



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO DE SISTEMAS SCADA CON CONEXIÓN A BASES DE
DATOS, VISUALIZACIÓN Y CONTROL EN DISPOSITIVOS
MÓVILES**

Carlos Alejandro Guzmán Solano

Asesorado por el Ing. Byron Odilio Arrivillaga Méndez

Guatemala, mayo de 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE SISTEMAS SCADA CON CONEXIÓN A BASES DE
DATOS, VISUALIZACIÓN Y CONTROL EN DISPOSITIVOS
MÓVILES**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

CARLOS ALEJANDRO GUZMÁN SOLANO
ASESORADO POR EL ING. BYRON ODILIO ARRIVILLAGA MÉNDEZ

AL CONFERÍRSE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, MAYO DE 2011

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Ing. Miguel Angel Dávila Calderón
VOCAL IV	Br. Luis Pedro Ortiz de León
VOCAL V	P.A. José Alfredo Ortiz Herincx
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Jaime Humberto Batten Esquivel
EXAMINADOR	Ing. César Ernesto Urquizú Rodas
EXAMINADORA	Inga. Sigrid Alitza Calderón de León
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE SISTEMAS SCADA CON CONEXIÓN A BASES DE DATOS, VISUALIZACIÓN Y CONTROL EN DISPOSITIVOS MÓVILES

Tema que se me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha noviembre de 2010.



Carlos Alejandro Guzmán Solano

Guatemala 2 de Octubre del 2010

Ing. Carlos Eduardo Guzmán Salazar

Coordinador de área

Escuela de Mecánica Eléctrica

Presente

Por este medio me dirijo a usted, para hacer de su conocimiento que he asesorado el proyecto de graduación titulado: **DISEÑO DE SISTEMAS SCADA CON CONEXIÓN A BASES DE DATOS, VISUALIZACION Y CONTROL EN DISPOSITIVOS MOVILES**, elaborado por el estudiante Carlos Alejandro Guzmán Solano, quien se identifica con número de carné: 2004-12853

El presente trabajo de graduación, cumple con los requisitos establecidos en el reglamento de graduación profesional, por lo que emito la aprobación para que se prosiga con los trámites correspondientes.

Al agradecer su atención me es grato suscribirme

Atentamente


Byron Odilio Arrivillaga Méndez
Ingeniero Electrónico
Col. 5217
Ing. Byron Odilio Arrivillaga Méndez
Ingeniero Electrónico
Colegiado activo: 5217



Ref. EIME 47. 2010
Guatemala, 15 de noviembre 2010.

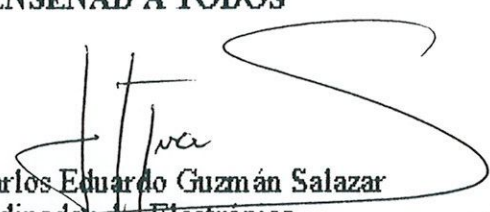
Señor Director
Ing. Guillermo Antonio Puente Romero
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado:
**DISEÑO DE SISTEMAS SCADA CON CONEXIÓN A BASES DE
DATOS, VISUALIZACIÓN Y CONTROL EN DISPOSITIVOS
MÓVILES**, del estudiante, **Carlos Alejandro Guzmán Solano**, que
cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS


Ing. Carlos Eduardo Guzmán Salazar
Coordinador de Electrónica

CEGS/sro





REF. EIME 48. 2010.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; Carlos Alejandro Guzmán Solano titulado: DISEÑO DE SISTEMAS SCADA CON CONEXIÓN A BASES DE DATOS, VISUALIZACIÓN Y CONTROL EN DISPOSITIVOS MÓVILES, procede a la autorización del mismo.


Ing. Guillermo Antonio Puentes Romero



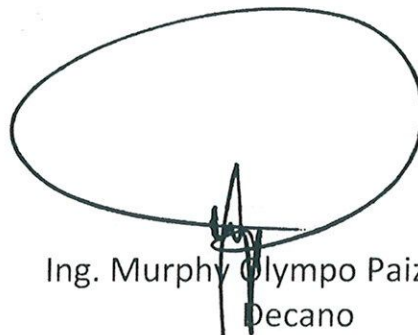
GUATEMALA, 22 DE NOVIEMBRE 2010.



DTG. 128.2011

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al trabajo de graduación titulado: **DISEÑO DE SISTEMAS SCADA CON CONEXIÓN A BASES DE DATOS, VISUALIZACIÓN Y CONTROL EN DISPOSITIVOS MÓVILES**, presentado por el estudiante universitario **Carlos Alejandro Guzmán Solano**, autoriza la impresión del mismo.

IMPRÍMASE:



Ing. Murphy Olympo Paiz Recinos
Decano

Guatemala, 3 de mayo de 2011.



/gdech

ACTO QUE DEDICO A:

Dios

Por la fortaleza, sabiduría, paciencia y compañía que me brinda todos los días.

Mis padres

Carlos Fernando Guzmán Hermann, Ingrid Ivette Castillo Solano de Guzmán, por forjar mi camino y fortalecer mi vida con sus consejos para lograr mis sueños que poco a poco se van cumpliendo. Infinitas gracias.

Mi hermana

Porque a través de la vida me ha enseñado y apoyado incondicionalmente.

Mi novia

Por apoyarme, darme aliento y alegría en las tardes de trabajo.

Mis amigos

Por haber compartido tantos momentos de esfuerzo, estrés y alegría. Al final todo esfuerzo tiene su recompensa.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
GLOSARIO	XI
RESUMEN	XVII
OBJETIVOS	XIX
INTRODUCCIÓN	XXI
1. DISPOSITIVOS MÓVILES	
1.1. Definición	1
1.1.1. PDA	1
1.1.1.1. Marcas más comunes de PDA	2
1.1.2. <i>Smartphone</i>	3
1.1.3. Sistemas operativos para dispositivos móviles	4
1.1.3.1. <i>Android</i>	4
1.1.3.2. <i>Iphone OS</i>	5
1.1.3.3. <i>Palm OS</i>	6
1.1.3.4. <i>BlackBerry OS</i>	9
1.1.3.5. <i>Windows mobile</i>	9
1.1.3.6. <i>Symbian</i>	12
1.1.4. Características importantes de los dispositivos móviles	13
1.1.4.1. Microprocesador	13
1.1.4.2. Memoria RAM	13
1.1.4.3. Memoria ROM	14
1.1.4.4. Puerto para tarjetas <i>Flash</i> , <i>PCMCIA</i> , <i>Secure Digital</i> , <i>Multimedia Card</i>	14

1.1.4.5.	Estructura por teclado o por método <i>Stylus</i>	14
1.1.4.6.	Reconocimiento de escritura de mano	15
1.1.4.7.	Equipo operado por baterías	15
1.1.4.8.	Tamaño reducido	15
1.1.4.9.	Conectividad con redes	15
1.1.5.	Limitaciones de los dispositivos móviles	16
1.1.5.1.	De la interfaz	16
1.1.5.2.	De la tecnología	16
1.1.5.3.	De la conexión de red	16
1.1.5.4.	Costo acumulativo	17
2.	LIGERO VISTAZO A REDES	
2.1.	Red	19
2.1.1.	Conceptos y componentes básicos	20
2.1.1.1.	Tarjeta de red	20
2.1.1.2.	Dirección IP	22
2.1.1.3.	Servidor	23
2.1.1.4.	<i>Switch</i> o conmutador	26
2.1.2.	Métodos de conexión remota	30
2.1.2.1.	Por dirección IP	30
2.1.2.2.	Red privada virtual	30
2.1.2.3.	<i>Dinamic DNS (DDNS)</i>	33
3.	SISTEMAS SCADA	
3.1.	SCADA	35
3.2.	Pasos para la creación de un sistema SCADA	35
3.2.1.	Adquisición de datos	36
3.2.1.1.	Tipos de Sensores	36

3.2.1.2.	Equipo para recopilación de datos	40
3.2.2.	Análisis de los datos adquiridos	42
3.2.2.1.	Algunos tipos de análisis	43
3.2.3.	Presentación de los datos en la interfaz gráfica	43
4.	BASES DE DATOS Y VISUALIZACIÓN WEB	
4.1.	Base de datos	45
4.2.	Gestores de bases de datos	45
4.3.	Páginas web	48
5.	DISEÑO EN <i>LABVIEW</i>	
5.1.	<i>Labview</i>	49
5.1.1.	Partes básicas de <i>Labview</i>	50
5.1.2.	Programación en <i>Labview</i>	56
5.1.2.1.	Creación de un control a la medida	61
5.1.2.2.	Conexión de <i>Labview</i> a <i>Mysql</i>	70
5.1.2.3.	Ingreso de datos a <i>Mysql</i> a través de <i>Labview</i>	80
5.1.2.4.	Acceso a datos en <i>Mysql</i> a través de <i>Labview</i>	85
5.1.2.5.	<i>Labview</i> en <i>Windows Mobile</i>	89
6.	DISEÑO DE UN SISTEMA SCADA CON CONEXIÓN A BASE DE DATOS VISUALIZACIÓN Y CONTROL EN DISPOSITIVO MÓVIL	
6.1.	Simulación de sensores	99
6.2.	Creación de la base de datos	101
6.3.	Programación de la pantalla de ingreso mediante usuario y contraseña	102
6.4.	Programación del VI que controlará el proceso	106

6.5. Programación del VI móvil y unificación con el VI en la PC	111
6.6. Verificación del usuario en el teléfono	116
6.7. Automatización de la creación de respaldos de la base de datos	117
CONCLUSIONES	121
RECOMENDACIONES	123
BIBLIOGRAFÍA	125

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	<i>PDA</i>	2
2.	<i>Smartphone</i>	3
3.	<i>Android</i>	5
4.	<i>iPhone OS</i>	6
5.	<i>Palm OS</i>	8
6.	<i>BalckBerry OS</i>	9
7.	Logotipo <i>Windows Mobile</i>	12
8.	Logotipo <i>Symbian OS</i>	13
9.	Tarjeta de Red	21
10.	Conexión VPN.....	32
11.	Ejemplo SCADA	35
12.	DAQ	41
13.	PLC	42
14.	Logotipo <i>MySql</i>	46
15.	Logotipo <i>Oracle</i>	47
16.	<i>Access</i>	48
17.	<i>Labview</i>	49
18.	Pantalla principal de <i>Labview</i>	51
19.	Panel frontal	52
20.	Diagrama de bloques	52
21.	Paleta de herramientas	53
22.	Paleta de funciones.....	54
23.	Paleta de controles	54

24.	Indicadores	55
25.	Controles	56
26.	Panel frontal programación básica 1.....	57
27.	Conexión bloques programación básica 1	57
28.	Diagrama de bloques programación básica 1.....	58
29.	Corriendo programación básica 1	59
30.	Panel frontal programación básica 2.....	60
31.	Diagrama de bloques programación básica 2.....	60
32.	Corriendo programación básica 2.....	61
33.	Control a modificar.....	62
34.	Acceso al menú de modificación	63
35.	Ventana de modificación.....	64
36.	Biblioteca de imágenes.....	65
37.	Control antes	66
38.	Control después.....	66
39.	Reemplazo del cursor	66
40.	Control completo.....	67
41.	Pregunta de reemplazo.....	68
42.	Control terminado en el panel frontal.....	68
43.	Programación de control a la medida	69
44.	Corriendo control a la medida.....	69
45.	Instalando <i>Xampp</i>	70
46.	Iniciando los servidores <i>Apache</i> y <i>MySql</i>	71
47.	Herramienta de configuración de <i>MySql</i>	72
48.	Origen de datos ODBC	73
49.	Asistente de conexión.....	74
50.	Parámetros para la conexión	75
51.	Prueba de conexión	76
52.	Datos para la conexión	77

53.	Creación del archivo udl.....	78
54.	Asistente de conexión	79
55.	Configuración del directorio.....	79
56.	Configuración bloque de conexión	79
57.	Controles para la conexión a <i>MySql</i>	80
58.	Bloques de programación para la conexión	81
59.	Programación del ingreso	81
60.	Configurar el control del case.....	82
61.	Conexión del directorio al bloque conexión.....	82
62.	Interconexión de bloques	83
63.	Expansión bloque <i>build array</i>	83
64.	Interconexión total del VI	83
65.	Ingreso de la tabla.....	84
66.	Programación completa del ingreso de datos	84
67.	Ejecución e inserción de datos.....	85
68.	Verificación de los datos	85
69.	Bloques para el acceso a datos	86
70.	Programación para el acceso a datos básico.....	86
71.	Indicador y ejecución del código	87
72.	Indexado de los datos	87
73.	Modificación de los datos extraídos	88
74.	Programación completa del acceso y modificación de datos	88
75.	Ejecución del VI de acceso a datos	89
76.	Proyecto nuevo	90
77.	Agregar un nuevo dispositivo móvil.....	90
78.	Dispositivo móvil agregado	91
79.	Solicitud de permiso de ejecución.....	91
80.	Selección del lugar de instalación	92

81.	Ventana de creación de una nueva variable	93
82.	Front panel móvil	94
83.	Front panel en la PC	94
84.	Programación del VI móvil	94
85.	Programación del VI de la PC.....	95
86.	Asistente para la creación de ejecutable	96
87.	Botón activado en el móvil	96
88.	Led encendido en la PC.....	97
89.	Botón activado en la PC	97
90.	Led encendido en el móvil	97
91.	Controles para la simulación.....	100
92.	Almacenamiento de la simulación en variables	100
93.	Programación del VI de simulación.....	101
94.	Tabla dentro de la base usuarios.....	101
95.	Panel frontal para el ingreso	102
96.	Adquisición de los datos	103
97.	Instrucción de búsqueda.....	103
98.	Comparación de los datos	104
99.	Petición de reingreso de información.....	104
100.	Invocar al otro VI.....	105
101.	Secuencia de apertura del VI de control	105
102.	Creación del tanque.....	106
103.	Creación del fuego.....	107
104.	Creación de los ventiladores.....	107
105.	Panel frontal del VI control.....	108
106.	Tablas del VI de control	109
107.	Configuración de los PID	109
108.	Programación del encendido y apagado de la bomba.....	110
109.	Encendido y apagado de ventiladores y fuego	111

110.	Sección móvil de los <i>setpoints</i>	112
111.	Sección móvil de las ganancias	112
112.	Sección de gráficos en el VI móvil	113
113.	Programación de la simulación de sensores.....	114
114.	Programación en la PC que es controlada por el móvil	114
115.	Programación en la PC que permite el cambio de ganancias vía dispositivo móvil	115
116.	Ingreso de ganancias en el móvil.....	115
117.	Selección de qué variables se graficarán.....	116
118.	Logueo en el móvil	116
119.	Selección de respaldo día jueves.....	117
120.	Creación del nombre del archivo de respaldo	118
121.	Creación del archivo de respaldo.....	118
122.	Configuración del envío de FTP	119
123.	Borrado del archivo de respaldo en la PC local	119
124.	Secuencia de respaldo.....	120

GLOSARIO

**Asistente
Personal Digital**

Es un dispositivo electrónico que posee funciones tales como agenda, recordatorios, conexión de red y tiene varias aplicaciones adicionales con las cuales nos asiste en las actividades diarias.

Bases de datos

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

**Controlador lógico
programable**

Llamado PLC por sus siglas en inglés, es un equipo electrónico programable que permite almacenar una secuencia de órdenes (programa) en su interior y ejecutarlo de forma cíclica, con el fin de realizar una tarea.

**Dirección de
Protocolo de
Internet**

Es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP (*Internet Protocol*), que corresponde al nivel de red del protocolo TCP/IP. Dicho número no se ha de confundir con la dirección MAC que es un número hexadecimal fijo, que es asignado a la tarjeta o dispositivo de red por el fabricante.

Labview

Herramienta de *software* diseñado para el control de procesos industriales que nos permitirá crear los programas que se requiera.

Memoria de acceso aleatorio RAM

Memoria que se usa para almacenar datos y, accederlos de manera aleatoria, es decir, que se puede acceder cualquier *byte* de memoria sin acceder los *bytes* precedentes. Los datos cambiarán constantemente y no se perderán en un corte de energía.

Memoria de solo lectura

También llamada memoria ROM por sus siglas en inglés, es una memoria que sólo permite leer sus datos, mas no escribir en ella. En esta memoria se encuentra toda la información que no se debe perder después de un corte de energía, como podría ser la configuración del equipo.

Memoria Secure Digital

Es un formato de tarjeta de memoria inventado por Panasonic, que se utiliza en dispositivos portátiles, tales como cámaras fotográficas, asistentes personales digitales, reproductores de música.

En esta memoria se puede escribir y leer; y su información no se perderá si se corta el suministro de energía.

Scada

Es el acrónimo de *supervisory control and data acquisition* (supervisión, control y adquisición de datos). Un SCADA es un sistema que funciona con base de computadoras, que permite supervisar y controlar a distancia una instalación de cualquier tipo. A diferencia de los Sistemas de Control Distribuido, el lazo de control es generalmente cerrado por el operador.

Sensor

Dispositivo que es capaz de detectar la presencia o variación de magnitudes físicas o químicas como la temperatura, densidad, velocidad, humedad, pH, fuerza, corriente eléctrica, etc.

Servidor

En informática, un servidor es un tipo de *software* que realiza ciertas tareas en nombre de los usuarios. El término servidor, también se utiliza para referirse a la computadora en la cual funciona ese *software*. Es una máquina cuyo propósito es proveer datos que puedan ser utilizados por otras máquinas para las distintas tareas que estas últimas ejecuten.

Sistema de nombres de dominio

Es un servidor que contiene una tabla de nombres asignados a las direcciones IP, para que el usuario recuerde nombres y no números.

Sistema operativo

Programa que permite la interacción de los seres humanos con las máquinas, ya que traducen del nivel más alto en la programación hasta el nivel máquina, constituido de unos y ceros.

Smartphone

Es un teléfono que posee características de una PDA como lo es calendario, red WiFi, procesador de texto, y que acepta la instalación de aplicaciones de elevada complejidad; por ende, es como una PC en pequeño.

Stylus

Dispositivo que tiene forma de lapicero sin tinta, que puede estar compuesto de varios materiales. Se utiliza para tocar la pantalla del dispositivo móvil o para escribir en la pantalla.

Switch

Dispositivo electrónico dentro de una red TCP/IP que transmite la información solicitada hacia alguno de sus puertos en específico.

Tarjeta de red

Dispositivo electrónico dentro de una red TCP/IP que se encarga de establecer la comunicación y la transmisión de datos entre dos o más equipos. Cabe mencionar que también se encarga de que no se pierdan datos dentro de la comunicación y si algún dato se pierde, automáticamente pide la retransmisión del mismo.

VPN

Llamada así por sus siglas en inglés, significa red privada virtual. Este tipo de red, permite la interconexión de los equipos, por lo general computadoras, utilizando redes públicas como lo es el internet. Todo esto mediante la creación de túneles que se crean en la red pública contando con seguridad o encriptación.

RESUMEN

Este trabajo explica la creación de un sistema SCADA, utilizando el programa para desarrollo de aplicaciones *Labview*, el cual requirió conexión hacia bases de datos, específicamente a bases creadas utilizando el programa de computo *MySql*. También explica cómo se logró ver y controlar ese sistema a través de un teléfono inteligente, con lo cual, implementando estas medidas en un entorno físico, se puede disminuir el tiempo de falla dentro de una empresa y alcanzar una mayor productividad.

El desarrollo del sistema SCADA comienza con la creación de un algoritmo que controla el nivel y la temperatura de líquido en un tanque, utilizando la herramienta *Labview*. Luego se procede a crear la base de datos de los usuarios que tendrán acceso al sistema SCADA, los cuales podrán ver y cambiar los parámetros del sistema para su funcionamiento óptimo.

A continuación se procede a crear la interfaz gráfica del programa para el teléfono inteligente y a introducir la programación en la computadora, que permitirá que el teléfono controle el proceso. Por último se programa el teléfono para que haga lo que el usuario le solicite.

OBJETIVOS

General

Diseñar un sistema SCADA con conexión a bases de datos, visualización y control en dispositivos móviles.

Específicos

1. Crear sistemas con bases de datos que sean capaces de hacer sus respaldos automáticamente utilizando *Labview database connectivity toolkit*
2. Controlar inalámbricamente un sistema deseado.
3. Definir usuarios específicos los cuales serán los únicos que tendrán acceso a la información por medios inalámbricos

INTRODUCCIÓN

Actualmente nuestro mundo está teniendo grandes avances en todos los campos de la ciencia, y eso se debe en gran parte a que la tecnología ha cambiado, permitiéndonos romper barreras tales como la velocidad de producción y la calidad de los productos. Esta tecnología cada día se hace más accesible, llegando así a todos los países, comenzando por los más desarrollados hasta llegar a los menos desarrollados. Los avances tecnológicos permiten ahora llegar a nuestras metas cada vez más rápido, por lo mismo ahora las empresas capacitan a su personal de una mejor manera previo a comprar nuevas tecnologías y darles el uso adecuado.

Una parte muy importante que influye en el mercado son las empresas y sus prioridades, las cuales se mencionan a continuación:

- Aumento de su productividad
- Entrega de producto de alta calidad al usuario final
- Minimizar los costos de producción

Esta última, en especial, ha sido una de las más importantes a lo largo de los años, ya que una empresa con menos costos de producción, y que mantiene o mejora su calidad, posee más ventajas para competir en el mercado, logrando así más ganancias.

Las empresas más destacadas a nivel mundial cuentan con sistemas automatizados de producción, controlados por sistemas SCADA, los cuales son sistemas que se encargan de:

- La adquisición de una amplia gama de información, que se da en cada una de las etapas del proceso dentro de la planta;
- Analizar la información recaudada en el paso anterior;
- Desplegar la información al usuario para que sea interpretada;
- Tomar decisiones y comandar acciones para modificar parámetros dentro del proceso, y así obtener los resultados adecuados.

Estos sistemas poseen un cierto nivel de inteligencia propia, lo cual les permite tomar decisiones determinantes cuando alguna variable del proceso cambia, notificando al usuario a cargo si ha ocurrido una falla, o si es necesario realizarle mantenimiento a una máquina en específico, lo anterior elimina probabilidades de fallas, disminuye tiempos de paro por fallas y disminuye pérdidas por productos mal elaborados.

Los capítulos de este trabajo detallarán algunas de las herramientas que nos permiten crear estos sistemas inteligentes, que además nos proporcionarán la movilidad que todo empresario desea y el control que es vital, con el fin de evitar las problemáticas cotidianas y poder así percibir los beneficios que se han expuesto anteriormente.

El capítulo uno dará una introducción sobre las PDA y qué capacidades poseen, además de qué tipo de sistema operativo pueden tener e incluimos algunos de los distintos servicios que nos proveen.

En el capítulo dos se muestran conceptos básicos de redes, tanto alámbricas como inalámbricas así como algunas configuraciones; además de las ventajas del uso de éstas dentro de la industria y los distintos ambientes que existen.

En el capítulo tres se explican los sistemas de adquisición, análisis y presentación de datos dentro de la industria, las ventajas que se tienen a partir de estos, tomando en cuenta varias necesidades que surgen en la programación del mismo.

En el capítulo cuatro se desarrollan las bases de datos, así como sus usos más comunes y las bases de datos más difundidas a nivel mundial, pero la que tomará más importancia en este trabajo es la base de datos en programación SQL específicamente *MySQL*.

En el capítulo cinco, se presenta la herramienta que se utilizó para el desarrollo del sistema SCADA.

En el capítulo seis se presenta el diseño de un sistema SCADA, el cual podrá ser observado y controlado desde un teléfono inteligente, utilizando las herramientas que se presentaron en los capítulos anteriores.

1. DISPOSITIVOS MÓVILES

1.1. Definición

Un dispositivo móvil puede definirse como todo aquel *hardware* que tenga características similares a las computadoras de escritorio, con la diferencia principal, que todo es reducido y normalmente integrado en una sola pieza.

Estos cuentan con un sistema operativo que es instalado de fábrica, este software por lo general es una versión similar al sistema operativo de una computadora de escritorio, con funcionalidad reducida, por lo que les llaman ediciones compactas. Sobre estos sistemas operativos a su vez se pueden instalar programas que pueden ser utilitarios o herramientas de desarrollo.

Se hace diferencia entre los principales dispositivos móviles como sigue:

- Celulares o *Smartphone*
- Asistente Digital Personal o PDA (*Personal Digital Assistant*)

1.1.1. PDA

El asistente digital personal mostrado en la Figura 1, es un computador de mano originalmente diseñado como agenda electrónica, calendario, lista de contactos, bloc de notas y recordatorios, el cual posee un sistema de reconocimiento de escritura.

Figura 1. PDA



Fuente: <http://bemovil.com/images/PALM%20TX.jpg>

Hoy en día estos dispositivos pueden realizar muchas de las funciones de una computadora de escritorio, pero con la ventaja de ser portátiles.

1.1.1.1. Marcas más comunes de PDA

En el mercado existe una amplia gama de distribuidores de PDA, pero como siempre, las personas prefieren a algunas marcas más que otras. Entre las preferidas se puede mencionar:

- *Palm* con una amplia gama
- *Hp* con la serie de *IPAQ*
- *Dell* con la serie *Axim*

Existen muchas marcas, pero estas son algunas de las más vendidas.

1.1.2. **SMARTPHONE**

Teléfono inteligente, Fig. 2, es un dispositivo electrónico que funciona como un teléfono celular con características similares a las de una computadora personal. Casi todos los teléfonos inteligentes son teléfonos celulares que soportan un cliente de correo electrónico con la funcionalidad completa de un organizador personal.

Figura 2. **Smartphone**



Fuente:http://www.photokina-show.com/news_images/00508_htc-smartphone.jpg

Los teléfonos inteligentes se distinguen por muchas características, que incluyen, entre otras, pantallas táctiles, un sistema operativo así como la conectividad a Internet. El soporte completo al correo electrónico parece ser una característica indispensable encontrada en todos los modelos existentes y anunciados en 2007, 2008 y 2009.

Casi todos los teléfonos inteligentes permiten, al usuario, instalar programas adicionales, inclusive de terceros; pero algunos vendedores gustan

de tildar a sus teléfonos como inteligentes, aun cuando no tienen esa característica. Un claro ejemplo de teléfonos inteligentes son los Motorola de la serie MOTO Q.

Entre las características más importantes están el acceso a Internet y al correo electrónico, a los programas de agenda, las cámaras integradas, administración de contactos, acelerómetros y algunos programas de navegación, así como ocasionalmente la habilidad de leer documentos de negocios en variedad de formatos como PDF y *Microsoft Office*.

1.1.3. Sistemas operativos para teléfonos inteligentes

Existen varios sistemas operativos para los teléfonos inteligentes, cada uno de ellos diseñado con diferencias importantes. Dentro de estos se puede mencionar:

- *Android*
- *iPhone OS*
- *Palm OS*
- *BlackBerry OS*
- *Windows Mobile*
- *Symbian*

1.1.3.1. Android

Es un sistema operativo para dispositivos móviles y computadoras basado en el núcleo *Linux*. Inicialmente fue desarrollado por *Android Inc.*, compañía que fue comprada después por *Google*. En la actualidad lo desarrollan los miembros de la *Open Handset Alliance* (liderada por *Google*). La presentación de la

plataforma *Android* se realizó el 5 de noviembre de 2007 junto con la fundación *Open Handset Alliance*, un consorcio de 48 compañías de *hardware*, *software* y telecomunicaciones comprometidas con la promoción de estándares abiertos para dispositivos móviles.

Esta plataforma permite el desarrollo de aplicaciones por terceros (personas ajenas a *Google*). Los desarrolladores deben escribir código gestionado en lenguaje de programación Java a través de SDK proporcionada por el mismo *Google*. Una alternativa es el uso de la NDK (*Native Development Kit*) de *Google* para hacer el desarrollo en lenguaje C como código fuente.

El logotipo que identifica a este sistema operativo es el que se presenta en la figura 3.

Figura 3. **Android**



Fuente:

<http://thearcherblog.files.wordpress.com/2009/10/android.jpg>

1.1.3.2. **iPhone OS**

Es el sistema operativo que utiliza el *iPod touch* y el *iPhone*, diseñado por 175 ingenieros de Apple, entre los cuales figuran Rafael Perez, Danel Abreu,

entre otros. Está basado en una variante del *Mach kernel* que se encuentra en *Mac OS X*. El *iPhone OS* incluye el componente de *software* “*Animation Core*” de *Mac OS X v10.5* que, junto con el *PowerVR MBX* el *hardware* de 3D, es responsable de las animaciones usadas en el interfaz de usuario. *iPhone OS* tiene 4 capas de abstracción: la capa del núcleo del sistema operativo, la capa de Servicios Principales, la capa de Medios de comunicación y la capa de *Cocoa Touch*. El sistema operativo ocupa bastante menos de medio GB del total del dispositivo, 8 GB o el almacenaje de 16 GB. Esto se realizó para soportar futuras aplicaciones de *Apple*.

El logotipo que identifica a este sistema operativo es el que se muestra en la figura 4.

Figura 4. ***iPhone OS***



Fuente: <http://carrero.es/wp-content/uploads/2009/03/iphone-os-3.jpg>

1.1.3.3. *Palm OS*

Es un sistema operativo hecho por *PalmSource, Inc.* Para computadores de mano (PDAs) fabricados por varios licenciatarios. El sistema operativo *Palm* fue desarrollado originalmente por Jeff Hawkins para el *Pilot* PDA de *U.S.*

Robotics. La versión 1.0 se vendía con los primeros *Pilot* 1000 y 5000 y la versión 2.0 se introducía con el *Palm Pilot* Personal y Profesional.

Cuando salieron los *Palm* de la serie III se introdujo la versión 3.0 del sistema operativo. Posteriormente, salieron las versiones 3.1, 3.3 y 3.5, que añadían apoyo para color, puertos de expansión múltiples, nuevos procesadores y otras prestaciones.

La versión 4.0 salió con la serie m500, y más tarde salió la actualización para aparatos anteriores. Esto añadía una interfaz estándar para el acceso del sistema de archivos externo (como tarjetas SD) y mejoraba las bibliotecas de telefonía, seguridad y mejoras de IU.

La versión 5 (*Garnet*) fue la primera versión que soportó los dispositivos ARM. Anunciado como paso importante por apoyar a los procesadores ARM, las aplicaciones *Palm* se ejecutan en un entorno emulado denominado el Entorno de Compatibilidad de Aplicaciones Palm (PACE, en inglés), disminuyendo velocidad pero permitiendo gran compatibilidad con programas antiguos.

El *software* nuevo puede aprovechar los procesadores de ARM con ARMlets, pequeñas unidades de código ARM. Era también aproximadamente entonces cuando *Palm* empezaba a separar sus divisiones de *hardware* y de sistemas operativos, y finalmente se convierten en dos compañías *PalmSource, Inc.* (sistemas operativos) y *palmOne, Inc.* (*hardware*).

Las siguientes versiones de *Palm OS* 5 han tenido un API estándar para alta resolución y áreas de entrada dinámicas, junto con un cierto número de mejoras menores.

Palm OS 4.1.2, 5.2.1 y posteriores, incluyen Graffiti 2, debido a la pérdida de un pleito de violación con Xerox. Graffiti se basa en Jot de CIC.

PalmSource, Inc. presentó Palm OS Cobalt (también denominado Palm OS 6) a los licenciarios el 29 de diciembre de 2003. Esto completaría la migración a aparatos con ARM, y permitiría apoyar a las aplicaciones nativas ARM junto con apoyo multimedia mejorado. Actualmente NO existen equipos que usen el Palm OS 6 o Cobalt. No está muy claro el futuro de esta versión de Palm OS, derivado de la compra de PalmSource por la compañía japonesa Access Co.

Aparentemente, en algún momento será posible tener nuevos equipos PDA con Palm OS cuyo núcleo sea un *Linux* completamente funcional.

El logotipo que identifica a este sistema operativo es el que se presenta en la figura 5.

Figura 5. ***Palm OS***



Fuente: <http://gizmodo.com/assets/resources/2007/10/PalmLogo.png>

1.1.3.4. **BlackBerry OS**

Es una plataforma de *software* diseñada por la compañía *Research in Motion* hecha para las PDA. Este sistema es capaz de trabajar con aplicaciones multi-tarea y hace un uso extenso de las interfaces de entrada como lo es la pantalla táctil o el teclado. Este *software* está enfocado en la conectividad del usuario, siendo así las tareas más importantes, el correo electrónico y la conectividad a internet.

El logotipo que identifica a este sistema operativo es el que se muestra en la figura 6.

Figura 6. **BlackBerry OS**



Fuente: <http://www.mobiletor.com/images/rim-blackberry-logo07.jpg>

1.1.3.5. **Windows Mobile**

Es un sistema operativo compacto, con una suite de aplicaciones básicas para dispositivos móviles basados en la API Win32 de *Microsoft*.

Los dispositivos que llevan *Windows Mobile* son *Pocket PC*, *Smartphone* y *Media Center* portátil. Ha sido diseñado para ser similar a las versiones de escritorio de *Windows*.

Tanto *Windows Mobile* para *Pocket PC*, como *Windows Mobile* para *Smartphone*, poseen bastantes aspectos parecidos:

- En la pantalla Hoy nos mostrará la fecha actual, la información del dueño, las citas próximas, los mensajes *e-mail*, y las tareas. En la parte inferior aparecerá, generalmente, una barra con dos botones. También incluye una barra que incluye iconos para notificar el estado del *bluetooth*, batería, cobertura, etc. Este tema predeterminado puede ser cambiado añadiendo o eliminando complementos, como por ejemplo, alarma, temperatura, estado de la batería.
- En la barra de tareas muestra: la hora actual, el volumen y el estado de la conectividad. Cuando un programa o un mensaje están abiertos el espacio en blanco, en el que estaba el reloj se convierte en una "ok" o un icono de cerrar (x). La característica principal de la barra de tareas es el botón de inicio, que está diseñado para que sea parecido al botón de Inicio de las versiones de escritorio de *Windows*. El menú de Inicio ofrece programas abiertos recientemente, nueve entradas del menú personalizadas, y accesos directos a programas, ajustes, búsquedas, y ayuda.
- Las versiones *Pocket PC* incluyen en *Windows Mobile* aplicaciones de *Microsoft Office*. Éstos incluyen *Pocket Word* y *Pocket Excel*. En *Windows Mobile 5.0* se incluye *Pocket PowerPoint*.

Estas versiones incluyen muchas de las características que se utilizan en versiones de escritorio, pero algunas otras características como la inserción de las tablas e imágenes no se han incluido en versiones anteriores a *Windows 5.0. ActiveSync*, ya que tiene la capacidad de convertir archivos de versiones de escritorio a archivos compatibles con *Pocket PC*.

- *Outlook Mobile* es también un programa que viene con *Windows Mobile*. Esto incluye tareas, calendario, contactos y la bandeja de entrada. *Microsoft Outlook* para las versiones de escritorio se incluye a veces en los CD-ROM's del fabricante del *Pocket PC*.
- *Windows Media Player* para *Windows Mobile* se añade con el *software*. Actualmente, todas las *Pocket PC* incluyen la versión 9 del reproductor, pero la versión 10 se ha incluido con un *hardware* más nuevo y con las nuevas versiones de *Windows Mobile*. Para algunos dispositivos, la versión 10 está disponible para su descarga solo para determinados dispositivos - éstos incluyen los dispositivos de la gama de Dell Axim. *Windows Media Player* reproduce: WMA, WMV , MP3, y AVI. Los archivos MPEG actualmente no están soportados, y se debe descargar un programa de terceros para reproducirlos, y los archivos de WAV se reproducen en un reproductor por separado. Algunas versiones son también capaces de reproducir M4A.
- Cliente para RPV's PPTP

El logotipo que identifica a este sistema operativo es el que se presenta en la figura 7.

Figura 7. **Logotipo *Windows Mobile***



Fuente: <http://www.imprescindible.es/wp-content/uploads/2009/07/windows-mobile.png>

1.1.3.6. *Symbian*

Es un sistema operativo que fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encuentran Nokia, Sony Ericsson, *PSION*, *Samsung*, *Siemens*, *Arima*, *Benq*, *Fujitsu*, *Lenovo*, LG, Motorola, *Mitsubishi Electric*, *Panasonic*, *Sharp*, etc. Sus orígenes provienen de su antepasado EPOC32, utilizado en PDA's y *Handhelds* de *PSION*.

El objetivo de *Symbian* fue crear un sistema operativo para terminales móviles que pudiera competir con el de *Palm* o el *Windows Mobile* de *Microsoft*, el logotipo de *symbian* se presenta en la figura 8.

Figura 8. **Logotipo Symbian**



Fuente: <http://nokiatouch.me/images/stories/symbian-os.jpg>

1.1.4. Características importantes de los dispositivos móviles

Los dispositivos móviles constan de varias partes, la mayoría de éstas, se listan a continuación.

1.1.4.1. Microprocesador

Se encuentran de diversos fabricantes y con velocidades que generalmente van de los 33Mhz hasta los 600Mhz

1.1.4.2. Memoria RAM

Aquí se almacenan nuestras aplicaciones y datos. Es de tipo no-volátil, o sea que los datos no se pierden aunque este apagado el dispositivo, siempre y cuando tenga colocada la batería.

1.1.4.3. Memoria ROM

Aquí se encuentra almacenado el sistema operativo del dispositivo. Presenta la particularidad de que este sistema se puede actualizar, para mejorar la versión del sistema operativo instalado de fábrica.

1.1.4.4. Puerto para tarjetas *Flash*, *PCMCIA*, *Secure Digital*, *Multimedia Card*

Estos periféricos incrementan la funcionalidad del dispositivo de muchas formas tal como puede ser:

- Incrementar la capacidad de almacenamiento del dispositivo a cantidades como *gigabytes*, si la necesidad del transporte de datos es alta.
- Agregar funcionalidades multimedia como cámaras fotográficas, procesamiento de audio.
- Conectividad inalámbrica

1.1.4.5. Estructura por teclado o por método *Stylus*

Para hacer el método de escritura algo familiar, se puede incluir un teclado incorporado o agregar un teclado expansible. El método *Stylus* se basa en escribir con un pequeño lápiz plástico sobre un área especial para el reconocimiento de un alfabeto simplificado, con el fin de acelerar la escritura

1.1.4.6. Reconocimiento de escritura a mano

Algunos dispositivos vienen con software de fábrica que permite el reconocimiento de escritura a mano, como si se estuviese escribiendo en papel.

1.1.4.7. Equipo operado por baterías

Esta es una consideración importante, pues el equipo debe funcionar sin una alimentación de energía, por lo que el dispositivo debe optimizar el uso de las baterías.

1.1.4.8. Tamaño reducido

Esto es lo que hace a los dispositivos móviles prácticos de transportar, como si llevara un celular.

1.1.4.9. Conectividad con redes

Existen tres modos básicos de conexión:

- Dos cuerpos: se trata de un dispositivo que se conecta con otro dispositivo que le brinda la conexión a las redes.
- Desmontable: si el dispositivo móvil accede por medio de tarjetas para acceso a redes cableadas o inalámbricas.
- Integrado: el PDA incluye la conectividad ya sea para redes inalámbricas, infrarrojos, o redes celulares, ahora también Wi-fi.

1.1.5. Limitaciones de los dispositivos móviles

Por ser de carácter compacto y portátil, los dispositivos móviles constan de varias limitaciones, las cuales se presentan a continuación.

1.1.5.1. De la interfaz

Las pantallas aunque con buena resolución, pueden ser muy pequeñas, por lo que puede limitar el contenido que se puede ver de una sola vez. Los medios que brindan pueden ser lentos ciertas veces para la entrada de datos, por lo que las aplicaciones deben facilitar el ingreso, por ejemplo, el relleno automático de formularios.

1.1.5.2. De la tecnología

Con fuentes de energía limitadas y menor velocidad de procesamiento que una PC, las aplicaciones se ven limitadas en su funcionalidad, causando así la necesidad de programas que consuman pocos recursos.

1.1.5.3. De la conexión de red

Se ve limitada en acceso a redes debido a tiempos de respuesta prolongados, la disponibilidad de la red y el ancho de banda altamente variable, esto por influencia de factores como el cambio de celda, condiciones de tráfico, competencia con la señal de voz, etc.

1.1.5.4. Costo acumulativo

Puede que se requiera adquirir tecnologías y complementos necesarios para ciertos tipos de aplicaciones, como las tarjetas de memoria y tarjetas inalámbricas.

Nota: al momento de la compra, es muy importante tomar en cuenta las características mencionadas anteriormente, junto con el trabajo que será desarrollado por el dispositivo.

2. LIGERO VISTAZO A REDES

2.1. Red

Conjunto de técnicas, conexiones físicas y programas empleadas para conectar dos o más computadoras. Los usuarios de una red pueden compartir archivos, impresoras y otros recursos, enviar mensajes electrónicos y ejecutar programas en otras computadoras.

Una red tiene tres niveles de componentes: *software* de aplicaciones, *software* de red y *hardware* de red.

El *software* de aplicaciones está formado por programas informáticos que se comunican con los usuarios de la red y permiten compartir información (como archivos de bases de datos, de documentos, gráficos o vídeos) y recursos (como impresoras o unidades de disco). Un tipo de *software* de aplicaciones se denomina cliente-servidor. Las computadoras cliente envían peticiones de información o de uso de recursos a otras computadoras, llamadas servidores, que controlan el flujo de datos y la ejecución de las aplicaciones a través de la red

El *software* de red consiste en programas informáticos que establecen protocolos o normas, para que las computadoras se comuniquen entre sí. Estos protocolos se aplican enviando y recibiendo grupos de datos formateados denominados paquetes.

El *hardware* de red está formado por los componentes materiales que unen las computadoras. Dos componentes importantes son los medios de transmisión que transportan las señales de los ordenadores, típicamente cables estándar o de fibra óptica.

También hay redes sin cables que realizan la transmisión por infrarrojos o por radiofrecuencias y el adaptador de red, que permite acceder al medio material que conecta a los ordenadores, recibir paquetes desde el *software* de red y transmitir instrucciones y peticiones a otras computadoras. La información se transfiere en forma de dígitos binarios, o *bits* (unos y ceros), que pueden ser procesados por los circuitos electrónicos de los ordenadores.

2.1.1. Conceptos y componentes básicos

Las computadoras poseen varios componentes, se abordaran los siguientes componentes, ya que ellos se encuentran en casi todas las computadoras.

2.1.1.1. Tarjeta de red

Para lograr el enlace entre las computadoras y los medios de transmisión (cables de red o medios físicos para redes alámbricas e infrarrojos o radiofrecuencias para redes inalámbricas), fue necesaria la intervención de una tarjeta de red o NIC (*Network Card Interface*) con la cual se puedan enviar y recibir paquetes de datos desde y hacia otras computadoras, empleando un protocolo para su comunicación y convirtiendo esos datos a un formato que pueda ser transmitido por el medio.

Cabe señalar que a cada tarjeta de red le es asignado un identificador único por su fabricante, conocido como dirección MAC (*Media Access Control*), que consta de 48 bits (6 bytes). Dicho identificador permite direccionar el tráfico de datos de la red del emisor al receptor adecuados.

El trabajo del adaptador de red es el de convertir las señales eléctricas que viajan por el cable o las ondas de radio en una señal que pueda interpretar el ordenador. Estos adaptadores son unas tarjetas PCI que se conectan en las ranuras de expansión del ordenador. En el caso de ordenadores portátiles, estas tarjetas vienen en formato PCMCIA. En algunos ordenadores modernos, tanto de sobremesa como portátiles, estas tarjetas ya vienen integradas en la placa base.

Adaptador de red es el nombre genérico que reciben los dispositivos encargados de realizar dicha conversión. Esto significa que estos adaptadores pueden ser tanto *Ethernet* como *Wireless*, así como de otros tipos como fibra óptica, coaxial, etc. También las velocidades disponibles varían según el tipo de adaptador; éstas pueden ser, en *Ethernet*, de 10, 100 ó 1000 Mbps, y en los inalámbricos de 11 ó 55 Mbps.

Figura 9. **Tarjeta de red**



Fuente: <http://www.bloginformatico.com/hardware-interno-de-una-computadora.php>

2.1.1.2. Dirección IP

Es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (*Internet Protocol*), que corresponde al nivel de red del protocolo TCP/IP. Dicho número no se ha de confundir con la dirección MAC que es un número hexadecimal fijo que es asignado a la tarjeta o dispositivo de red por el fabricante, mientras que la dirección IP se puede cambiar.

Esta dirección puede cambiar 2 ó 3 veces al día; y a esta forma de asignación de dirección IP se denomina una dirección IP dinámica (normalmente se abrevia como IP dinámica).

Los sitios de internet que por su naturaleza necesitan estar permanentemente conectados, generalmente tienen una dirección IP fija (comúnmente, IP fija o IP estática), es decir, no cambia con el tiempo. Los servidores de correo, DNS, FTP públicos, y servidores de páginas web necesariamente deben contar con una dirección IP fija o estática, ya que de esta forma se permite su localización en la red.

A través de Internet, las computadoras se conectan entre sí mediante sus respectivas direcciones IP. Sin embargo, a los seres humanos nos es más cómodo utilizar otra notación más fácil de recordar y utilizar, como los nombres de dominio; la traducción entre unos y otros se resuelve mediante los servidores de nombres de dominio DNS.

2.1.1.3. Servidor

En informática, un servidor es un tipo de *software* que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese *software*, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen *hosting* u hospedaje.

Alternativamente, el servidor web podría referirse al *software*, como el servidor de http de *Apache*, que funciona en la máquina y maneja la entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

Dependiendo del trabajo a desempeñar, se puede tener una clasificación amplia de los servidores, como la presentada a continuación:

- Servidor de archivos: almacena varios tipos de archivo y los distribuye a otros clientes en la red.
- Servidor de impresiones: controla una o más impresoras y acepta trabajos de impresión de otros clientes de la red, poniendo en cola los trabajos de impresión (aunque también puede cambiar la prioridad de las diferentes impresiones), y realizando la mayoría o todas las otras funciones que en un sitio de trabajo se realizaría para lograr una tarea de

impresión si la impresora fuera conectada directamente con el puerto de impresora del sitio de trabajo.

- Servidor de correo: almacena, envía, recibe, enruta y realiza otras operaciones relacionadas con *e-mail*, para los clientes de la red.
- Servidor de fax: almacena, envía, recibe, enruta y realiza otras funciones necesarias para la transmisión, la recepción y la distribución apropiadas de los fax.
- Servidor de la telefonía: realiza funciones relacionadas con la telefonía, como es la de contestador automático, realizando las funciones de un sistema interactivo para la respuesta de la voz, almacenando los mensajes de voz, encaminando las llamadas y controlando también la red o el Internet; p. ej., la entrada excesiva del IP de la voz (VoIP), etc.
- Servidor *proxy*: realiza un cierto tipo de funciones a nombre de otros clientes en la red, para aumentar el funcionamiento de ciertas operaciones (p. ej., depositar documentos u otros datos que se soliciten muy frecuentemente). También brinda seguridad; esto es, tiene un *firewall*. Permite administrar el acceso a internet en una red de computadoras permitiendo o negando el acceso a diferentes sitios web.
- Servidor del acceso remoto (RAS): controla las líneas de módem de los monitores u otros canales de comunicación de la red, para que las peticiones conecten con la red de una posición remota, responden llamadas telefónicas entrantes o reconocen la petición de la red y realizan los chequeos necesarios de seguridad y otros procedimientos necesarios para registrar a un usuario en la red.

- Servidor de uso: realiza la parte lógica de la informática o del negocio de un uso del cliente, aceptando las instrucciones para que se realicen las operaciones de un sitio de trabajo y sirviendo los resultados a su vez al sitio de trabajo, mientras que el sitio de trabajo realiza el interfaz operador o la porción del GUI del proceso (es decir, la lógica de la presentación) que se requiere para trabajar correctamente.
- Servidor web: almacena documentos HTML, imágenes, archivos de texto, escrituras, y demás material Web compuesto por datos (conocidos colectivamente como contenido), y distribuye este contenido a clientes que la piden en la red.
- Servidor de reserva: tiene el *software* de reserva de la red instalado y tiene cantidades grandes de almacenamiento de la red en discos duros u otras formas del almacenamiento (cinta, etc.) disponibles para que se utilice con el fin de asegurarse de que la pérdida de un servidor principal no afecte a la red. Esta técnica también es denominada *clustering*.
- Impresoras: muchas impresoras son capaces de actuar como parte de una red de ordenadores sin ningún otro dispositivo, tal como un *print server*, a actuar como intermediario entre la impresora y el dispositivo que está solicitando un trabajo de impresión de ser terminado.
- Terminal tonto: muchas redes utilizan este tipo de equipo en lugar de puestos de trabajo para la entrada de datos. En estos sólo se exhiben datos o se introducen. Este tipo de terminales, trabajan contra un servidor, que es quien realmente procesa los datos y envía pantallas de datos a los terminales.

- Servidor de autenticación: es el encargado de verificar que un usuario pueda conectarse a la red en cualquier punto de acceso, ya sea inalámbrico o por cable, basándose en el estándar 802.1x y puede ser un servidor de tipo *RADIUS*.
- Servidor DNS: este tipo de servidores resuelven nombres de dominio sin necesidad de conocer su dirección IP, es un sistema de nombres que permite traducir de nombre de dominio a dirección IP y vice-versa.

2.1.1.4. Switch o conmutador

Un conmutador o *switch* es un dispositivo digital de lógica de interconexión de redes de computadores que opera en la capa 2 (nivel de enlace de datos) del modelo OSI.

Su función es interconectar dos o más segmentos de red, de manera similar a los puentes, pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red.

Los conmutadores se utilizan cuando se desea conectar múltiples redes, fusionándolas en una sola. Al igual que los puentes, dado que funcionan como un filtro en la red, mejoran el rendimiento y la seguridad de las LANs (Local Area Network- Red de Área Local).

Los conmutadores pueden conectarse unos a los otros pero siempre hay que hacerlo de forma que exista un único camino entre dos puntos de la red. En caso de no seguir esta regla, se forma un bucle o *loop* en la red, que produce la transmisión infinita de tramas de un segmento al otro. Generalmente estos

dispositivos utilizan el algoritmo de *spanning tree* para evitar bucles, haciendo la transmisión de datos de forma segura.

Según el método del direccionamiento de las tramas pueden ser:

- *Store-and-Forward*: los conmutadores *Store-and-Forward* guardan cada trama en un *buffer* antes del intercambio de información hacia el puerto de salida. Mientras la trama está en el *buffer*, el *switch* calcula el CRC y mide el tamaño de la misma. Si el CRC falla, o el tamaño es muy pequeño o muy grande la trama es descartada. Si todo se encuentra en orden es encaminada hacia el puerto de salida.

Este método asegura operaciones sin error y aumenta la confianza de la red. Pero el tiempo utilizado para guardar y chequear cada trama añade un tiempo de demora importante al procesamiento de las mismas. La demora o *delay* total es proporcional al tamaño de las tramas: cuanto mayor es la trama, mayor será la demora.

- *Cut-Through*: los *Switch Cut-Through* fueron diseñados para reducir esta latencia. Esos *switch* minimizan el *delay* leyendo sólo los 6 primeros *bytes* de datos de la trama, que contiene la dirección de destino MAC, e inmediatamente la encaminan.

El problema de este tipo de *switch* es que no detecta tramas corruptas causadas por colisiones (conocidos como *runts*), ni errores de CRC. Cuanto mayor sea el número de colisiones en la red, mayor será el ancho de banda que consume al encaminar tramas corruptas.

Existe un segundo tipo de *switch cut-through*, los denominados *fragment free*, fue proyectado para eliminar este problema. El *switch* siempre lee los primeros 64 bytes de cada trama, asegurando que tenga por lo menos el tamaño mínimo, y evitando el encaminamiento de *runts* por la red.

- *Adaptative Cut-Through*: los *switch* que procesan tramas en el modo adaptativo soportan tanto *store-and-forward* como *cut-through*. Cualquiera de los modos puede ser activado por el administrador de la red, o el *switch* puede ser lo bastante inteligente como para escoger entre los dos métodos, basado en el número de tramas con error que pasan por los puertos.

Cuando el número de tramas corruptas alcanza un cierto nivel, el *switch* puede cambiar del modo *cut-through* a *store-and-forward*, volviendo al modo anterior cuando la red se normalice.

Los *switch cut-through* son más utilizados en pequeños grupos de trabajo y pequeños departamentos. En esas aplicaciones es necesario un buen volumen de trabajo o *throughput*, ya que los errores potenciales de red quedan en el nivel del segmento, sin impactar la red corporativa.

Los *switch store-and-forward* son utilizados en redes corporativas, donde es necesario un control de errores.

- *Switch* de capa 2: son los *switch* tradicionales, que funcionan como puentes multi-puertos. Su principal finalidad es dividir una LAN en múltiples dominios de colisión, o en los casos de las redes en anillo,

segmentar la LAN en diversos anillos. Basan su decisión de envío en la dirección MAC destino que contiene cada trama.

Los *switch* de nivel dos posibilitan múltiples transmisiones simultáneas sin interferir en otras sub-redes. Los *switch* de capa dos no consiguen, sin embargo, filtrar difusiones o *broadcasts*, *multicasts* , ni tramas cuyo destino aún no haya sido incluido en la tabla de direccionamiento.

- *Switch* de capa 3 o *layer 3 switch*: son los *switch* que, además de las funciones tradicionales de la capa 2, incorporan algunas funciones de enrutamiento o *routing*, como por ejemplo la determinación del camino basado en informaciones de capa de red (capa 3 del modelo OSI), validación de la integridad del cableado de la capa 3 por checksum y soporte a los protocolos de *routing* tradicionales (RIP, OSPF, etc)

Los *switch* de capa 3 soportan también la definición de redes virtuales (VLAN's), y según modelos posibilitan la comunicación entre las diversas VLAN's sin la necesidad de utilizar un *router* externo.

Por permitir la unión de segmentos de diferentes dominios de difusión o *broadcast*, los *switch* de capa 3 son particularmente recomendados para la segmentación de redes LAN muy grandes, donde la simple utilización de *switch* de capa 2 provocaría una pérdida de rendimiento y eficiencia de la LAN, debido a la cantidad excesiva de *broadcasts*.

Se puede afirmar que la implementación típica de un *switch* de capa 3 es más escalable que un *router*, pues éste último utiliza las técnicas de enrutamiento a nivel 3 y encaminamiento a nivel 2 como complementos,

mientras que los *switch* sobreponen la función de enrutamiento encima del encaminamiento, aplicando el primero donde sea necesario

- *Switch* de capa 4 o *layer 4 switch*: están en el mercado hace poco tiempo y hay una controversia en relación con la adecuada clasificación de estos equipos. Muchas veces son llamados de *Layer 3+* (*Layer 3 Plus*).

Básicamente, incorporan a las funcionalidades de un *switch* de capa 3 la habilidad de implementar la políticas y filtros a partir de informaciones de capa 4 o superiores, como puertos TCP/UDP, SNMP, FTP, etc.

2.1.2. Métodos de conexión remota

Los métodos que se presentan a continuación, son los métodos más usados para comunicarse a una computadora, que se encuentra fuera de nuestra red, utilizando el internet como medio de comunicación.

2.1.2.1. Por dirección IP

Este método es el más sencillo de todos, y consiste en saber la dirección del servidor al cual queremos conectarnos, el cual está en el internet y no en una subred, y utilizar esta dirección en la configuración del sistema.

2.1.2.2. Red privada virtual (VPN)

Una red privada virtual o VPN (siglas en inglés de *virtual private network*), es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo internet.

Ejemplos comunes son: la posibilidad de conectar dos o más sucursales de una empresa utilizando como vínculo internet, permitir a los miembros del equipo de soporte técnico la conexión desde su casa al centro de cómputo.

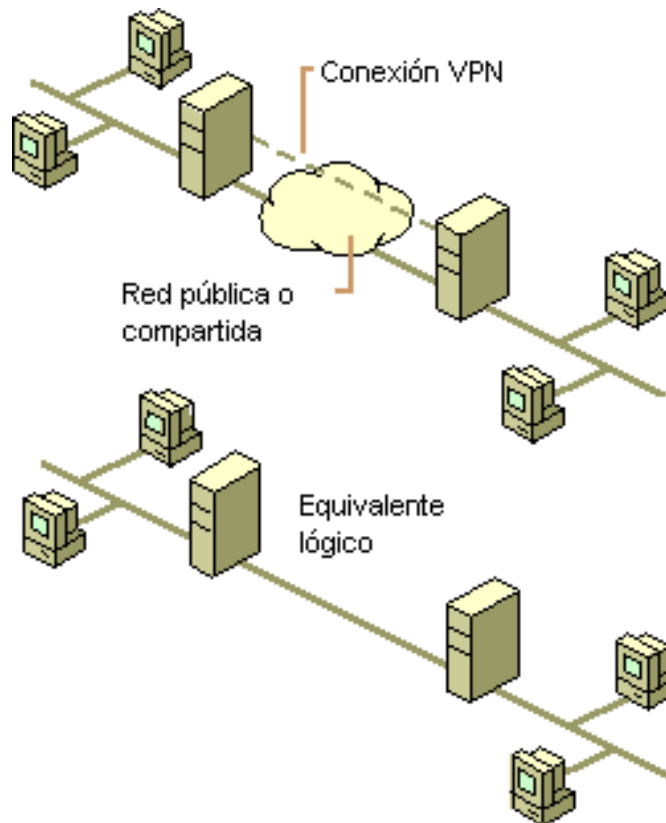
La principal ventaja de usar una VPN es que permite disfrutar de una conexión a red con todas las características de la red privada a la que queremos acceder. El cliente VPN adquiere totalmente la condición de miembro de esa red, con lo cual se le aplican todas las directrices de seguridad y los permisos de un ordenador en esa red privada.

Así se puede acceder a la información publicada para aquella red privada: bases de datos, documentos internos, etc. a través de un acceso público.

En ese momento, todas las conexiones de acceso a internet desde el ordenador cliente VPN se llevarán a cabo con los recursos y las conexiones que tenga la red privada.

Entre los inconvenientes se puede mencionar una mayor carga en el cliente VPN, ya que ha de realizar la tarea adicional de encapsular los paquetes de datos una vez más. Esta situación se agrava cuando además, se hace una encriptación de los datos que produce una mayor ralentización de la mayoría de conexiones. También se produce una mayor complejidad en el tráfico de datos, que puede producir efectos no deseados en cambiar la numeración asignada al cliente VPN y que puede requerir cambios en las configuraciones de aplicaciones o programas (proxy, servidor de correo, permisos basados en nombre y número IP).

Figura 10. **Conexión VPN**



Fuente: <http://www.uv.es/siuv/cas/zxarxa/INTCH0901es.gif>

El uso de encriptación en la conexión VPN puede ser necesario en los casos en los que la información que vaya a pasar por el túnel sea sensible y requiera privacidad.

La conexión encriptada VPN requiere bastantes recursos, tanto al servidor de túnel como al ordenador cliente de VPN, aparte de requerir la instalación de programas especiales al cliente, como se indica en el apartado correspondiente. Hay que recordar que muchas aplicaciones y muchos programas ya hacen encriptación, y cuando lo hacen, el túnel VPN no aporta seguridad adicional.

Los principales protocolos que se utilizan en este caso son:

- PPTP (Protocolo de túnel punto a punto): es un protocolo de capa 2 desarrollado por *Microsoft, 3Com, Ascend, US Robotics y ECI Telematics*.
- L2F (Reenvío de capa dos): es un protocolo de capa 2 desarrollado por *Cisco, Northern Telecom y Shiva*. Actualmente es casi obsoleto.
- L2TP (Protocolo de túnel de capa dos): el resultado del trabajo del IETF (RFC 2661), incluye todas las características de PPTP y L2F. Es un protocolo de capa 2 basado en PPP.
- IPSec: es un protocolo de capa 3 creado por el IETF que puede enviar datos cifrados para redes IP.

2.1.2.3. DINAMIC DNS (DDNS)

Este método está diseñado para que las computadoras se conecten a un servidor y no alreves, que es como por funciona generalmente, además se encarga de conectarlas entre ellas. Es necesario que cada computadora cliente tenga un *software* instalado, que se encarga de estar enviando, al servidor, la dirección IP que posee cada cliente en cada instante, se asegura que la conexión se establezca a pesar de que las direcciones IP cambien.

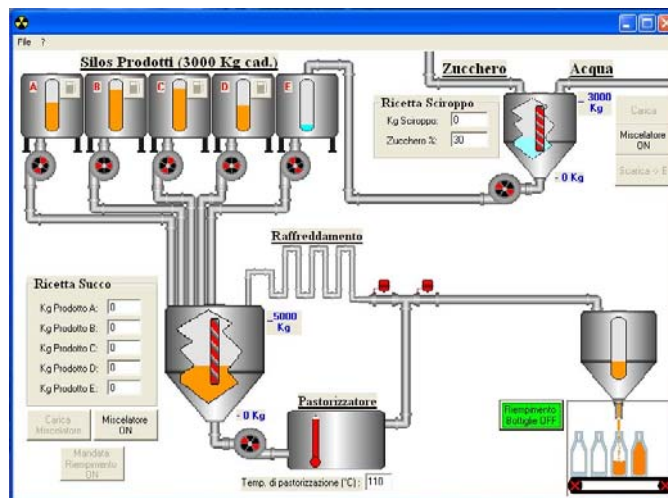
Existen varias empresas que pueden proveer este tipo de servicio. Por lo general son empresas ajenas que lo hacen para asegurar la disponibilidad del servicio en un 99.99% del tiempo

3. SISTEMAS SCADA

3.1. SCADA

SCADA es el acrónimo de *Supervisory Control And Data Acquisition* (supervisión, control y adquisición de Datos). Un SCADA es un sistema basado en computadoras que permiten supervisar y controlar a distancia una instalación de cualquier tipo. A diferencia de los sistemas de control distribuido, el lazo de control es generalmente cerrado por el operador.

Figura 11. Ejemplo SCADA



Fuente: http://www.serycon.com.ar/IMAGENES/scada_2.jpg

3.2. Pasos para la creación de un sistema SCADA

Los pasos para la creación de un sistema SCADA son tres, los cuales se listan a continuación:

3.2.1. Adquisición de datos

La adquisición de datos consiste en obtener señales adecuadas que representen los fenómenos físicos, que suceden en cada instante del proceso. Para la adquisición de estas señales es necesario que se tengan en cuenta los fenómenos físicos que se desean medir, tomando en cuenta que no todos los transductores o sensores son útiles para la misma medición.

3.2.1.1. Tipos de sensores

Existe una amplia gama de sensores y varios subtipos de los mismos. En este trabajo se presentarán los sensores más utilizados y los que son más fáciles de encontrar en el mercado, como lo son los que se muestran a continuación:

- Detectores de ultrasonidos: los detectores de ultrasonidos resuelven los problemas de detección de objetos de prácticamente cualquier material. Trabajan en ambientes secos y polvorientos. Normalmente se usan para control de presencia/ausencia, distancia o rastreo.
- Interruptores básicos: se consiguen interruptores de tamaño estándar, miniatura, subminiatura, herméticamente sellados y de alta temperatura.

Los mecanismos de precisión se ofrecen con una amplia variedad de actuadores y características operativas. Estos interruptores son idóneos para aplicaciones que requieran tamaño reducido, poco peso, repetitividad y larga vida.

- Interruptores final de carrera: El *microswitch* es un conmutador de dos posiciones con retorno a la posición de reposo y viene con un botón o con una palanca de accionamiento, la cual también puede traer una ruedita.

Funcionamiento: en estado de reposo la patita común (COM) y la de contacto normal cerrado (NC), están en contacto permanente hasta que la presión aplicada a la palanca del *microswitch* hace saltar la pequeña platina acerada interior y entonces el contacto pasa de la posición de normal cerrado a la de normal abierto (NO), se puede escuchar cuando el *microswitch* cambia de estado, porque se oye un pequeño clic, esto sucede casi al final del recorrido de la palanca.

- Interruptores manuales: estos son los sensores más básicos, incluye pulsadores, llaves, selectores rotativos y conmutadores de enclavamiento. Estos productos ayudan al técnico e ingeniero con ilimitadas opciones en técnicas de actuación y disposición de componentes.
- Productos encapsulados: diseños robustos, de altas prestaciones y resistentes al entorno o herméticamente sellados. Esta selección incluye finales de carrera miniatura, interruptores básicos estándar y miniatura, interruptores de palanca y pulsadores luminosos.
- Productos para fibra óptica: el grupo de fibra óptica está especializado en el diseño, desarrollo y fabricación de componentes optoelectrónicos activos y submontajes para el mercado de la fibra óptica.

- Productos infrarrojos: la optoelectrónica es la integración de los principios ópticos y la electrónica de semiconductores. Los componentes optoelectrónicos son sensores fiables y económicos. Se incluyen diodos emisores de infrarrojos (IREDs), sensores y montajes.
- Sensores para automoción: se incluyen sensores de efecto Hall, de presión y de caudal de aire. Estos sensores son de alta tecnología y constituyen soluciones flexibles a un bajo costo. Su flexibilidad y durabilidad hace que sean idóneos para una amplia gama de aplicaciones de automoción.
- Sensores de caudal de aire: los sensores de caudal de aire contienen una estructura de película fina aislada térmicamente, que contiene elementos sensibles de temperatura y calor. La estructura de puente suministra una respuesta rápida al caudal de aire u otro gas que pase sobre el chip.
- Sensores de corriente: los sensores de corriente monitorizan corriente continua o alterna. Se incluyen sensores de corriente lineales ajustables, de balance nulo, digitales y lineales. Los sensores de corriente digitales pueden hacer sonar una alarma, arrancar un motor, abrir una válvula o desconectar una bomba.

La señal lineal duplica la forma de la onda de la corriente captada, y puede ser utilizada como un elemento de respuesta para controlar un motor o regular la cantidad de trabajo que realiza una máquina.

- Sensores de humedad: los sensores de humedad relativa/temperatura y humedad relativa están configurados con circuitos integrados que

proporcionan una señal acondicionada. Estos sensores contienen un elemento sensible capacitivo en base de polímeros que interacciona con electrodos de platino. Están calibrados por láser y tienen una intercambiabilidad de +5% HR, con un rendimiento estable y baja desviación.

- Sensores de presión y fuerza: los sensores de presión son pequeños, fiables y de bajo costo. Ofrecen una excelente repetitividad y una alta precisión y fiabilidad bajo condiciones ambientales variables. Además, presentan unas características operativas constantes en todas las unidades y una intercambiabilidad sin recalibración.
- Sensores de temperatura: los sensores de temperatura se catalogan en dos series diferentes: TD y HEL/HRTS.

Estos sensores consisten en una fina película de resistencia variable con la temperatura (RTD) y están calibrados por láser para una mayor precisión e intercambiabilidad. Las salidas lineales son estables y rápidas.

- Sensores magnéticos: los sensores magnéticos se basan en la tecnología magnetoresistiva SSEC. Ofrecen una alta sensibilidad. Entre las aplicaciones se incluyen brújulas, control remoto de vehículos, detección de vehículos, realidad virtual, sensores de posición, sistemas de seguridad e instrumentación médica.
- Sensores de presión: los sensores de presión están basados en tecnología piezoresistiva, combinada con microcontroladores que proporcionan una alta precisión, independiente de la temperatura, y

capacidad de comunicación digital directa con PC. Las aplicaciones afines a estos productos incluyen instrumentos para aviación, laboratorios, controles de quemadores y calderas, comprobación de motores, tratamiento de aguas residuales y sistemas de frenado.

3.2.1.2. Equipo para recopilación de datos

Estos equipos de procesamiento, generalmente constan de alta capacidad, ya que es necesario tener una velocidad adecuada de muestreo, según el proceso a medir.

Entre estos equipos figuran:

- DAQ: la adquisición de datos o adquisición de señales, consiste en la toma de muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por una computadora. Es decir, consiste en tomar un conjunto de señales físicas, convertirlas en tensiones eléctricas y digitalizarlas de manera que se puedan procesar en una computadora o PAC. Se requiere una etapa de acondicionamiento, que adecua la señal a niveles compatibles con el elemento que hace la transformación a señal digital. El elemento que hace dicha transformación es el módulo de digitalización o tarjeta de Adquisición de Datos (DAQ).

DAQ *hardware* son por lo general las interfaces entre la señal y un PC. Podría ser en forma de módulos que pueden ser conectados a la computadora de los puertos (paralelo, serie, USB, etc...) o ranuras de las tarjetas conectadas a (PCI, ISA) en la placa madre.

Las tarjetas DAQ a menudo contienen múltiples componentes (multiplexores, ADC, DAC, TTL-IO, temporizadores de alta velocidad, memoria RAM). Estos son accesibles a través de un bus por un micro controlador, que puede ejecutar pequeños programas.

Como ejemplo se muestra la figura 12

Figura 12. **DAQ**



Fuente: <http://www.checkitsystems.com/images/Compact%20Daq3.jpg>

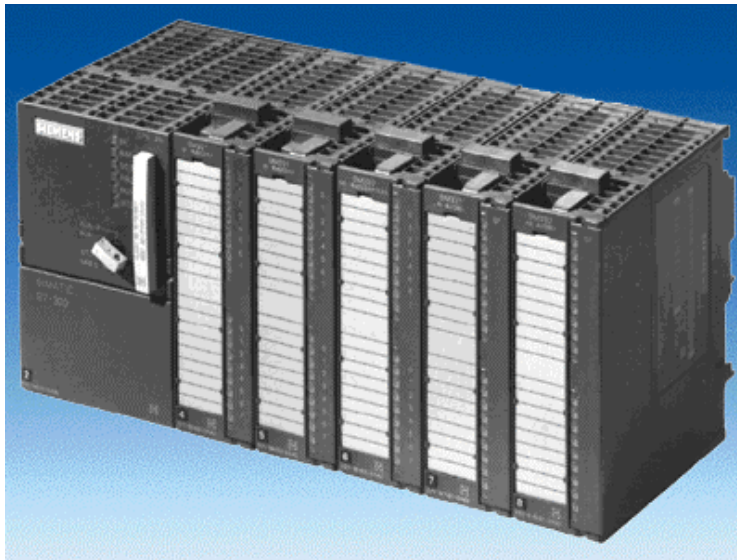
- PLC: controlador de lógica programable, son dispositivos electrónicos muy usados en Automatización Industrial.

Es un *hardware* industrial, que se utiliza para la obtención de datos. Una vez obtenidos, los pasa a través de bus (por ejemplo, por *ethernet*) en un servidor.

Hoy en día, los PLC no sólo controlan la lógica de funcionamiento de máquinas, plantas y procesos industriales, sino que también pueden

realizar operaciones aritméticas, manejar señales analógicas para realizar estrategias de control, tales como controladores proporcional integral derivativo (PID).

Figura 13. **PLC**



Fuente:

http://images.industrial.omron.es/IAB/Products/Automation%20Systems/PLCs/Compact%20PLC%20Series/CP1L/CP1L%20CPU%20Units/images/CP1L_left2400x400.jpg

3.2.2. Análisis de los datos adquiridos

Luego de obtener los datos es necesario hacer que estos sean entendidos por los operarios, lo que obliga a hacerle transformaciones, escalamientos, operaciones matemáticas, etc.

3.2.2.1. Algunos tipos de análisis

- Análisis estadístico: utiliza la estadística y sus herramientas para hacer inferencias sobre las variables del sistema y por lo tanto por su comportamiento, algunas herramientas utilizadas en el análisis estadístico son:

- Varianza
- Correlación
- Covarianza
- Histograma
- Jerarquía y percentil
- Regresión
- Muestreo
- Prueba T
- Prueba Z

- Análisis matemático: utiliza las herramientas matemáticas para hacer los ajustes necesarios a los datos para su futuro despliegue, algunas de las herramientas son:

- Adición
- Sustracción
- Multiplicación
- División
- Porcentajes
- Transformada Z
- Transformada de Laplace
- Transformada de Fourier
- Series complejas
- Etc.

3.2.3. Presentación de los datos en una interfaz gráfica

Esta parte consiste en la presentación de los datos al usuario final, es decir al operario. Los datos deberán estar en unidades adecuadas, con imágenes que representen adecuadamente cada etapa del proceso con sus

respectivas variables; además la interfaz entre humano y máquina, HMI (*Human machine interface*), debe ser capaz de desplegar al operario, las alarmas que sucedan en el instante preciso de su acontecimiento ya que de esa forma el usuario puede hacer los ajustes necesarios para que el proceso siga o si es necesario, detener el proceso.

4. BASES DE DATOS Y VISUALIZACION WEB

4.1. Base de Datos

Una base de datos o banco de datos (en ocasiones abreviada BB.DD.) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

4.2. Gestores de bases de datos

Un gestor de base de datos es un programa diseñado para la manipulación de los datos almacenados en la base de datos, algunos de los gestores más populares son:

- *MySql*: es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita su uso. Este sistema desarrollado en su mayor parte en ANSI C. el logotipo del programa es el que se encuentra en la figura 14

Figura 14. Logotipo MySQL



Fuente: <http://cmop17.files.wordpress.com/2009/06/logo-mysql.jpg>

- *Oracle*: es un sistema de gestión de base de datos relacional multiplataforma desarrollado por *Oracle Corporation* con logotipo como el que se encuentra en la figura 15.

Se considera a *Oracle* como uno de los sistemas de bases de datos más completos, destacando:

- soporte de transacciones
- estabilidad
- escalabilidad
- soporte multiplataforma

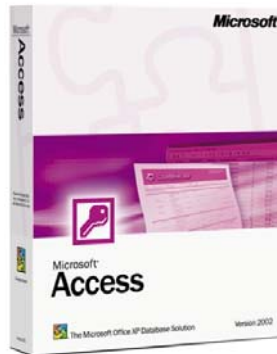
Figura 15. **Logotipo Oracle**



Fuente: <http://7safe.files.wordpress.com/2009/10/oracle.jpg>

- Access: es un programa sistema de gestión de base de datos, con logotipo en la figura 16, relacional creado y modificado por Microsoft para uso personal en pequeñas organizaciones. Es un componente de la suite Microsoft Office, aunque no se incluye en el paquete "básico". Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas. Este programa permite manipular datos en forma de tablas (lo que es la unión de filas y columnas), realizar cálculos complejos con fórmulas y funciones, e incluso dibujar distintos tipos de gráficas.

Figura 16. **Logotipo MS Access**



Fuente:http://www.ctec.com.ar/Cursos_pc_cursos/microsoft_access_2002.jpg

4.3. **Página web**

Una página web es una página sin estructura en la que se pueden insertar texto, imágenes, tablas, e incrustar hojas de datos, presentaciones, vídeos y otros materiales. La página web tiene unos controles de formato estándar como negrita, cursiva, subrayado, selección de fuentes, color del texto y resaltar texto. Se pueden crear listas con viñetas, listas numeradas y establecer con facilidad enlaces a otras páginas del sitio.

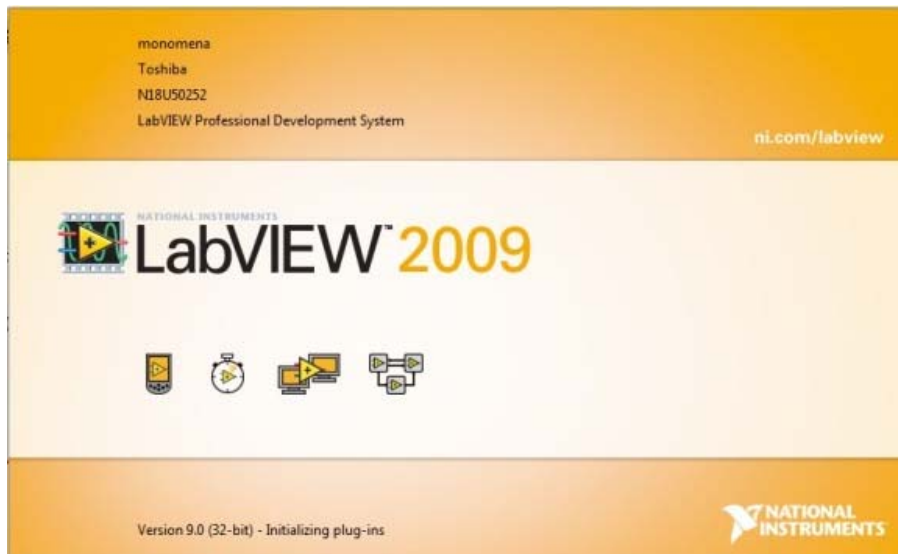
Existen varios lenguajes de programación de páginas web, entre los cuales figuran:

- HTML
- Java Script
- Php
- Flash

5. DISEÑO EN LABVIEW

La implementación del sistema scada y todos los ejemplos que a continuación se muestran, serán realizados en el *software* de *National Instruments* llamado *Labview*, específicamente la versión 2009 v9.0 para sistemas de 32bits. Se puede observar la versión en la figura 17

Figura 17. **Labview 2009**



5.1. **Labview**

Labview es un *software* diseñado para una amplia gama de objetivos, es un lenguaje tipo G, es decir que es de programación gráfica, no con pseudocódigo, aunque no implica que no se puede introducir pseudocódigo en el lenguaje.

Este *software* fue diseñado para la automatización industrial, por eso es que tiene muchas herramientas, las que también se pueden utilizar en el análisis de otros procesos que no son puramente industriales.

5.1.1. Partes básicas de *Labview*.

Labview consta de varias partes básicas que se explican a continuación, para que el lector pueda entender a los términos a los que nos referiremos más adelante

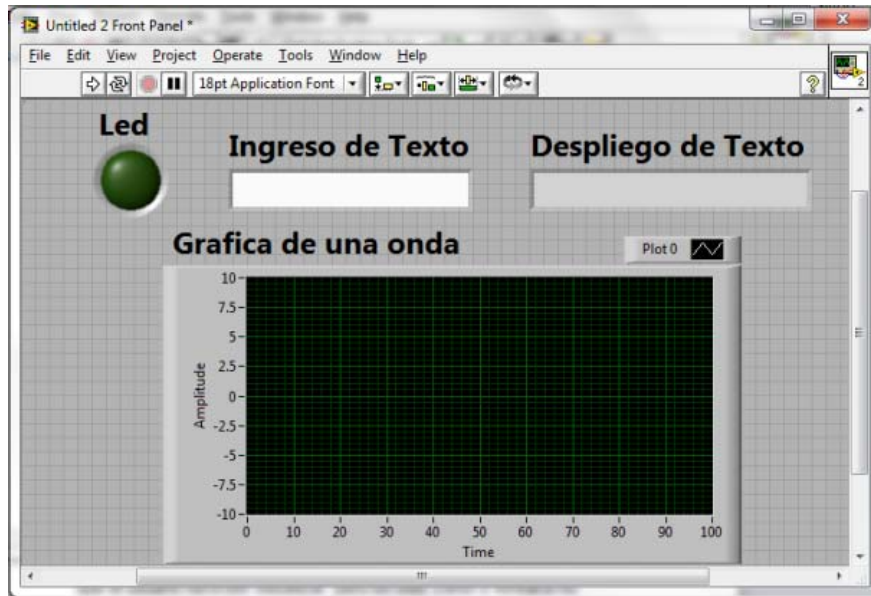
- *Labview core*: se refiere a la parte más vital del programa, es decir, su núcleo. Esta parte es la que primero se debe instalar incluye los *toolkits* básicos para poder trabajar con él.
- *Toolkit*: se refiere a un agregado que se le puede hacer al *core*, para poder tener más funciones, es decir que son herramientas extras que se le pueden agregar al programa, como por ejemplo el *Database conectiviti toolkit*, el que provee a *Labview* las herramientas necesarias para conectar todos los programas que se realicen en *Labview* hacia una base de datos donde se puedan almacenar los datos procesados.
- Pantalla de inicio: es la ventana que aparece al iniciar *Labview*, la cual tiene varias opciones como: iniciar un proyecto, o abrir un VI en blanco, además de mostrar los últimos programas que se han editado. Véase figura 18.

Figura 18. Pantalla principal de *Labview*



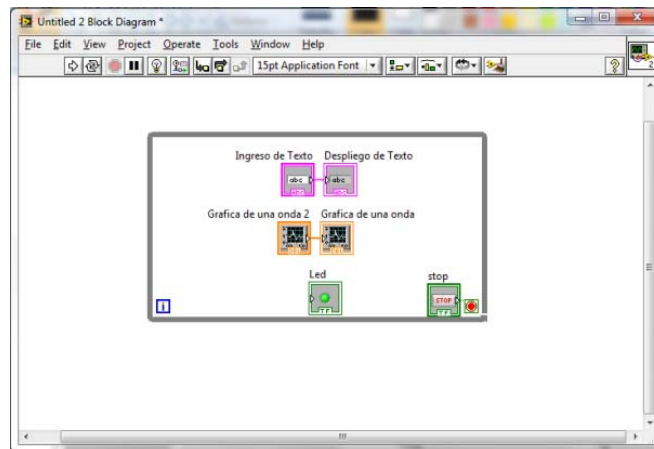
- VI: con este término se conoce a un instrumento virtual por sus siglas en inglés *Virtual Instrument*, con lo que se refiere a todos los programas que se creen, que serán llamados instrumentos virtuales, ya que los programas son como herramientas, solo que como no existen físicamente, como lo puede ser un barreno, son llamados virtuales.
- Panel Frontal: se refiere a la ventana, la que es una de las dos partes necesarias del VI, en la cual debe ir todas las partes gráficas como *leds*, casillas para desplegar texto, casillas para ingresar texto, gráficas, etc. Además se incluyen una amplia gama de cosas que el usuario necesite visualizar, pero en este panel o ventana no se programa, como se observa en la figura 19.

Figura 19. Panel frontal



- Diagrama de bloques: esta es la segunda parte necesaria del VI y es la parte que une los elementos gráficos del panel frontal y les da la funcionalidad necesaria, es decir, que en esta ventana del VI es donde se programa, como se observa en la figura 20.

Figura 20. Diagrama de bloques



- Paleta de herramientas: este menú muestra las diferentes funciones que puede tener el cursor dentro de *Labview*, ya sea para conectar bloques, hasta seleccionar un color específico para cada parte. Se puede acceder a él desde *VIEW-TOOLS PALLET*. Se puede observar la paleta de herramientas en la figura 21.

Figura 21. **Paleta de herramientas**



- Paleta de funciones: este menú es el más utilizado al momento de programar; en él se encuentran todas las funciones las cuales posee *Labview* y a través del mismo, es posible acceder a este menú con simplemente dar un clic derecho del mouse dentro del diagrama de bloques y se obtendrá la siguiente paleta que se muestra en la figura 22.
- Paleta de control: esta paleta permite acceder a todos los elementos gráficos que se pueden poner dentro del panel frontal, como lo son controles, indicadores, gráficas, formulas, etc. Esta paleta se muestra en la figura 23.

Figura 22. **Paleta de funciones**

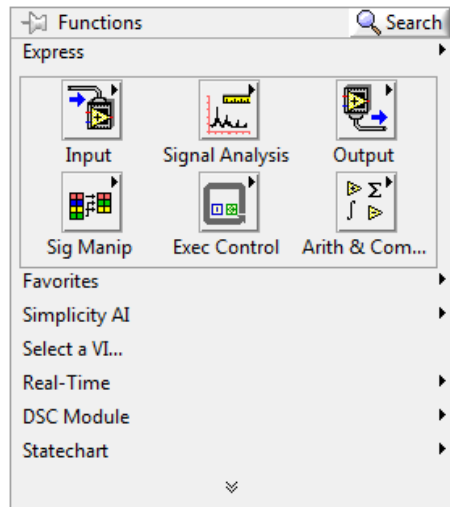
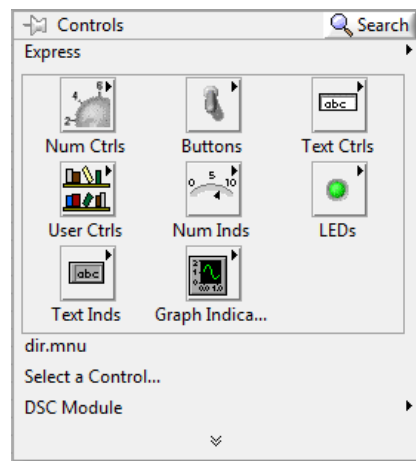


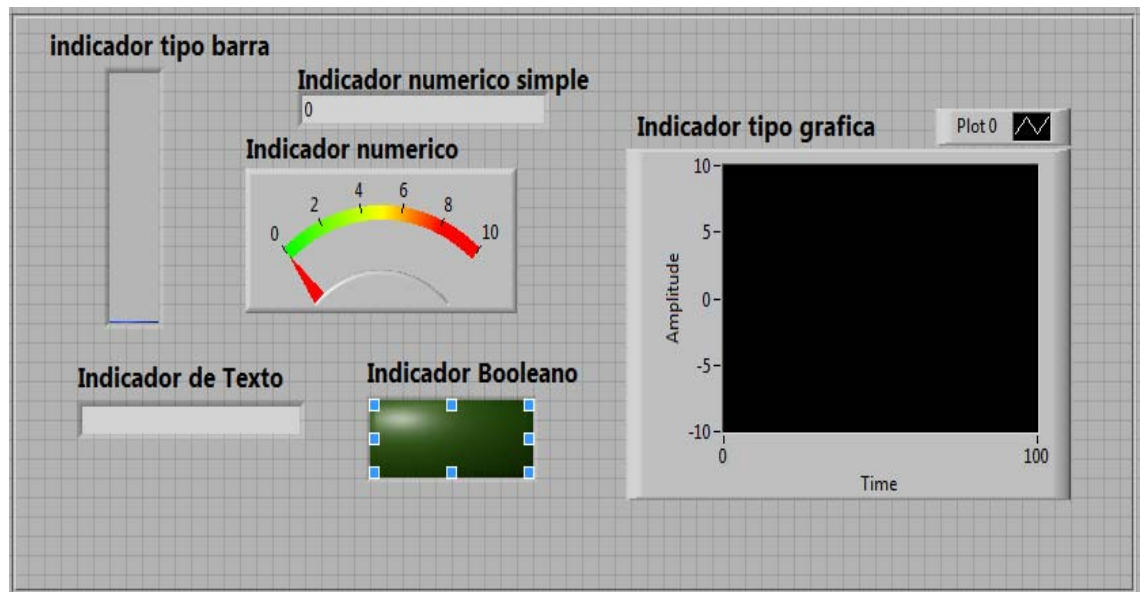
Figura 23. **Paleta de controles**



- Indicadores: son todos los instrumentos gráficos que muestran el estado actual de las diferentes partes del sistema, como pueden ser leds, indicadores numéricos, indicadores de texto, matrices, etc. En la figura 24 se muestran algunos de los indicadores que posee *Labview*

Todos estos elementos se instalan en el panel frontal, pero también aparecen en el diagrama de bloques, para ser utilizados en la programación. Se pueden encontrar dándole *click* derecho en el panel frontal e ingresando donde dice *Indicators*.

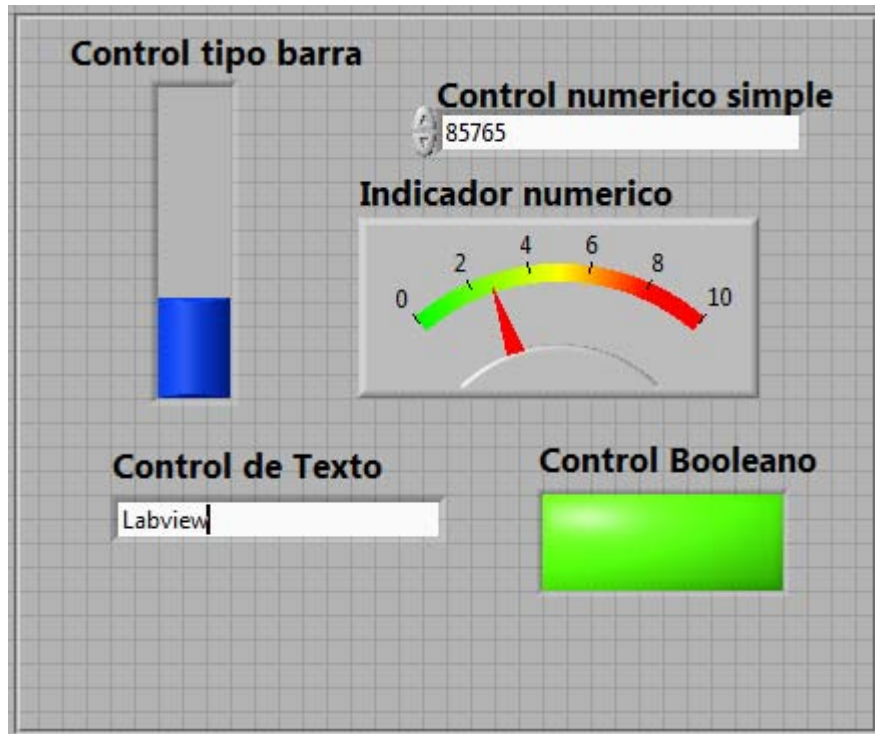
Figura 24. **Indicadores**



- **Controles:** estos elementos sirven para ingresar valores, letras, oraciones, o cambiar estados dentro del programa. Existe una amplia gama de controles, y de ser necesario es posible crear un control hecho a la medida. Como ejemplo vea figura 25.

Se pueden encontrar dándole clic derecho en el panel frontal e ingresando donde dice *Controls*,

Figura 25. **Controles**



5.1.2. **Programación en *Labview***

Esta programación es muy sencilla. El principio básico es que cada salida es mostrada si todas sus entradas están listas y presentes.

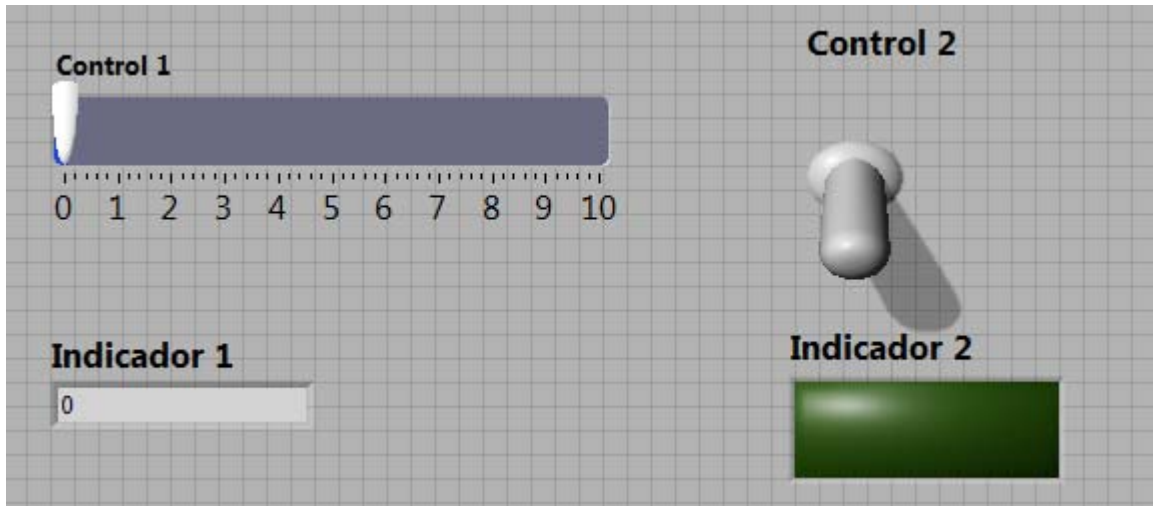
Se mostrará un pequeño ejemplo, en el cual, lo que se escribe en un control es desplegado en un indicador.

Ejemplo: programación básica 1

Utilización de los controles e indicadores para el ingreso y despliegue de datos en el programa.

Primero se colocan los indicadores y controles correspondientes, todos estos se encuentran en la paleta de herramientas o *tools palette*.

Figura 26. **Panel frontal programación básica 1**



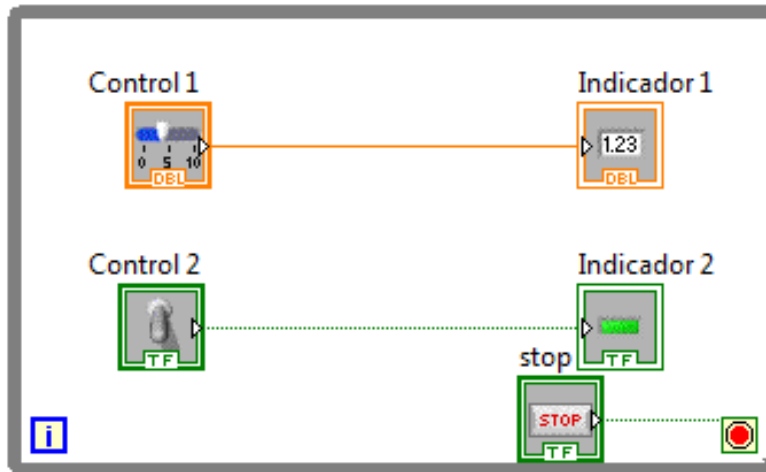
Se unen los bloques del diagrama de bloques con una línea que se puede hacer, dándole clic a las terminales del primer bloque y luego uniendo la línea hacia la terminal del otro bloque y de esta forma la salida del control será conectada a la entrada del indicador, como se observa en la figura 27.

Figura 27. **Conexión bloques programación básica 1**



Ahora solo falta encerrar esos bloques en un ciclo *while* para que se ejecute y no se detenga el programa hasta que se presione el botón de paro. Este ciclo se puede encontrar en la paleta de herramientas en *Express-Exec Control-While*. En la figura 28 puede observar cómo debe quedar.

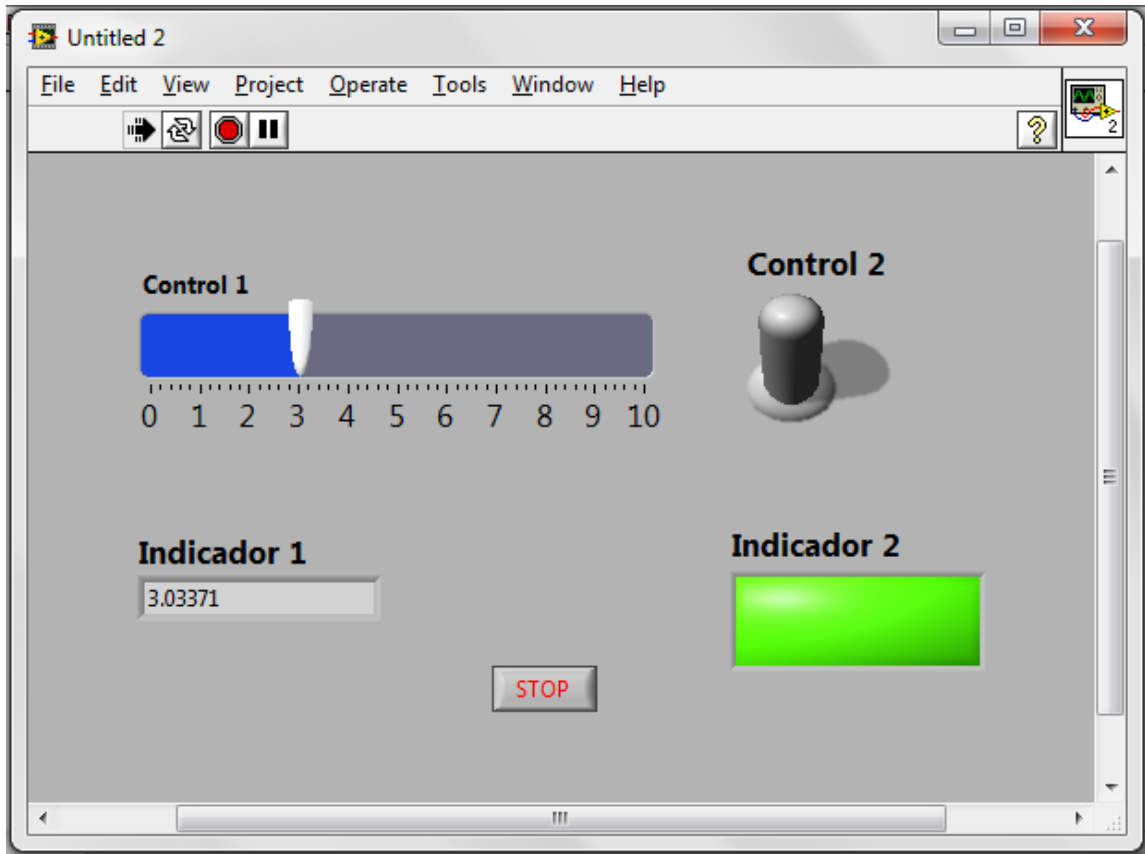
Figura 28. Diagrama de bloques programación básica 1



Y de esta forma al correr el vi al presionar la flecha *RUN* en la barra de herramientas, o simplemente pulsando *ctrl+R* comenzará a ejecutarse el programa y se podrá observar que lo que se ingrese en el control, será desplegado en el indicador.

El VI de la figura 29 está ejecutándose y si se observa detenidamente, se puede ver que lo que está en la entrada se presenta en la salida

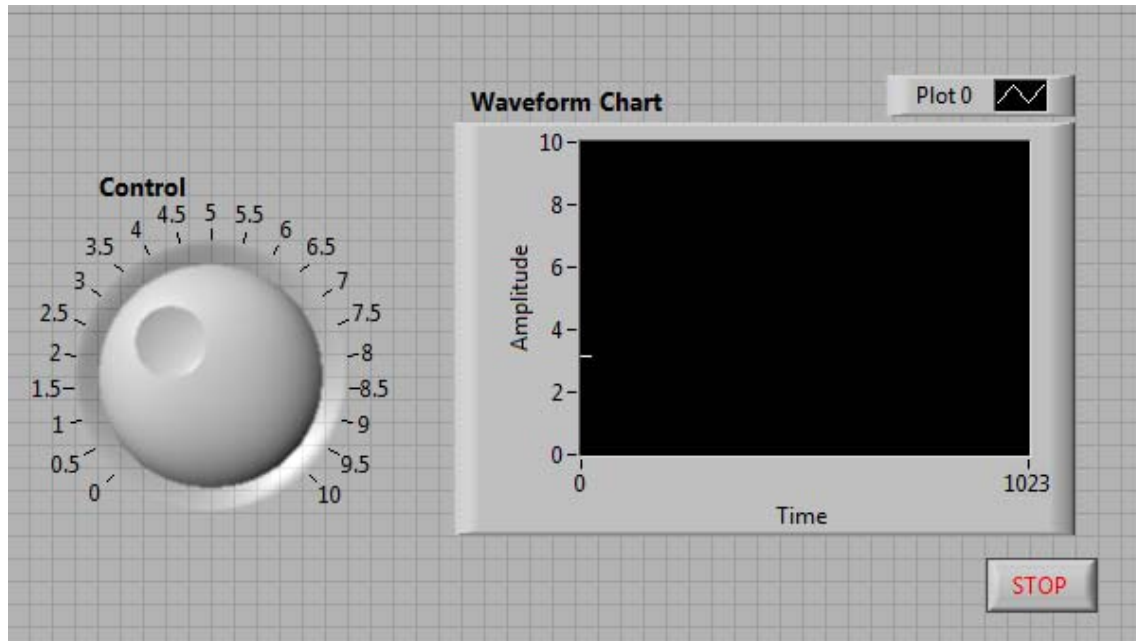
Fig. 29. Corriendo programación básica 1



Ejemplo: programación básica 2

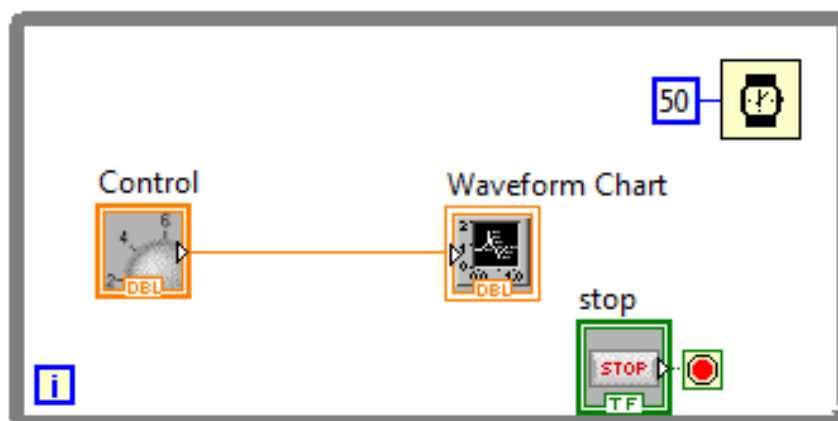
Vea la figura 30, la función de este VI será que al mover el *dimmer*, se almacenen los datos en la gráfica, para así observar cómo varía el estado del mismo.

Figura 30. Panel frontal programación básica 2



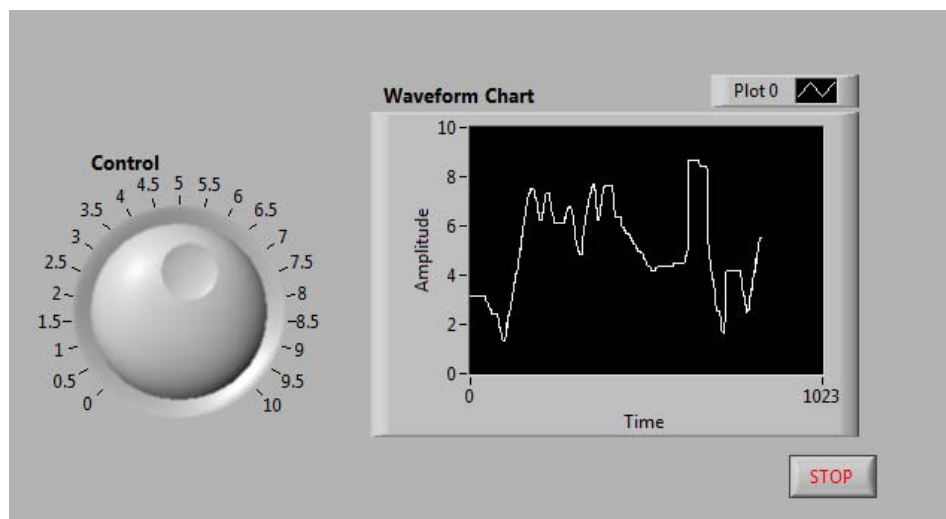
El diagrama de bloques deberá quedar así como se muestra en la figura 31. En esta figura se muestra que es sencillo presentar los datos de un control en un indicador, tan sencillo que sólo es necesario unirlos mediante una línea. El bucle en el cual se encuentran encerrados los elementos, es un bloque tipo *while*, es decir, que se ejecutará hasta que el usuario presione el botón *STOP*

Figura 31. Diagrama de bloques programación básica 2



El bloque del reloj, se puede encontrar dentro de la paleta de herramientas en *Programming – Timming – Wait*. Este bloque sirve para crear un retardo entre cada ejecución del ciclo y así observar los cambios realizados. Al ir moviendo la perilla, con el paso del tiempo se generará una gráfica como se observa en la figura 32.

Fig. 32. **Corriendo programación básica 2**



5.1.2.1. **Creación de un control a la medida**

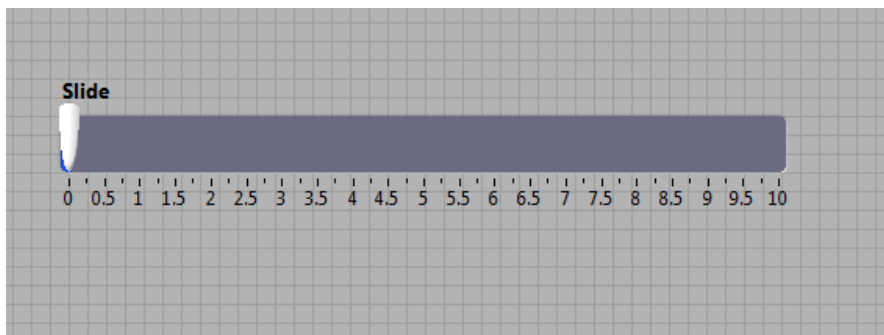
Los controles o indicadores que se desean no siempre se encuentran dentro de la paleta de controles, como lo podría ser una banda transportadora pero, los controles que dentro de la paleta pueden ser modificados para adquirir una forma distinta y también una función distinta.

Ejemplo: control a la medida

Lo que se hará a continuación será modificar un control de los que vienen con *Labview* para crear una banda transportadora, simulando así, que una botella va caminando encima de la banda.

Primero se coloca un *Horizontal Pointer Slide* que se encuentra dentro de la paleta de funciones, en *Moder-Numeric-Horizontal Pointer Slide* de manera que el panel frontal quede de la siguiente manera. Véase figura 33.

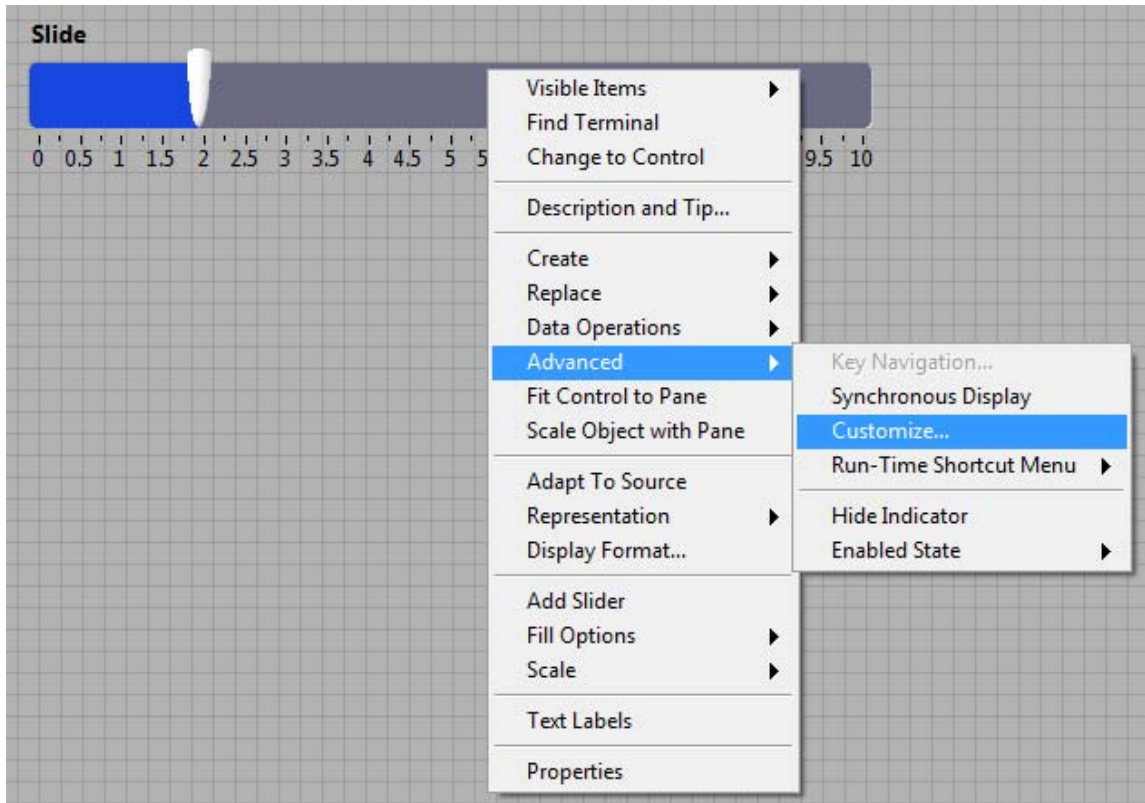
Figura 33. **Control a modificar**



Luego se selecciona el control y se hace clic derecho sobre él, y se selecciona la opción *Change to Indicator* de tal forma que ahora se convirtió en un indicador.

Se selecciona nuevamente la barra que ahora es un indicador, y se hace clic derecho sobre ella y se da *click* en *Advance –Customize* como se observa en la figura 34.

Fig. 34. Acceso al menu de modificación



Entonces aparecerá la ventana de la figura 35

Esta ventana ofrece opciones que ninguna otra ventana ofrece, por eso es que a través de ella se puede modificar los controles e indicadores para que realicen las funciones necesarias.



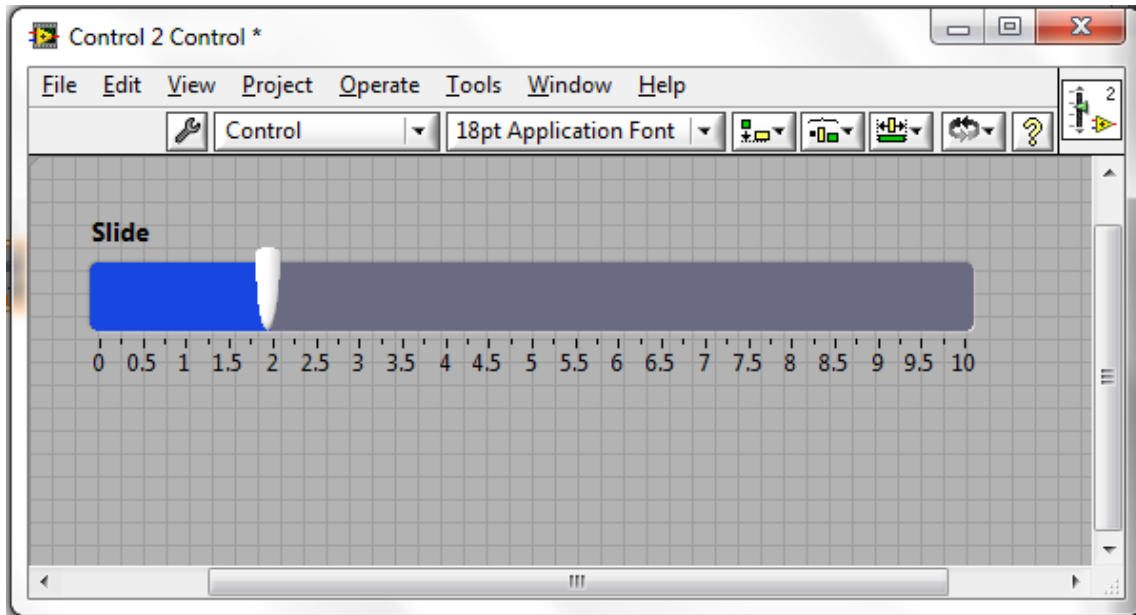
Una vez en la ventana de modificación, se da clic sobre el ícono que tiene una llave de herramientas ; al hacer esto, se le indica a *Labview* que cambie al modo de edición de ese instrumento. Para corroborar que todo esté bien, la llave tuvo que haber cambiado a esto . Si es así, todo está en orden.

Fig. 35. **Ventana de modificación**



Ahora se procederá a buscar la imagen de la banda dentro de la librería de imágenes que se encuentra integrada en *Labview*.

La imagen no tiene que estar sólo en la librería puesto que podría adquirirse la imagen desde cualquier archivo, pero por cuestión de comodidad y didáctica se hará desde la librería. Se selecciona el menú *TOOLS-DSC MODULE- IMAGE NAVIGATOR* y aparecerá el navegador de imágenes

A través del navegador mostrado en la figura 36, se puede encontrar un montón de imágenes que pueden servir para realizar el efecto deseado.

Como siguiente paso, se procede a sustituir la imagen del control por una banda transportadora y también se moverá la barrita indicadora, para luego sustituirla por una botella.

Figura 36: Biblioteca de imágenes



En el navegador es necesario ubicarse donde dice *conveyors* y se escoge una banda cualquiera, se selecciona y se le hace clic derecho y copiar, luego se ubica la ventana de modificación y se mueve la cuña blanca del indicador hacia arriba para preparar el control los respectivos reemplazos de imágenes.

Se selecciona la barra central y se le hace clic derecho y se selecciona *Import from Clipboard at the same size* de esta forma, se pegará la imagen seleccionada en el lugar donde estaba la barra central, se pueden observar realizados en las figuras 37 y 38.

Ahora se hará el mismo procedimiento, sustituyendo el puntero blanco por una botella cualquiera de manera que quede como en la figura 39.

Figura 37. **Control antes**

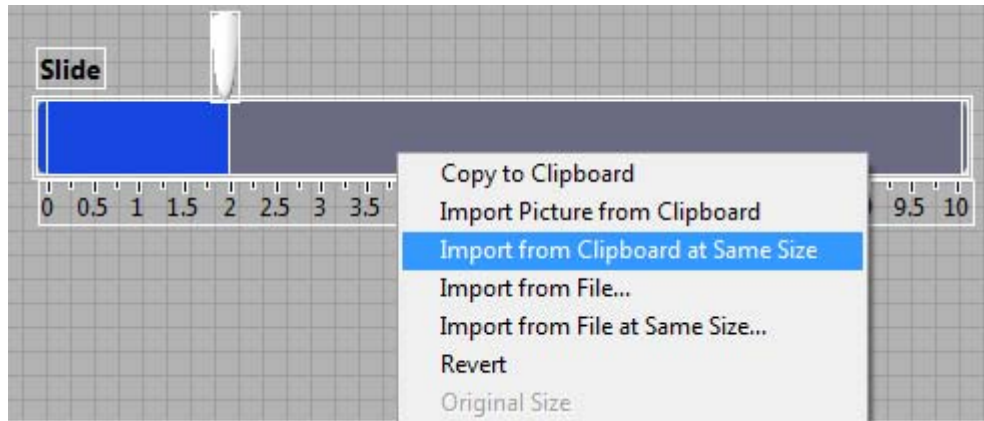


Figura 38. **Control después**

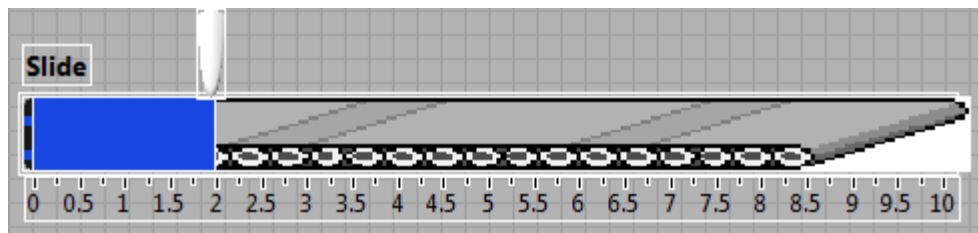
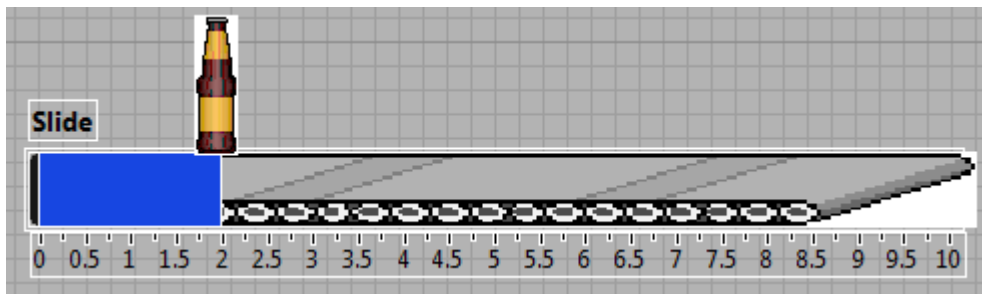




Fig. 39. **Reemplazo del cursor**



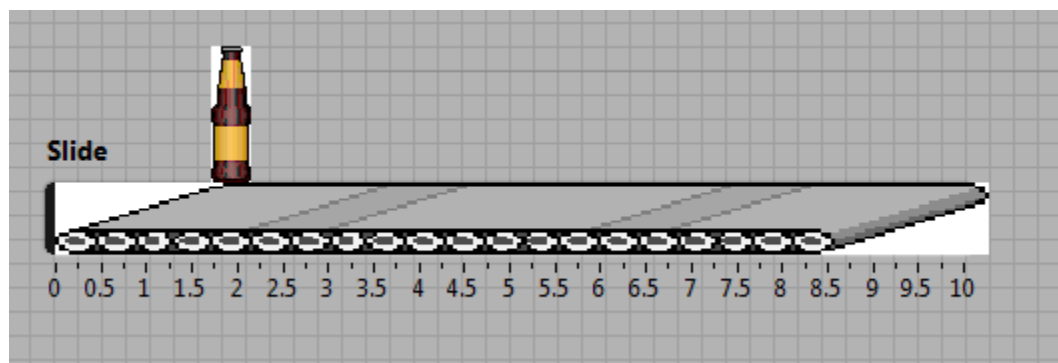
Ahora casi parece una banda transportadora; entonces se hace clic en el ícono con las pinzas , para indicar que ya se terminaron los cambios y cambiara de nuevo por el ícono con la llave de tuercas  para mostrar que

ahora no se puede modificar. Cabe mencionar que si se desea volver a modificar, puede hacerse dándole clic de nuevo sobre la llave de tuercas.

Para eliminar la barra azul, se hace clic derecho sobre el indicador modificado y luego en *Fill Options- No Fill*.

Con los pasos realizados en los párrafos anteriores, se ha modificado un control tipo slider para que se vea como una banda transportadora, como la que se observa en la figura 40; de la misma forma se pueden utilizar todos los controles que provee *Labview* para crear nuevos, pero más que todo, el límite que se tiene es la creatividad de cada programador.

Figura 40. **Control completo**



Ahora se debe guardar el control en la carpeta que se desee por medio del menú Archivo y salir de la ventana de modificación. Al salir, se preguntará si se desea reemplazar el indicador del panel frontal con el que ya se modificó, luego se selecciona la opción “sí” la cual aparece en la figura 41.

De tal manera que se vea el nuevo indicador en el panel frontal, como se muestra en la figura 42.

Figura 41. **Pregunta de remplazo**

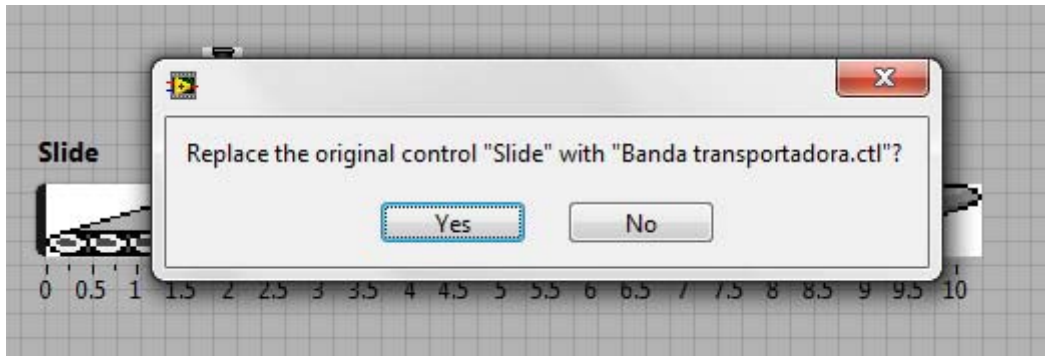
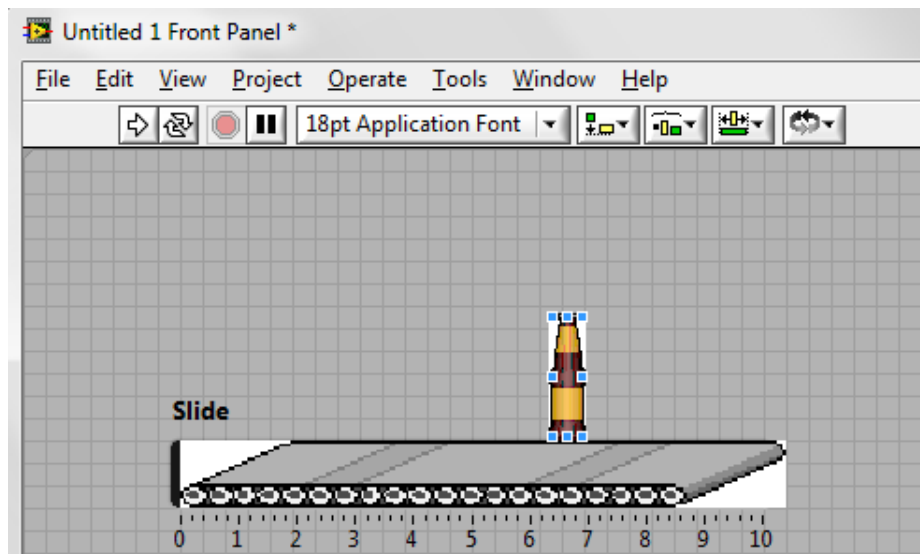


Figura 42. **Control terminado en el panel frontal**



Se remueven los números de la parte inferior, seleccionando el indicador, dándole clic derecho y en *Scale – Style* y se selecciona el cuadro en blanco.

Para esconder el nombre del control, se hace clic derecho sobre el control, y donde dice *visible* se deselecciona el nombre.

De esta manera se tiene un indicador que al momento de ingresarle datos, la botella indicará el número, pero si hacemos un ciclo *While* se podrá observar

Se pueden hacer muchas modificaciones, como lo podría ser que la botella parezca que se va llenando a lo largo de la banda, pero eso ya depende del ingenio del programador.

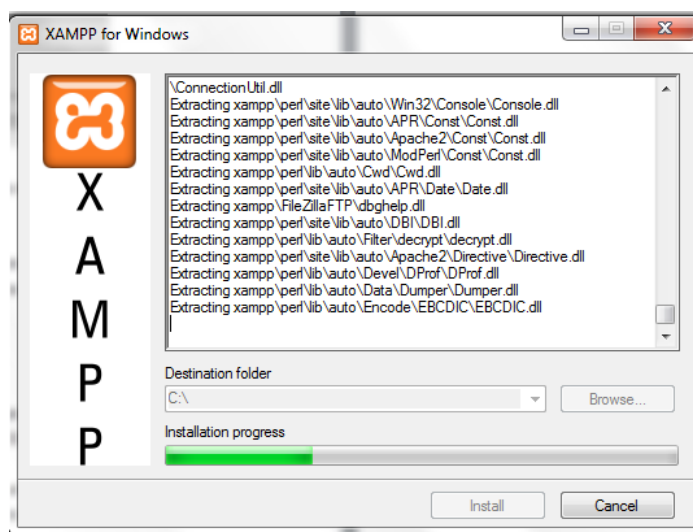
5.1.2.2. Conexión de *Labview* a *MySql*

En esta sección se observará cómo conectarse a la base de datos a través de *Labview* mediante el uso los conectores ODBC provistos por *MySql*.

Se instalara primero *Xampp*, el cual es un paquete de *software* que trae *Apache*, *Filezilla*, *Mercury* y *MySql*, siendo este último el de nuestro interés.

Instalar *Xampp* es bien sencillo, se ejecuta el instalador y se hace clic en *install*; básicamente este no es un instalador, sino más bien es un desempaquetador que al final corre unos archivos *.bat* que agregan los registros necesarios, como se muestra en la figura 45.

Figura 45. Instalando Xampp

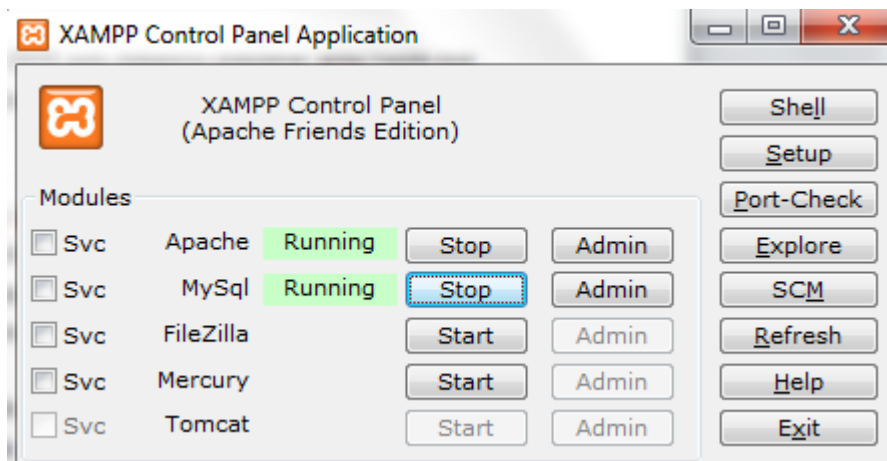


Cuando salga la consola en DOS sólo debemos presionar enter hasta que dé la opción para salir con “X”, para así dejar todos los valores por predeterminados.

Y así de sencillo se instala *Xampp*. Cuando termine la instalación lo que se debe correr es el panel de control que trae *Xampp*, en este panel podemos arrancar todos los servidores, además, provee los enlaces necesarios para poder ingresar a las herramientas que permitirán configurar *MySQL*.

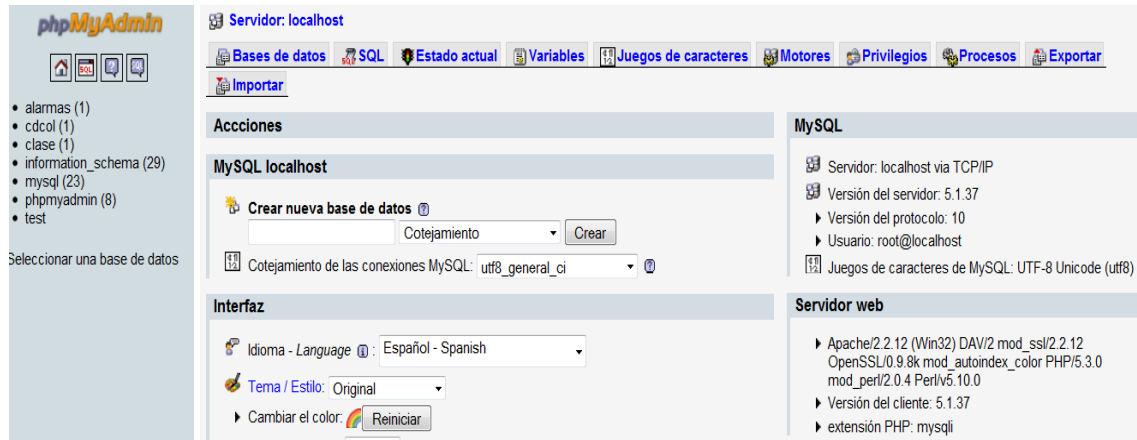
Se ingresa al panel de control de *Xampp* se levantan el servidor de *Apache*, para que corra la herramienta de configuración de *MySQL*, y el servidor de *MySQL*. El panel de control de *Xampp* se muestra en la figura # 46.

Figura 46. **Iniciando los servidores *Apache* y *MySQL***



Luego se hace clic donde dice Admin, a la par de *MySQL* y se abrirá la herramienta de configuración de *MySQL* que se muestra en la figura 47.

Figura 47. **Herramienta de configuración de *MySQL***



Una vez en ella, se crea una base de datos, donde dice crear una base de datos y se crean campos de ejemplo para poder utilizarlos más adelante.

Para fines didácticos la base de datos se llamara “clase” y contendrá una tabla llamada “estudiantes” con campos “Nombre, Apellido, Carnet, Nota”

Base de datos:

- clase

Tabla:

- estudiantes

Campos:

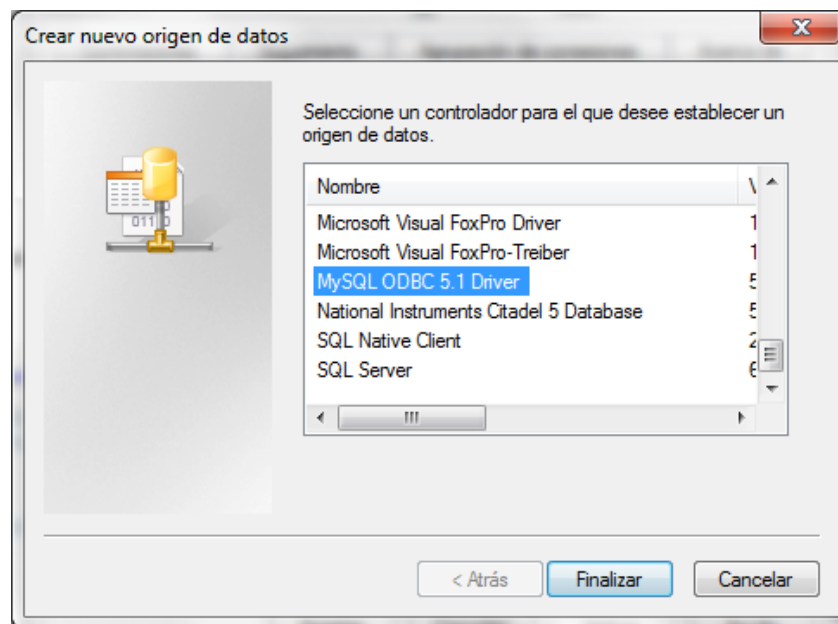
- ✓ Nombre
- ✓ Apellido
- ✓ Carnet
- ✓ Nota

Una vez que se tiene la base ya lista, se puede instalar el conector ODBC provisto. El instalador del conector ODBC no requiere más que aceptar una licencia y listo, motivo por el cual no se explicara su instalación.

Una vez instalado *Xampp* y el conector ODBC se procede a configurar *Windows* para que sepa que tiene instalada una base de datos, esto se hace de la siguiente manera:

En el panel de control de *Windows*, se hace clic en Herramientas Administrativas – orígenes de datos ODBC. Una vez abierta la ventana de orígenes de datos se posiciona en la pantalla de DSN de Sistema y se hace clic en Agregar. Se selecciona la opción *MySQL ODBC 5.1 Driver* y se hace clic en Finalizar.

Figura 48. Origen de datos ODBC



Luego aparecerá un asistente de configuración en el cual se debe llenar los campos solicitados.

Figura 49. **Asistente de conexión**



The image shows a screenshot of the MySQL Connector/ODBC 'Connection Parameters' dialog box. The window title is 'MySQL Connector/ODBC'. The dialog contains several input fields: 'Data Source Name' (empty), 'Description' (empty), 'Server' (empty), 'Port' (3306), 'User' (empty), 'Password' (empty), and 'Database' (dropdown menu). A 'Test' button is located to the right of the 'Database' field. At the bottom, there are four buttons: 'Details >>', 'OK', 'Cancel', and 'Help'.

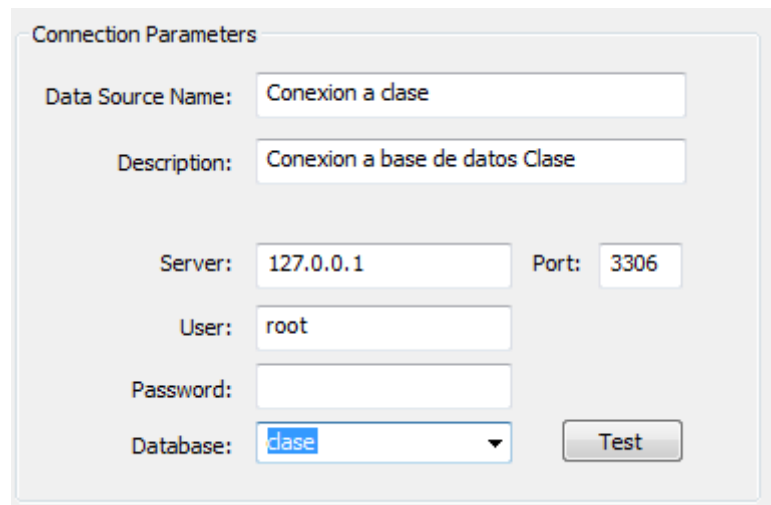
En *Data Source Name*, introducirá el nombre con el cual se quiere que aparezca esa conexión.

En *Server* pondrá la dirección IP del servidor de *MySQL*, en este caso 127.0.0.1 o escribir *localhost*.

En usuario y contraseña, se escribe el usuario y la contraseña que fueron configuradas en *MySQL*. En este caso el usuario es *root* y la contraseña se dejará en blanco.

Ahora en la pestaña database, aparecerá las bases de datos publicadas por el servidor y se selecciona la que es de interés, en este caso la base de datos que se llama clase.

Figura 50. **Parámetros para la conexión**



The image shows a 'Connection Parameters' dialog box. It contains the following fields and values:

- Data Source Name: Conexion a clase
- Description: Conexion a base de datos Clase
- Server: 127.0.0.1
- Port: 3306
- User: root
- Password: (empty)
- Database: clase (selected in a dropdown menu)

A 'Test' button is located at the bottom right of the dialog.

Luego se hace clic en test y si todo está bien se podrá ver que desplegará un mensaje donde dice que se logró conectar con éxito. Por ultimo hacemos clic en *OK*.

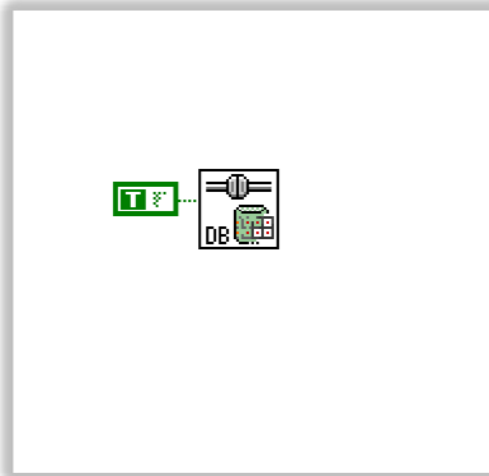
Ahora que ya está configurado *Xampp* y que *Windows* también ya está configurado, se puede empezar a programar en *Labview*.

Se inicia *Labview* y se crea un nuevo VI. Ahora lo que se hará será probar si se tiene acceso a la conexión configurada anteriormente.

Dentro del diagrama de bloques se coloca primero la función *DB TOOL OPEN CONECTION* la cual servirá para abrir una conexión a la base de datos llamada clase.

Se coloca una constante tipo *bool* en la entrada del bloque que se llama *prompt* y se le coloca un valor verdadero, como se muestra en la figura 51. Se ejecuta el VI y se observará lo que sucede.

Figura 51. **Prueba de conexión**



Al correr el VI se puede observar que abre un asistente de conexión en el cual solicita el nombre de la conexión a la cual se conectará, el usuario, la contraseña y la base de datos a la cual se desea conectar como se observa en la figura 52.

Ingresando los datos que fueron preparados anteriormente y presionando el botón de probar conexión observará que los datos son correctos, y terminará la ejecución del programa sin errores, eso es indicador que todo salió bien.

Considerando que un usuario final no sabrá todos esos datos, y que sería tedioso estar ingresando los datos de cada conexión, cada vez que se inicie un programa, lo que se puede hacer es ingresar todos esos datos en un archivo con extensión udl para luego llamarlo con *Labview*.

Figura 52. Datos para la conexión

1. Especifique el origen de datos:

Usar el nombre del origen de datos
conexionaclase

Usar la cadena de conexión
Cadena de conexión:

2. Escriba la información para iniciar sesión en el servidor

Nombre de usuario:

Contraseña:

Contraseña en blanco Permitir guardar contraseña

3. Escriba el catálogo inicial:

clase

Comenzando con la creación del archivo con extensión udl, se abrirá el programa bloc de notas y se creará un nuevo archivo e inmediatamente se guarda, solo que con la variante que tendrá extensión “.udl”.

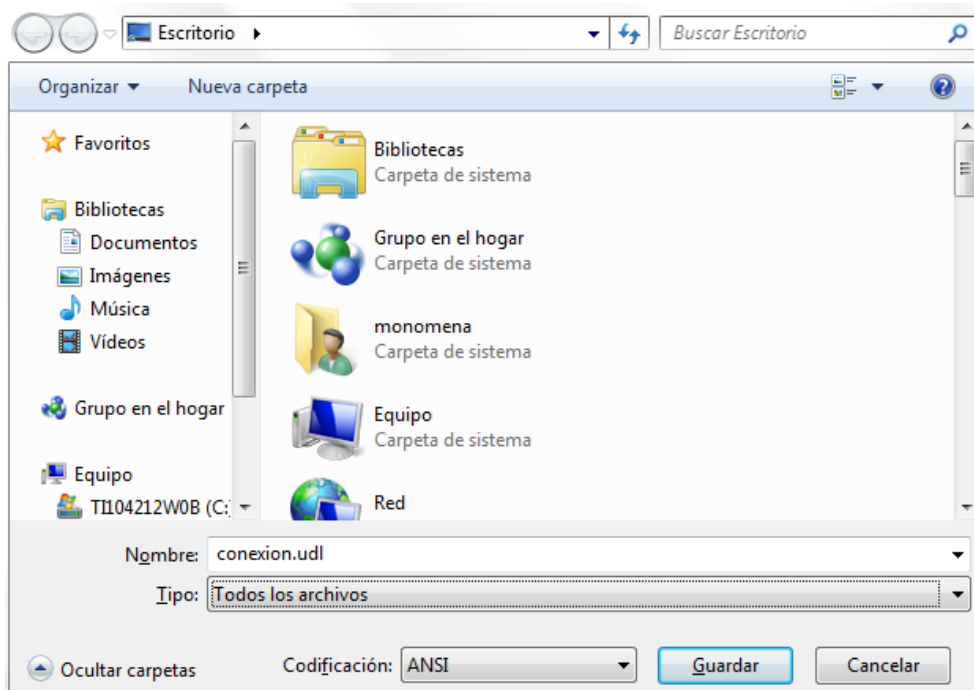
En este caso el archivo será guardado con nombre conexión.udl. Luego de escribir el nombre, en la pestaña donde dice tipo seleccionamos “todos los archivos” y pulsamos el botón de guardar como se observa en la figura 53.

El archivo será creado en la carpeta deseada y se observará que el ícono no será de un archivo de texto.

Al darle doble clic al archivo conexión.udl se abrirá un asistente para la conexión a una base de datos, como el mostrado en la figura 54, este asistente

será igual al que *Labview* mostró, así que ahora se llenan los campos nuevamente como se hizo el asistente de *Labview* y se presiona aceptar

Figura 53. Creación del archivo udl



Ahora los datos de la conexión se encuentran almacenados dentro del archivo conexión.udl.

En el panel frontal se pone un control tipo *file path control* que se encuentra en la paleta de controles, en *Modern – String & path*, y se busca el directorio en el cual se tiene el archivo conexión.udl, quedando como en la figura 55.

En el diagrama de bloques se unen el control de directorio, hacia la entrada del bloque de conexión en la entrada llamada *Connection Information* y se cambia la constante tipo *bool* hacia falso, como en la figura 56.

Figura 54. **Asistente de conexión**

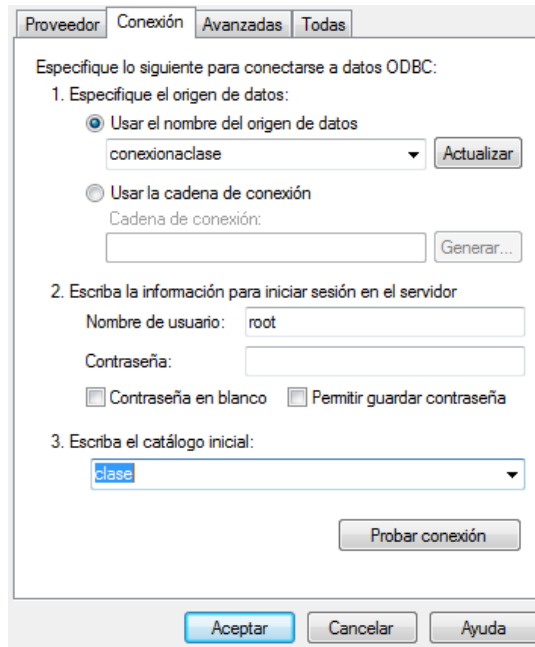


Figura 55. **Configuración del directorio**

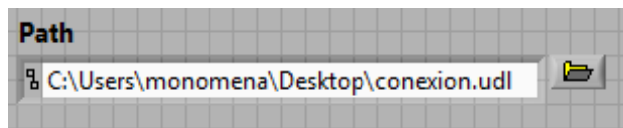
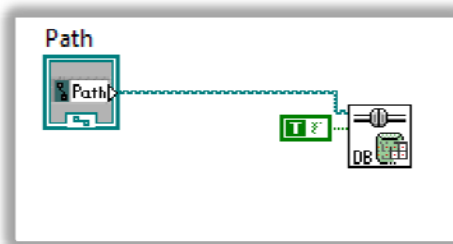


Figura 56. **Configuración bloque conexión**



Ahora se puede ejecutar el VI y se observará que no se produce error alguno, lo cual indica que la conexión fue realizada con éxito.

5.1.2.3. Ingreso de datos a *MySql* a través de *Labview*

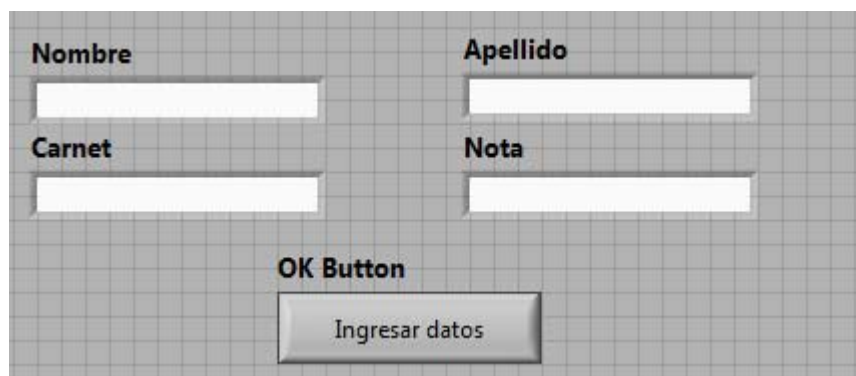
En esta sección se explicará cómo ingresar datos a *MySql* a través de *Labview*, siempre se trabajará sobre lo que ya se ha hecho en las secciones anteriores.

En un VI en blanco se insertan cuatro controles tipo string, llamados:

- Nombre
- Apellido
- Carnet
- Nota.

Estos controles llevan los nombres de los campos en *MySql*. También se agregará un control tipo botón para que solo cuando se presione se realice el ingreso de los datos, como se muestra en la figura 57, de esta manera solo se escribirá en la base de datos una vez, ya que de lo contrario *Labview* escribiría una gran cantidad de registros dentro de la base.

Figura 57. Controles para la conexión a *MySql*



The image shows a LabVIEW front panel with a grid background. It contains four string controls arranged in a 2x2 grid. The top-left control is labeled 'Nombre', the top-right is 'Apellido', the bottom-left is 'Carnet', and the bottom-right is 'Nota'. Below these controls is a button labeled 'OK Button' with the text 'Ingresar datos' inside it.

Ahora lo que se llevara a cabo será insertar bloques de conexión e inserción a la base de datos, los cuales se encuentran en la paleta de funciones en *Connectivity – Database*.

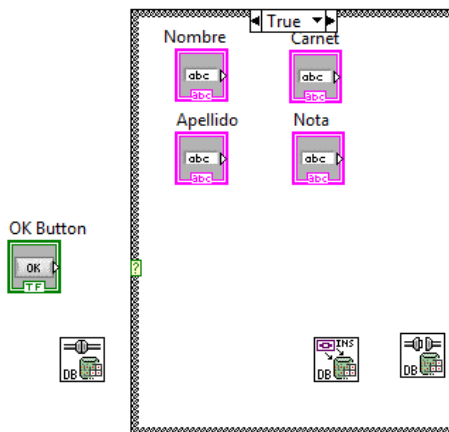
Primero se inserta una función de conexión, segundo un bloque llamado *Insert Data*, y por último se agrega un bloque de desconexión, estos bloques pueden observarlos en la figura 58.

Figura 58. **Bloques de programación para la conexión**



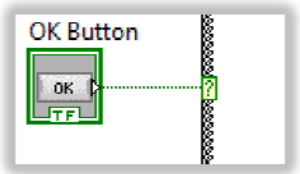
Ahora se encierran los controles y las funciones a excepción de la *open connection* dentro de un *case*, como se observa en la figura 59.

Figura 59. **Programación del ingreso**



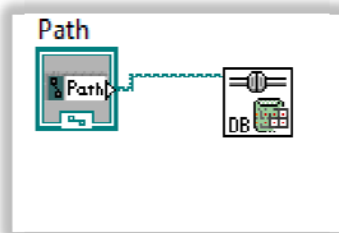
Ahora se une el control tipo botón hacia el case, específicamente en el cuadrado que tiene el signo de interrogación, de esta manera el case sólo se ejecutará si y solo si es presionado el botón.

Figura 60. **Configurar el control del case**



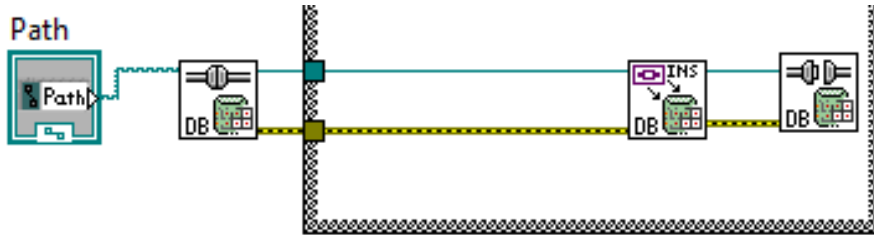
Se procede a inserta y conectar un control tipo *path* en el cual se ingresará el directorio hacia donde esté el archivo conexión.udl como se observa en la figura 61

Figura 61. **Conexión del directorio al bloque conexión**



Ahora se conectan las salidas que tienen el mismo nombre con las entradas del mismo nombre de los bloques de conexión, inserción y de desconexión, como se observa en la figura 62.

Figura 62. **Interconexión de bloques**



A continuación se inserta un bloque llamado *Build Array* el cual se expande para que sea de cuatro entradas para verse como en la figura 63.

Figura 63. **Expansión bloque *build array***

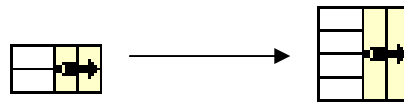
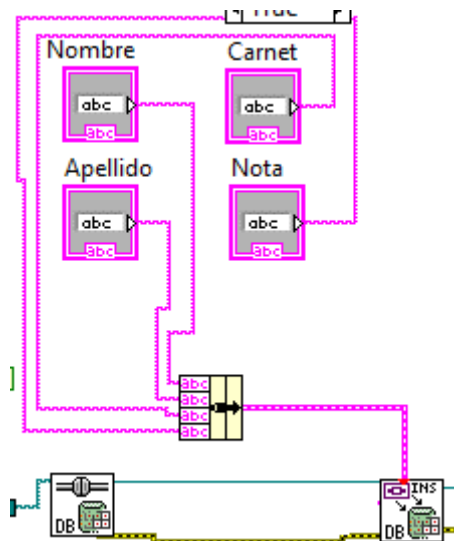


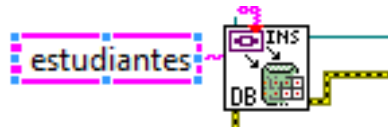
Figura 64. **Interconexión total del VI**



Se conectan los controles a las entradas pertinentes, en orden: Nombre, Apellido, Carnet y por último Nota. Luego se une la salida del arreglo a la entrada *Data* del bloque de inserción como se observa en la figura 64.

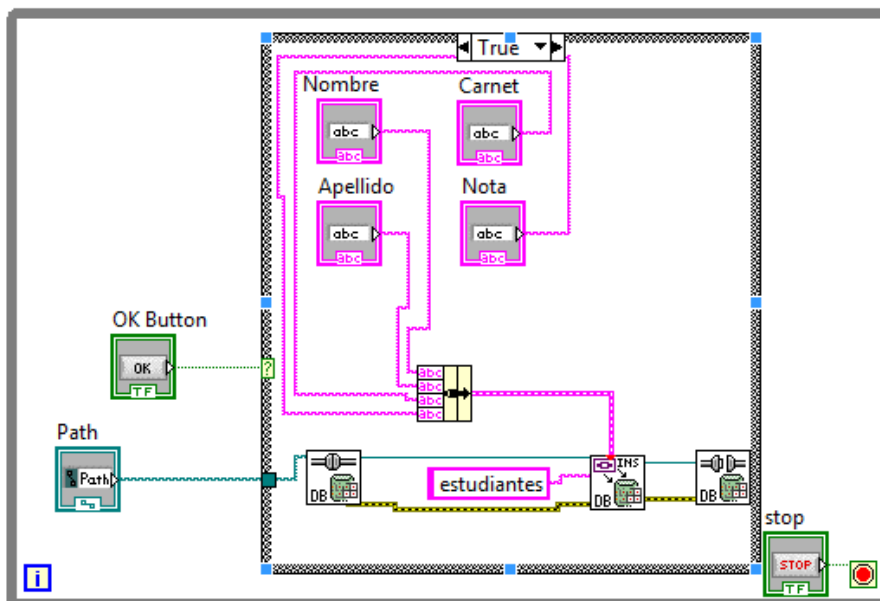
Ahora en la entrada tabla, del bloque de inserción se crea una constante llamada *estudiantes* y se hace la conexión como se muestra en la figura 65.

Figura 65. Ingreso de la tabla



Ahora se necesita encerrar todo en un ciclo *while* como en la figura 66.

Figura 66. Programación completa del ingreso de datos



Cuando se ejecuta, primero se llenan los campos, como en la figura 67, y se presiona el botón “ingresar datos”, luego se dirige al administrador de *Mysql* y se observará que los datos sean ingresados correctamente.

Figura 67. Ejecución e Inserción de datos

Insertar datos en una base de datos

Nombre: Carlos Apellido: Guzman

Carnet: 200412853 Nota: 100

OK Button: Ingresar datos, Salir

Se puede observar en la base de datos llamada “Clase” en la tabla “estudiantes” los datos almacenados, observe la figura 68 para comprobar.

Figura 68. Verificación del ingreso de los datos

Mostrar : 30 filas empezar en modo horizontal y repetir

+ Opciones

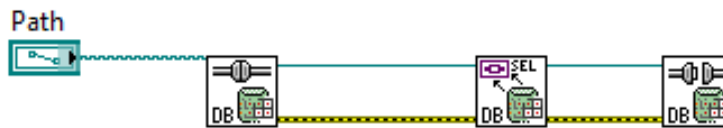
	Nombre	Apellido	Carnet	Nota
	Carlos	Guzman	200412853	100

5.1.2.4. Acceso a los datos en *MySQL* a través de *Labview*

En el siguiente ejercicio se observará cómo extraer información de la base de datos, se utilizarán los datos de las secciones anteriores.

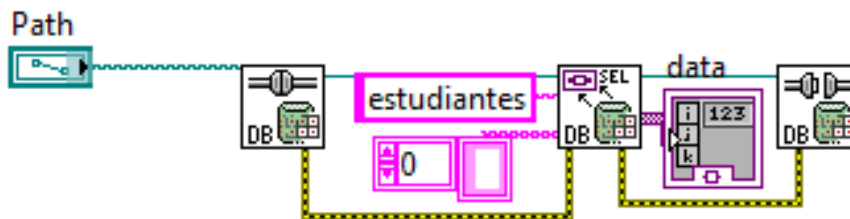
Primero se crea una conexión a la base de datos y en lugar de un bloque de inserción es necesario insertar un bloque llamado *DB Tools Select Data* con el cual se logrará extraer la información de la base de datos.

Figura 69. **Bloques para el acceso a datos**



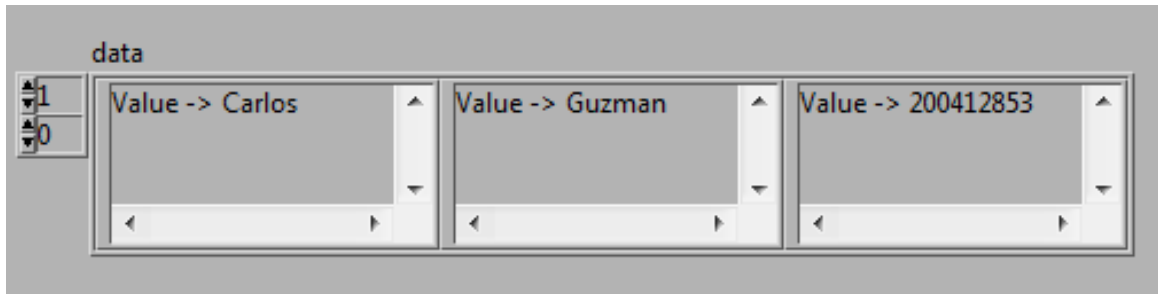
A continuación se agrega una constante de la tabla y una constante de la columna en la cual están los datos, además se agrega un indicador a la salida de ese bloque para que entregue los datos y se puedan visualizar. Puede observarse lo anterior explicado en la figura 70.

Figura 70. **Programación para el acceso a datos básico**



Ahora aparecerá en el panel frontal el indicador deseado y se debe agrandarlo, en la esquina superior derecha, para así observar todas las columnas.

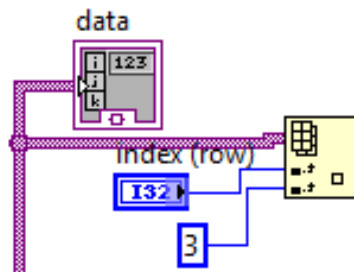
Figura 71. **Indicador y ejecución del código**



La información mostrada en la figura 71, sólo es útil para visualizar, pero no se puede operar matemáticamente nada, ya que está en un tipo de variable llamada *variant*. Si se desea usar esta información, como podía ser la nota, para mostrar la nota total del usuario, más algunos puntos extras, se hace necesaria la conversión de la información que es lo que se hará ahora.

Se agrega una función llamada *index array*, la cual permite acceder a los datos uno por uno. A sus entradas ponemos controles o constantes para la selección de columna y fila como se observa en la figura 72.

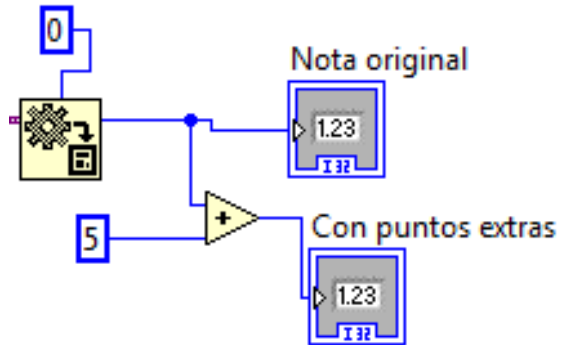
Figura 72. **Indexado de los datos**



Se agrega un bloque llamado *variant to data*, el cual hará la conversión. A este bloque se le ingresa la salida de la función *index* y en su entrada donde

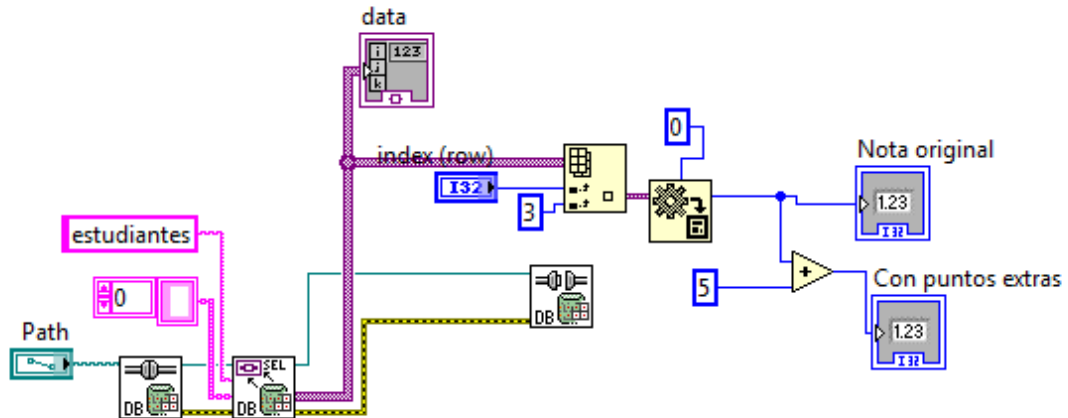
dice *type*, se le ingresa una constante numérica y a la salida se agrega una función matemática de suma y le sumamos cinco puntos como en la figura 73.

Figura 73. **Modificación de los datos extraídos**



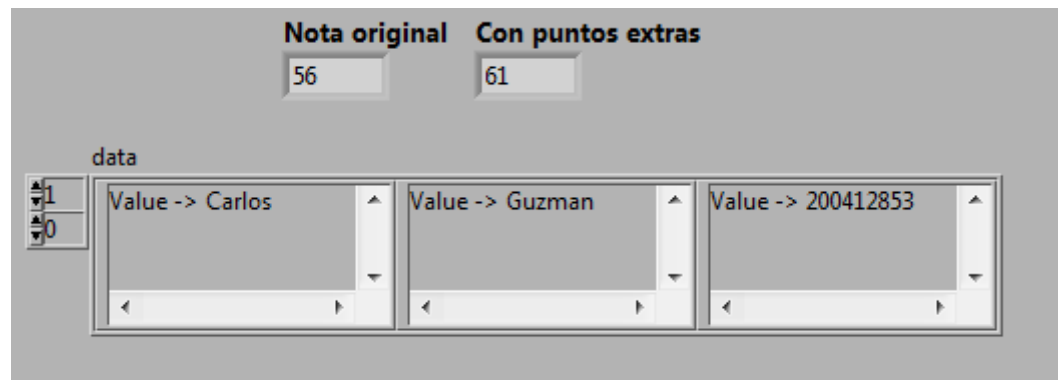
La programación completa deberá verse como en la figura 74

Figura 74. **Programación completa del acceso y modificación de datos**



Al ejecutarlo se observan los resultados como se muestra en la figura 75.

Figura 75. **Ejecución del VI de acceso a datos**



5.1.2.5. **Labview en *Windows Mobile*.**

Labview tiene la capacidad de generar archivos ejecutables para ser usados en el sistema operativo *Windows Mobile* o WM. Al momento de crear un programa para WM es necesario tomar en cuenta que estos dispositivos no tienen una gran capacidad de procesamiento y que no son tan robustos como son las computadoras, por lo anterior ha de ser tomado sobre éstos la visualización de los procesos y el control de algún proceso que de preferencia no sea tan crítico para el sistema.

Se comienza con lo que es *Labview* en WM, primero se debe instalar el soporte de lo que son variables compartidas en el dispositivo móvil. Es necesario generar un nuevo proyecto como la figura 76.

Ahora se debe agregar el dispositivo que vamos a utilizar, se hace dando clic derecho sobre *Project: Untitled* y luego clic en *New – Target and devices* entonces aparecerá una ventana, como la mostrada en la figura 77, en la que se selecciona la opción *Existing target and device* y se expande la opción *Mobile*, donde se seleccionará en este caso, un dispositivo *Windows Mobile 6*

Device, ya que se cuenta con un dispositivo físico que tiene ese sistema operativo.

Figura 76. **Proyecto Nuevo**

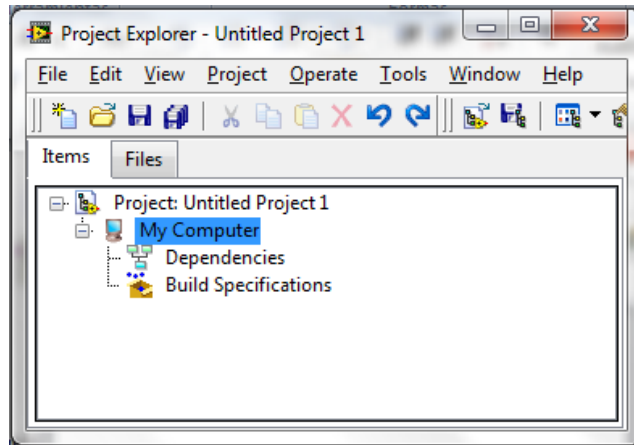
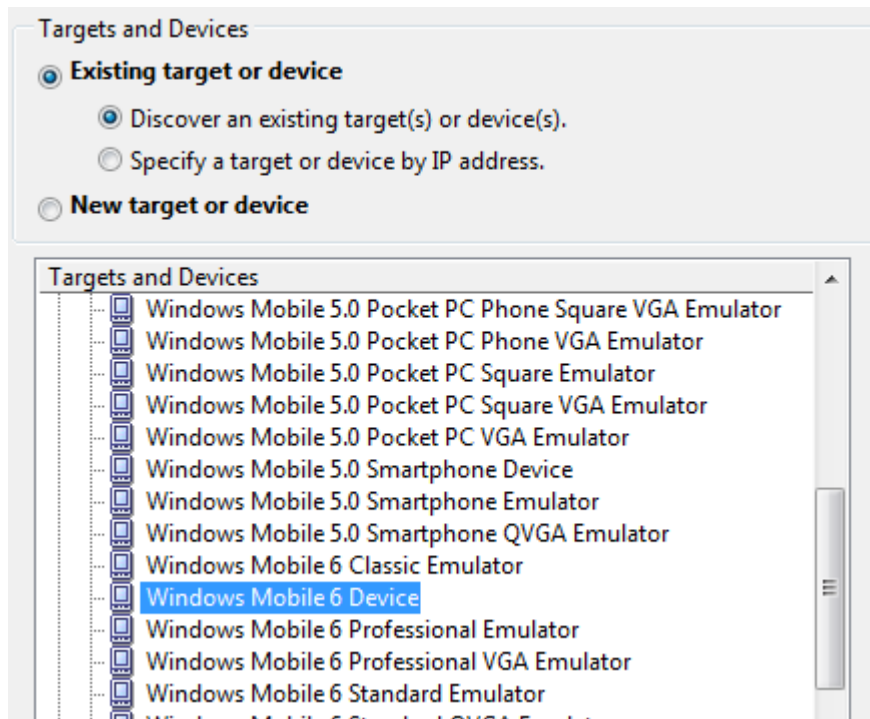
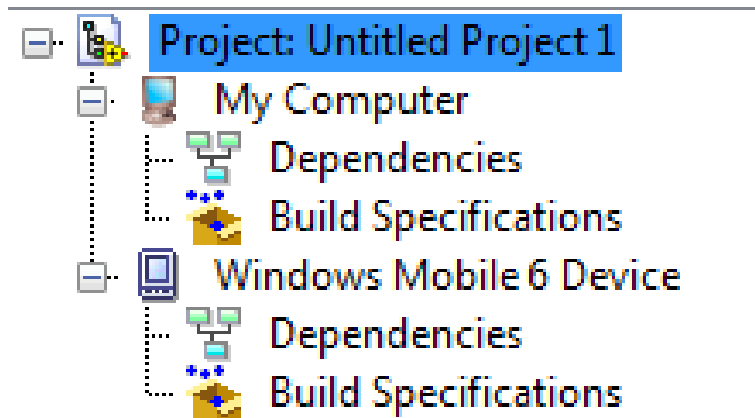


Figura 77. **Agregar un nuevo dispositivo móvil**



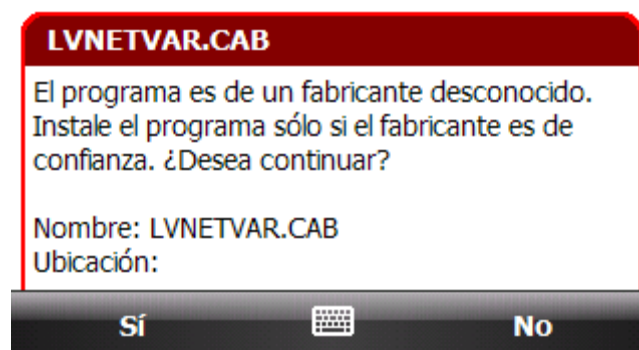
Ahora se observará que en el proyecto se ha agregado ese dispositivo tal como se muestra en la figura 78.

Figura 78. **Dispositivo móvil agregado**



Para darle soporte al dispositivo para manejar variables compartidas, lo cual es necesario para manejar variables a distancia, se puede hacer dando clic derecho sobre el dispositivo WM y se selecciona *Support for shared variables*. Esta opción abrirá un asistente de instalación en el teléfono, como el que se muestra en la figura 79, que preguntará si se quiere ejecutar un archivo ejecutable que *Labview* descargó en el teléfono.

Figura 79. **Solicitud de permiso de ejecución**



Luego preguntará dónde se quieren instalar los archivos, si en la memoria del teléfono o en la tarjeta de almacenamiento, como se muestra en la figura 80. Es preferible instalarlo en la memoria del teléfono.

Figura 80. **Selección de lugar de instalación**

Seleccione una ubicación para instalar "NI Shared Variable Support":

Dispositivo

Tarjeta de almacenamiento

Espacio necesario: 2786 KB

Espacio disponible: 41808 KB



Por último, finalizada la instalación, se puede ver que se solicita reiniciar el dispositivo.

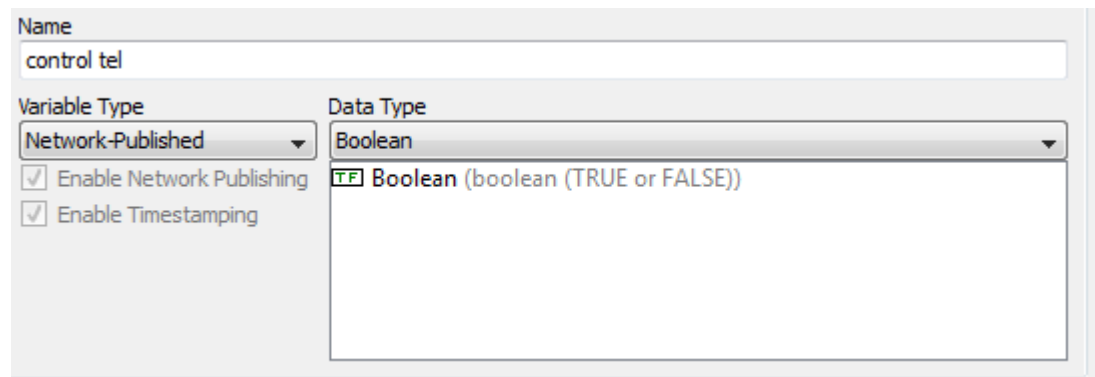
Hecho esto último, se ha finalizado con la instalación de soporte para variables compartidas y ahora solo queda crear un ejecutable para WM.

Ahora se realizará un ejercicio sencillo que a través de la computadora se encenderá un led en el teléfono y a través del teléfono se encenderá un led en la computadora, por medio de variables compartidas.

Se comenzará creando dos variables compartidas, una llamada “control tel” y la otra “control compu”.

Para crear una variable compartida hay que hacer clic derecho en *My computer – New – Variable*, al abrirse el asistente que se observa en la figura 81, se selecciona el tipo de variable *boolean*, se le coloca el nombre y se hace clic a *ok*.

Figura 81. **Ventana de creación de una nueva variable**



Ahora se observa que esta variable aparece en el proyecto dentro de una librería que se llama *Untitled Library*, ya que aún no se ha guardado esa librería con el nombre deseado.

Luego se crea un *Mobile VI* en el teléfono y un *VI* en la computadora, los cuales contendrán un *led* y un control tipo *bool* cada uno.

En el *VI* móvil se observará el panel frontal, véase la figura 82 y en *VI* de la computadora se observará el panel frontal de la figura 83.

Figura 82. **Front panel móvil**

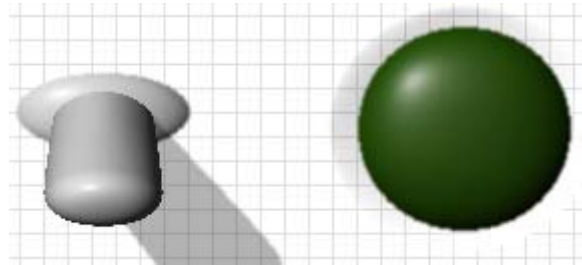


Figura 83. **Front panel en la PC**

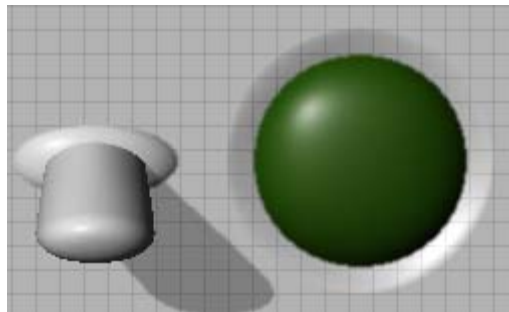
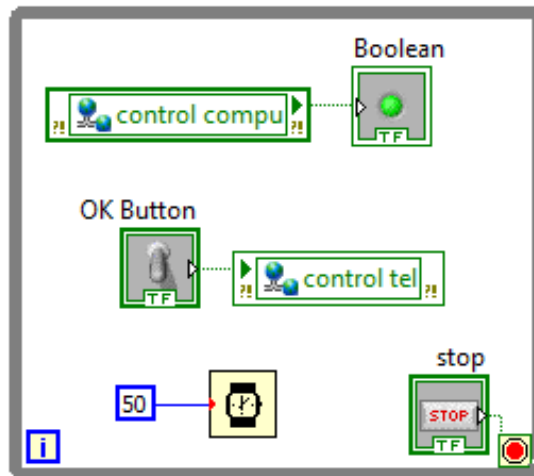


Figura 84. **Programación del vi móvil**



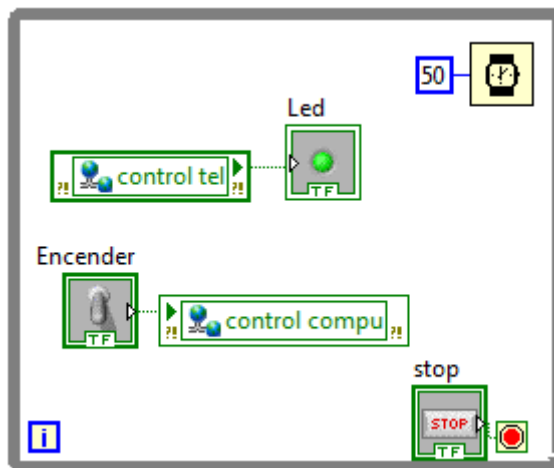
En la sección de la programación, se arrastran las variables directamente desde el proyecto hacia cada uno de los VI correspondientes haciendo que

cada una sea de escritura o lectura a través de sus opciones, accesibles desde un clic derecho, luego se encierran cada uno en un ciclo *while*, para así hacer las pruebas.

En el Vi móvil, la programación debe de quedar como en la fig. 84.

La programación en la computadora debe quedar como en la figura 85.

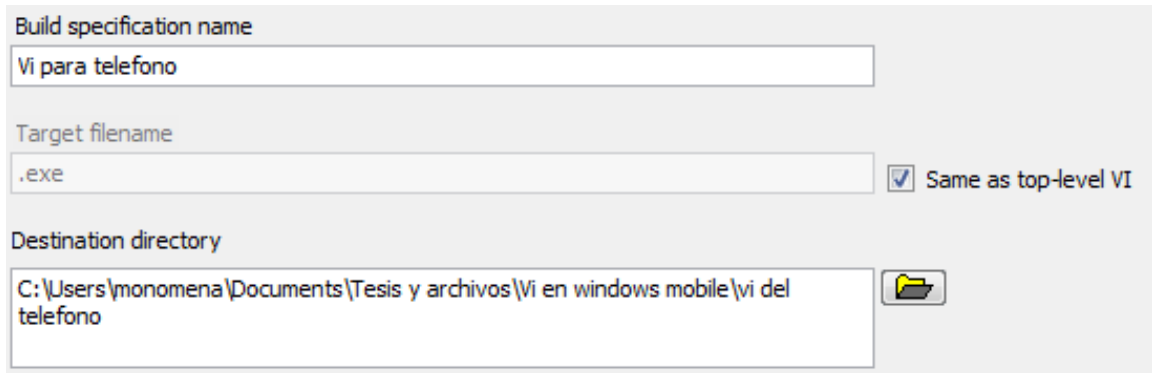
Figura 85. Programación del vi de la PC



Se crea el ejecutable para el teléfono y para ello hay que darle clic derecho en donde dice *Build specifications*, debajo del dispositivo WM, y se hace clic en *new - Mobile application*.

En el asistente que se abre, hay que poner el nombre con el cual se quiere que se cree el ejecutable y el directorio en el cual se almacenará; tal como se muestra en la figura 86.

Figura 86. **Asistente para la creación del ejecutable**



Build specification name
Vi para telefono

Target filename
.exe Same as top-level VI

Destination directory
C:\Users\monomona\Documents\Tesis y archivos\Vi en windows mobile\vi del telefono

Se presiona *ok* y ahora aparecerá el ejecutable en el proyecto, se hace clic derecho y se hace clic en *build*. Ahora ya se tiene el ejecutable listo para trasladar al dispositivo móvil.

Una vez trasladado el ejecutable, solo es de abrirlo y probarlo, se observará que el control booleano en el teléfono enciende el led de la computadora y que el control de la computadora enciende el led del teléfono.

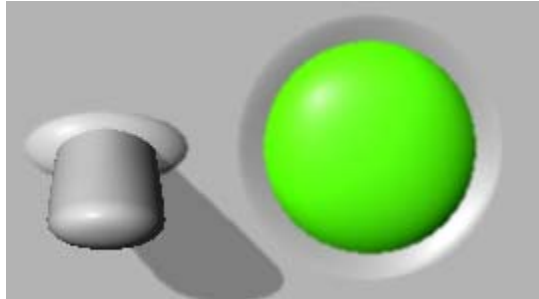
Si se levanta el *switch* del teléfono como en la figura 87

Figura 87. **Botón activado en el móvil**



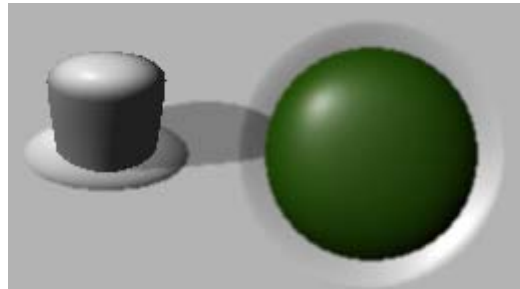
Se observará que se enciende el *led* en la PC como se muestra en la figura 88.

Figura 88. **Led encendido en la PC**



Y si se levanta el *switch* de la computadora, así como se observa en la figura 89.

Figura 89. **Botón activado en la PC**



Se observará que se enciende el *led* en el teléfono, como en la figura 90.

Figura 90. **Led encendido en el móvil**



Como se observa es un control cruzado de los *leds*.

6. DISEÑO DE UN SESTEMA SCADA CON CONEXIÓN A BASE DE DATOS, VISUALIZACION Y CONTROL EN DISPOSITIVO MÓVIL

En este capítulo se desarrollará un sistema SCADA para controlar la temperatura y el nivel del agua de un tanque. El control será mediante PID, y para ajustar la temperatura usaran resistencias de calentamiento y ventiladores para el enfriamiento.

6.1. Simulación de sensores

Primero se prepararán los controles que servirán para funcionar como entradas de temperatura y nivel. Estos se simularán ya que no se cuenta con el equipo físico necesario. Se agregarán dos controles tipo numérico llamados *Pointer slide* como se muestra en la figura 91.

Luego se generaran dos variables compartidas las que servirán para retener los valores que se le ingresen, a través de los controles que representan los sensores; luego se conectan los sensores con las variables correspondientes llamadas *sensor temp* y *sensor fill*. La programación correspondiente se puede observar en la figura 92.

Figura 91. Controles para la simulación

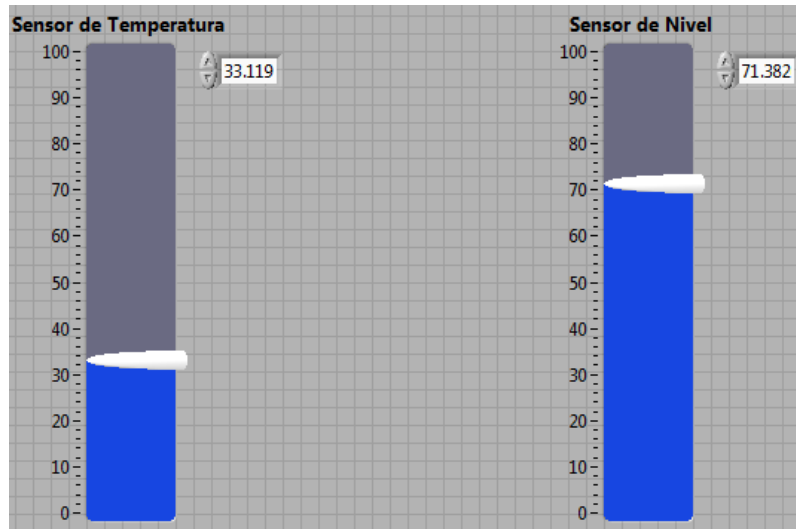
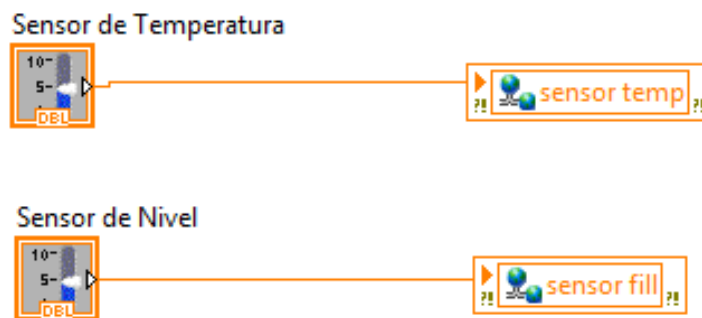
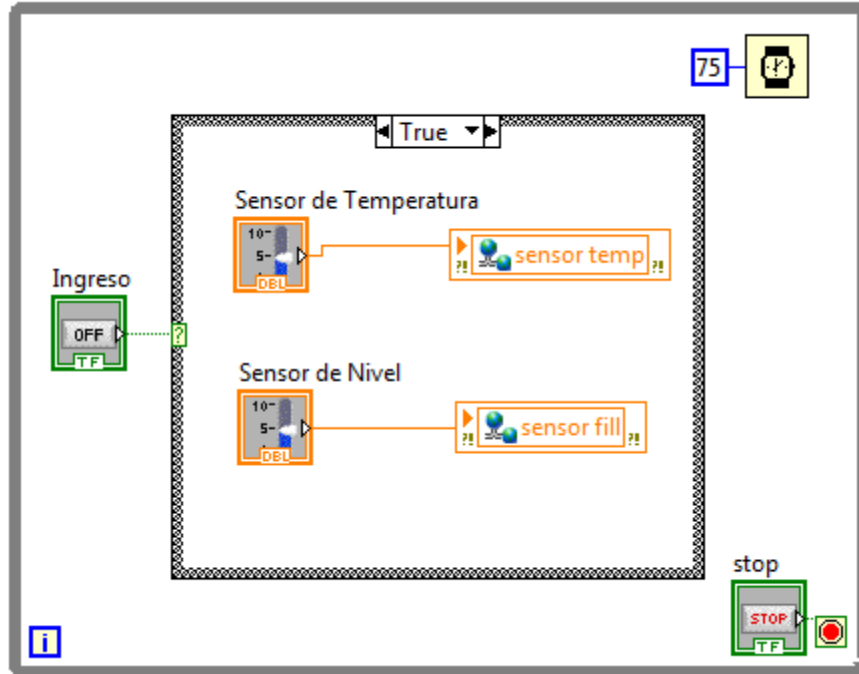


Figura 92. Almacenamiento de la simulación en variables



A continuación se encierra toda la programación dentro de un ciclo *case* y luego dentro del ciclo *while*. Lo encerrado en un *case* será controlado por un botón, para que solo cuando se pulse ese botón, se ingresen los datos, tomando en cuenta que es necesario conectar el botón de ingreso con el signo de interrogación del ciclo *case*, todo esto lo pueden observar en la figura 93.

Figura 93. Programación del VI de simulación



6.2. Creación de la base de datos

Ahora se deberá crear una base de datos nueva, la que contendrá los datos del sistema, comenzando por una tabla de usuarios.

En este momento la base de datos se llama SCADA y contiene una tabla usuarios, como se muestra en la figura 94.

Figura 94. Tabla dentro de la base usuarios

SCADA (1)	
usuarios	
Campo	Tipo
<input type="checkbox"/> usuario	varchar(50)
<input type="checkbox"/> password	varchar(10)

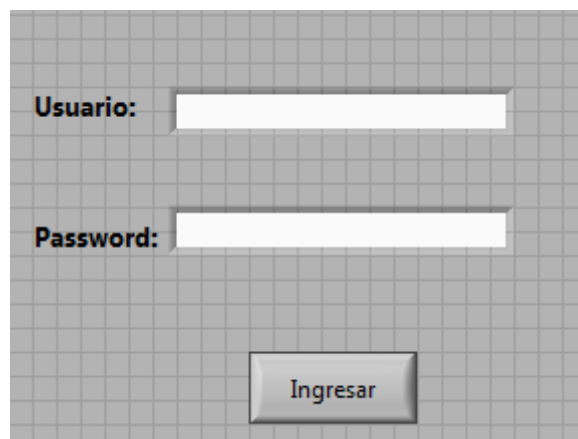
Se configuran los datos de conexión en el sistema, con nombre “SCADA”, y se crea el archivo `scada.udl` que será el que almacene los datos de la conexión.

6.3. Programación de la pantalla de ingreso mediante usuario y contraseña

Se genera un nuevo VI, el cual será la pantalla de inicio, donde se solicitará el ingreso de un usuario y su contraseña, como una medida de seguridad. Esta ventana deberá tener dos controles tipo *string* y un botón de ingreso, tal como se puede ver en la figura 95.

En la parte de la programación, se debe verificar que la existencia de ese usuario corresponda con su contraseña. Esto se realizará mediante un proceso de comparación.

Figura 95. **Panel frontal para el ingreso**



Primero es necesario extraer los datos de la base. Lo que se hará es conectarse a la base y pedirle a *Mysql* que despliegue el usuario deseado; y si existe, que lo compare con su clave.

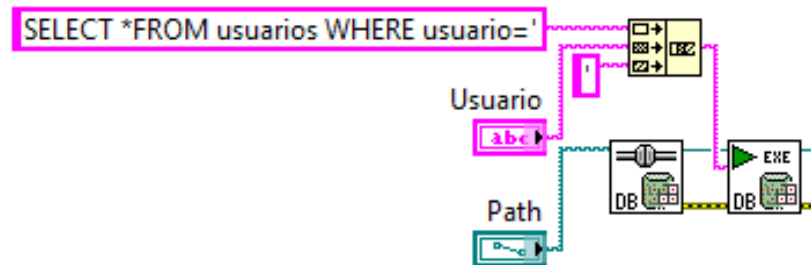
En esta etapa se usarán un par de funciones nuevas: *Fetch data* la que desplegará los datos de la búsqueda, y *Execute Query* la cual ejecutará la búsqueda. Estas dos funciones las puede observar en la figura 96.

Figura 96. **Adquisición de los datos**



Ahora se dan las instrucciones a *Labview* para que busque de la tabla usuarios, lo que se encuentre en el control usuario. Se concatena el texto y se le introduce al bloque que corresponde como se muestra en la figura 97

Figura 97. **Instrucción de búsqueda**



Ahora se separan del arreglo recibido, los datos obtenidos y se compara la siguiente columna, con lo que se tiene en el control *password*, como se muestra en la figura 98.

Figura 98. Comparación de los datos

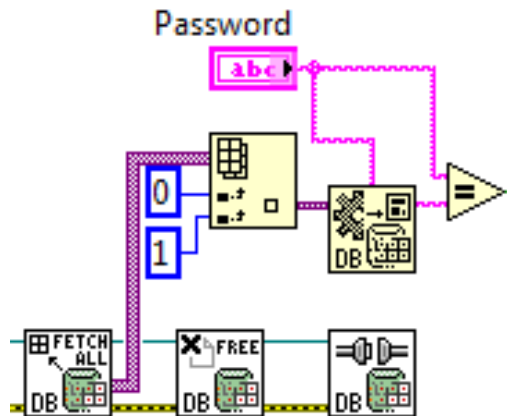
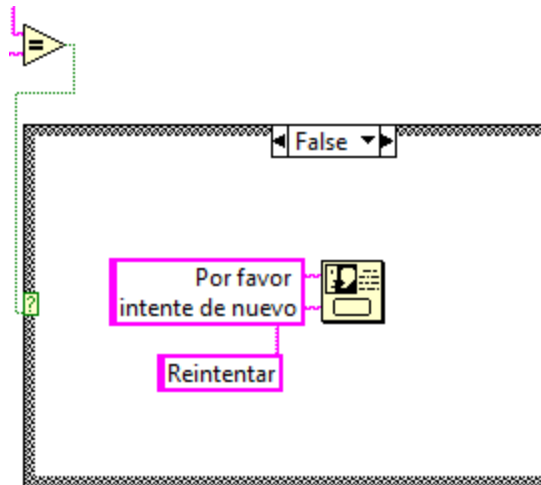


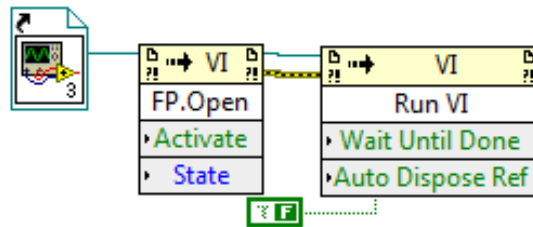
Figura 99. Petición de reingreso de información



Si la salida de la comparación anterior es verdadera, es posible usarla para llamar al VI donde se encontrará la interfaz de control. De lo contrario que despliegue un mensaje solicitando el chequeo de sus datos. Esto se hará con otro *case* y una función *one button dialog* como se muestra en la figura 99.

Para llamar a otro VI, se debe abrir una referencia a otro VI y usar dos *invoke notes* para que abra el panel frontal y que corra, se observa la programación en la figura 100.

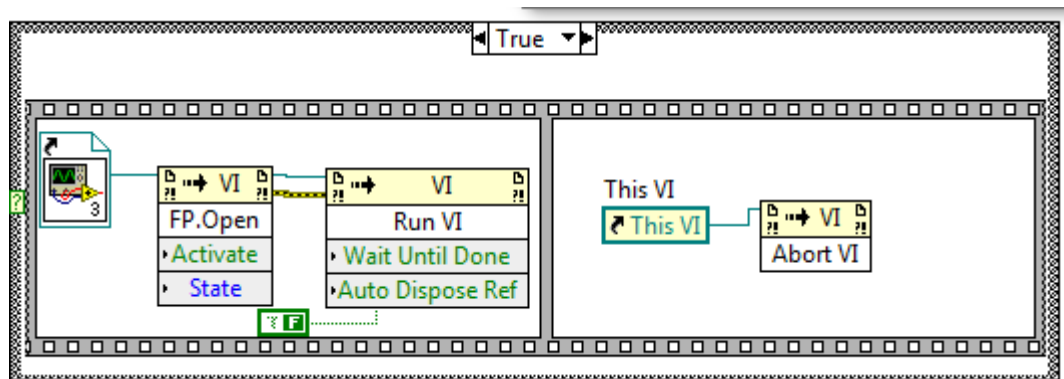
Figura 100. **Invocar a otro VI**



En este instante, cada vez que sean ciertas las condiciones de ingreso se abrirá el VI llamado control, que es donde se ejercerá el control de todo el sistema.

Por último, se agrega una secuencia para hacer que luego de abierto el VI de control, se cierre el VI que lo llamo, es decir el VI de *login*.

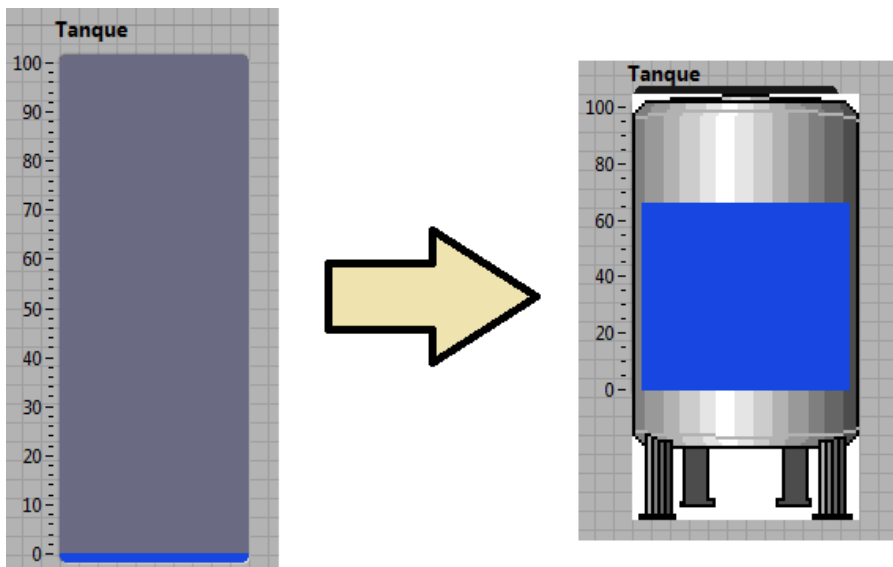
Figura 101. **Secuencia de apertura del VI de control**



6.4. Programación del VI que controlara el proceso.

En la etapa de control, se agrega un control tipo slider, el cual servirá para simular el tanque, el control a usar específicamente se llama *Fill Slid*. Este control se modifica como se mostró en el capítulo 5 para que sea un tanque y se vea de la siguiente manera, vea la figura 102.

Figura 102. Creación del tanque



En este caso, el nivel del agua será proporcional al área azul, que representa el agua. Cómo se debe controlar la temperatura, se necesita de ventiladores y alguna forma para calentar el agua, se comienza con un control led cuadrado que se modifica para quedar así como en la figura 103.

También se cambia un botón tipo led cuadrado para que tome la apariencia de lo que sería una torre de ventiladores para que se vea así como en la fig.104. Recuerde que sólo es un indicador tipo led, el que se modifica, ya que en la imagen aparecen dos ventiladores.

Figura 103. **Creación del fuego**

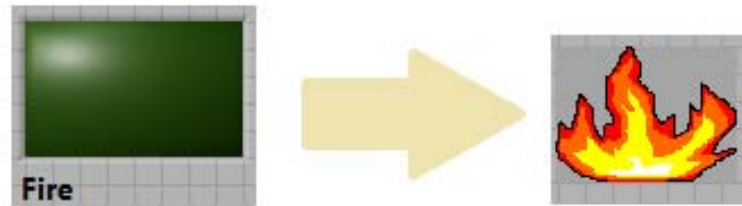
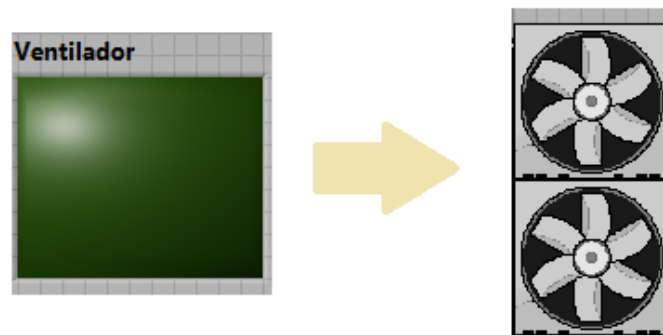


Figura 104. **Creación de los ventiladores**

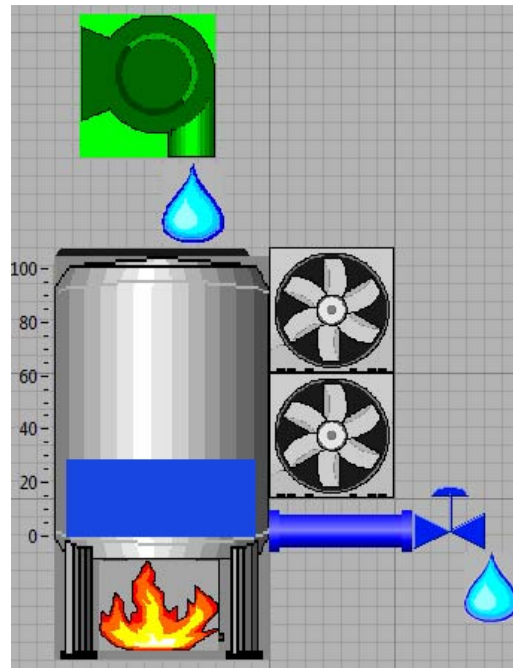


Aún falta un control con el que se simulará la bomba de agua y su funcionamiento, así que se coloca un *led*, que es modificado para crear la bomba de agua solamente cambiando el color de fondo para cada estado, indicando así si la bomba está activada o no.

Se agrega una llave y tubería desde la paleta de controles en el menú DSC para darle un poco más sentido y lógica al panel.

Todos los controles colocados se observan en la figura 105.

Figura 105. **Panel frontal del VI de control**



En el diagrama de bloques se insertan dos funciones PID y se crea un control para cada entrada de los bloques, generando así entradas para ganancias y *setpoints*.

Lo que se hará en el panel frontal será colocar los *setpoints* e indicadores de funcionamiento en una tabla y se agregarán las ganancias en otra tabla de tal forma que, se vea como en la figura 106.

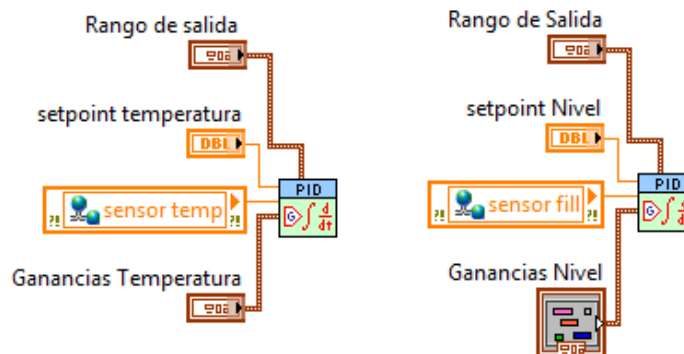
De esta forma se agruparán los controles de todas las funciones en pestañas, pero a su vez, estas pestañas tendrán orden según tipo de entrada.

Figura 106. Tablas del VI de control



La variable de proceso que se solicita en los PID, son las dos variables compartidas *sensor temp* y *sensor fill* las cuales serán los sensores, de tal forma que la programación de los PID es así como se muestra en la fig.107

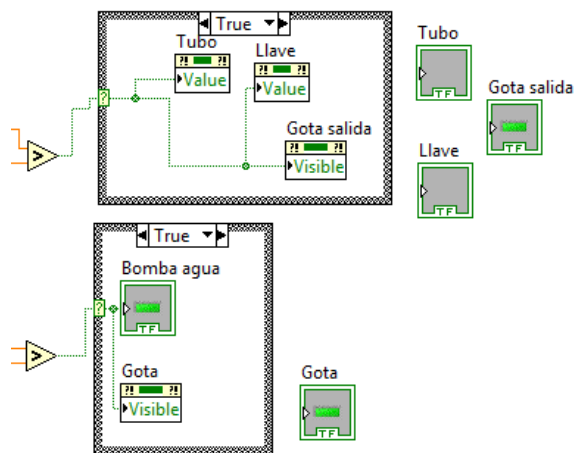
Figura 107. Configuración de los PID



Ahora lo que se hará será comparar si el *setpoint* es igual al valor censado, entonces si es mayor a cero, que active la bomba y que aparezca la

gota de agua; de lo contrario que la gota desaparezca y que la bomba se ponga en rojo. Se logrará esto mediante un ciclo *case* y unas *Property Node*. Si la salida del PID es menor a cero debe activarse la llave y debe aparecer la gota de agua a la salida de la tubería de salida, programación que se puede observar en la figura 108

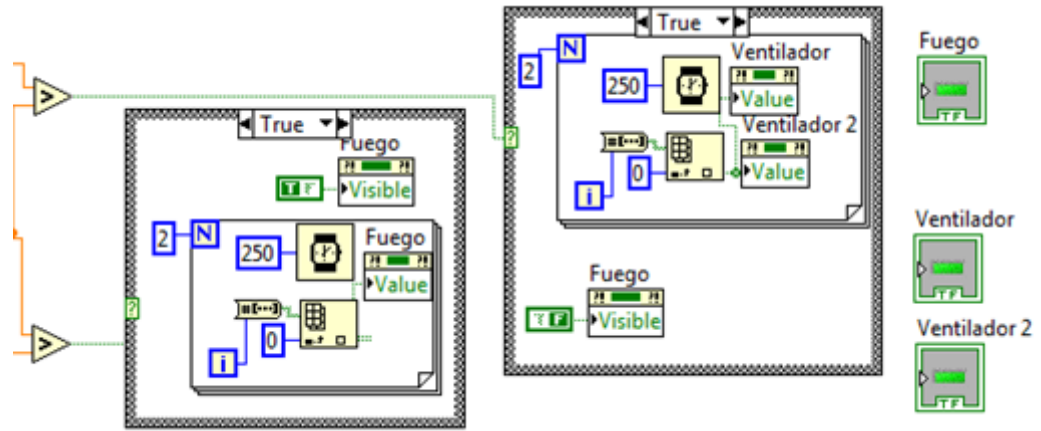
Figura 108. Programación del encendido y apagado de la bomba



Ahora se realizara casi lo mismo con la temperatura.

La parte de la programación de la temperatura se diferencia en que se usan ciclos para animar el fuego y los ventiladores, pueden observar esos cambios en la figura 109.

Figura 109. Encendido y apagado de ventiladores y fuego



6.5. Programación del VI móvil y unificación con el VI en la PC

En el teléfono se diseñará un panel frontal compuesto también de pestañas, una de *setpoint*, otra de ganancias y una donde se muestre la salida de los PID. Para que no sea controlado el sistema desde dos puntos distintos, se deshabilitara el menú del VI control en la computadora al instante de ingresar los datos mediante el celular

Comenzando con el panel frontal, solo se deben colocar las pestañas e ingresar los controles, dos numéricos uno para cada *setpoint*, seis más para las ganancias de los PID, y por último dos gráficas, la sección de los *setpoint* se pueden observar en la figura 110.

Ahora se colocan los controles de ganancias con su opción de ingresar, como se muestra en la figura 111.

Figura 110. Sección móvil de los *setpoints*



Figura 111. Sección móvil de las ganancias



Y en la última pestaña se pondrá un indicador de gráfica y un par de controles booleanos, como se muestra en la figura 112.

Figura 112. Sección de gráficas en el VI móvil

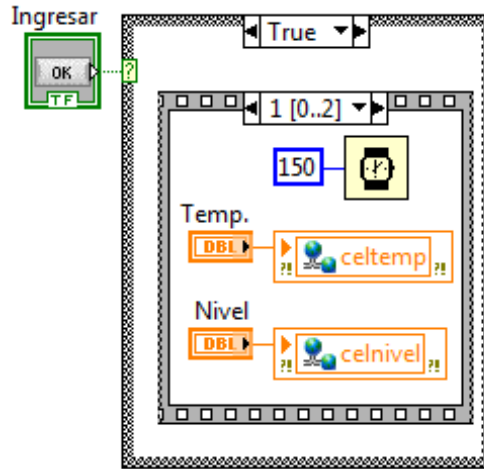


Comenzando con la programación, lo primero que se hará será hacer que se puedan introducir los *setpoints* deseados a través del teléfono, creando una variable compartida llamada Bandera, la cual será utilizada como bandera para indicar que las variables son escritas desde el teléfono; y que las debe cambiar en la computadora y una variable llamada desactivar, la cual desactivará el menú en la computadora, para así evitar que se escriba a una variable al mismo tiempo desde dos puntos distintos.

La idea principal es que al momento de que el botón ingresar tome posición de verdadero, Bandera pasará a ser verdadero, y se cambiará el valor de dos variable auxiliares llamadas “celtemp” y “celnivel”, las cuales cambiarán los valores de los controles en la PC se observa la programación de la sección anterior en la figura 113.

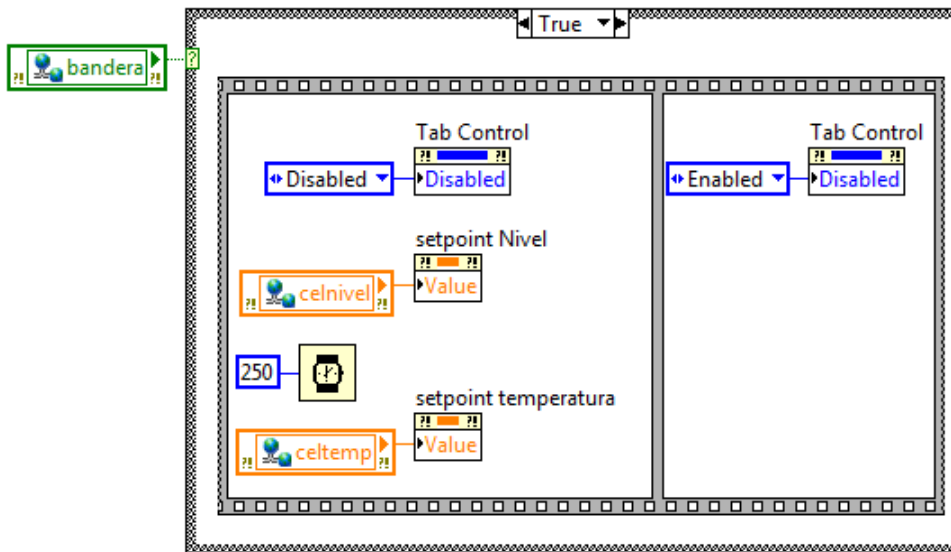
Entonces se agrega en el VI control, un ciclo case, el cual cuando ingresar sea verdadero, que deshabilite el panel, que cambie los datos, y que después lo habilite de nuevo, esta sección de la programación la pueden encontrar en la figura 114.

Figura 113. Programación de la simulación de sensores



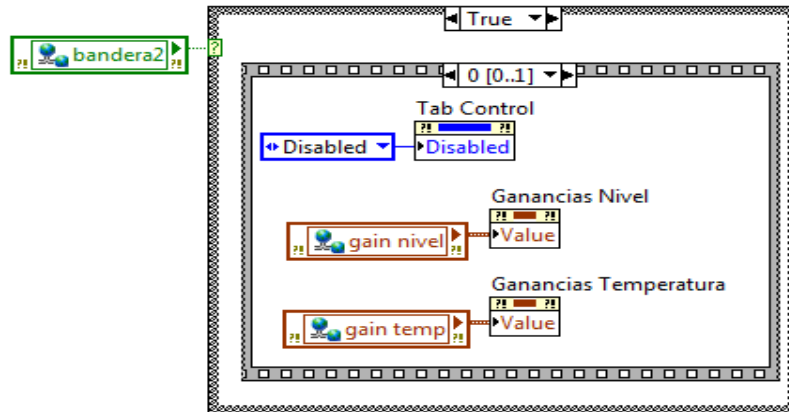
Ya se tiene el ingreso de los *setpoints* y se agregará una variable llamada *bandera2*, la cual indicará que los datos a cambiar son las ganancias de los PID y se hará lo mismo en el VI de control.

Figura 114. Programación en la PC que es controlada por el móvil



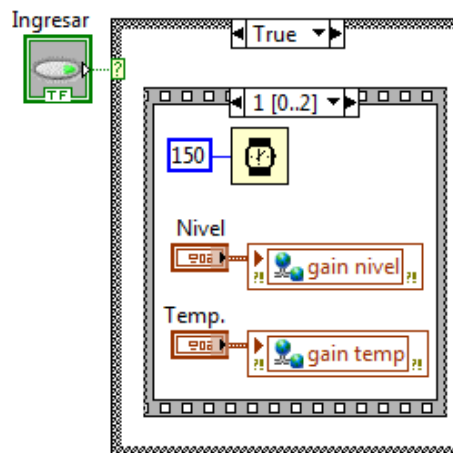
En el VI control se encuentra la programación observada en la figura 115.

Figura 115. Programación en la PC que permite el cambio de ganancias vía dispositivo móvil



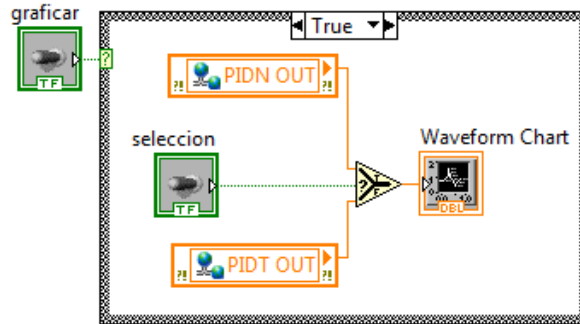
La programación para el cambio de ganancias en el dispositivo móvil, se encuentra en la figura 116.

Figura 116. Ingreso de ganancias en el móvil



Con las variables que anteriormente se crearon se pretende introducir a la gráfica, según lo deseado. Todo esto se logrará introduciendo un case en el VI del teléfono, quedando como se muestra en la figura 117.

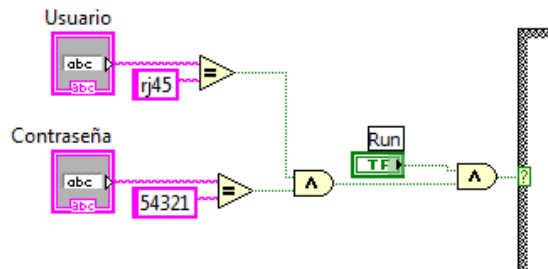
Figura 117. Selección de que variable se graficara



6.6. Verificación de Usuario en el teléfono

Ahora todo lo que se cambie en el teléfono, se percibirá en el VI de la computadora. Pero aún resta hacer que solo los usuarios que se desean se conecten a este servicio, lo que se hará será agregar una nueva pestaña en el teléfono, la cual contendrá dos campos tipo *string*, uno de usuario y el otro de clave, después agregamos un control tipo *bool*, llamado *login*, de tal forma que cuando presionen el botón se genere una rutina de comparación que le dé acceso al sistema. Esta programación se observa en la figura 118.

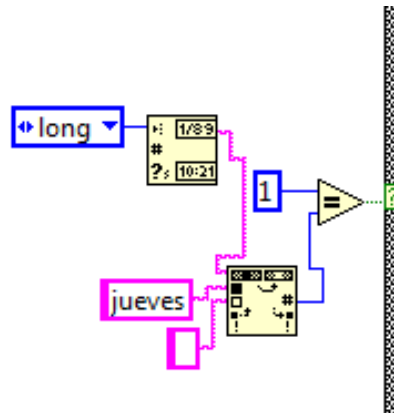
Figura 118. Logeo en el móvil



6.7. Automatización de la creación de respaldos de la base de datos

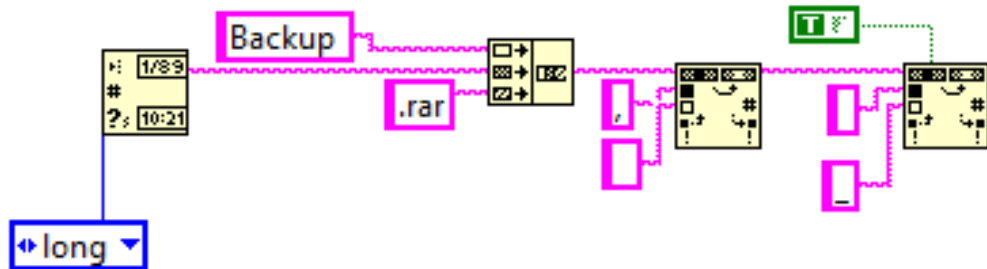
Por último, sólo resta hacer que el sistema realice automáticamente los respaldos de la base de datos. Este proceso será sencillo y la programación se llevará a cabo en el VI Control. Primero se insertará un bloque de *Get Date* el cual proveerá la fecha, después se busca si el día es jueves, para que si es jueves realice el respaldo y lo suba a un servidor ftp, con lo cual se generara un respaldo una vez por semana. En la figura 119 puede observar esta parte de la programación.

Figura 119. Selección de respaldo día jueves



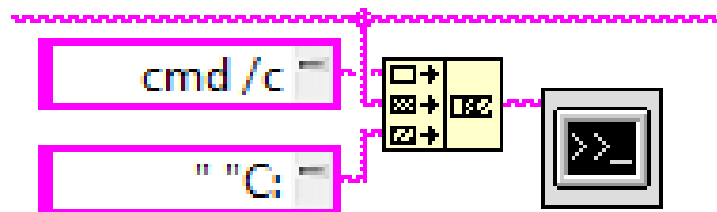
A continuación se obtendrá nuevamente la fecha, pero ahora se reemplazará la coma por un espacio, y luego se reemplazan todos los espacios con guiones y por último se concatena esa fecha con el nombre del archivo que será *backup*, como se muestra en la figura 120

Figura 120. Creación del nombre del archivo de respaldo



Ahora lo que se llevará a cabo, es decirle al sistema operativo que compacte los archivos donde se encuentra la base de datos; para esto, se utilizó *Winrar* y una función llamada *System exec* la cual ejecuta una línea de comandos en DOS. La programación la pueden ver en la fig. 121

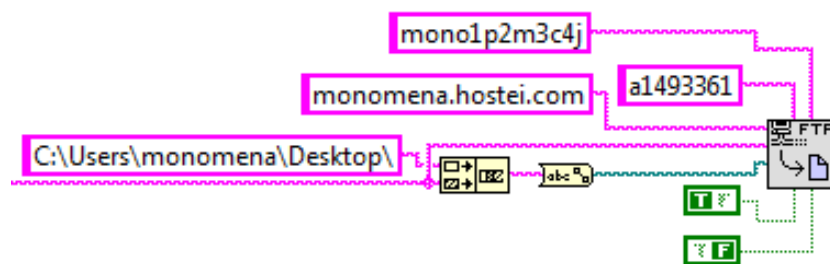
Figura 121. Creación del archivo de respaldo



Una vez creado el archivo de respaldo, se le dice a *Labview* que envíe ese archivo vía ftp hacia un servidor externo. Esta parte de la programación

requiere de un bloque llamado *Put Ftp* que se encuentra en la paleta de funciones y la configuración queda como la de la figura 122.

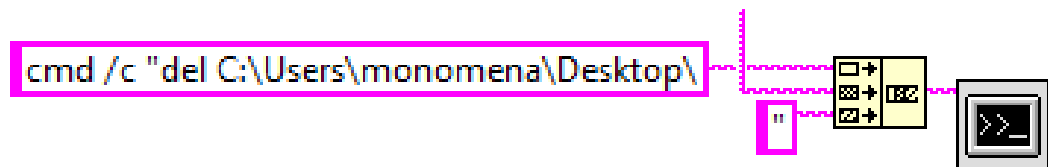
Figura 122. **Configuración del envío por FTP**



En la configuración se necesitan los campos de: servidor, ya sea como IP o dominio, usuario, contraseña y el directorio donde será almacenado el archivo. Ha de tomarse en cuenta que debe enviarse el dato como binario, ya que de lo contrario subirá un archivo vacío al servidor.

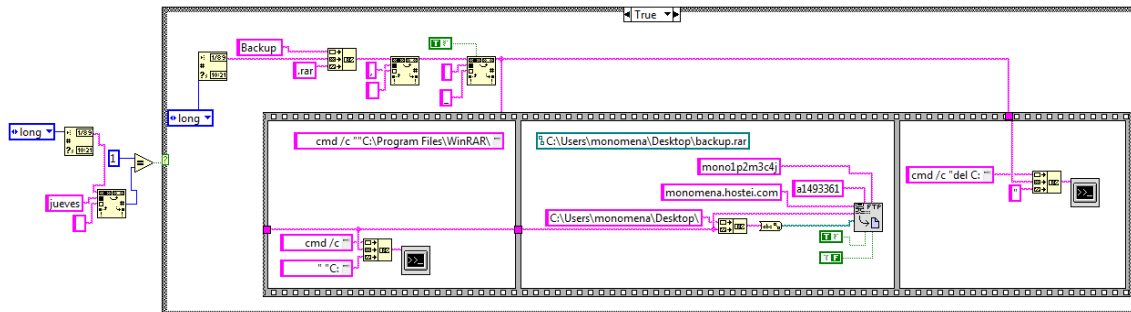
Aún resta eliminar el archivo de respaldo de la computadora local, y se hace de igual manera que la creación, solo que ahora se ejecuta el siguiente comando: `cmd /c "del C:\Users\monomena\Desktop\backup.rar` donde el nombre *backup.rar* representa el nombre que en ese instante posee el archivo, esta programación la puede observar en la figura 123.

Figura 123. **Borrado del archivo de respaldo en la PC local**



El último paso será introducir la programación en una secuencia para que se cree primero el archivo, luego que sea subido al servidor, y por último que sea eliminado de la computadora local, vea la figura 124 para su mejor comprensión.

Figura 124. Secuencia de respaldo



Debido a la gran cantidad de espacio, la figura 124 es muy pequeña, pero se puede observar que el case es controlado por la selección del día jueves y que después se realiza la rutina de respaldo comenzando por la creación del archivo de respaldo, luego pasa a subir el archivo a la red, y por último elimina el archivo de respaldo del disco local.

Con este último paso, se da por finalizado este sistema SCADA que controla el nivel y temperatura de agua dentro de un tanque, este sistema se conecta a una base de datos en *MySql* para extraer los datos de los usuarios permitidos y también es capaz de ser controlado por medio de un dispositivo móvil, como lo es un celular, en este caso un celular con *Windows Mobile 6.5*

El último capítulo fue un pequeño ejemplo de lo que se puede lograr, pero realmente depende del programador y su creatividad. Con esto doy por terminado el capítulo 6 y todo el desarrollo del trabajo de graduación.

CONCLUSIONES

- 1 Los sistemas de control deben ser de fácil visualización y sobre todo ser amigables con el usuario, ya que el mismo no debe ser un experto para entender lo que sucede en la fábrica.
- 2 Un sistema SCADA debe tener la capacidad de establecer un registro de actividades, de alarmas y mediciones realizadas durante los procesos, para que en caso de tener que reemplazar parte del equipo de la producción, se tenga un patrón de comportamiento deseado que deberá ser provisto por el equipo nuevo; de esta premisa surge la necesidad de la conexión del sistema a una base de datos como por ejemplo: una base de datos en *MySql*, la cual fue utilizada en este Sistema.
- 3 Por medio del sistema SCADA que se realizó, se logró que el sistema fuera capaz de crear automáticamente respaldos de la base de datos en la cual se encuentran usuarios y alarmas que el sistema ha presentado con anterioridad.

Estos respaldos son posteriormente almacenados en un servidor FTP en prevención de un fallo de la computadora donde se encuentra el sistema. Es por ello que se tienen dichos datos, los cuales se consideran de vital importancia, en una localidad segura; ahorrando de esta manera tiempo y dinero en una nueva configuración del sistema, que a su vez deberá ser reinstalado en el equipo nuevo.

RECOMENDACIONES

- 1 Es indispensable aprender métodos de programación visual, debido a que es la próxima generación de programación, la cual además de ser sencilla es completa y puede realizar las tareas más complejas.
- 2 Es necesario que el programador del sistema SCADA tenga, previo a iniciar con la programación, un amplio conocimiento en cuanto al proceso que se necesita controlar, porque le facilitará el diseño.
- 3 El sistema debe contar con medios de seguridad adecuados, para proteger la información, por lo tanto es necesario que los usuarios se registren antes de utilizar el sistema; para que se restrinja el acceso al mismo y no sea viable la modificación de variables.
- 4 Es recomendable tener la base de datos en un servidor provisto de seguridad adecuada para que no sufra ataques víricos.

BIBLIOGRAFÍA

1. COMER, Douglas. *Redes Globales de Información con Internet y TCP/ IP*.
. 3ª edición. México: Prentice Hall, 1997. 621 p. ISBN 968-880-541-6
2. GROTH, David; SKANDIER, Toby. *Guía del estudio de redes*. 4ª edición.
Estados Unidos: Sybex Inc, 1999. 491 p. ISBN 0-7821-4406-3
3. JEFFERY, Travis. *LabVIEW for Everyone*. 3ª edición Estados Unidos:
Prentice Hall, 2006. 1032 p. ISBN: 0131856723
4. PIATTINI, Mario. *Diseño de bases de datos relacionales*. 3ª edición.
Mexico: Alfaomega, 2000. 600p. ISBN: 8478973850
5. STALLINGS, William. *Comunicaciones y Redes de Computadores*. 7ª
edición. Madrid: Pearson Educación, 2004. 868 p. ISBN 84-205-
4110-9
6. _____. *Sistemas Operativos*. 4ª edición. Madrid: Prentice Hall,
2001. 847 p. ISBN: 978-84-205-4462