

## CAPITULO 3

### DISEÑO DEL SOFTWARE

#### 3.1 INTRODUCCIÓN

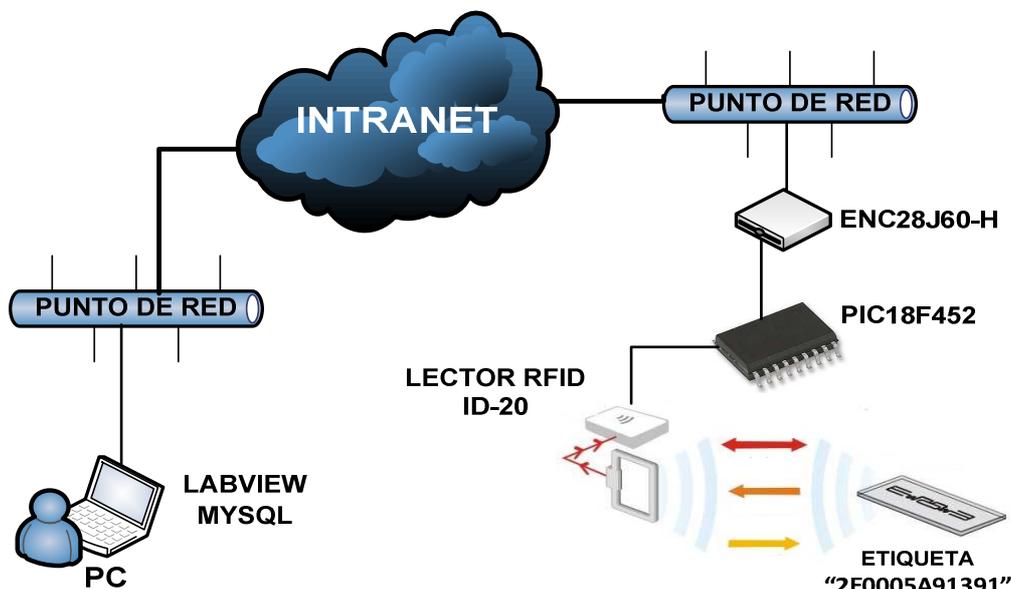
En este capítulo detallaremos el diseño general del control de personal para el colegio universitario “UTN”, mediante la utilización de un controlador ethernet y la tecnología RFID, en donde se especifica como interaccionan sus diferentes componentes entre sí. Además se explica la estructura del programa realizado. Se tiene por una parte la programación del microcontrolador y por otra, la programación del HMI<sup>46</sup> realizado en labview.

#### 3.2 DISEÑO GENERAL DEL SISTEMA DE CONTROL DE PERSONAL MEDIANTE LA UTILIZACIÓN DE UN CONTROLADOR ETHERNET Y LA TECNOLOGÍA DE IDENTIFICACIÓN POR RADIO FRECUENCIA

El presente proyecto tiene como finalidad realizar un control de personal para el colegio universitario “UTN”, que permita un registro de manera inalámbrica mediante el uso de tecnología RFID. Además un HMI elaborado en labview que admite el monitoreo del personal de forma remota.

---

<sup>46</sup> **HMI** Interfaz Hombre Maquina



**Figura 3.1** Diseño general del control de personal con tecnología RFID y controlador ethernet

Fuente: Darwin Marcelo Pillo G.

El proceso de control de personal empieza cuando la etiqueta RFID de tipo pasiva se acerca al lector ID-20, donde la etiqueta RFID elegida según las necesidades del sistema anteriormente analizadas, contiene un código único de identificación el cual permite reconocer a cada uno de los usuarios. Dicho código es transmitido al lector ID-20 mediante ondas de radiofrecuencia, posteriormente el lector ID-20 enviara el código único de identificación hacia el microcontrolador PIC18F452 de forma serial.

El código de cada etiqueta RFID es una cadena de caracteres ASCII de 12 byte con el formato siguiente:

$0 \leftarrow \text{-----} \rightarrow 11$   
 2E0005A91391

**Figura 3.2** Código único de identificación de una etiqueta pasiva RFID

Fuente: Darwin Marcelo Pillo G.

En el microcontrolador PIC18F452 se implementan algoritmos que permitan una correcta lectura y almacenamiento del código de identificación proveniente de la

etiqueta RFID. Como se menciona anteriormente el PIC18F452 y el lector RFID ID-20 se comunican mediante una interfaz serial a una velocidad de transmisión de 9600 baudios según consta en su hoja de datos.

Para la comunicación entre el microcontrolador PIC18F452 y la red ethernet, utilizamos el controlador ENC28J60-H, que nos permite acceder al protocolo TCP/IP. El método de conexión utilizado entre el microcontrolador y el ENC28J60-H es una interfaz de bus SPI el cual usa pocos pines del microcontrolador.

Los paquetes enviados desde el microcontrolador hacia la red ethernet son del tipo UDP, que es un protocolo de transporte no orientado a conexión. Lo cual contribuye a reducir el tráfico de la red. El puerto UDP elegido para el envío de la información es el número 200 que usualmente es utilizado para aplicaciones desarrolladas por IBM<sup>47</sup>, pero tomando en cuenta que en la red no existe aplicaciones IBM no hay ningún inconveniente en utilizar dicho puerto.

El modulo ENC28J60-H usa como medio de comunicación con la red ethernet del colegio universitario "UTN" un cable par trenzado a una velocidad de transmisión de hasta 10Mbits. A través del cual se envía el código único de identificación proveniente de la etiqueta RFID, almacenada en un paquete UDP. Dicho paquete es capturado por el HMI del sistema de control de personal almacenado en la computadora del administrador de personal.

Cuando el paquete UDP que contiene el código RFID llega correctamente al HMI, este realiza una búsqueda en la base de datos y comprueba si el usuario se encuentra o no registrado.

Si el usuario se encuentra registrado, el HMI envía un mensaje al microcontrolador indicando que se ha almacenado correctamente el código RFID, posteriormente el microcontrolador mediante la pantalla GLCD indica al usuario que su registro de ingreso o salida ha sido exitosamente realizado.

Cuando el usuario no se encuentra almacenado en la base de datos, el HMI envía un mensaje de error al microcontrolador, y mediante la pantalla GLCD se indica que el usuario no existe en la base de datos del sistema de control de personal.

---

<sup>47</sup> **IBM** International Business Machines

El HMI es una interfaz gráfica realizada en LabVIEW, el cual permite recibir los paquetes enviados desde el microcontrolador y almacenarlos en una base de datos elaborada en MySQL.

El sistema de control de personal utiliza la red ethernet para el envío del código RFID de cada usuario, pero en ocasiones debido a problemas de la red inalámbrica existe pérdida de conectividad entre el hardware del sistema de control de personal y el HMI. Para este caso el microcontrolador detecta que ya no tiene conectividad con el HMI del sistema de control de personal y almacena los códigos RFID de los alumnos maestros en la memoria EEPROM del microcontrolador.

Luego cuando se restablece la conectividad entre el hardware del sistema de control y el HMI, el administrador de personal mediante una etiqueta RFID indica al microcontrolador que envíe hacia el HMI, los códigos RFID de los alumnos maestros registrados cuando se perdió la conectividad. Los códigos RFID almacenados en la memoria EEPROM del microcontrolador se visualizan en la pantalla GLCD y en un subVI del software del sistema de control de personal.

El colegio universitario posee una red inalámbrica mediante el uso de un router D-Link DIR-600, dicha red tiene el nombre de ADMINTRACION\_UTN y permite tener el acceso remoto entre la computadora del administrador y el hardware del sistema de control de personal. Dicha red es utilizada solo para sistemas o programas de administración del colegio universitario. A continuación indicamos las características del router D-Link DIR-600:

- Tecnología de flujo 802.11n entregando velocidad de hasta 150 Mbps
- Cumple con estándares IEEE 802.11g/b y es compatible con 802.11n
- Soporta función WMM para satisfacer los requerimientos de banda ancha de datos multimedia
- Configuración Protegida Wi-Fi
- Cifrado de datos WEP y WPA/WPA2 (TKIP y AES)
- Compatible con Windows 7

- Switch de 4 puertos para incorporar a red dispositivos cableados



**Figura 3.3 Router inalámbrico D-Link DIR-600**

**Fuente:** <http://www.alamedacomputacion.cl/computacion/router-d-link-dir-600/>

Para evitar un ingreso no autorizado a la red ADMINTRACION\_UTN, el router D-Link DIR-600, solicita una contraseña para acceder a la red. En el anexo C se indica las configuraciones realizadas al router inalámbrico.

### **3.3 REQUERIMIENTOS DEL SOFTWARE**

Una vez conocido los antecedentes para el control de personal en el colegio universitario utilizando la tecnología RFID, a cada alumno maestro se le asigna una tarjeta RFID agilizando así el proceso de registro del personal. Además para brindar flexibilidad al sistema de control de personal utilizamos un controlador ethernet, que permitirá al inspector de área del colegio acceder al HMI del sistema de control de personal desde cualquier computador personal que utilice el estándar ethernet.

El sistema debe cumplir con lo siguiente:

- Ingreso de nuevos Usuarios
- Eliminar usuarios
- Visualizar la información del usuario con imagen incluida
- Registro del personal
- Consulta de atrasos
- Consulta de inasistencia o falta

- Recobrar códigos RFID almacenadas en memoria EEPROM del Microcontrolador

### **3.4 PROGRAMACIÓN DEL MICROCONTROLADOR**

Una vez detallados los requerimientos, que deben ser satisfechos en la programación del software para el sistema de control de personal. Se procede primero a explicar la programación del microcontrolador. Para la programación del microcontrolador se utilizó el software mikrobasic que utiliza un lenguaje de programación similar al BASIC. Donde una vez creado el código, este se compila y genera un archivo hexadecimal, el cual es descargado al microcontrolador.

Mikrobasic es un compilador distribuido por la empresa mikroelectrónica, el cual contiene librerías específicas para ser utilizadas con microcontroladores de microchip, como por ejemplo la utilización del puerto SPI, serial, I2C, PWM, ADC, etc.

Además de las librerías específicas, mikrobasic contiene librerías para comunicarse con otros dispositivos; como por ejemplo, teclados, memorias flash, pantallas de cristal líquido alfanuméricas, puerto ethernet SPI utilizando el ENC28J60, que es el integrado utilizado en la aplicación.

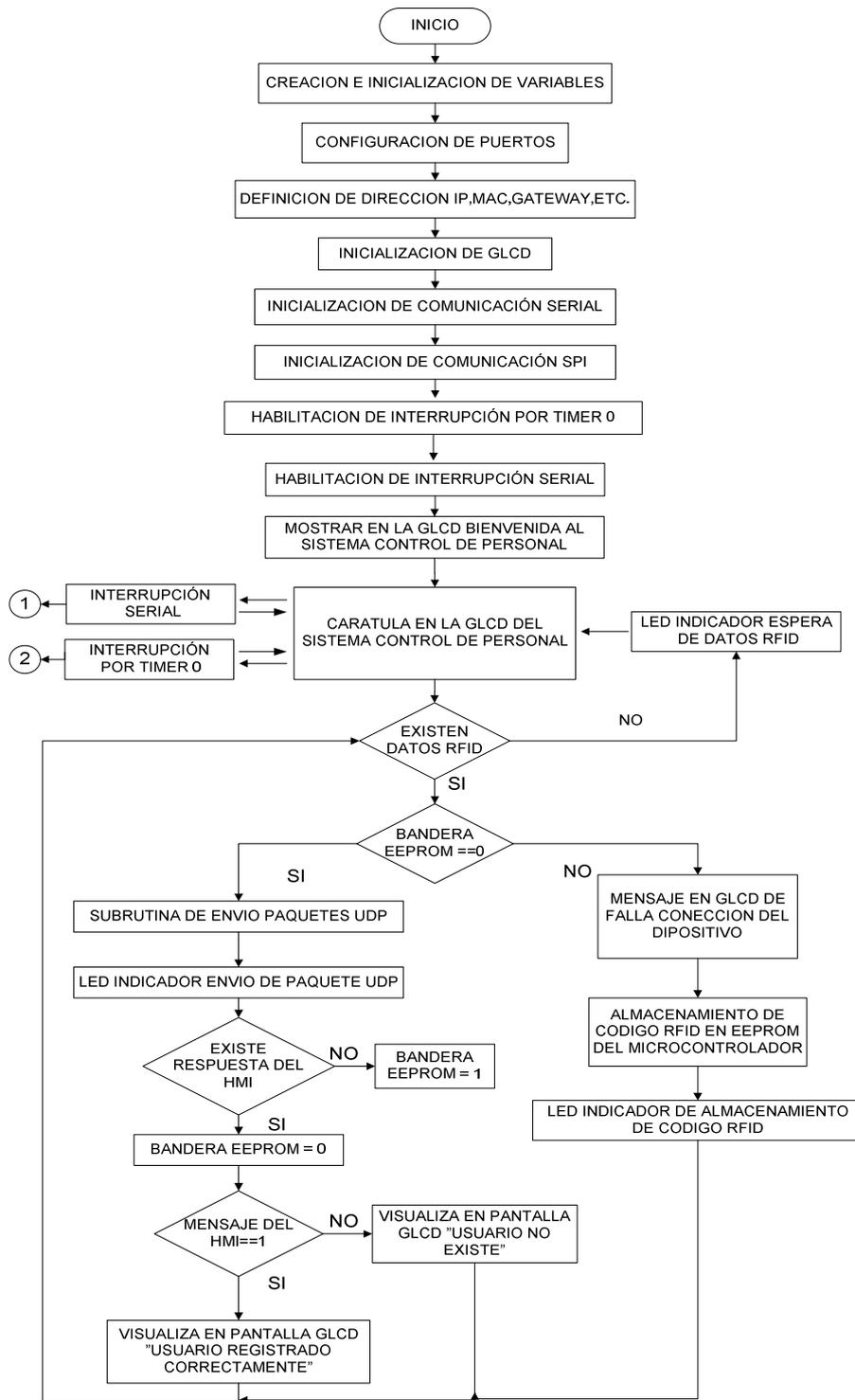
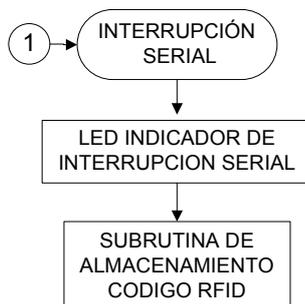


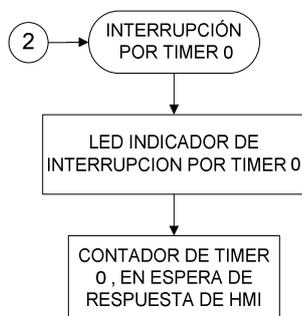
Figura 3.4 Diagrama de flujo del programa del microcontrolador

Fuente: Darwin Marcelo Pillo G.



**Figura 3.5 Diagrama de flujo de interrupción por comunicación serial**

Fuente: Darwin Marcelo Pillo G.



**Figura 3.6 Diagrama de flujo de interrupción por Timer 0**

Fuente: Darwin Marcelo Pillo G.

### 3.4.1 CONFIGURACIÓN DE PUERTOS

El microcontrolador PIC18F452 posee 5 puertos a través de los cuales se maneja los diferentes dispositivos del sistema de control de personal, según consta en la tabla 2.14. En general los puertos de un PIC de gama alta constan de tres registros para su operación.

- El registro TRIS, el cual controla la dirección de funcionamiento del puerto.
- El registro PORT, que permite leer y escribir en el puerto del microcontrolador.
- El registro LAT, el cual es el LATCH<sup>48</sup> de salida del puerto.

```

trisa=0           //configura el porta como salida
porta=0          //inicia el porta en cero
  
```

<sup>48</sup>**LATCH** Es un circuito electrónico usado para almacenar información en sistemas lógicos.

```

trise=0          //configura el porte como salida para leds indicadores
porte=0         //inicia el porte en cero
trisb=0         //configura el portb como salida para pantalla GLCD
portb=0        //inicia el portb en cero
trisd=0        //configura el portd como salida para pantalla GLCD
portd=0        //inicia el portd en cero
adcon1=%00001110 //Apaga el modulo de conversión analógico digital

```

### 3.4.2 DEFINICIÓN DE DIRECCIÓN IP, MAC, GATEWAY, ETC.

Para la conexión del microcontrolador con una red ethernet mediante el uso del modulo ENC28J60-H, es necesario agregar algunas características como son la dirección MAC, IP origen, DNS, GATEWAY, MASCARA, IP destino.

```

//Definición de la dirección física
mymacaddr[0]=0xA3
mymacaddr[1]=0x12
mymacaddr[2]=0x45
mymacaddr[3]=0xB3
mymacaddr[4]=0x22
mymacaddr[5]=0xF0
//Definición de la dirección IP del microcontrolador
myipaddr[0]=192
myipaddr[1]=168
myipaddr[2]=1
myipaddr[3]=10
//Definición de la máscara del microcontrolador
ipmask[0]=255
ipmask[1]=255
ipmask[2]=255
ipmask[3]=0
//Definición del DNS del microcontrolador
dnsipaddr[0]=192

```

```

dnsipaddr[1]=168
dnsipaddr[2]=1
dnsipaddr[3]=1
//Definición del Gateway del microcontrolador
gwipaddr[0]=192
gwipaddr[1]=168
gwipaddr[2]=1
gwipaddr[3]=1
//Definición de la IP destino
myipaddr2[0]=192
myipaddr2[1]=168
myipaddr2[2]=1
myipaddr2[3]=20

```

### 3.4.3 INICIALIZACIÓN E INTERRUPCIÓN DE COMUNICACIÓN SERIAL

El microcontrolador PIC18F452 tiene la posibilidad de comunicación serial mediante el puerto USART. Si se utiliza esta herramienta se puede configurar el sistema de forma que se obtenga interrupciones por recepción o transmisión serial.

Los parámetros de comunicación para el desarrollo del presente trabajo son:

- Comunicación asincrónica
- 9600 baudios
- 8 bits de datos
- 1 bit de parada
- Sin paridad

Como se mencionó anteriormente el compilador mikrobasic tiene varias librerías específicas como la USART Library, que permite configurar con pocas líneas de código la comunicación serial en el PIC, así también las interrupciones por recepción de datos vía serial.

Para la habilitación de la comunicación serial con los parámetros anteriormente planteados se utiliza:

```
USART_Init(9600)           //Inicializa la comunicación serial del microcontrolador
```

Como el microcontrolador se comunica vía serial con el ID-20, la lectura del código de la etiqueta RFID se realiza mediante la interrupción por recepción serial del PIC, a continuación se detalla los comandos utilizados para leer el código RFID vía comunicación serial.

```

sub procedure Interrupt()           //Función de Interrupción por recepción serial
  if (PIR1.RCIF=1)then             //Pregunta si a sucedido la interrupción
    while true                      //ciclo infinito
      if Usart_Data_Ready = 1 then   //Pregunta si existe datos en el canal serial
        received_byte[i3+1]= Usart_Read // Almacena código RFID
        i3=i3+1
        bandera13=1
        if i3=16 then               //pregunta si ya ha terminado de almacenar código RFID
          PORTE.1=1 //indicador que ha terminado de almacenar código RFID
          PORTE.2=0
          PIR1.RCIF=0               //Habilita de nuevo la interrupción serial
          i3=0
          delay_ms(50)              //Retardo
          break                      //Termina ciclo infinito
        end if
      end if
    wend
  end if                             //Termina la interrupción

```

#### **3.4.4 INICIALIZACIÓN E INTERRUPCIÓN POR TIMER 0**

El microcontrolador PIC18f452 dispone de un potente conjunto de temporizadores para manejar eficientemente todas las operaciones que involucran al tiempo y al contaje. Dichos temporizadores son tres y se denominan TMR0, TMR1 y TMR2.

El Timer0 es un temporizador/contador ascendente de 16 bits y cuando trabaja con el reloj del PIC se le suele llamar temporizador.

El temporizador del microcontrolador es el encargado de determinar el tiempo de espera máximo que tiene el hardware de control de personal, para recibir una respuesta del HMI. Si la respuesta o mensaje proveniente del HMI está dentro del tiempo de espera el sistema sigue funcionando de manera normal, pero si el mensaje proveniente del HMI no llega al hardware del sistema de control de personal, el microcontrolador a través del timer 0 pasa al modo de registro de personal con almacenamiento en memoria EEPROM.

```

sub procedure Interrupt()           //Función de Interrupción por Timer0
if ( intcon.TMR0IF=1 ) then        //Pregunta si a sucedido la interrupción de Timer0
  cnt = cnt+1                       //contador de timer
  if ( cnt =1500) then              //formando interval de tiempo
    cnt = 0 ' reset cnt            //Reiniciando contador de timer
    if(ban_timer=1)then //si no existe respuesta del HMI
      PORTE.0= PORTE.0 xor 1 //toggle led
      cont_timer=cont_timer+1 //incrementa contador de timer espera de respuesta
      if (cont_timer=6) then //formado intervalo de tiempo
        ban_eeprom=1 //cambia bandera y empieza almacenamiento en EEPROM
        cont_timer=0 //reinicia contador timer
      end if //fin de sentencia if
    end if //fin de sentencia if
  end if //fin de sentencia if
  TMR0L = 96 //reinicio de timer 0
  INTCON.TMR0IF=0 //habilita de nuevo interrupción por timer 0
  INTCON = $20
end if
end sub //termina interrupción por timer 0

```

### 3.4.5 BIENVENIDA Y CARATULA DEL SISTEMA CONTROL DE PERSONAL MEDIANTE GLCD

La pantalla GLCD permite visualizar los diferentes estados del sistema de control de personal, al inicio una bienvenida mediante una secuencia de imágenes almacenadas en el microcontrolador y luego una caratula de presentación

```

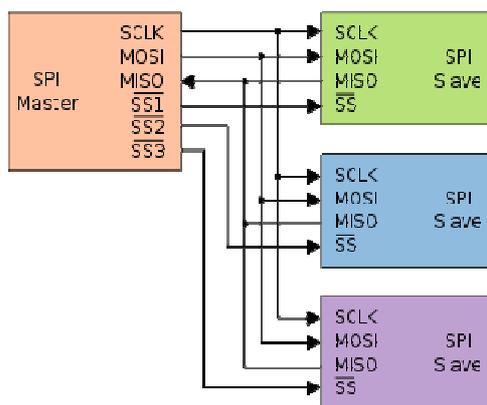
glcd_init(portb,0,1,2,3,5,4,portd) //Inicializa la pantalla GLCD
Glcd_Image(truck_bmp2) //Secuencia de imágenes de bienvenida al
delay_ms(50) //sistema de control de personal
Glcd_Image(truckt1_bmp) //Imagen 1
delay_ms(320) //retardo
Glcd_Image(truckt2_bmp) //Imagen 2
delay_ms(320) //retardo
Glcd_Image(truckt3_bmp) //Imagen 3
delay_ms(320) //retardo
Glcd_Image(truckt4_bmp) //Imagen 4
delay_ms(320) //retardo
Glcd_Image(truckt5_bmp) //Imagen 5
delay_ms(380) //retardo
Glcd_Image(truckt6_bmp) //Imagen 6
delay_ms(900) //retardo
Glcd_Image(truck_bmp2) //Imagen 7
delay_ms(50) //retardo
//////////////////////////////////// Caratula Presentación //////////////////////////////////////
Glcd_Write_Text("CONTROL DE PERSONAL ",10, 1, 1)
Glcd_Write_Text("COLEGIO UNIVERSITARIO",2, 2, 1)
Glcd_Write_Text("-UTN-",48, 3, 1)
Glcd_Write_Text("AUTOR: DARWIN PILLO G",2, 6, 1)

```

### 3.4.6 COMUNICACIÓN SPI [40]

El bus SPI (serial peripheral interface), es un estándar de comunicación de datos serial síncrono diseñado por Motorola que opera en modo full dúplex. Los

dispositivos conectados se comunican en modo maestro/esclavo y transmiten paquetes de información de 8 bits. El maestro es aquel que inicia la transferencia de la información sobre el bus y genera las señales de reloj y control, mientras que el esclavo es un dispositivo controlado por el maestro. Donde esclavo es controlado sobre el bus a través de una línea selectora llamada chip select o select slave.



**Figura 3.7 Comunicación SPI maestro/esclavo**

**Fuente:** [http://es.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface](http://es.wikipedia.org/wiki/Serial_Peripheral_Interface)

Para realizar la comunicación de este tipo, solamente se requieren tres pines ubicados en el puerto C del microcontrolador como se describe a continuación:

- RCD5/SD0 Salida de datos seriales
- RC4/SDI Entrada de datos seriales
- RC6/SCK Señal de reloj para sincronización

En el microcontrolador PIC18F452, el bus de comunicación SPI está asociado con 4 registros los mismos que son:

- SSPCON1 Registro de control
- SSPSTAT Registro de estado
- SSPBUF Buffer de transmisión o recepción
- SSPSR Registro de cambios

#### 3.4.6.1 Inicialización de comunicación SPI

Como se mencionó anteriormente mikrobasic posee una librería SPI-ethernet que sirve para activar el puerto SPI del microcontrolador e inicializar el modo de

operación del controlador ethernet (velocidad, modo de transmisión, dirección IP y MAC<sup>49</sup>).

```
spi_init()                //Inicialización del hardware SPI del PIC
spi_ethernet_init(portc,0,portc,1,mymacaddr,myipaddr,1) //Ponemos RC0 como
// Reset,RC1 Como el CS, indicamos la MAC y la IP del microcontrolador
spi_ethernet_confnetwork(ipmask,gwipaddr,dnsipaddr) //Configuramos los
parámetros de red del microcontrolador como son: dirección IP, máscara de subred,
DNS.
```

#### 3.4.6.2 Subrutina de Envío de Paquetes UDP

Una vez configurado el bus SPI mediante la ayuda de la librería SPI - Ethernet, se utiliza el comando Spi\_Ethernet\_sendUDP para enviar el código único de identificación RFID en un paquete UDP, de la siguiente forma:

```
spi_ethernet_sendudp(myipaddr2,200,200,@received_byte,14) //Envía
received_byte(contiene código RFID) a la dirección IP myipaddr2(IP destino) desde
el puerto UDP 200 al puerto UDP 200.
```

#### 3.4.6.3 Subrutina de Recepción de Paquetes UDP

La librería SPI-Ethernet mediante el comando spi\_ethernet\_getbyte(), permite recibir paquetes UDP y almacenarlo en una variable. El HMI del sistema de control de personal siempre que reciba un código RFID mediante un paquete UDP, envía una respuesta indicando si un usuario existe o no en la base de datos del sistema de control de personal. Dependiendo de la respuesta el microcontrolador indica mediante un mensaje en la pantalla GLCD si el usuario ha sido registrado correctamente, o caso contrario indicando que el usuario no existe.

```
If destport = 200 then // Indica el puerto UDP
for ii=0 to 15 //ciclo for para leer cadena UDP proveniente del HMI
getRequest[ii]=spi_ethernet_getbyte() // Almacenamiento de paquete UDP
Glcd_Write_Text("CONTROL DE PERSONAL ",10, 1, 1) //caratula GLCD
```

<sup>49</sup> **MAC** Media Access Control (Control de Acceso al Medio)

```

Glcd_Write_Text("COLEGIO UNIVERSITARIO",2, 2, 1) //caratula GLCD
FOR n3=1 TO 15 //ciclo for para visualización de mensaje del MHI
Glcd_Write_Char(getRequest[n3],p3, 4, 1) //visualiza mensaje HMI
  p3=p3+6 // incrementa variable
NEXT n3 //cierra estructura de ciclo for
p3=20 //reinicia variable
if (getRequest[0]="A")then //si respuesta del HMI es "A"
  Glcd_Write_Text("USUARIO NO EXISTE",15, 5, 1) //mensaje en GLCD
end if // termina ciclo if
if (getRequest[0]="B") then //si respuesta del HMI es "A"
  Glcd_Write_Text("CORRECTO",40, 5, 1) //mensaje en GLCD
end if // termina ciclo if
delay_ms(1000) //retardo de 1 segundo
Glcd_Write_Text(" ",2, 4, 1) //Borra pantalla GLCD
Glcd_Write_Text(" ",2, 5, 1) //Borra pantalla GLCD
ban_timer=0 //reinicia variable
cont_timer=0 //reinicia variable
end if // termina ciclo if

```

### 3.4.7 ALMACENAMIENTO DE CODIGO RFID EN MEMORIA EEPROM

En ocasiones es necesario guardar la información que se genera durante un proceso de una forma permanente, es decir, esos datos han de permanecer incluso cuando el sistema se desconecta de la alimentación. El microcontrolador PIC18F452 posee una memoria EEPROM de 256 bytes que permite el almacenamiento de códigos RFID cuando no existe conectividad entre el hardware y software del sistema de control de personal.

Al igual que la librería SPI-ethernet el compilador mikrobasic posee una librería EEPROM que nos permite almacenar y leer datos de la memoria EEPROM del PIC18F452.

A través de las rutinas EEPROM\_READ y EEPROM\_WRITE podemos leer y almacenar datos en la memoria EEPROM del PIC18F452.

```

If (bandera13=1) then          // Si no existe respuesta del HMI
    k3=15                      //Inicialización de variable
    PORTE.2=1                 //led indicador de modo almacenamiento en EEPROM
    Glcd_Write_Text("        ",1,5, 1)          //caratula GLCD
    Glcd_Write_Text(" LECTURA CORRECTA ",1, 6, 1) //caratula GLCD
    FOR j3=1 TO 13            //ciclo for
        Glcd_Write_Char(received_byte[j3],k3,7, 1) // código RFID en GLCD
        Eeprom_Write(20+t,received_byte[j3]) // Almacenamiento en EEPROM
        k3=k3+6                //Incremento variable para GLCD
        t=t+1                  //Incremento de posición de memoria
        Eeprom_Write(17,t)     //escritura de tamaño total de datos
        delay_ms(15)          //Retardo
    NEXT j3                    // Terminación ciclo For
    wordtostr(t,tStr)          //conversion a tipo string
    Glcd_Write_Text(tStr, 20, 5, 1) //visualización de tamaño total de EEPROM
    k3=20                      //reinicio de variable
    bandera13=0                //borra bandera para almacenamiento en EEPROM
    delay_ms(900)              //Retardo
    Glcd_Fill(0)               //Borrar GLCD
end if

PORTE.1=0                      //apaga led indicador
PORTE.2=0                      //apaga led indicador
end if

```

Cuando se restablece la conexión entre el hardware del sistema de control de personal y el HMI, el administrador mediante una etiqueta RFID recupera los datos almacenados en la memoria EEPROM y los envía hacia el HMI.

```

if(received_byte[j3+1]=Eeprom_Read(200+j3)) then // Compara con etiqueta
                                                    // RFID del administrador

    conta_id=conta_id+1 // incrementa contador de longitud de código RFID
    if(conta_id>=12) then // si longitud de código RFID es 12

```

```

Glcd_Fill(0)           //Borra pantalla GLCD
Glcd_Write_Text("- EEPROM READ -",15, 0, 1) //caratula en pantalla GLCD
x1=14                  //inicialización de variable
for x=1 to 15         //ciclo for
  teeprom_byte[x]=Eeprom_Read(x) //comparación con etiqueta de administrador
  Glcd_Write_char(teeprom_byte[x],x1,1, 1) //visualización de etiqueta RFID
  x1=x1+6              //Incrementa variable para pantalla GLCD
next x                //termina ciclo for
x1=15                 //Inicialización de variable
x2=2                  //Inicialización de variable
x3=0                  //Inicialización de variable
for x=36 to 110      //ciclo for para lectura de datos en EEPROM
  teeprom_byte[x]=Eeprom_Read(x) //Lectura de memoria EEPROM
  Glcd_Write_char(teeprom_byte[x],x1,x2, 1) //visualización de códigos RFID
almacenados en memoria EEPROM
  x1=x1+6              //Inicialización de variable
  x3=x3+1              //Inicialización de variable
  if(x3>=13)then      // Ciclo if para inicialización de variables
    x1=15              //Inicialización de variable
    x3=0               //Inicialización de variable
    x2=x2+1            //Inicialización de variable
  end if               //fin de ciclo if
  delay_ms(10)         //retardo
next x                //termina ciclo for
for x=1 to 2          // ciclo for para envio de etiquetas RFID hacia el HMI
  spi_ethernet_dopacket()//inicialización de comunicación ethernet
  spi_ethernet_sendudp(broadcast,200,200,@teeprom_byte,120) //envio de
etiquetas RFID hacia el HMI
next x                //Ciclo for
delay_ms(1000)        // retardo
conta_id=0            //Inicialización de variable

```

```

goto inicio2
end if          //fin de ciclo if
end if        //fin de ciclo if

```

### 3.5 PROGRAMACIÓN DEL HMI EN LABVIEW

El HMI se desarrolló en el paquete computacional LabVIEW 10.0, el cual permite un fácil manejo del protocolo TCP/IP y el posterior tratamiento de la información.

Para almacenar la información se utiliza el motor de base de datos MySQL, para lo cual se utiliza el toolkit de Labview Database Connectivity toolset. Que en conjunto con la librería IMAQ visión permiten crear un sistema completo para el control de personal con tecnología RFID y un controlador ethernet.

#### 3.5.1 DATABASE CONNECTIVITY TOOLSET DE LABVIEW [41]

El labview database connectivity toolset es un conjunto de herramientas que permite crear y eliminar tablas de una base de datos, así como insertar y obtener información de las mismas. Donde estas herramientas de labview son muy fáciles de manejar solo hay que saber interpretar sus entradas y salidas, además de tener conocimientos básicos de bases de datos.

Antes de usar las herramientas de labview database connectivity, tenemos que crear un conector ODBC<sup>50</sup>. Este conector sirve para enlazar labview y el motor de base de datos MySQL. Esto es necesario ya que el objeto open connection del database connectivity recibe un parámetro llamado connection information el cual justamente es el nombre que le pondremos a nuestro ODBC.



**Figura 3.8 Instalador mysql-connector-odbc-5.1.8-win32**

**Fuente:** Darwin Marcelo Pillo G.

<sup>50</sup> **ODBC** Open Data Base Connectivity (Conectividad Abierta de Bases de Datos).

Para este proyecto se utiliza el `mysql-connector-odbc-5.1.8-win32`, que es una nueva versión del controlador ODBC para base de datos en MySQL. En el anexo G y H se detalla la creación del conector ODBC y una breve descripción de las herramientas utilizadas para la creación del HMI en labview respectivamente.

### 3.5.2 TABLAS CREADAS EN MySQL

La base de datos del sistema de control de personal está desarrollada en MySQL utilizando el MySQL Query Browser, Donde " *MySQL Query Browser es una herramienta gráfica proporcionada por MySQL AB para crear, ejecutar, y optimizar consultas en un ambiente gráfico, donde el MySQL Administrator está diseñado para administrar el servidor MySQL*"[42].

Para el manejo de la información se han creado tres tablas:

- Usuarios
- Horarios
- Registros

#### 3.5.2.1 Tabla Usuarios

La tabla usuarios contiene la información de todos los usuarios registrados en el sistema de control de personal. Tiene los siguientes campos:

- RFID
- CI
- Nombre
- Apellido
- N\_foto
- ID\_Horario

A continuación se muestra el código necesario para la creación de la tabla usuarios y sus campos:

```

18 DROP TABLE IF EXISTS `ibarra_registros`,`usuarios`;//borrar tabla usuarios si existe
19 CREATE TABLE `ibarra_registros`,`usuarios` ( //crear tabla usuarios
20 `RFID` varchar(12) NOT NULL, //crear campo RFID
21 `CI` varchar(10) NOT NULL, //crear campo CEDULA IDENTIDAD
22 `NOMBRE` varchar(100) NOT NULL, //crear campo NOMBRE
23 `APELLIDO` varchar(100) DEFAULT NULL, //crear campo APELLIDO
24 `N_FOTO` varchar(100) DEFAULT NULL, //crear campo NOMBRE FOTO
25 `ID_Horario` int(3) NOT NULL, //crear campo ID. HORARIO
26 PRIMARY KEY (`RFID`,`CI`) //clave primaria RFID,CI
27 ) ENGINE=MyISAM DEFAULT CHARSET=latin1; //Indica motor de almacenamiento MYSQL

```

### Figura 3.9 Creación de tabla registros

Fuente: Darwin Marcelo Pillo G.

#### 3.5.2.2 Tabla Horarios

Contiene los horarios de los usuarios del sistema de control de personal, se almacena la hora de entrada y salida en el siguiente formato: HH: MM – HH: MM. Esta tabla contiene los siguientes campos:

- ID\_Horario
- Dia
- Hora\_Ingreso
- Hora\_salida

A continuación se muestra el código necesario para la creación de la tabla horarios y sus campos:

```

29 DROP TABLE IF EXISTS `ibarra_registros`.`horarios`;//borrar tabla horarios si existe
30 CREATE TABLE `ibarra_registros`.`horarios` ( //borrar tabla horarios si existe
31 `ID_horario` int(10) NOT NULL, //crear campo ID. HORARIO
32 `Dia` varchar(45) NOT NULL, //crear campo DIA
33 `H_ingreso` time NOT NULL, //crear campo HORA INGRESO
34 `H_salida` time NOT NULL //crear campo HORA SALIDA
35 ) ENGINE=InnoDB DEFAULT CHARSET=latin1; //Indica motor de almacenamiento MYSQL

```

### Figura 3.10 Creación de la tabla Horarios

Fuente: Darwin Marcelo Pillo G.

#### 3.5.2.3 Tabla Registros

Contiene información acerca de los usuarios válidos que se han registrado y utilizan el sistema de control de personal. Esta tabla contiene los siguientes campos:

- RFID
- Nombre
- Apellido
- Fecha
- Hora
- Ingreso
- Min\_in
- Salida

- Min\_Salida
- Falta

A continuación se muestra el código necesario para la creación de la tabla registros y sus campos:

```

4 DROP TABLE IF EXISTS `ibarra_registros`.`registros`; //borrar tabla registros si existe
5 CREATE TABLE `ibarra_registros`.`registros` ( //crear tabla registros
6   `RFID` varchar(14) NOT NULL, //crear campo RFID
7   `NOMBRE` varchar(100) NOT NULL, //crear campo NOMBRE
8   `APELLIDO` varchar(100) DEFAULT NULL, //crear campo APELLIDO
9   `FECHA` date NOT NULL, //crear campo FECHA
10  `HORA` time NOT NULL, //crear campo HORA
11  `INGRESO` int(1) unsigned DEFAULT '0', //crear campo INGRESO
12  `MIN_IN` int(10) DEFAULT '0', //crear campo MINUTO ENTRADA
13  `SALIDA` int(1) unsigned DEFAULT '0', //crear campo SALIDA
14  `MIN_SALIDA` int(10) DEFAULT '0', //crear campo MINUTO SALIDA
15  `FALTA` int(1) unsigned DEFAULT '0' //crear campo FALTA
16 ) ENGINE=MyISAM DEFAULT CHARSET=latin1; //Indica motor de almacenamiento MYSQL

```

**Figura 3.11 Creación de tabla registros**

**Fuente:** Elaborado por Darwin Marcelo Pillo G.

Las tablas utilizadas en la base de datos para el sistema de control de personal pueden ser accesibles al usuario mediante el programa MySQL Query Browser, pero la información no se maneja directamente, sino únicamente a través de labview.

### 3.5.3 MENU DE INICIO

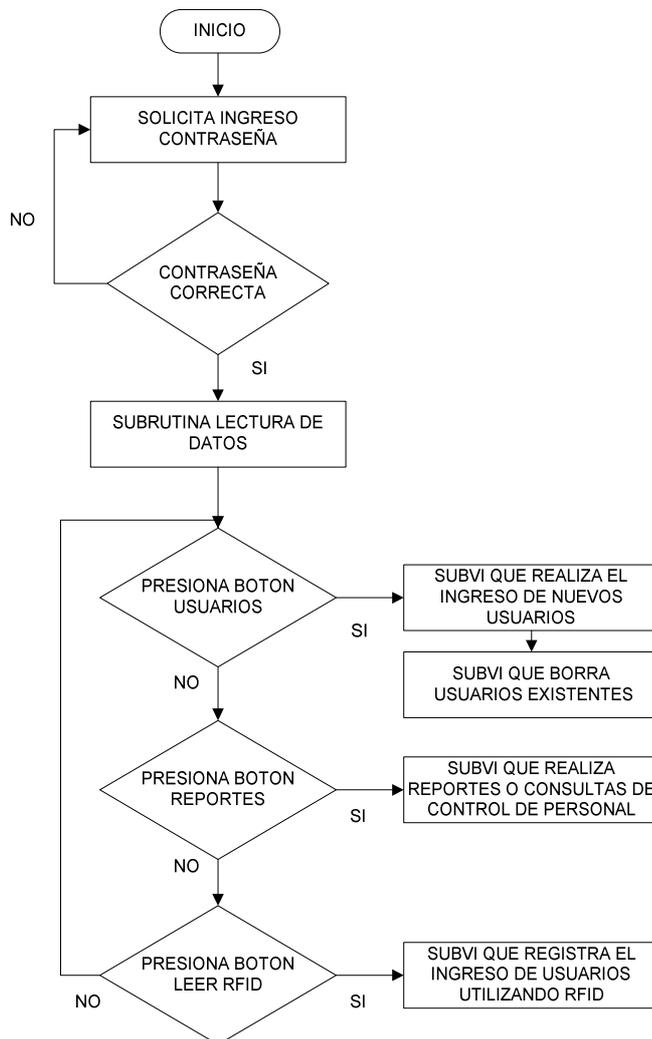
El menú de inicio es la pantalla principal del sistema de control de personal, mediante la utilización de un controlador ethernet y la tecnología de identificación por radio frecuencia.



**Figura 3.12 Menú de Inicio**

Fuente: Darwin Marcelo Pillo G.

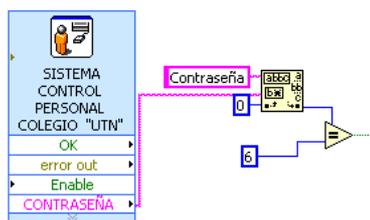
El menú inicio es el Vi que contiene los demás subVI que componen todo el HMI del sistema de control de personal como por ejemplo agregar usuarios, eliminar usuarios, reportes, etc. En la Figura 3.13 se puede apreciar la estructura del programa menú inicio realizado en Labview.



**Figura 3.13 Diagrama de Flujo del Programa en labview para el Menú Inicio**

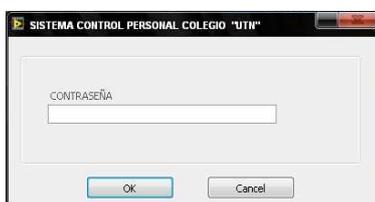
**Fuente:** Darwin Marcelo Pillo G.

El programa contiene una subrutina inicial que solicita la contraseña de administrador para evitar un ingreso no autorizado al sistema de control de personal, cuando la contraseña es correcta prosigue con la ejecución del programa, pero cuando la contraseña es incorrecta se indica un error a través de un mensaje como se indica en la figura 3.16 y sale de la ejecución del programa menú inicio.



**Figura 3.14** Subrutina solicitud de ingreso de contraseña

Fuente: Darwin Marcelo Pillo G.



**Figura 3.15** Solicitud de contraseña para ingreso al sistema

Fuente: Darwin Marcelo Pillo G.



**Figura 3.16** Clave incorrecta

Fuente: Darwin Marcelo Pillo G.

En el menú inicio cada botón de la interfaz está asociado a una nueva ventana o subVI, los cuales se explican en las siguientes secciones. La lectura de los botones se realiza mediante un *build array* el cual direcciona a los diferentes subVI que se encuentran almacenado dentro de un *case structure*. A continuación se muestra el diagrama en labview del programa realizado.

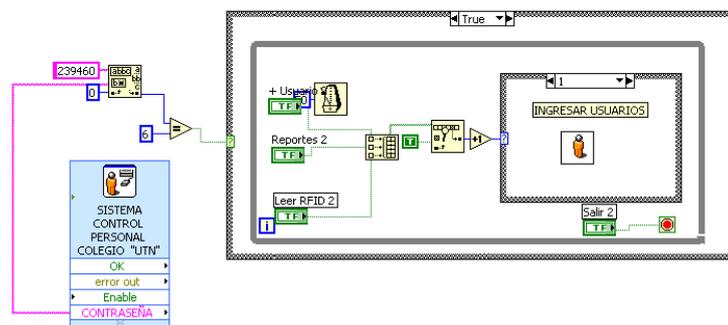


Figura 3.17 Diagrama de bloques del programa menú inicio

Fuente: Darwin Marcelo Pillo G.

### 3.5.4 INGRESO DE USUARIOS

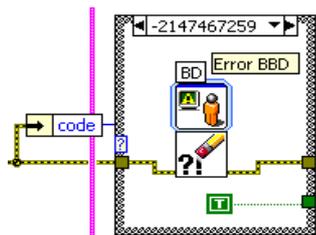
Esta pantalla permite crear o ingresar un nuevo usuario al sistema de control de personal para el colegio universitario UTN.



Figura 3.18 Ingreso nuevo usuario

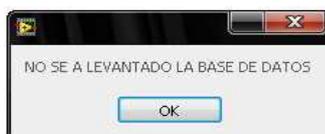
Fuente: Darwin Marcelo Pillo G.

Para el ingreso de la información a la base de datos del sistema de control de personal el inspector de área que está a cargo del los alumnos maestros, utiliza este programa para el ingreso del nombre, apellido, cedula, horario, foto y el código RFID asignado a cada usuario. En caso de que la base de datos no esté levantada el programa genera un error a través de un mensaje, el cual se indica en la figura 3.20.



**Figura 3.19 Subrutina de comprobación de la base de datos**

Fuente: Darwin Marcelo Pillo G.



**Figura 3.20 Mensaje no se ha levantado la base de datos**

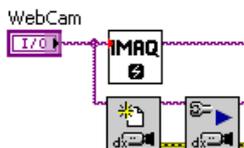
Fuente: Darwin Marcelo Pillo G.

Una de las partes más importantes del programa es el enlace entre la base de datos realizada en MySQL y Labview mediante la creación del conector ODBC como se indica en la figura 3.21. Así también se indica el inicio de sesión de la webcam que toma la imagen del nuevo usuario en dimensiones de 340x255 pixeles para un correcto funcionamiento del programa.



**Figura 3.21 Conexión de la Base de datos en MySQL y Labview**

Fuente: Darwin Marcelo Pillo G.

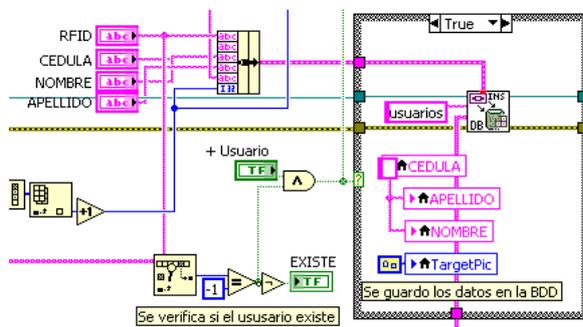


**Figura 3.22 Inicio de sesión de la webcam**

Fuente: Darwin Marcelo Pillo G.

La información personal del usuario es ingresada a través de un arreglo de string, los cuales son reunidos mediante un *bundle* y enviados a un *case structure*, el cual contiene una subrutina con *DB Tools Insert Data* que permite almacenar la información del usuario en la base de datos.

Al momento del envío de la información del usuario para ser almacenada en la base de datos, el programa realiza una búsqueda o comprobación de si el usuario ya existe dentro de la base de datos. Si el usuario ya existe el programa enciende un led indicador, tal como se indica en la figura 3.24.



**Figura 3.23 Subrutina de ingreso y verificación de nuevo usuario**

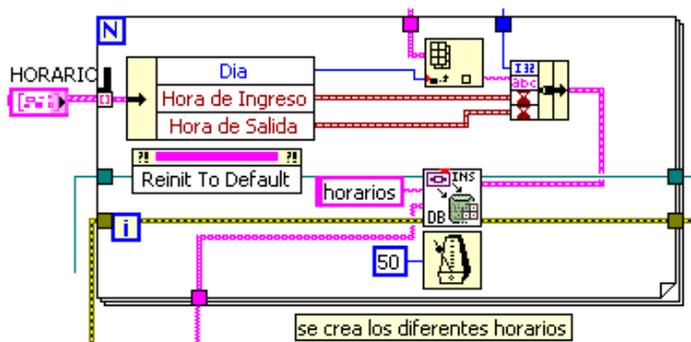
Fuente: Darwin Marcelo Pillo G.



**Figura 3.24 Led Indicador de la existencia de un usuario**

Fuente: Darwin Marcelo Pillo G.

El horario de cada alumno maestro es almacenado en la base de datos del sistema de control de personal, mediante la utilización de un *cluster*, el cual permite contener un conjunto de datos de diferentes tipos. Tal como se muestra en la figura 3.25.



**Figura 3.25 Subrutina de almacenamiento del horario en la base de datos**

Fuente: Darwin Marcelo Pillo G.

El programa permite almacenar la imagen del usuario de dos formas: la primera a través de una webcam que posee el computador, las especificaciones de la cámara no son especiales por tal motivo para el desarrollo del presente proyecto se utiliza la webcam Genius Eye 312, la cual posee las siguientes características:

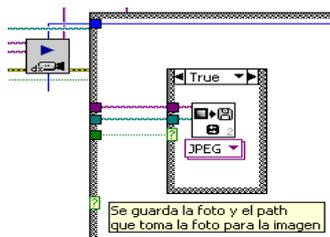
- 1,3 Megapíxeles (imagen fija).
- Resolución VGA 1280x960.
- Micrófono integrado.
- Una base clip multifuncional
- Inclina y gira hasta 360 grados.



**Figura 3.26 webcam Genius Eye 312**

Fuente: [http://mke.com.co/product.php?id\\_product=661](http://mke.com.co/product.php?id_product=661)

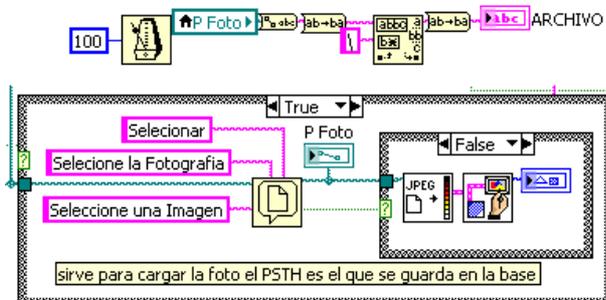
Para almacenar la imagen tomada por la webcam, se utiliza la herramienta *IMAQdx Grab* que permite adquirir la imagen del usuario, posteriormente mediante el *IMAQ Write File* se escribe la imagen en un archivo en formato BMP<sup>51</sup>.



**Figura 3.27 Subrutina para almacenar la imagen del usuario mediante webcam**

Fuente: Darwin Marcelo Pillo G.

En la segunda forma el programa permite buscar la imagen del usuario en un directorio específico, mediante el icono *File Dialog Express* se muestra un cuadro de diálogo que permite seleccionar un archivo que contiene la imagen de un usuario del sistema de control de personal.

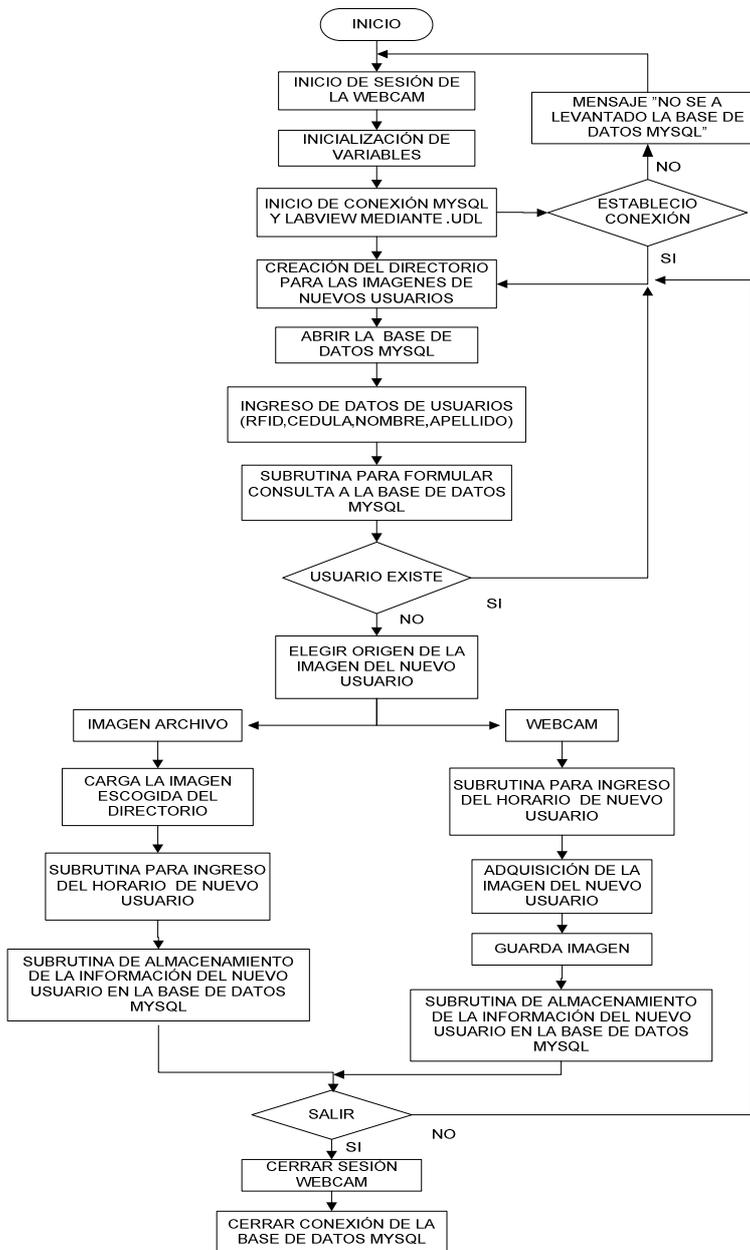


**Figura 3.28 Subrutina que permite seleccionar la imagen del usuario de un directorio específico**

Fuente: Darwin Marcelo Pillo G.

En la Figura 3.29 se puede apreciar la estructura del programa ingresar usuarios realizado en Labview.

<sup>51</sup> **BMP** Bit Mapped Picture (Mapa de Bits).



**Figura 3.29 Diagrama de Flujo del Programa en labview para el Menú Inicio**

Fuente: Darwin Marcelo Pillo G.

A continuación en la figura 3.30 se muestra el diagrama completo en labview del programa realizado.

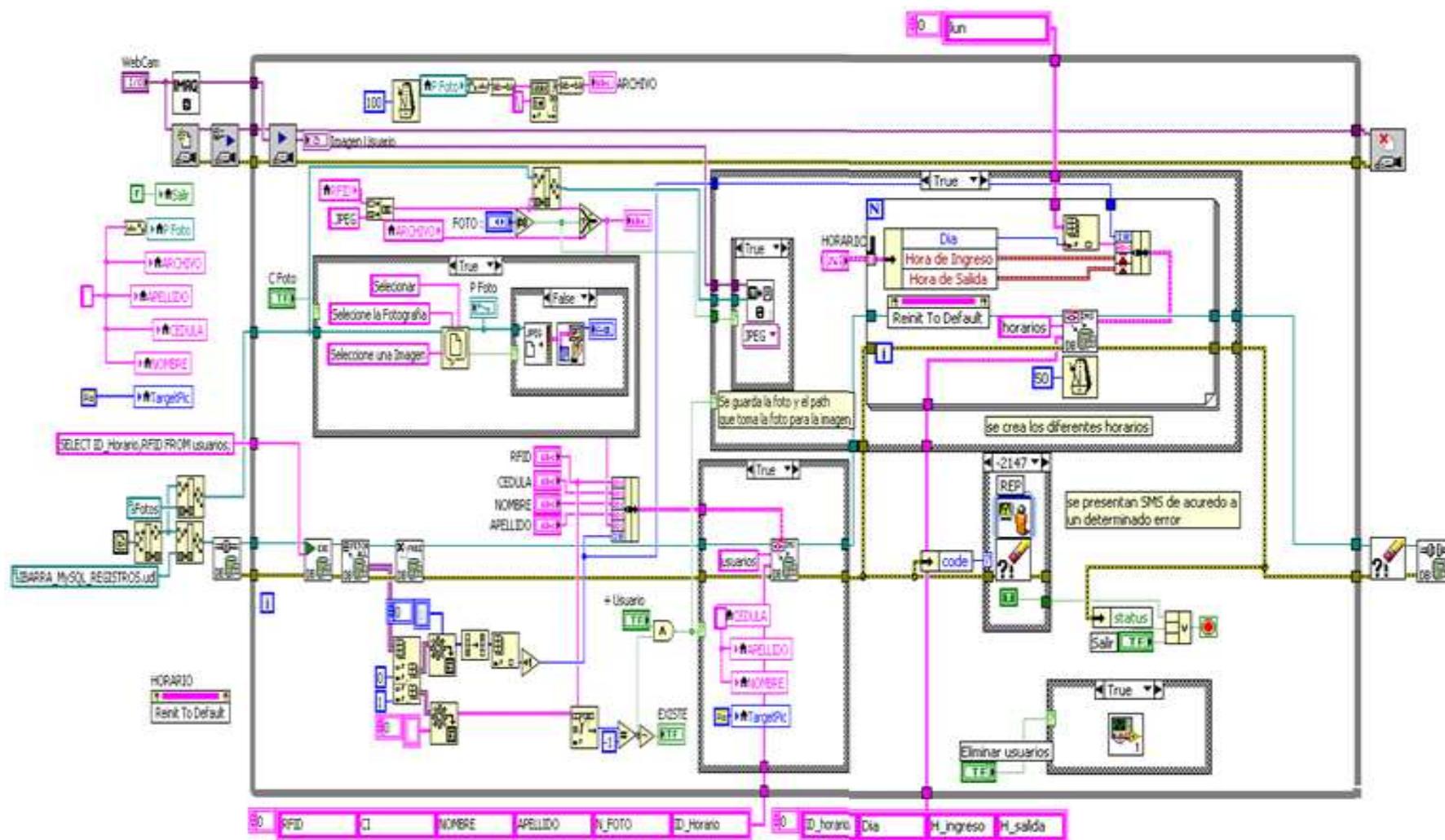
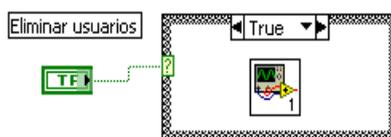


Figura 3.30 Diagrama de bloques del programa ingresar usuario

Fuente: Darwin Marcelo Pillo G.

El programa de ingresar usuarios posee un subVI que permite borrar usuarios que ya no vayan a usar el sistema de control de personal.



**Figura 3.31 SubVI Borrar usuarios**

Fuente: Darwin Marcelo Pillo G.

### 3.5.5 BORRAR USUARIOS

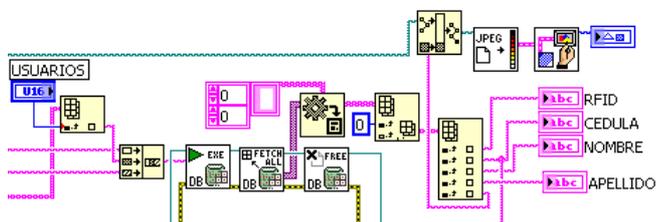
Este programa permite borrar usuarios que estén previamente almacenados en la base de datos, el usuario que se desee borrar se escoge de lista que aparece al presionar el botón usuarios que aparece en la interfaz , el cual contiene todos los usuarios almacenados en la base de datos.



**Figura 3.32 Borrar usuarios**

Fuente: Darwin Marcelo Pillo G.

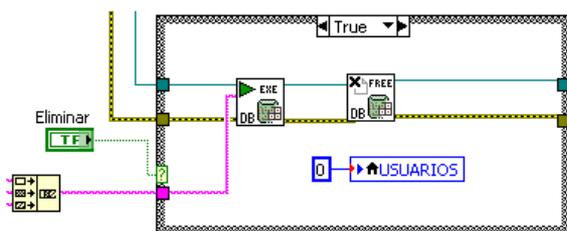
Mediante el botón usuarios se puede visualizar a todos los alumnos maestros almacenados dentro de la base de datos, a través de este botón se puede elegir al usuario que se desee eliminar. El icono *DB Tools Execute Query* permite realizar una consulta a la base de datos y mediante *DB Tools Fetch Recordset Data* se libera la información de los usuarios que se encuentren almacenados en la base de datos. Además, utilizando el icono *Read JPEG File* se importa la imagen correspondiente a cada usuario del sistema de control de personal, Con esto cuando se elimine a un usuario específico se elimina su imagen y toda su información personal.



**Figura 3.33 Subrutina de búsqueda de todos los usuarios almacenado en la base de datos**

Fuente: Darwin Marcelo Pillo G.

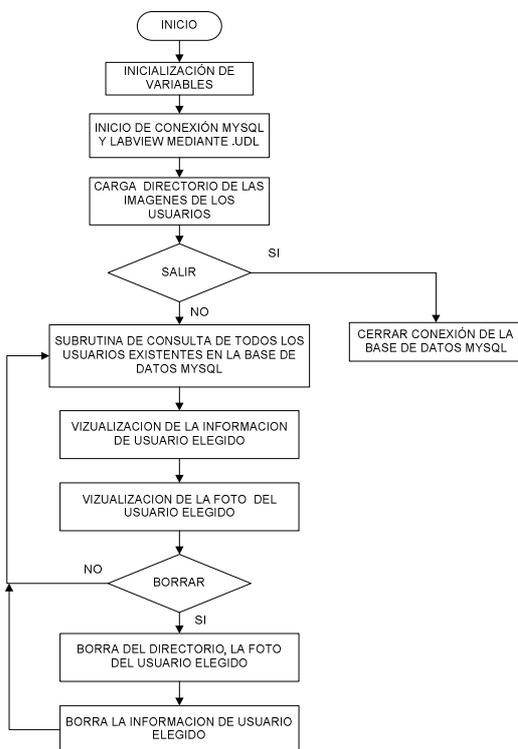
Luego de haber elegido al usuario que se desea eliminar, mediante el uso de los iconos *DB Tools Execute Query* y *DB Tools Free Object* dentro de un *case structure* se elimina al usuario de la base de datos.



**Figura 3.34 Subrutina que permite eliminar un usuario**

Fuente: Darwin Marcelo Pillo G.

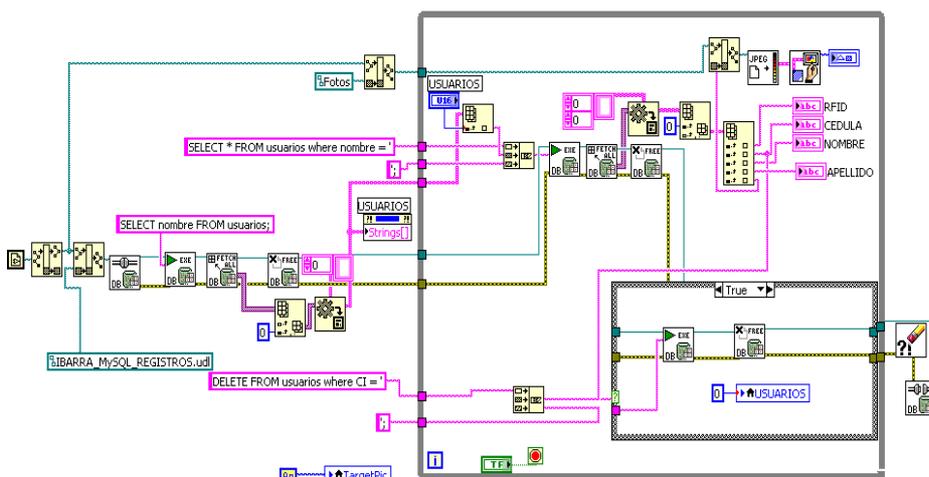
En la Figura 3.35 se puede apreciar la estructura del programa borrar usuarios realizado en Labview.



**Figura 3.35 Diagrama de Flujo del Programa en labview para borrar usuarios**

Fuente: Darwin Marcelo Pillo G.

A continuación en la figura 3.36 se muestra el diagrama completo en labview del programa realizado.



**Figura 3.36 Diagrama de bloques del programa borrar usuarios**

Fuente: Darwin Marcelo Pillo G.

### 3.5.6 REGISTRO DE USUARIOS

El programa de registro de usuarios nos permitirá adquirir y observar en la pantalla los datos que son enviados desde el lector RFID a la PC, mediante el uso del controlador ethernet.



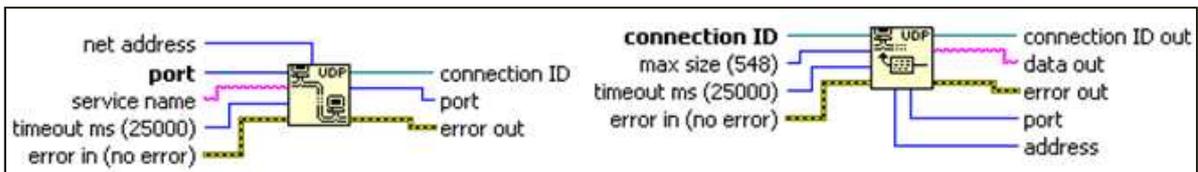
**Figura 3.37 Registro de Usuarios**

Fuente: Darwin Marcelo Pillo G.

Los datos enviados desde el hardware del sistema de control de personal hacia el HMI realizado en labview, se lo hace a través del protocolo UDP. Para esta configuración labview tiene implementada funciones para crear aplicaciones que usen TCP o UDP.

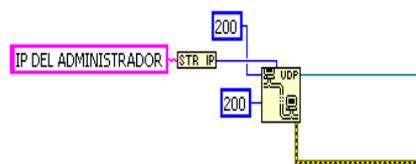
Para la configuración del puerto UDP se hace uso de los funciones *UDP Open* y *UDP read*. Donde la función *UDP open* permite abrir una conexión UDP con otro host, y que como parámetros de entrada debe tener el puerto UDP de destino. Los parámetros de entrada opcionales son la dirección IP de destino, el puerto local y el tiempo de espera para confirmación del cliente.

La función *UDP read* permite leer los datos que se encuentre en el puerto UDP local ya especificado, para indicar cuantos datos se deben leer se usa el parámetro max size. En la figura 3.38 se muestra las dos funciones necesarias para la comunicación con el protocolo UDP.



**Figura 3.38 Funciones UDP**

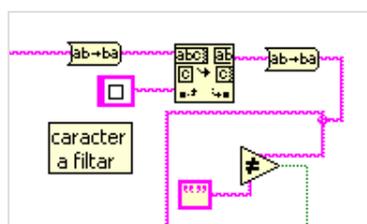
**Fuente:** [http://zone.ni.com/reference/en-XX/help/371361E-01/lvcomm/udp\\_vi\\_descriptions/](http://zone.ni.com/reference/en-XX/help/371361E-01/lvcomm/udp_vi_descriptions/)



**Figura 3.39 Subrutina de abrir una conexión UDP con otro host**

**Fuente:** Darwin Marcelo Pillo G.

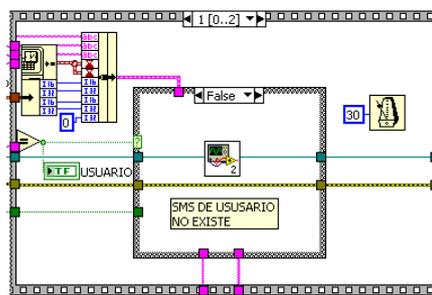
Debido a interferencias del cable de red, componentes electrónicos o el medio ambiente en el que se encuentra el sistema, provoca que se introduzcan caracteres que no pertenecen al código de la etiqueta RFID, por tal motivo el programa de registro de usuarios contiene una subrutina que permite filtrar los datos producidos por dichas interferencias.



**Figura 3.40 Filtrado de Datos**

**Fuente:** Darwin Marcelo Pillo G.

Cuando el lector RFID detecta una etiqueta dentro de su rango de lectura, el hardware del sistema de control de personal envía en paquetes UDP el código único de identificación, hacia la computadora de administrador de personal. Antes de almacenar la hora de entrada o salida conjuntamente con el código RFID, se realiza una consulta a la base de datos para determinar si el usuario trabaja ese día o para verificar si el usuario está almacenado en la base de datos del sistema de control de personal.



**Figura 3.41 Subrutina de comprobación de la existencia de un usuario**

Fuente: Darwin Marcelo Pillo G.

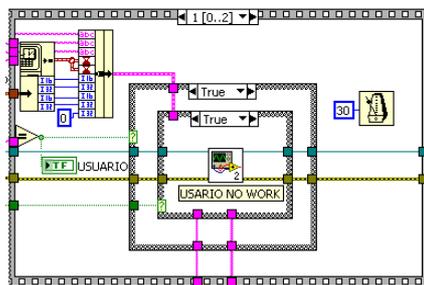
Cuando el sistema detecta que el usuario no se encuentra almacenado en la base de datos de los alumnos maestros, se produce un mensaje tanto para el microcontrolador como para el HMI indicando que el usuario no existe, como se indica en la figura 3.42.



**Figura 3.42 Mensaje cuando usuario no existe**

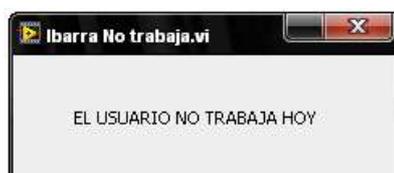
Fuente: Darwin Marcelo Pillo G.

Pero cuando el sistema detecta que el usuario no trabaja ese día el programa produce un mensaje indicando que el alumno maestro no trabaja hoy. Tal como se indica en la figura 3.44.



**Figura 3.43 Subrutina de comprobación cuando usuario no trabaja ese día**

Fuente: Darwin Marcelo Pillo G.



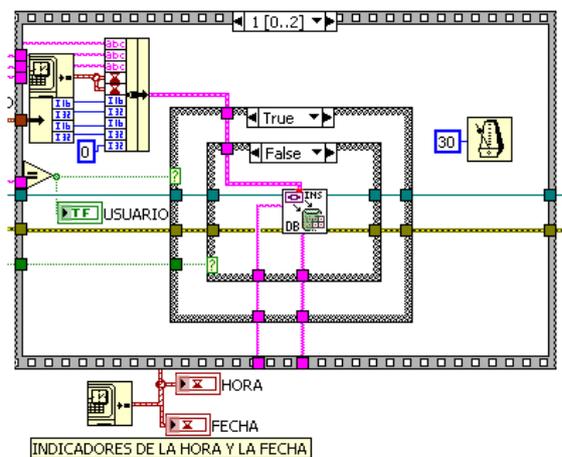
**Figura 3.44 Mensaje cuando el usuario no trabaja hoy**

Fuente: Darwin Marcelo Pillo G.

El programa de registro de usuarios contiene dos partes importantes, la primera es cuando el programa está en modo de solo registro de personal como se muestra en la figura 3.48 y la segunda es cuando se termina la jornada laboral, la interfaz realiza una consulta a la base de datos y asigna falta o inasistencia a los alumnos maestros que se no han registrado ese día. Tal como se muestra en la figura 3.49.-

En el modo registro cuando se ha comprobado que el usuario si existe y si trabaja ese día, el programa procede a almacenar el código RFID y la hora de ingreso o salida del alumno maestro mediante el uso del icono *DB Tools Insert Data*. La hora almacenada en la base de datos es la misma hora que se tiene en la computadora del administrador de personal esto se realiza utilizando el icono *Get Date/Time In Seconds Function*.

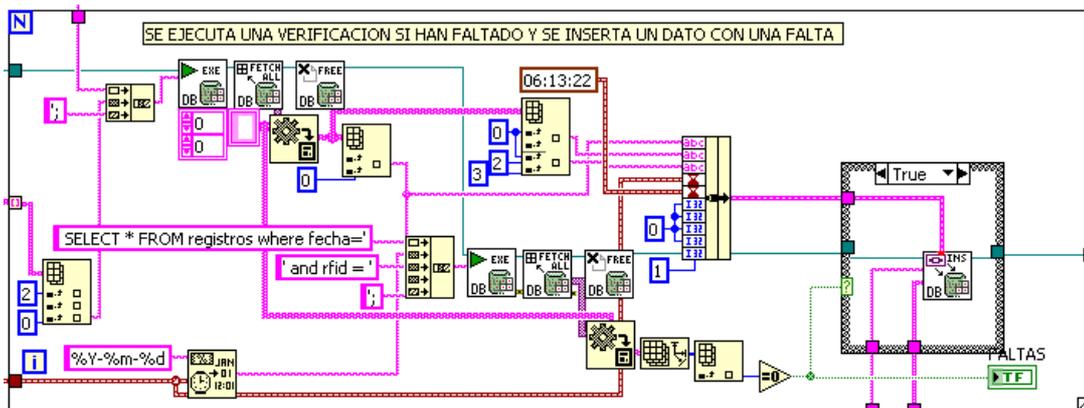
Cuando se almacena correctamente toda la información del registro de un usuario, en la pantalla del subVI de registro de personal se visualiza toda la información del usuario como es: nombre, apellido, cedula, horario, foto y el código RFID.



**Figura 3.45 Subrutina de registro de la hora de ingreso y salida del código RFID**

Fuente: Darwin Marcelo Pillo G.

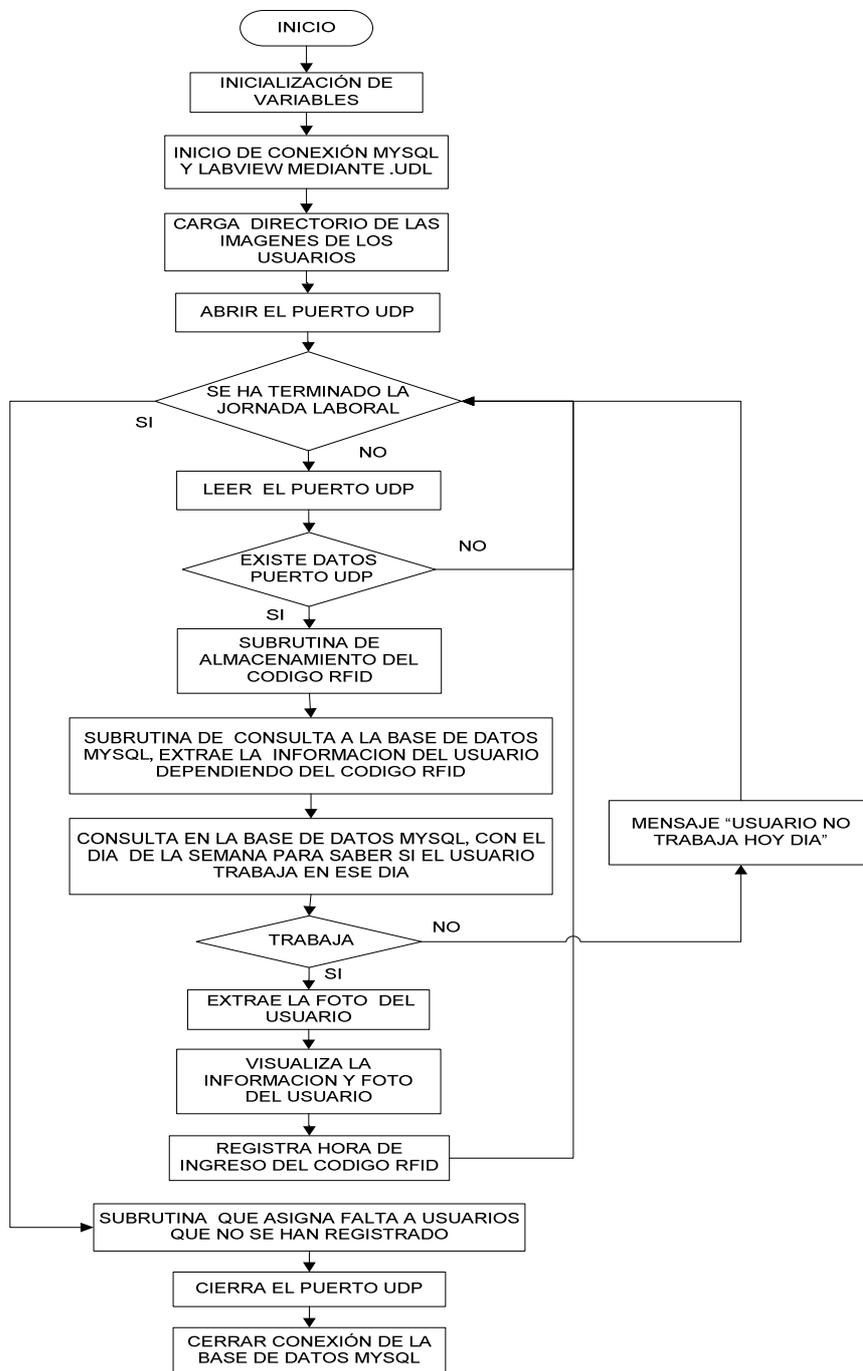
Cuando se termina la jornada laboral, se realiza una consulta a la base de datos de los usuarios que no se han registrado ese día, y mediante el uso del icono *DB Tools Insert Data* se ingresa una falta a dichos usuarios, esto nos servirá en el subVI de consultas para generar los reportes de inasistencia.



**Figura 3.46 Subrutina que asigna falta a usuarios que no se han registrado**

Fuente: Elaborado por Darwin Marcelo Pillo G.

En la Figura 3.47 se puede apreciar la estructura del programa registrar usuarios realizado en Labview.



**Figura 3.47 Diagrama de flujo del programa en labview para el registro de usuarios**

Fuente: Darwin Marcelo Pillo G.

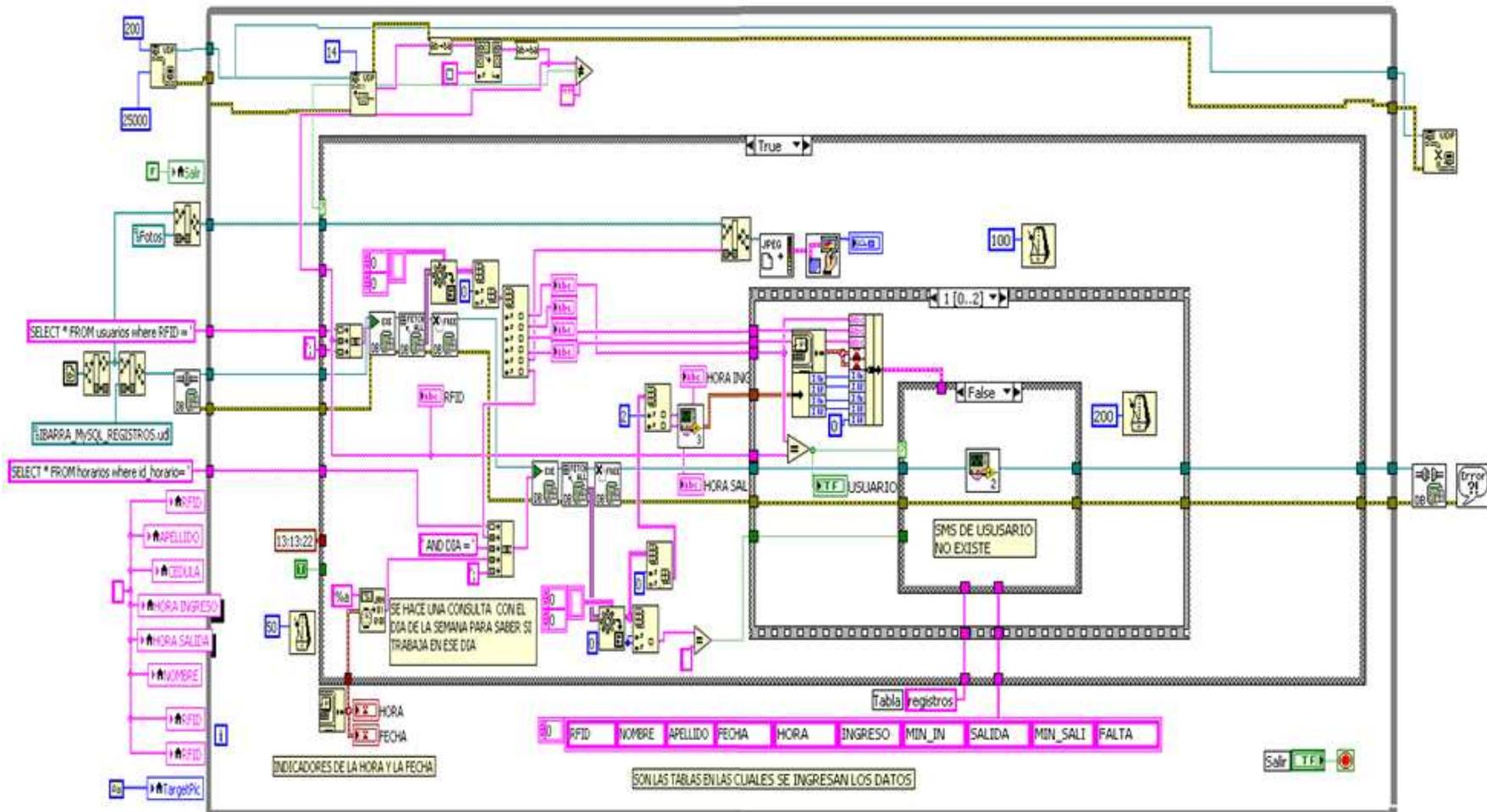


Figura 3.48 Diagrama de bloques 1 del programa registro usuarios

Fuente: Darwin Marcelo Pillo G.

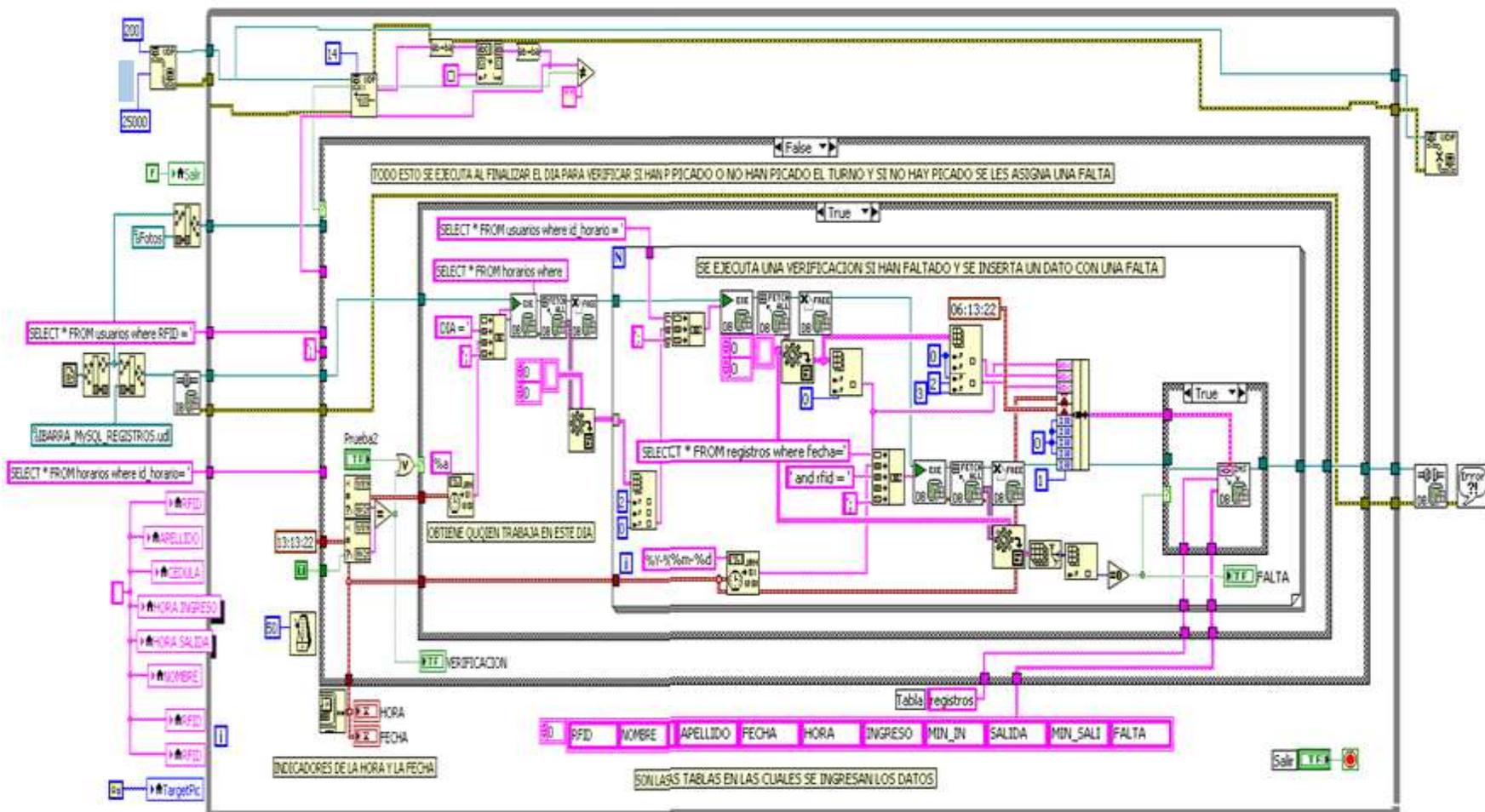


Figura 3.49 Diagrama de bloques 2 del programa registro usuarios

Fuente: Darwin Marcelo Pillo G.

### 3.5.7 REPORTE DE USUARIOS

Este programa permite al inspector general tener reportes de atrasos, adelantos, faltas de los alumnos maestros que usan el sistema de control de personal.

**REPORTES DE USUARIOS**

USUARIOS: [Input Field] RFID: [Input Field]

INICIO: 01/04/2011 [Calendar Icon]

07/04/2011 [Calendar Icon]

COMPLETO  
 ADELANTOS  
 ATRASOS  
 FALTAS

CEDULA: [Input Field]  
 NOMBRE: [Input Field]  
 APELLIDO: [Input Field]

CONSULTAR

RFID	FECHA	HORA	NOMBRE	APELLIDO	SALIDA	MIN_ADEL	ENTRADA	MIN	FALTA
27008AF7603A	19/03/2011	17:35:00							
27008AF7603A	19/03/2011	17:35:04							
27008AF7603A	19/03/2011	17:35:04							
27008AF7603A	19/03/2011	17:35:04							

HORARIOS

Dia: [Input] Hora Ingreso: [Input] Hora Salida: [Input]

LUNES

Dia: [Input] Hora Ingreso: [Input] Hora Salida: [Input]

LUNES

Dia: [Input] Hora Ingreso: [Input] Hora Salida: [Input]

LUNES

Dia: [Input] Hora Ingreso: [Input] Hora Salida: [Input]

LUNES

Salir

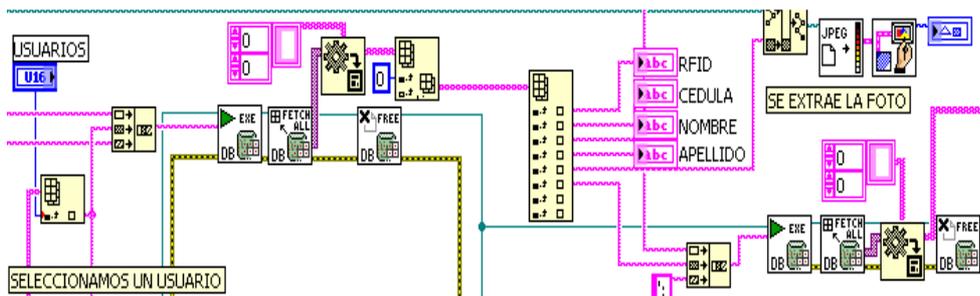
AUTOR: DARWIN MARCELO PILLO G.

**Figura 3.50 Reporte de Usuarios**

Fuente: Darwin Marcelo Pillo G.

Los reportes de cada usuario se pueden visualizar mediante un arreglo de string el cual contiene el código RFID, nombre, apellido, fecha, hora, faltas, etc. Al igual que en la interfaz ingresar usuarios el programa reportes nos indica la foto del usuario del cual estamos generado el reporte.

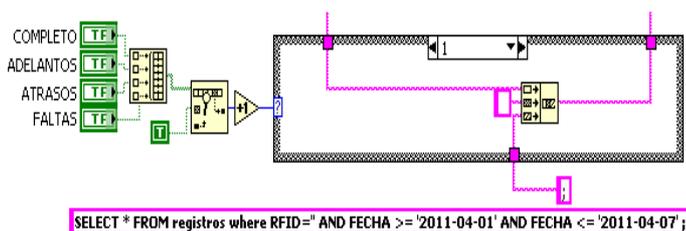
La lista de usuarios del los cuales podemos generar una consulta se puede elegir a través de un *text ring* llamado usuarios, el cual contiene todos los usuarios validos almacenados en la base de datos, para esto se realiza una consulta a la base de datos mediante el uso de los iconos *DB Tools Execute Query*, *DB Tools Fetch Recordset Data* y *DB Tools Free Object* los cuales ya hemos detallado anteriormente.



**Figura 3.51 Subrutina consulta todos los usuarios existentes**

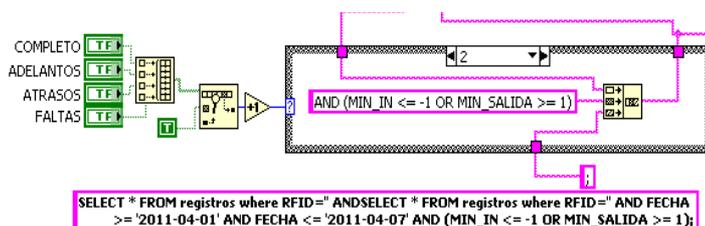
Fuente: Darwin Marcelo Pillo G.

En muchas ocasiones no es necesario generar todos los reportes, por lo tanto mediante *boolean text* podemos elegir entre adelantos, atrasos, faltas o un reporte completo. Además la interfaz reportes a través de un *time stamp* nos permite indicar la fecha desde la cual deseamos generar el reporte o consulta.



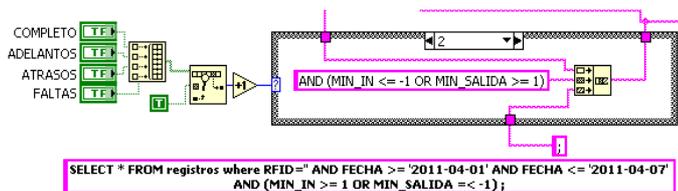
**Figura 3.52 Subrutina de bloques para consulta completa**

Fuente: Darwin Marcelo Pillo G.



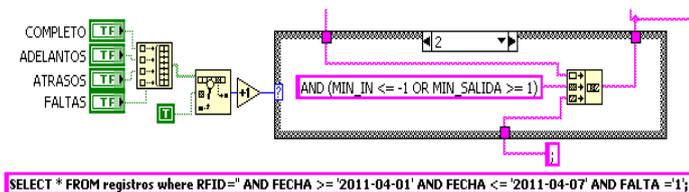
**Figura 3.53 Subrutina de bloques para consulta de adelantos**

Fuente: Darwin Marcelo Pillo G.



**Figura 3.54 Subrutina de bloques para consulta de atrasos**

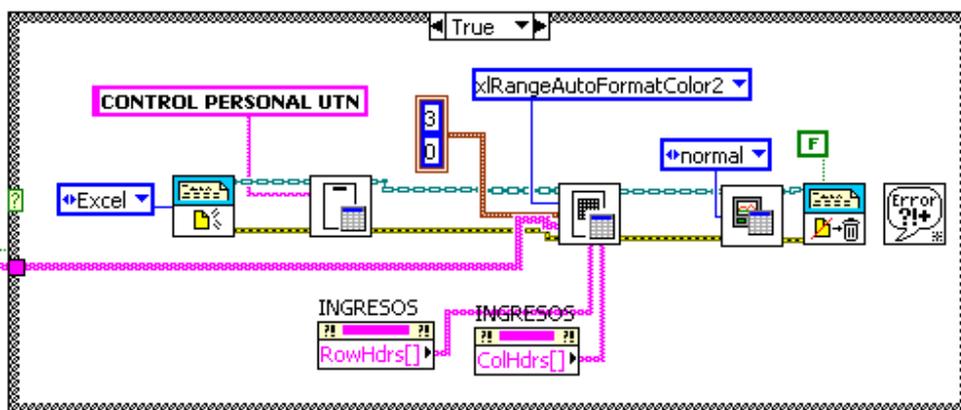
Fuente: Darwin Marcelo Pillo G.



**Figura 3.55 Subrutina de bloques para consulta de faltas**

Fuente: Darwin Marcelo Pillo G.

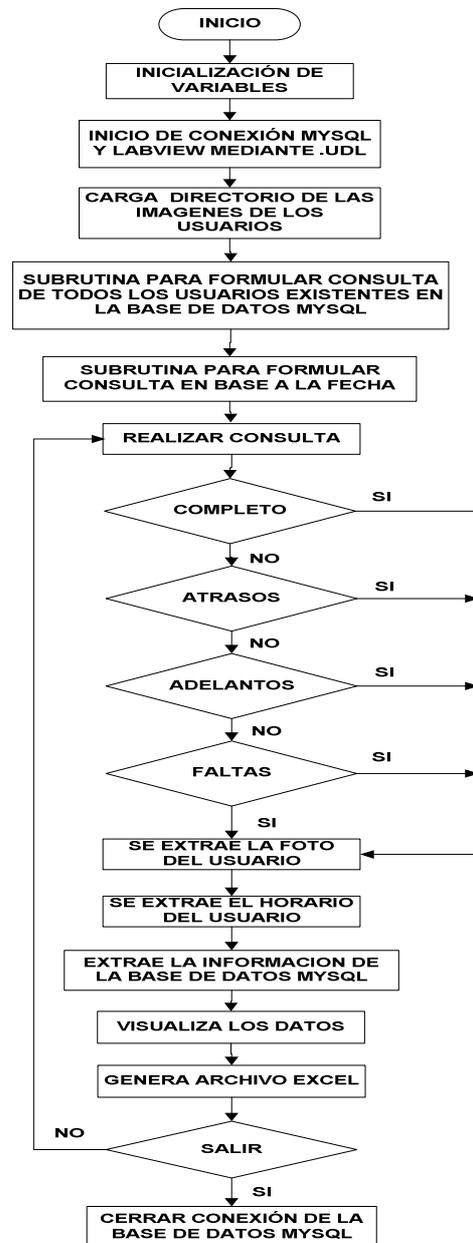
Como se menciona anteriormente las consultas que se realizan a la base de datos se pueden visualizar en la misma interfaz de reportes, pero también existe la opción de que dichos reportes se generen en un archivo de excel utilizando las herramientas de *Report Generation Toolkit* de labview , tal como se indica en la siguiente figura.



**Figura 3.56 Subrutina de generación de reportes en un archivo de Excel**

Fuente: Darwin Marcelo Pillo G.

En la Figura 3.57 se puede apreciar la estructura del programa reportes de usuarios realizados en Labview.



**Figura 3.57** Diagrama de flujo del programa en labview para el reporte de usuarios

**Fuente:** Elaborado por Darwin Marcelo Pillo G.

A continuación en la figura 3.58 se muestra el diagrama completo en labview del programa realizado para generar reportes o consultas.

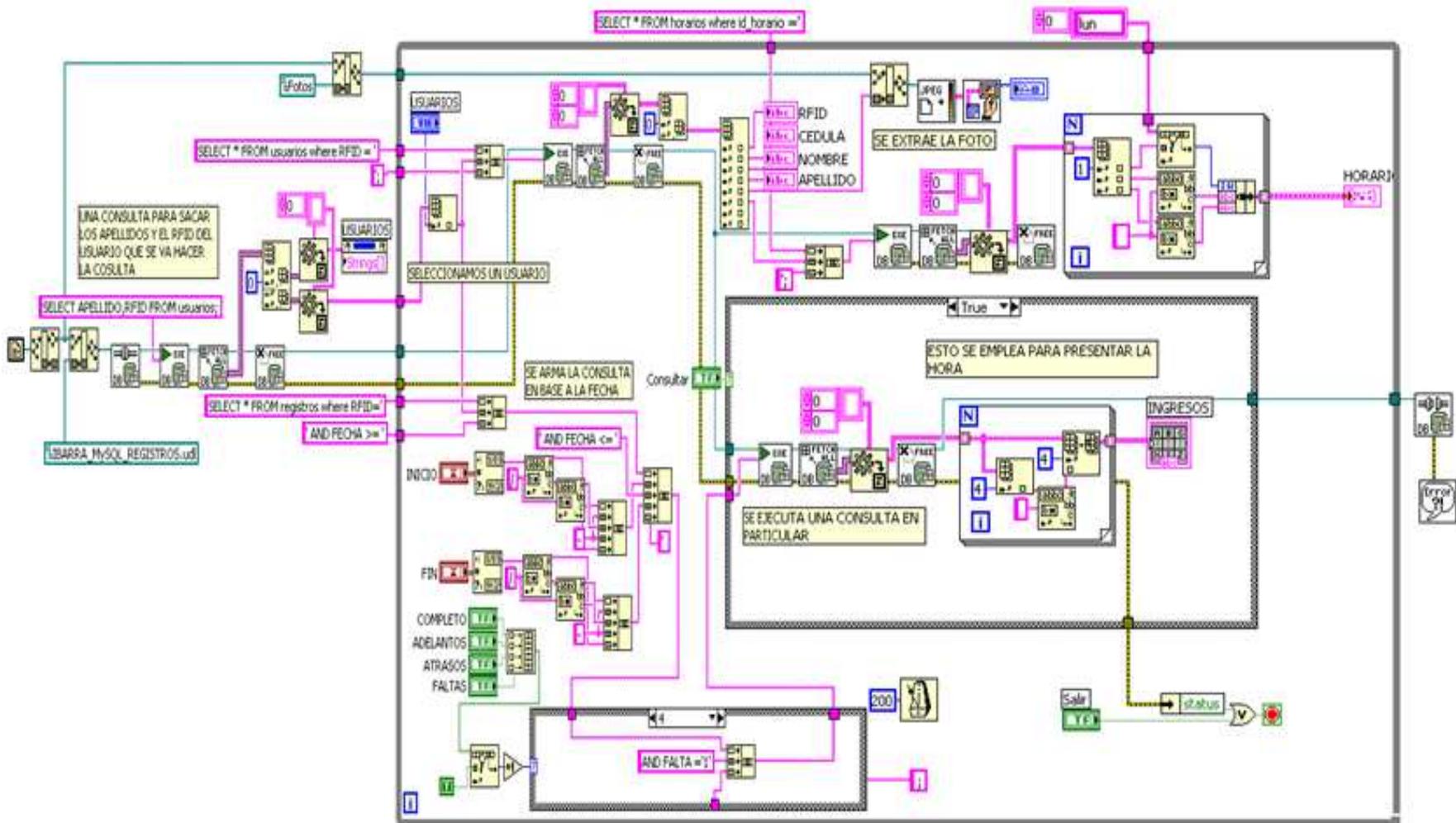


Figura 3.58 Diagrama de bloques del programa de reporte de usuario

Fuente: Elaborado por Darwin Marcelo Pillo G.

### 3.5.8 RECUPERACIÓN DE CÓDIGOS RFID ALMACENADOS EN MEMORIA EEPROM DEL MICROCONTROLADOR

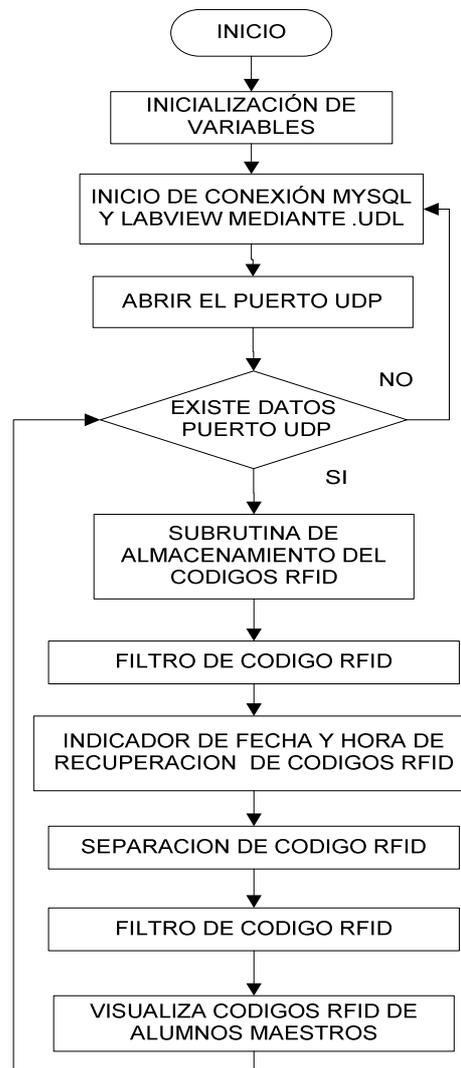
Este programa permite visualizar los códigos RFID almacenados en la memoria EEPROM del PIC18F452. Los códigos únicos de identificación representan el registro de cada usuario cuando el sistema se encuentra sin conexión entre el hardware del sistema de control de personal y el HMI.



**Figura 3.59 SubVI de recuperación de códigos RFID almacenados en memoria EEPROM del microcontrolador**

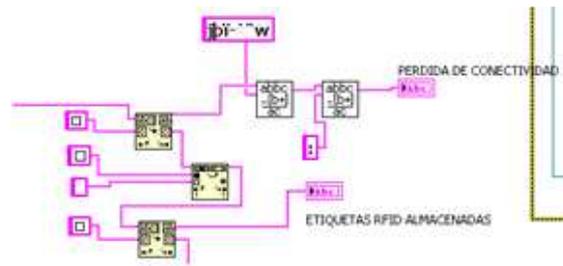
**Fuente:** Elaborado por Darwin Marcelo Pillo G.

En la Figura 3.60 se puede apreciar la estructura del programa de recuperación de códigos RFID almacenados en memoria EEPROM del microcontrolador realizado en Labview.



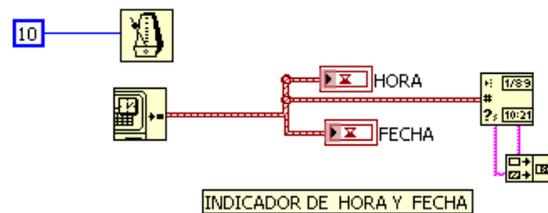
**Figura 3.60 Diagrama de flujo recuperación de códigos RFID almacenados en memoria EEPROM del microcontrolador**

Los códigos RFID que se envían desde el microcontrolador PIC18F452 hacia el HMI, en ocasiones presentan caracteres que no pertenecen al código de la etiqueta. Esto se debe a interferencias del cable de red o a información almacenada en la memoria EEPROM del microcontrolador. Por tal motivo este subVI contiene una subrutina que permite filtrar los datos producidos por dichas interferencias.



**Figura 3.61 Filtrado de Datos**

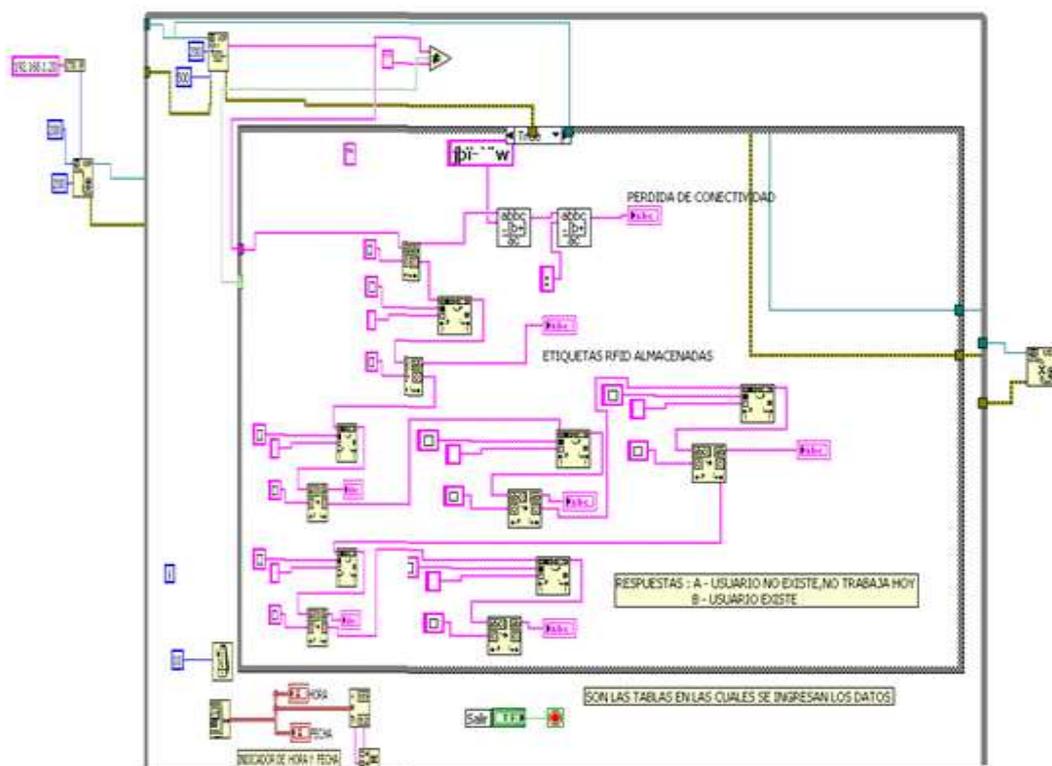
Fuente: Darwin Marcelo Pillo G.



**Figura 3.62 Indicador de Fecha y hora**

Fuente: Darwin Marcelo Pillo G.

Cuando lleguen los códigos RFID al HMI, mediante un arreglo de fecha y hora se visualiza el momento que llegaron los datos, para posteriormente ordenarse y visualizarse en una variable de tipo string en el panel frontal del subVI. En la figura 3.63 se muestra el diagrama completo en labview del programa realizado para recuperar los códigos RFID almacenados en la memoria EEPROM el PIC18F452.



**Figura 3.63 Diagrama de bloques del programa para recuperación de códigos RFID**

**Fuente:** Elaborado por Darwin Marcelo Pillo G.