuestos.
Reconocimiento de las estructuras de control	Se indicó como se debe estructurar las fichas para que su secuencia de instrucciones se repita por dos o tres veces o se ejecute cuando encuentre un obstáculo.	En un principio se les dificulto mucho las utilización de las estructuras de control principalmente a los más pequeños por lo que se notaba que evitaban su utilización sin embargo con el avance de los retos se daban cuenta que necesitaban la utilización de estructuras repetitivas y de control en base al obstáculo.
Relacionamiento del juego con la programación	Con el afán de que los niños vayan relacionando las actividades del juego se le nombraba a cada una de las fichas instrucciones, al conjunto de instrucciones se le denominaba secuencia y a las repeticiones ciclos.	Con el pasar del tiempo y a las repetidas veces que se le nombraba a las fichas como instrucciones los niños fueron reconociendo que las fichas eran instrucciones y el conjunto de fichas era una secuencia por lo que algunos niños ya llamaban las cosas por su nombre.

Fuente: El Autor. Resultados basados en observación directa

La *Tabla 31* establece resultados y observaciones principales de las pruebas realizadas con los 16 niños en sus diferentes edades.

Tabla 31 Resultados y observaciones en las distintas edades

EDAD	RESULTADOS	OBSERVACIONES
4 Años (4 niños)	No todos los niños de esta edad lograron perfeccionar en su totalidad las funcionalidades de JED-Pro	Dificultad con el uso de las estructuras de control y confundían entre los giros de derecha e izquierda.
5 Años (4 niños)	Todos los niños de esta edad lograron perfeccionar las funcionalidades de JED-Pro	Dificultad con las estructuras de control
6 Años (4 niños)	Todos los niños de esta edad lograron perfeccionar las funcionalidades de JED-Pro	Dificultad en entender la estructura “si hay obstáculo”
7 Años (3 niños)	Todos los niños de esta edad lograron perfeccionar las funcionalidades de JED-Pro	Sin dificultades
Discapacidad (Autismo)	La niña con la que se realizó la prueba no perfeccionó todas las funcionalidades de JED-Pro	Solamente perfeccionó el uso de las funciones básicas.

Fuente: El Autor. Resultados obtenidos mediante observación directa

En la *Tabla 32* se detalla el tiempo aproximado que tardaron los niños con los que se realizaron las pruebas en perfeccionar el uso y las funcionalidades de JED-Pro, clasificándolos por edades.

Tabla 32 Tiempos de perfeccionamiento del uso de JED-Pro

ACTIVIDAD	4 AÑOS	5 AÑOS	6 AÑOS	7 AÑOS	Discapacitados
Presentación de los elementos de JED-Pro	15 Minutos	15 Minutos	10 Minutos	10 minutos	20 Minutos
Puesta en marcha del juguete	5 Minutos				
Reconocimiento de instrucciones básicas (Avance hacia adelante, Giro a la derecha y Giro a la Izquierda)	15 Minutos	15 Minutos	18 Minutos	10 Minutos	18 Minutos
Generación de secuencias con instrucciones básicas	30 minutos	20 minutos	25 minutos	15 minutos	60 minutos
Reconocimiento de las estructuras de control	de 120 a 240 minutos	de 120 a 240 minutos	de 60 a 120 minutos	de 60 a 120 minutos	No logrado
Relacionamiento del juego con la programación	inmediato	Inmediato	inmediato	inmediato	No relacionó
Tiempo Total	305 minutos (5 horas aproximadamente)	295 minutos (5 horas aproximadamente)	178 minutos (3 horas aproximadamente)	160 minutos (3 horas aproximadamente)	103 minutos (2 horas aproximadamente)
Tiempo Promedio	4 horas				

Fuente: El Autor. Los datos visualizados en esta tabla son tiempos aproximados del perfeccionamiento de JED-Pro en las diferentes edades.

CAPÍTULO IV. ANÁLISIS ECONÓMICO

En el presente capítulo se realiza un análisis económico del Jugete Electrónico Didáctico para la enseñanza de lógica de programación planteado en este proyecto, con la finalidad de hacer una relación costo-beneficio que justifique los costos asociados del mismo con los beneficios que este representaría si se incorporara en el proceso de enseñanza y aprendizaje de los niños.

4.1 COSTO DEL HARDWARE

En la *Tabla 32* se detalla los costos de los elementos y dispositivos electrónicos utilizados en la implementación del prototipo.

Tabla 33. Costos de Hardware

ELEMENTO	CANTIDAD	PRECIO UNITARIO (DÓLARES)	PRECIO TOTAL (DÓLARES)
Arduino Mini –Pro	2	8	16
Chasis Robot	1	20	20
Resistencias	48	0,05	2,4
Modulo bluetooth HC-05	2	11	22
Sensor ultrasónico SR04	1	4,5	4,5
Multiplexor 74hc4051	2	1	2
Regulador LM7805	1	0,5	0,5
Batería Ion de Litio 7.4V	1	10	10
Pulsador	1	1	1
Borneras de 2 pines	4	0,25	1
Baquelita de cobre	1	1,80	1,80
Espadines macho	1	0,75	0,75
Espadines hembra	1	0,75	0,75
Cable termo contraíble (1m)	1	0,60	0,60
Diodos LED	2	0,15	0,30
Otros	1	10	10
		TOTAL	93,60

Fuente: El Autor

Como se puede observar en la *Tabla 32*, el costo total de los materiales utilizados en la construcción del prototipo es de \$ 93,60. Es importante mencionar que el valor total del costo puede variar dependiendo del proveedor de los materiales electrónicos y de la cantidad de elementos adquiridos.

4.2 COSTO DEL SOFTWARE UTILIZADO

En la *Tabla 33* se muestra los programas utilizados en el desarrollo del presente proyecto. El más importante en el presente trabajo es el Entorno de Desarrollo Integrado de Arduino en el cual se desarrolló la programación del prototipo, este programa es de código abierto por lo tanto tiene distribución libre y es gratuito. Los programas Eagle y Proteus fueron utilizados una sola vez para la creación de circuitos impresos y simulación respectivamente del prototipo, solo se hizo uso de las versiones gratuitas que estos programas ofrecen.

Tabla 34. Costo del Software

SOFTWARE	CANTIDAD	VALOR TOTAL
Arduino IDE	1	\$0
EAGLE 6.5 (versión de prueba)	1	\$0
ISIS Proteus 7.9 (versión de prueba)	1	\$0
	TOTAL	\$0

Fuente: El Autor

4.3 BENEFICIOS

A continuación se detallan los beneficios que conlleva la implementación del Jugete Electrónico Didáctico para la enseñanza de programación a niños y niñas de 4 a 7 años.

- Promueve la enseñanza y el aprendizaje de programación

JED-Pro además de ser un juguete divertido también brinda pautas importantes para la introducción a la programación, siendo una opción de enseñanza individualizada desde tempranas edades. La simplicidad que conlleva la acción de jugar para los niños facilita su aprendizaje y aumenta la motivación y el gusto por adquirir conocimientos.

- Ideal para personas que deseen tener un inicio en la programación

Aunque el público tentativo de JED-Pro son niños de 4 a 7 años, este puede ser utilizado por niños, jóvenes o adultos de cualquier edad que tengan ganas de jugar y adquirir conocimientos de programación.

- Promueve la cultura del desarrollo tecnológico desde tempranas edades

Uno de los objetivos principales de JED-Pro es promover en los niños el desarrollo tecnológico mediante la enseñanza de programación y de esta forma que los niños dejen de ser consumidores de tecnología y se conviertan en creadores.

- Desarrolla la creatividad de los niños

Mediante el juego los niños son capaces de desarrollar su creatividad y mucho más si sus ideas se pueden plasmar en la realidad, JED-Pro al permitir que las ideas de los niños sean plasmadas en un programa y que este programa se ejecute en tiempo real motiva e influye en el desarrollo de la creatividad de los niños.

- Enseña razonamiento sistemático en resolución de problemas

Los niños al realizar programas y al poder ver su ejecución pueden desarrollar la capacidad de resolución de problemas en caso de que el Robot no ejecute las acciones deseadas.

4.4 MODELO DE NEGOCIO CANVAS

Según Osterwalder y Pigneur (2011), el modelo de negocio CANVAS describe la fórmula con la que una organización crea, entrega y captura valor. Crear valor sobre un producto o servicio necesario para alguien, entregar valor vinculando lo que se produce con el mercado y capturar beneficios económicos o de otros tipos que permitan la sostenibilidad del proyecto dentro del sistema universitario o científico.

El modelo CANVAS se representa mediante un diagrama esquemático basado en nueve indicadores:

- Segmentación de clientes, para identificar oportunidades de negocio
- Definición de la propuesta de valor o de diferenciación
- Definición de los canales de comunicación
- Establecer la relación con los clientes
- Determinar las fuentes económicas
- Identificar los recursos claves
- Conocer las actividades claves que agregan valor al producto
- Tener en cuenta a los socios para establecer las alianzas claves
- Marcar las estructuras de costos

A continuación se detalla el modelo de negocio CANVAS del Prototipo JED-Pro

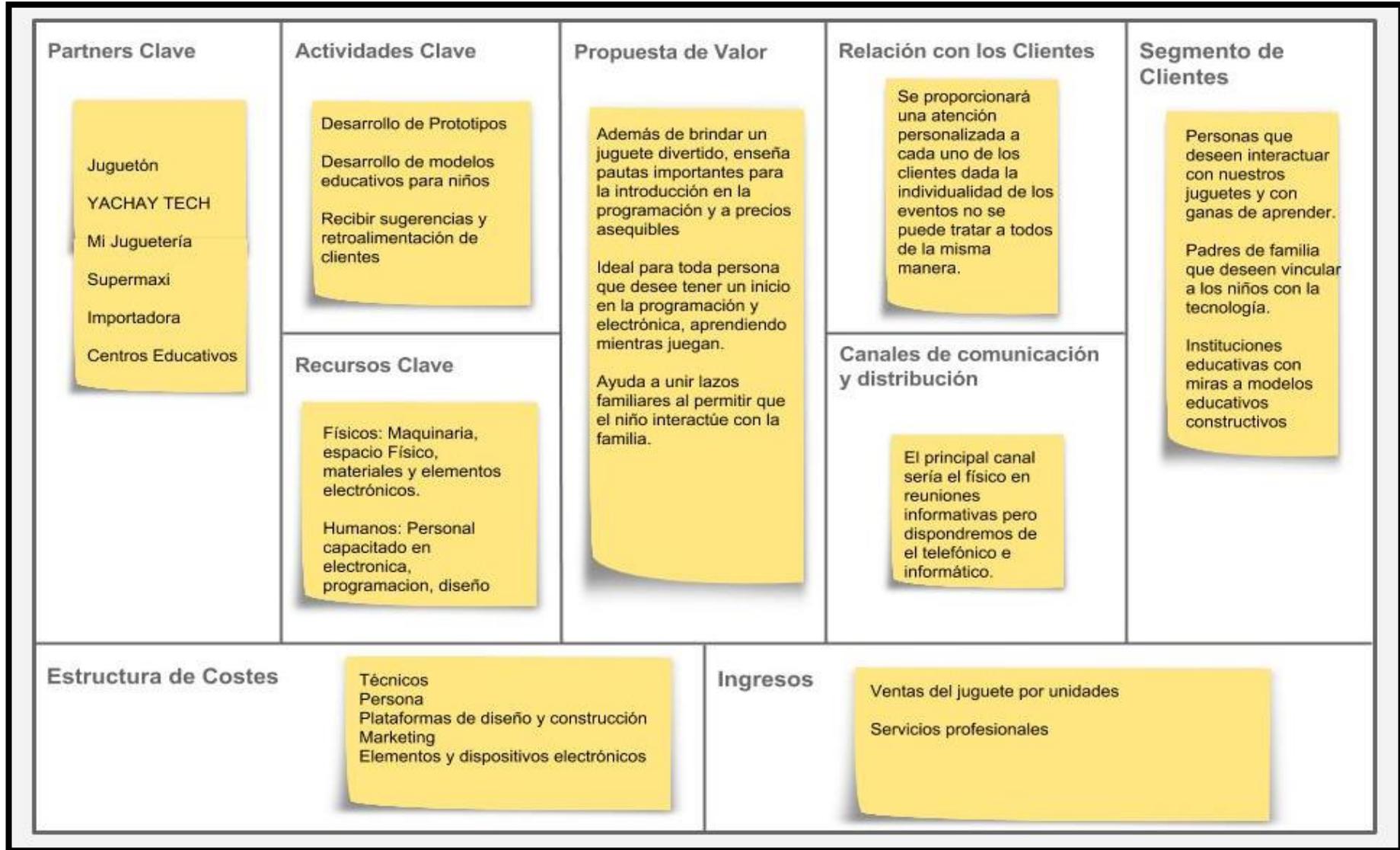


Figura 136. Modelo CANVAS de JED-Pro
Fuente: El Autor

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

Finalizado el proyecto titulado “Juguete Electrónico Didáctico, como elemento de apoyo para la enseñanza de programación a niños y niñas de 4 a 7 años”; se presentan las siguientes conclusiones y recomendaciones.

5.1 CONCLUSIONES

- Para conocer la capacidad de aprendizaje de los niños se realizó una investigación de las teorías del aprendizaje y de las características evolutivas de los niños de 4 a 7 años, la misma que permitió entender que los niños que oscilan entre estas edades se encuentran en una etapa del pensamiento intuitivo, en la cual desarrollan habilidades de aprendizaje por lo tanto si es posible que los niños adquieran conocimientos, entre ellos de programación, tomando en cuenta que unos niños pueden aprender con mayor facilidad que otros.
- El elemento principal del presente trabajo es un juguete, por lo que se investigó cual es la incidencia de los juguetes didácticos en el proceso de aprendizaje de los niños y se llegó a la conclusión de que son de vital importancia ya que el juego es algo innato de los niños y es la forma más adecuada de impartir conocimientos a los menores.
- La investigación realizada en este proyecto acerca de la incidencia de la tecnología en los niños conllevó a pensar que aunque la tecnología puede resultar de gran ayuda en los diferentes aspectos de la vida, se debe tomar precauciones en la forma como se usa

y como es proporcionada a los niños ya que puede afectar la integridad física y psicológica con un uso inapropiado e irresponsable.

- JED-Pro no solo fue diseñado para enseñar programación sino que también fue diseñado con el propósito de que se constituya un elemento que entretenga a los niños y junte a la familia sin necesidad de involucrarlos con dispositivos tecnológicos que al proporcionarse a los niños desde muy tempranas edades pueden causar dependencia y consumismo.
- Se desarrolló una lógica de programación tangible basada en la plataforma Arduino, la cual se introdujo en el juguete con el afán de familiarizar a los niños con conceptos básicos como lo son instrucción, secuencia y estructuras de control de una forma lúdica y sin necesidad de dispositivos inteligentes.
- Se logró diseñar un Juguete Electrónico Didáctico como elemento de apoyo a la enseñanza de programación y las nuevas tendencias de educación, que busca inculcar el desarrollo tecnológico desde tempranas edades; Basándose en conocimientos previos de electrónica y tecnologías inalámbricas adquiridos durante la carrera de ingeniería en electrónica y redes de comunicación y a la vez aportando con un granito de arena al cambio de la matriz productiva del país.
- Para potencializar el uso de JED-Pro se realizó un manual de usuario que indica cómo deben ser presentados cada uno de los elementos del juguete a los niños, además contiene la explicación de las funcionalidades y posibles retos que ayudan

principalmente en la inicialización y familiarización de la programación tangible de juguete.

- Las pruebas de JED-Pro se realizaron con un total de 16 niños comprendidos entre las edades de 4 a 7 años. Estas pruebas determinaron la gran aceptación por parte de los padres de familia y principalmente de los niños mismo, donde se pudo observar su gran predisposición por jugar y aprender; En un principio se notó la dificultad de entender el uso de estructuras de control principalmente en los niños más pequeños, sin embargo poco a poco fueron incorporándolas en sus programas.
- Las pruebas determinaron que el tiempo promedio para perfeccionar las funcionalidades de JED-Pro fue de 4 horas, sin embargo para unos niños resulto mucho más fácil que para otros, lo que ratifica la teoría socio-cultural de Vygotsky que establece que el aprendizaje no solo depende del intelecto sino también de la motivación y el ambiente en el que se aprende.
- Mediante el transcurrir de las pruebas se pudo observar que los niños empezaron a empoderarse de sus ideas, lo que les permitía sentirse más seguros de ellos mismo haciendo que mejoren su liderazgo y a la vez su comunicación tanto con sus amigos como con su propia familia.
- JED-Pro también fue probado con una niña con autismo de 4 años y aunque no logró perfeccionar todas las funcionalidades del juguete, JED-Pro llamo su atención y estimuló la acción de jugar en la niña logrando estimular su socialización y sus habilidades motrices.

- El presente proyecto denominado “JUGUETE ELECTRÓNICO DIDÁCTICO COMO ELEMENTO DE APOYO PARA LA ENSEÑANZA DE PROGRAMACIÓN A NIÑOS Y NIÑAS DE 4 A 7 AÑOS” fue presentado en el Concurso Nacional de Telecomunicaciones Primer Edición donde obtuvo el premio al segundo lugar gracias a la utilidad e innovación tecnológica que presenta.

5.2 RECOMENDACIONES

- El presente trabajo busca colaborar con el cambio de la matriz productiva del país, no solo proporcionando un producto 100% ecuatoriano sino que también aporta con un propósito didáctico que promueve la creación de tecnología desde tempranas edades; Por lo que se recomienda difundir este tipo de propuestas para motivar a más estudiantes a colaborar desarrollando investigaciones y prototipos electrónicos que tengan un fin educativo o resuelvan un problema de la sociedad.
- La carrera de ingeniería en electrónica y redes de comunicación brinda pautas para el diseño y construcción de dispositivos electrónicos por lo que se recomienda en base a los conocimientos adquiridos crear más alternativas mediante la investigación e implementación de dispositivos electrónicos basados en la niñez que son los que podrían cambiar la productividad del país en un futuro no muy lejano.
- En lo que se refiere a diseño y construcción de dispositivos electrónicos se recomienda conocer las especificaciones de las hojas de datos de los elementos, principalmente de corrientes y voltajes para poder evitar inconvenientes en el momento de la implementación de los circuitos.

- Las plataformas de software y hardware libre han permitido que países de bajos recursos como el nuestro, incursionen en el mundo de la creación de prototipos electrónicos y desarrollo de sistemas informáticos con mayor facilidad; por eso se recomienda promover su uso ya que presentan infinidad de beneficios.
- Referente al uso de JED-Pro (Juguete Electrónico Didáctico Programable) se recomienda seguir los pasos descritos en el manual de usuario para sacar un mayor provecho y que los niños se familiaricen con mayor facilidad antes de crear secuencias más complejas.

BIBLIOGRAFÍA

LIBROS

Cisneros Estupiñán, M. (2012). *Cómo elaborar trabajos de grado (2a. ed.)*. Ecoe Ediciones.

Cormack Lynch, M. (2004). *Estrategias de aprendizaje y enseñanza en la educación del menor de 6 años*. *Acción Pedagógica*. 13(2): 154-161, 2004. D - Universidad de los Andes Venezuela.

Galeano Atehortúa, G. A. (2009). *Programación de sistemas embebidos en C, teoría y prácticas aplicadas a cualquier microcontrolador*. Alfaomega Grupo Editor.

García Prósper Beatriz, Songel Gabriel. (2004). *Factores de innovación para el diseño de nuevos productos en el sector juguetero*. Editorial de la Universidad Politécnica de Valencia.

Martín Biezma, C. (Diciembre 2013). *Didáctica de la educación infantil*. Macmillan Iberia, S.A.

Purificación, S. G. (2003). *Las ideas de niños y adolescentes acerca de algunos aspectos del mundo laboral*. *Psicología Educativa. Revista de los Psicólogos de la Educación*. 1995, Vol. 1 (1) 19 páginas. Madrid: Colegio Oficial de Psicólogos de Madrid.

Reyes Cortés Fernando, Cid Monjaraz Jaime, Vargas Soto Emilio. (2013). *Mecatrónica: control y automatización*. Alfaomega Grupo Editor.

Rosales, G. (2009). *Niñez - Resiliencia - Creatividad*. Argentina: Brujas.

Salsa, A. M. (2005). *Desarrollo simbólico en niños pequeños: el rol de la instrucción en la comprensión y el uso de símbolos*. Red Interdisciplinaria.

Siraj-Blatchford, J. (2005). *Nuevas tecnologías para la educación infantil y primaria*. Ediciones Morata, S. L.

PAGINAS WEB Y REPOSITARIOS VIRTUALES

Almeida, I., & Ochoa, J. (2013). Repositorio UPS. Obtenido de <http://dspace.ups.edu.ec/bitstream/123456789/4498/1/UPS-GT000403.pdf>

Andrade, L. P. (Diciembre de 2010). Repositorio UTE. Obtenido de http://repositorio.ute.edu.ec/bitstream/123456789/10256/1/43437_1.pdf

Angela Alvarez C, E. O. (1979). Desarrollo de las funciones básicas para el aprendizaje de la lecto-escritura según la teoría de Piaget segunda parte. Nueva Universidad, 250-258.

Angelfire. (2015). Obtenido de Multiplexores: <http://www.angelfire.com/al2/Comunicaciones/Laboratorio/multiple.html>

Arcega, I. I. (2009). Conductismo, Cognitivismo y Diseño Instruccional. Buenos Aires.

Arduino, O. (21 de Abril de 2015). Arduino Oficial. Obtenido de <http://www.arduino.cc/>

Arroyo, L. (2014). Repositorio ESPOCH. Obtenido de <http://dspace.espoch.edu.ec/bitstream/123456789/3139/1/65T00104.pdf>

Artero, Ó. T. (2013). Arduino : curso práctico de formación. Madrid: RC Libros.

Benitez, R. (6 de Diciembre de 2012). BotScience. Obtenido de <https://botscience.wordpress.com/2012/06/20/tutorial-02-entradas-y-salidas-digitales-y-uso-de-temporizadores/>

Benitez, S. (22 de Julio de 2010). loquilladas. Obtenido de LA IMPORTANCIA DE LOS JUGUETES EN LA INFANCIA: <http://loquilladas.blogspot.com/2010/07/la-importancia-de-los-juguetes-en-la.html>

Bers, M. (10 de Diciembre de 2008). educ.ar. Obtenido de El portal educativo del Estado argentino: <http://portal.educ.ar/debates/educacionytic/inclusion-digital/ensinar-a-programar-a-los-nino.php>

Briconatur. (20 de Enero de 2014). Obtenido de <http://blog.briconatur.com/baterias-recargables-tipos-y-caracteristicas/>

Cabrera, A., & Delgado, G. (2014). Repositorio UAzúy. Obtenido de <http://dspace.uazuay.edu.ec/bitstream/datos/3610/1/10292.pdf>

- Cabrera, P., & Nato, E. (Diciembre de 2012). Repositorio ESPE. Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/6395/1/T-ESPEL-CDT-0991.pdf>
- Carmen, A. A. (4 de Abril de 2007). Diseño de un Manual de Estrategias Metodológicas para la Integración de la Informática en el Desarrollo Integral de los niños/as de 4 – 5 años, en el Centro de Desarrollo Infantil UTE. Obtenido de Repositorio virtual UTE: http://repositorio.ute.edu.ec/bitstream/123456789/11444/1/32589_1.pdf
- Casares, F. J. (2012). ELECTROTECNIA. Obtenido de http://www.uco.es/electrotecnia-etsiam/pdf-master/05-Comunicacion_Serie-paralelo.pdf
- Cevallos, O. (Septiembre de 2013). repositorio pucese. Obtenido de repositorio.pucese.edu.ec/bitstream/123456789/.../Oscar%20Cevallos.pdf
- Chacón, P. (2008). El Juego Didáctico como estrategia de enseñanza y aprendizaje. Nueva Aula Abierta , 16.
- Cortes, M. C. (2015). Servidor Proton. Obtenido de <http://proton.ucting.udg.mx/~redblade/Paginas/Robotica/Tareas/sensores/Sensores.html>
- DIY Makers. (26 de Noviembre de 2013). Obtenido de <http://diymakers.es/control-velocidad-y-sentido-de-motor-dc/>
- DONGO, A. (2008). LA TEORÍA DEL APRENDIZAJE DE PIAGET Y SUS CONSECUENCIAS PARA LA PRAXIS EDUCATIVA. REVISTA DE INVESTIGACIÓN EN PSICOLOGÍA - VOL. 11, Nº 1, 168-181.
- Elía, V. d. (12 de Noviembre de 2014). oei. Obtenido de <http://www.oei.es/congreso2014/memoriactei/469.pdf>
- Fernández, J. (Septiembre de 2011). joefernandez.me. Obtenido de <http://www.joefernandez.me/docs/MemoriaBlueWay.pdf>
- Fernando. (2 de Marzo de 2009). Universidad de Sevilla: un campus una ciudad. Obtenido de <http://departamento.us.es/edan/php/asig/LICFIS/LFIPC/Tema5FISPC0809.pdf>
- García, J. (2015). Asi Funciona. Obtenido de http://www.asifunciona.com/fisica/ke_led/ke_led_2.htm
- Gomez, M., & Coronel, K. (2011). ELABORACIÓN DE MATERIAL DIDACTICO EN EL ÁREA DE LAS MATEMÁTICAS DIRIGIDO A NIÑOS Y NIÑAS DE 2 A 4 AÑOS DE LA FUNDACIÓN SALECIANA PACES UBICADO EN EL SECTOR DE FERIA LIBRE. Cuenca: UNIVERSIDAD POLITECNICA SALECIANA SEDE CUENCA.
- Güidi, P., Rossaro, A. L., & Tellería, X. G. (Mayo de 2014). Fundacion Evolución. Obtenido de http://craig.com.ar/biblioteca/6/MS_AprenderAProgramarEnLaEscuela.pdf

- Guzhñay, A., & Calle, L. (Marzo de 2015). Repositorio UPSC. Obtenido de <http://dspace.ups.edu.ec/bitstream/123456789/8843/1/UPS-CT005042.pdf>
- Hernández, A. (2013). Diseño del PCB, Proceso de producción y calibración de un sensor de temperatura y conductividad para embarcaciones marinas. Mérida, Yucatán, México.
- Herrador, R. E. (13 de Noviembre de 2009). Guía de Usuario Arduino. Obtenido de http://www.jcarazo.com/tmp/Arduino_user_manual_es.pdf
- Kotch, J. (2012). North Carolina Child Care Health and Safety Resource Center. Obtenido de http://www.healthychildcarenc.org/PDFs/ccnews_2012_winter_span.pdf
- Lecaro, A., Gonzalez, J., & Valdivieso, C. (2011). Repositorio ESPOL. Obtenido de <http://www.dspace.espol.edu.ec/bitstream/123456789/19021/1/paper%20tesis%204567.pdf>
- Letrán, J. (Diciembre de 2011). Corto Circuito. Obtenido de Introducción a Arduino: <http://www.cortoc.com/2011/12/introduccion-arduino.html>
- Monsalve, N. (24 de Febrero de 2015). Robótica I. Obtenido de http://www.nelsonmonsalveo.com/2015_02_01_archive.html
- Montes, L. V. (15 de Abril de 2012). La importancia de los juguetes. Obtenido de Nuestros niños: <http://www.nuestrosninos.com/PDFs/130-importanciadejuguetes.pdf>
- Montesinos, J. (Septiembre de 2013). Red de sensores auto-configurable mediante tecnologías ZigBee y Arduino con monitorización por aplicación Android. Obtenido de <http://repositorio.bib.upct.es/dspace/bitstream/10317/3678/1/tfg176.pdf>
- Moreta, J. (Septiembre de 2009). Repositorio UTA. Obtenido de <http://repositorio.uta.edu.ec/bitstream/123456789/257/1/t418e.pdf>
- Muñoz, F. (1976). Psicología del desarrollo infantil. Quito: Centro Audiovisual de la Universidad Central del Ecuador.
- Newby, P. A. (1993). Una Comparación De Los Aspectos Críticos Desde La Perspectiva Del Diseño De Instrucción. En Conductismo, Cognitvismo Y Constructivismo (págs. 50-72).
- Orengo, D. J. (2013). Perspectiva Cognitivista. Obtenido de suagm: http://www.suagm.edu/umet/biblioteca/Reserva_Profesores/janette_orengo_educ_173/Teoria_%20Sociocultural_%20de_%20Vygotsky_agosto_2012.pdf
- Pérez, F. C. (2014). LA TEORÍA DEL DESARROLLO COGNITIVO DE PIAGET APLICADA COGNITIVO DE PIAGET APLICADA. Valladolid.
- Pérez, I. E. (s.f.). I-Micro. Obtenido de Ingeniería en Microcontroladores: <http://www.i-micro.com/pdf/articulos/rs-232.pdf>

- Pers, H. (2009). Guía de Metodologías Participativas y Juego Educativo. Bolivia: Paola Ballivián.
- Prada, M. (2015). Calameo. Obtenido de <http://es.calameo.com/read/0021048730b3c06fa7982>
- Roberto, C. (2014 de Diciembre de 2014). LA PROGRAMACIÓN ES UN JUEGO DE NIÑOS. Obtenido de Innovacion y ciencia: <http://www.bloglenovo.es/la-programacion-es-un-juego-de-ninos-o-eso-intentan-estas-catorce-propuestas/>
- Rojo, B. (13 de Marzo de 2013). ¿Niños de primaria aprendiendo a programar? Obtenido de <http://blog.uptodown.com/ninos-de-primaria-aprendiendo-a-programar/>
- Roman, A. (2013). avid-romangonzalez.com. Obtenido de http://www.avid-romangonzalez.com/Sensores_y_Actuadores.pdf
- Román, L. L. (2011). Programación estructurada y orientada a objetos: un enfoque algorítmico (3a. ed.). Mexico: Alfaomega Grupo Editor.
- Romero, V. (Julio de 2009). Creacion de entorno 3D para la simulaciond de trafico urbano. Madrid, España.
- Rubio, F. (2000). Cursillo de Electrónica Practica. Obtenido de <http://www.grupoautodisco.com>
- Saburido, A. (20 de Mayo de 2014). 10 Razones para usar Arduino. Obtenido de <http://www.modulo0tutoriales.com/10-razones-para-usar-arduino/>
- Sánchez, E. (Diciembre de 2012). Diseño de un sistema de control domótico basado en la plataforma Arduino. Obtenido de riunet: <https://riunet.upv.es/bitstream/handle/10251/18228/Memoria.pdf?sequence=1>
- Santana, M. S. (2005). LA ENSEÑANZA DE LAS MATEMÁTICAS Y LAS NTIC. UNA ESTRATEGIA DE FORMACIÓN PERMANENTE. Tarragona.
- TABERNER, R. (2015). ENCODERS ÓPTICOS. Obtenido de http://www.infoplac.net/files/documentacion/instrumentacion_deteccion/infoPLC_net_ENCODERS_OPTICOS.pdf
- Urgiles, E., & Colcha, D. (2015). Diseño e implementación de un sistema de comunicación Omron – Celular para medir la presión arterial. Obtenido de epoch.edu.ec: <http://dspace.epoch.edu.ec/bitstream/123456789/3794/1/98T000631.pdf>
- Zamora, I. (4 de Febrero de 2015). ABC Tecnología. Obtenido de Razones por las que deberíamos aprender programación siendo niños: http://www.abc.es/tecnologia/informatica/20140915/abci-programacion-201409111713_2.html

ANEXO 01 PROGRAMACIÓN DE LA INTERFAZ DE PROGRAMACIÓN

```
/*  
  
PROTOTIPO JED-PRO (JUGUETE ELECTRÓNICO DIDÁCTICO PROGRAMABLE)  
  
PROGRAMA INTERFAZ DE PROGRAMACIÓN TANGIBLE  
  
ELABORADO POR: EDUARDO CARTAGENA  
  
FECHA: 14 DE JULIO DEL 2015  
  
MODELO DE PLACA: ARDUINO MINI-PRO 5V  
  
*/  
  
////////////////////////////////////DECLARACIÓN DE VARIABLES////////////////////////////////////  
  
int estadoBoton; // guarda el estado del botón  
const int tiempoAntirebote = 20; // tiempo de espera para que no exista rebote  
int estadoBotonAnterior; // guarda el estado anterior del botón  
int lecturas[16]; // vector que guarda las lecturas de los pines analógicos  
int rango = 25; // Rango de error que puede tener la lectura analógica  
const int seleccion1 [] = {2, 3, 4}; // Pines de selección del mux1 (S1,S2,S3)  
const int seleccion2 [] = {10, 11, 12}; // Pines de selección del mux2 (S1,S2,S3)  
const int analogPin2 = A0; // Pin analógico conectado a la salida del mux1  
const int analogPin = A1; // Pin analógico conectado a la salida del mux2  
const int Boton = 8; // Pin digital 8 para la conexión del Botón de inicio  
////////////////////////////////////  
  
/*INICIALIZACIÓN DE FUNCIONES Y DECLARACIÓN DE PINES COMO ENTRADA O  
SALIDA (SE EJECUTA UNA SOLA VEZ)*/  
////////////////////////////////////  
  
/  
  
void setup () {  
  Serial.begin(9600); //inicializa monitor serial  
  for (int bit = 0; bit < 3; bit++) { // ciclo que declara los pines de selección como salidas  
    pinMode(seleccion1[ bit ], OUTPUT);  
    pinMode(seleccion2[ bit ], OUTPUT);  
  }  
  pinMode(Boton, INPUT); // Declara el pin del botón como entrada  
  
}  
////////////////////////////////////  
/
```

```
//////////////////////////////////SUB-RUTINAS//////////////////////////////////
```

```
////////////////////////////////// Retorna los valores leídos en el mux1//////////////////////////////////
```

```
int mux1( int channel) {  
  for (int bit = 0; bit < 3; bit++) {  
    int pin = seleccion1 [ bit ];  
    int isBitSet = bitRead(channel, bit);  
    digitalWrite(pin, isBitSet);  
    delay(10);  
  }  
  return analogRead(analogPin);  
}
```

```
//////////////////////////////////
```

```
////////////////////////////////// Retorna los valores leídos en el mux2//////////////////////////////////
```

```
int mux2( int channel2) {  
  for (int bit = 0; bit < 3; bit++) {  
    int pin2 = seleccion2 [ bit ];  
    int isBitSet2 = bitRead(channel2, bit);  
    digitalWrite(pin2, isBitSet2);  
    delay(10);  
  }  
  return analogRead(analogPin2);  
}
```

```
//////////////////////////////////
```

```
////////////////////////////////// ANTI-REBOTE//////////////////////////////////
```

```
/* Evita rebotes al momento de pulsar el botón de inicio*/
```

```
boolean antirrebote(int pin)  
{  
  int contador = 0;  
  boolean estado;  
  boolean estadoAnterior;  
  do {  
    estado = digitalRead(pin);  
    if (estado != estadoAnterior) {  
      contador = 0;  
      estadoAnterior = estado;  
    }  
    else {  
      contador++;  
    }  
    delay(1);  
  } while (contador < tiempoAntirebote);  
  return estado;  
}
```

```
//////////////////////////////////
```

```
////LECTURA DEL VALOR ANALÓGICO DE LAS FICHAS INSERTADAS//////
```

```
void LeerEntradas() {  
  for (int i = 0; i < 8; i++) {  
    int value = mux1(i);  
    lecturas[i] = value;  
    // Serial.println(lecturas[i]);    /* Activar para imprimir los valores (Entradas 1-8)  
  }  
  for (int j = 8; j < 16; j++) {  
    int value2 = mux2(j);  
    lecturas[j] = value2;  
    // Serial.println(lecturas[j]); /* Activar para imprimir los valores (9-16)  
  }  
}
```

```
////////////////////////////////////INSTRUCCIONES////////////////////////////////////
```

```
/*Transforma el valor analógico leído al carácter identificador de la instrucción*/
```

```
void Instrucciones(int i) {  
  if (lecturas[i] > (802 - rango) && lecturas[i] < (802 + rango)) { //Avance Adelante  
    Serial.println('A');  
    delay(100);  
  }  
  
  if (lecturas[i] > (930 - rango) && lecturas[i] < (931 + rango)) { //Giro a la Derecha  
    Serial.println('B');  
    delay(100);  
  }  
  
  if (lecturas[i] > (991 - rango) && lecturas[i] < ( 991 + rango )) { // Giro a la Izquierda  
    Serial.println('C');  
    delay(100);  
  }  
  
  if (lecturas[i] > (510 - rango) && lecturas[i] < (510 + rango)) { // Ciclo 2 repeticiones  
    Serial.println('D');  
    delay(100);  
  }  
  
  if (lecturas[i] > (88 - rango) && lecturas[i] < (88 + rango)) { // Ciclo 3 repeticiones  
    Serial.println('E');  
    delay(100);  
  }  
  
  if (lecturas[i] > (828) && lecturas[i] < (837 + rango)) { //Condición de Obstáculo  
    Serial.println('G');  
    delay(100);  
  }  
}
```

```

if (lecturas[i] > (32 - rango) && lecturas[i] < ( 32 + rango)) { // Abre estructura
  Serial.println('{}');
  delay(100);
}

if (lecturas[i] > (847 - rango) && lecturas[i] < (847 + rango)) { // Cierra estructura
  Serial.println('{}');
  delay(100);
}
}
}
////////////////////////////////////MÉTODO PRINCIPAL////////////////////////////////////
/*Se Ejecuta permanentemente */

void loop ( ) {
  estadoBoton = digitalRead(Boton); // Lee el estado del Botón
  if (estadoBoton != estadoBotonAnterior) { // Verifica estado del Botón
    if (antirrebote(Boton)) { // Llama la subrutina de Anti-Rebote
      LeerEntradas(); // Llama la subrutina para Leer entradas
      for ( int i = 0; i < 17; i++) { // ciclo que cambia el número de entrada leída
        if (lecturas[i] < 1010) { // si el valor es mayor a 1010, no hay ficha
          Instrucciones(i); // envía las ID de las instrucciones
        }
        else { // caso contrario // al finalizar de enviar las instrucciones envía
          Serial.println('0'); // un carácter de FIN
          delay(100); // retardo de 100 milisegundos
          break; // detiene el ciclo
        }
      }
    }
  }
}

estadoBotonAnterior = estadoBoton; //Guarda el estado anterior del Botón para que no
//existan rebotes y se lea una sola vez cuando es
// Presionado

delay(30);
}

/*
*****
*****NOTA*****
**Este Software está diseñado para la placa Arduino Mini-Pro 5V, **
*****
*****
*/

```

ANEXO 02. PROGRAMACIÓN DEL ROBOT

```
/*
PROTOTIPO JED-PRO (JUGUETE ELECTRÓNICO DIDÁCTICO
PROGRAMABLE)
PROGRAMA DEL ROBOT
ELABORADO POR: EDUARDO CARTAGENA
FECHA: 14 DE JULIO DEL 2015
MODELO DE PLACA: ARDUINO MINI-PRO 5V
*/
//////////////////////////////////DECLARACIÓN DE VARIABLES//////////////////////////////////
const int trigger = 3;           // Conectamos a Trigger del Sensor Ultrasónico
const int echo = 4;             // Conectamos al echo del sensor Ultrasónico
const int MOTOR1_ADELANTE = 5; // Salida1 para el Motor1
const int MOTOR1_ATRAS = 6;     // Salida2 para el Motor1
const int MOTOR2_ADELANTE = 10; //Salida1 para el Motor2
const int MOTOR2_ATRAS = 11;    //Salida2 para el Motor2
float distance = 5;             //
const int EncoderIzquierdo = A0; // encoder izquierdo conectado al pin A0
const int EncoderDerecho = A1;  //encoder derecho conectado al pin A1
//////////////////////////////////
char vectorPrincipal[16];       //Vector que almacena todos los datos recibidos
char funcion[16];               //Almacena los datos del Ciclo de 2 repeticiones
char funcion2[16];              //Almacena los datos del Ciclo de 3 repeticiones
char funcion3[16];              //Almacena los datos de la condición de obstaculo
int i, j, k, z, l, x, m, n;    // Variables Globales
char dato;                       // Almacena temporalmente el dato recibido
int pos;                          // Posición del vector función
int pos2;                          // Posición del vector función2
int pos3;                          // Posición del vector función 3
char iniciar;                     // Variable Para Iniciar El Proceso

//////////////////////////////////
/*INICIALIZACIÓN DE FUNCIONES Y DECLARACIÓN DE PINES COMO ENTRADA
O SALIDA (SE EJECUTA UNA SOLA VEZ)*/
//////////////////////////////////

void setup() {
  Serial.begin(9600);           //Inicializa la comunicación serial
  pinMode(trigger, OUTPUT);     // Pin trigger como salida
  pinMode(echo, INPUT);        // Pin Echo como entrada
  pinMode(MOTOR1_ADELANTE, OUTPUT); // Pines de los Motores como salida
  pinMode(MOTOR2_ADELANTE, OUTPUT);
  pinMode(MOTOR1_ATRAS, OUTPUT);
  pinMode(MOTOR2_ATRAS, OUTPUT);
  delay(2000);
}
```

```

////////////////////////////////////SUB-RUTINAS////////////////////////////////////
////////////////////////////////////AVANCE ADELANTE////////////////////////////////////
void adelante(int speed, int steps) {
  int vL, pvL = 0;
  int vR, pvR = 0;
  int counterL = 0, counterR = 0;
  pvL = digitalRead(EncoderIzquierdo);
  pvR = digitalRead(EncoderDerecho);
  analogWrite(MOTOR1_ATRAS, speed);
  digitalWrite(MOTOR1_ADELANTE, LOW);
  analogWrite(MOTOR2_ADELANTE, speed);
  digitalWrite(MOTOR2_ATRAS, LOW);
  while (counterL <= steps) {
    if (counterL <= steps) {
      vL = digitalRead(EncoderIzquierdo);
      if (vL != pvL)
      {
        counterL++;
        pvL = vL;
        //Serial.println(counterL++);
      }
    }
    stop();
  }
}
////////////////////////////////////
////////////////////////////////////GIRO A LA IZQUIERDA////////////////////////////////////
void derecha(int speed, int steps) {
  int vL, pvL = 0;
  int vR, pvR = 0;
  int counterL = 0, counterR = 0;
  pvL = digitalRead(EncoderIzquierdo);
  pvR = digitalRead(EncoderDerecho);
  digitalWrite(MOTOR1_ADELANTE, LOW);
  analogWrite(MOTOR1_ATRAS, speed);
  digitalWrite(MOTOR2_ADELANTE, LOW);
  analogWrite(MOTOR2_ATRAS, speed);
  while (counterL <= steps ) {
    if (counterL <= steps) {
      vL = digitalRead(EncoderIzquierdo);
      if (vL != pvL)
      {
        counterL++;
        pvL = vL;
      }
    }
  }
  stop();
}

```

```

//////////////////////////////////SUB-RUTINAS//////////////////////////////////
//////////////////////////////////GIRO A LA IZQUIERDA//////////////////////////////////
void izquierda(int speed, int steps) {
  int vL, pvL = 0;
  int vR, pvR = 0;
  int counterL = 0, counterR = 0;
  pvL = digitalRead(EncoderIzquierdo);
  pvR = digitalRead(EncoderDerecho);
  analogWrite(MOTOR1_ADELANTE, speed);
  digitalWrite(MOTOR1_ATRAS, LOW);
  analogWrite(MOTOR2_ADELANTE, speed);
  digitalWrite(MOTOR2_ATRAS, LOW);
  while (counterR <= steps) {
    if (counterR <= steps) {
      vR = digitalRead(EncoderIzquierdo);
      if (vR != pvR)
      {
        counterR++;
        pvR = vR;
        // Serial.println(counterL++);
      }}
  }
  stop();
}
//////////////////////////////////
//////////////////////////////////DETENER MOTOR IZQUIERDO//////////////////////////////////
void stopizquierda() {
  digitalWrite(MOTOR1_ADELANTE, LOW);
  digitalWrite(MOTOR1_ATRAS, LOW);
}
//////////////////////////////////DETENER MOTOR DERECHO//////////////////////////////////
void stopderecha() {
  digitalWrite(MOTOR2_ADELANTE, LOW);
  digitalWrite(MOTOR2_ATRAS, LOW);
}
////////////////////////////////// DETENER LOS DOS MOTORES//////////////////////////////////
void stop() {
  stopizquierda();
  stopderecha();
}

```

```
//////////Metodo Para Almacenar Las Instrucciones del Ciclo De 3 Repeticiones//////////
```

```
void LeerCiclo3 ()  
{  
  for (j = 0; j <= i; j++) {  
    if (vectorPrincipal[j] == 'E' && vectorPrincipal[j + 1] == '{')  
    {  
      {  
        pos = j + 1;  
        while (vectorPrincipal[pos] != '}') {  
          funcion[k] = vectorPrincipal[pos];  
          delay(10);  
          vectorPrincipal[pos] = ' ';  
          delay(10);  
          pos++;  
          k++;  
        }  
      }  
    }  
  }  
}
```

```
//////////Método Para Almacenar Las Instrucciones del Ciclo De 2 Repeticiones//////////
```

```
void LeerCiclo2 ()  
{  
  for (j = 0; j <= i; j++) {  
    if (vectorPrincipal[j] == 'D' && vectorPrincipal[j + 1] == '{')  
    {  
      pos2 = j + 1;  
      while (vectorPrincipal[pos2] != '}') {  
        funcion2[m] = vectorPrincipal[pos2];  
        delay(10);  
        vectorPrincipal[pos2] = ' ';  
        delay(10);  
        pos2++;  
        m++;  
      }  
    }  
  }  
}
```

```
//////////
```

```

//////////Método Para Almacenar Las Instrucciones de la estructura condicional//////////
void Obstaculo ()
{
for (j = 0; j <= i; j++) {
if (vectorPrincipal[j] == 'G' && vectorPrincipal[j + 1] == '{')
{
pos3 = j + 1;
while (vectorPrincipal[pos3] != '}') {
funcion3[n] = vectorPrincipal[pos3];
delay(10);
vectorPrincipal[pos3] = ' ';
delay(10);
pos3++;
n++;
}} }
}
//////////INSTRUCCIONES//////////
void instrucciones(){
switch (dato) {
case 'A':
digitalWrite(11, HIGH);
adelante(128, 16);
break;

case 'C':
izquierda(128, 9);
break;

case 'B':
derecha(138, 7);
break;

default:
stop();
break;
}
}

//////////

```

```
////////////////////////////////////MÉTODO PRINCIPAL////////////////////////////////////
```

```
void loop() {  
  
    if (Serial.available() > 0) { //verifica si existe algún dato de entrada en la Cx Serial  
        char letra = Serial.read(); //Lee el dato recibido  
  
        // Verifica si es un dato permitido antes de almacenarlo  
        if (letra == 'A' || letra == 'B' || letra == 'C' || letra == 'E' || letra == 'D' || letra == 'G' || letra  
            == '{' || letra == '}' )  
            vectorPrincipal[i] = letra; // almacena el dato recibido en el vector principal  
            delay(100);  
            i++; // cambia de posición en el vector  
        }  
        // Si el Dato es '0' inicia con la ejecución de instrucciones  
        else {  
            if (letra == '0') // espera que llegue el dato de que indica el final para iniciar  
                {  
                    iniciar = '1';  
                }  
            if (iniciar == '1') { // si iniciar es igual a 1 inicia la ejecución de instrucciones  
                LeerCiclo3();  
                LeerCiclo2(); // Llama las sub-rutinas  
                Obstaculo();  
  
                // Lee los datos del vector principal para identificar solo instrucciones simples  
                for (z = 0; z < i; z++) {  
                    if (vectorPrincipal[z] != 'D' && vectorPrincipal[z] != 'E' && vectorPrincipal[z] != 'G' )  
                    {  
                        dato = vectorPrincipal[z];  
                        delay(10);  
                        instrucciones();  
                    }  
                    else // Si no es instrucción simple llama la subrutina que le corresponda  
                    {  
                        if (vectorPrincipal[z] == 'D') { // Ciclo de 2 repeticiones  
                            for (l = 0; l < 2; l++) {  
                                for (x = 0; x < k; x++) {  
                                    dato = funcion[x];  
                                    instrucciones();  
                                    delay(200);  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```

```

else {
    if (vectorPrincipal[z] == 'E') {
        for (l = 0; l < 3; l++) {
            for (x = 0; x < m; x++) {                // Ciclo de 3 repeticiones
                dato = funcion2[x];
                instrucciones();
                delay(200);
            }
        }
    }
    else
    {
        if (vectorPrincipal[z] == 'G') {            // Condición "SI hay Obstáculo"
            while (distance > 0.25) {
                dato = 'A';
                instrucciones();
                digitalWrite(trigger, LOW);
                delayMicroseconds(5);
                digitalWrite(trigger, HIGH);
                delayMicroseconds(10);
                digitalWrite(trigger, LOW);
                distance = pulseIn(echo, HIGH);
                distance = distance * 0.0001657; //Calculo de la distancia en funcion del tiempo
            };
            for (x = 0; x < n; x++) {                //
                dato = funcion3[x];
                instrucciones();
                delay(200);
            };}}}}
            i = 0; k = 0; m = 0; n = 0; distance = 5; iniciar = '0'; Reinicia contadores y variables
        }}

/*
*****
*****NOTA*****
**Este Software está diseñado para la placa Arduino Mini-Pro 5V, **
*****
*****
*/

```