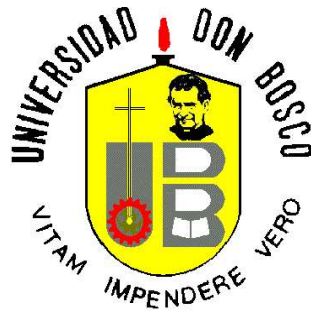


**UNIVERSIDAD DON BOSCO  
FACULTAD DE INGENIERIA  
ESCUELA DE INGENIERIA  
ELECTRÓNICA**



***“IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO  
INALÁMBRICO DE UNA VARIABLE ANALÓGICA POR  
MEDIO DE UNA INTERFAZ WEB UTILIZANDO GNU/LINUX,  
APACHE, PHP Y MYSQL”***

**TRABAJO DE GRADUACION  
PARA OPTAR AL TITULO DE  
INGENIERO EN  
ELECTRÓNICA**

**PRESENTADO POR:**

**DANIEL ALEJANDRO FLORES URBINA**

**RODRIGO EDUARDO BLANCO AVELAR**

**OCTUBRE 2005  
SOYAPANGO EL SALVADOR CENTROAMERICA**

# INDICE

<b>Introducción</b> .....	<b>1</b>
<b>Objetivos</b> .....	<b>3</b>
<b>Objetivo General</b> .....	<b>3</b>
<b>Objetivos Específicos</b> .....	<b>3</b>
<b>PARTE I</b> .....	<b>4</b>
a. Definición de Variable a Monitorear .....	4
b. Tecnologías Disponibles para Adquisición y Procesamiento de Datos .....	4
Detectores de Temperatura Resistivos (RTD) .....	5
Termistores .....	15
Pares Termoeléctricos.....	26
c. Selección de Tecnología para Adquisición de Datos .....	29
d. Tecnologías Disponibles para Transmisión Inalámbrica .....	30
e. Selección de Tecnología de Transmisión Inalámbrica .....	34
f. Selección del Puerto de la PC .....	36
g. Lectura de Datos desde el puerto de la Computadora. ....	36
¿Qué es RS-232? .....	36
Comunicación asíncrona.....	37
Abriendo el Puerto Serial.....	39
Opciones de Apertura .....	40
Escribiendo Datos al Puerto.....	40
Lectura de Datos desde el Puerto .....	41
Cerrando el Puerto Serial .....	41
La interfaz de Terminal POSIX .....	42
Opciones de Control .....	42
Configurando la taza de Baud (Baud Rate) .....	44
Configuración del tamaño del carácter.....	46
Configurando el chequeo de paridad .....	46
Escogiendo una entrada canónica .....	47
Escogiendo una entrada ordinaria .....	48
Esquema General del Proyecto .....	50
<b>PARTE II</b> .....	<b>53</b>
<b>A. Sistema Operativo Mandrake Linux</b> .....	<b>53</b>

a.	Descripción.....	53
b.	Consideraciones Previas a la Instalación.....	54
	Configuración del BIOS.....	54
	Hardware Soportado.....	54
c.	Discos y Particiones.....	55
	Estructura de una unidad de disco duro.....	55
	Particiones.....	55
	Particiones necesarias para Instalar Linux.....	55
d.	Instalación y Desinstalación de Programas.....	56
<b>B.</b>	<b>Servidor Web Apache.....</b>	<b>61</b>
a.	Descripción.....	61
b.	Instalación.....	62
	Requerimientos del Sistema para Instalar Apache.....	63
	Requerimientos para poner en marcha el Servidor Web Apache.....	63
	Ubicación de archivos y Directorios de <i>Apache</i> en <i>Mandrake Linux</i> .....	64
c.	Módulos del Servidor Web Apache.....	64
<b>C.</b>	<b>MySQL.....</b>	<b>66</b>
a.	Descripción.....	66
b.	Creación de Bases de Datos.....	66
	Ver Bases de Datos existentes.....	67
	Borrar una Base de Datos.....	68
	Crear una Base de Datos.....	68
	Crear una Tabla.....	68
	Insertar Datos en una Base de Datos.....	69
	Ver Datos Almacenados.....	70
c.	Administración de MySQL.....	70
<b>D.</b>	<b>PHP.....</b>	<b>72</b>
a.	Descripción.....	72
b.	Estructura del Lenguaje.....	72
	Sintaxis y Comandos Básicos.....	72
	Variables y Operadores.....	74
	Operadores Lógicos.....	75
	Incrementando y Decrementando una variable Entera.....	76
	Arreglos.....	76
	Estructuras de Control.....	77

Estructuras If .....	77
Uso de la cláusula else con If .....	78
Estructura if-elseif-else .....	78
Lazos .....	79
Lazos while .....	79
Lazos do while .....	80
Lazos for .....	80
break .....	81
Anidamiento de lazos.....	81
Formularios e Interacción con el usuario .....	82
Elementos en un formulario HTML .....	85
Listas Desplegables (SELECT).....	86
Casillas de Verificación (Check Boxes) .....	86
Funciones Especiales de PHP .....	86
isset .....	86
foreach.....	87
Die.....	87
c. Integración de PHP con MySQL .....	87
Conectándose a un servidor de Bases de Datos.....	87
Seleccionando una Base de Datos.....	88
Ejecutando comandos y realizando consultas a la base de datos.....	89
Determinando el número de Filas encontradas en un Consulta .....	89
Accediendo un Resultado .....	90
<b>E. Selección del Lenguaje para generación de Gráficos .....</b>	<b>90</b>
a. Determinación de Alternativas .....	90
Java .....	90
Perl.....	91
PHP.....	94
Creación de Imágenes con PHP .....	94
Adquiriendo Color.....	95
Dibujando Líneas.....	96
Rellenos de Imagen.....	96
Dibujar Arcos y Circunferencias .....	97
Dibujar rectángulos .....	97
Trabajando con Texto.....	97

b. Selección de Lenguaje a Utilizar.....	98
<b>PARTE III.....</b>	<b>100</b>
a. Instalación y Configuración de Linux Mandrake .....	100
b. Instalación del Servidor Apache y Módulos necesarios .....	120
c. Arranque y puesta en marcha de Apache .....	121
d. Pruebas de Configuración de Apache .....	122
e. Creación y Configuración de Autenticación en el Servidor.....	123
f. Desarrollo de Interfaz Web.....	126
g. Instalación del Servidor de Base de Datos MySQL .....	140
h. Creación de Base de Datos Temperatura .....	141
i. Desarrollo de Scripts para la construcción de Gráficos Dinámicos.....	143
<b>PARTE IV .....</b>	<b>160</b>
a. Implementación del Sistema de Adquisición y Procesamiento de la variable analógica .	160
b. Desarrollo del Programa para lectura del puerto.....	162
c. Compilación del Código fuente.....	166
<b>PARTE V.....</b>	<b>169</b>
Resumen de Costos del Proyecto .....	169
<b>Conclusiones.....</b>	<b>170</b>
<b>Recomendaciones.....</b>	<b>172</b>
<b>Referencias Bibliográficas.....</b>	<b>175</b>
<b>Glosario .....</b>	<b>177</b>
<b>Anexos.....</b>	<b>181</b>
<b>Anexo 1: Transmisor Point Sensor Analog 10V .....</b>	<b>181</b>
<b>Anexo 2: Point View Reciever.....</b>	<b>182</b>
<b>Anexo 3: Load Cell Amplifier Montalvo M3200-ce/UR .....</b>	<b>183</b>
<b>Anexo 4: Licencia Pública General (GPL) .....</b>	<b>184</b>
<b>Anexo 5: Placa de circuito impreso .....</b>	<b>185</b>

## TABLA DE FIGURAS

Figura 1. 1: Símbolo RTD.....	6
Figura 1. 2: Curvas para termómetros de resistencia.....	7
Figura 1. 3: Sonda termométrica de Platino.....	10
Figura 1. 4: Termómetro de Resistencia de Níquel.....	10
Figura 1. 5: Resistencia Metálica evaporada sobre una lámina aislante.....	12
Figura 1. 6: Método Potenciométrico.....	13
Figura 1. 7: Montaje de dos hilos.....	14
Figura 1. 8: Montaje de tres hilos.....	14
Figura 1. 9: Montaje a cuatro hilos.....	15
Figura 1. 10: Símbolo PTC y NTC.....	16
Figura 1. 11: Curva de Ajuste con cuatro parámetros.....	18
Figura 1. 13: Diferentes Tipos de NTC.....	21
Figura 1. 14: Circuitos de Medición con Termistores.....	24
Figura 1. 15: Curva Linealizada.....	25
Figura 1. 16: Representación Esquemática de Termopar.....	26
Figura 1. 17: Arreglo de Varios Termopares.....	27
Figura 1. 18 Aerocomm Connexlink 900MHZ Industrial.....	32
Figura 1. 19 Configuraciones para el Connexlink.....	32
Figura 1. 20 Point Sensor Analog 10V.....	33
Figura 1. 21 Point View Receiver.....	34
Figura 1. 22: Conectores DB9 y DB25.....	37
Figura 1. 23: Esquema de Conexiones.....	50
Figura 1. 24: Esquema de Proceso.....	51
Figura2. 1 Selección de RPMDRAKE.....	57
Figura2. 2: Cuadro de Verificación.....	58
Figura2. 3: Administrador de Paquetes RPM.....	58
Figura2. 4: Búsqueda de Paquetes de Software.....	59
Figura2. 5: Desinstalación de Programas.....	60
Figura2. 6: Formulario HTML.....	84
Figura2. 7: Formulario con dos campos vacíos.....	85
Figura 3. 1 Pantalla de Instalación Inicial.....	101
Figura 3. 2 Selección de Idioma.....	102

Figura 3. 3 Licencia de Distribución.....	102
Figura 3. 4 Selección de Ratón .....	103
Figura 3. 5 Selección de Tipo de Instalación .....	104
Figura 3. 6 Configuración de Teclado .....	105
Figura 3. 7 Nivel de Seguridad .....	106
Figura 3. 8 Particionamiento.....	107
Figura 3. 9 Formatear Particiones .....	108
Figura 3. 10 Selección de Paquetes.....	110
Figura 3. 11 Paquetes Individuales .....	111
Figura 3. 12 Inicio de Instalación.....	111
Figura 3. 13 Cuenta de Administrador .....	113
Figura 3. 14 Creación de Usuario .....	113
Figura 3. 15 Selección de Escritorio .....	114
Figura 3. 16 Instalación del cargador de Arranque .....	115
Figura 3. 17 Configuración Final .....	117
Figura 3. 18 Configuración de la interfaz Gráfica .....	118
Figura 3. 19 Instalación de Actualizaciones.....	119
Figura 3. 20 Fin de Instalación .....	120
Figura 3. 21 Página de Inicio.....	123
Figura 3. 22 Cuadro de Autenticación.....	126
Figura 3. 23: Esquema del Sitio .....	127
Figura 3. 24: Página index.html.....	131
Figura 3. 25: Página dina.html .....	133
Figura 3. 26: Página stat.html .....	134
Figura 3. 27 Diagrama de Flujo para graph.php .....	144
Figura 3. 28: Diagrama de Flujo main.php Parte A .....	148
Figura 3. 29: Diagrama de Flujo Parte main.php Parte B.....	149
Figura 3. 30: Diagrama de Flujo main.php parte C .....	150
Figura 3. 31: Diagrama de flujo main.php Parte D.....	151
Figura 4. 1: Circuito para adquisición de la variable analógica .....	160

## LISTA DE TABLAS

Tabla 1. 1: Características generales de los modelos de NTC más frecuentes .....	23
Tabla 1. 2 Tabla Comparativa entre los diferentes sensores .....	30
Tabla 1. 3 Tabla Comparativa Transmisores Inalámbricos .....	35
Tabla 1. 4 Nomenclatura para puerto serial en Linux .....	38
Tabla 1. 5: Miembros de la estructura de TERMIOS.....	42
Tabla 1. 6: Opciones de Control de c_flag .....	43
Tabla 1. 7: Constantes para tcsetattr .....	45
Tabla 1. 8 Constantes para el miembro c_lflag .....	47
Tabla 1. 9: Constantes para el miembro c_iflag .....	49
Tabla2. 1: Requerimientos para instalar Apache.....	63
Tabla2. 2 Operadores de Comparación.....	75
Tabla2. 3: Tipos de Datos manejados por PHP.....	75
Tabla2. 4: Tabla Dinámica generada con PHP y anidamiento de lazos.....	82
Tabla 5. 1: Resumen de Costos del Proyecto .....	169



## Introducción

En el presente el acceso a la información es una necesidad en casi todos los sectores productivos de nuestro país, específicamente en el sector industrial es de vital importancia que pueda conocerse el estado de la maquinaria en todo momento tanto desde la ubicación física como desde cualquier ubicación remota.

Desde las pequeñas empresas hasta la gran industria cuentan además con *Redes de Área Local (LAN)* o *Redes de Área Extensa (WAN)* por lo que se hace conveniente establecer el monitoreo a través de estas y de forma tal que pueda realizarse desde cualquier estación de trabajo y desde cualquier ubicación, esto sin la necesidad de infraestructura adicional.

En una época en la que el uso adecuado de recursos y el establecimiento de planes apropiados de mantenimiento son vitales se presenta este proyecto de graduación como una alternativa de monitoreo para aquella maquinaria y equipo que no posee esta funcionalidad y que se desea mantener actualizada con el resto de maquinaria brindando los conocimientos teóricos técnicos para su efectiva implementación.

Veremos que existen diferentes tecnologías disponibles tanto para la adquisición como para la transmisión de la variable analógica hacia el servidor.

El proyecto ha sido basado en las alternativas líderes en el mercado del software libre como son Apache, MySQL y PHP, que en el caso del primero es el líder indiscutido en el área de servidores Web, y en el caso de PHP comparte honores con ASP de Microsoft en el área de los lenguajes de Guión.

Muestra una forma básica de gráficar en tiempo real el comportamiento de una variable analógica utilizando las alternativas de software antes mencionadas y una sencilla interfase de hardware.

Con la implementación se pretende agregar nuevas funcionalidades a equipo o maquinaria que solo pueden ser monitoreadas en sitio.

# Objetivos

## Objetivo General

- Implementar un sistema de monitoreo inalámbrico de variables analógicas utilizando Software Libre el cual pueda ser accedido de forma remota.

## Objetivos Específicos

- Construir una interfaz Web basada en PHP para la visualización del estado de la variable a monitorear.
- Utilizar el servidor Web Apache para que albergue la Página de Monitoreo.
- Construir una base de datos con MySQL, que registre los eventos y estado de la variable analógica y la cual pueda ser consultada a través de la interfaz Web.
- Utilizar el sistema Operativo Linux como plataforma sobre la cual residirá el servidor de monitoreo.
- Transmitir de forma inalámbrica la variable analógica hasta el servidor de monitoreo.
- Desarrollar el sistema utilizando únicamente software libre.

## **PARTE I**

### **a. Definición de Variable a Monitorear**

Uno de los objetivos planteados con la implementación de este proyecto es que sea extensible al monitoreo de diferentes variables analógicas y no a una sola, pero para fines prácticos necesitamos definir una variable específica que nos servirá como variable de prueba. Para tal fin se ha seleccionado la temperatura por tratarse de una variable muy importante, ya que es de gran utilidad en procesos industriales, el clima, etc. Por lo que se hace necesario tener un monitoreo sobre la misma, para esto hay una amplia gama de sensores disponibles en el mercado así como el hardware para la adquisición y procesamiento de información. Pero siempre tomando en consideración que si se desea monitorear otra variable los cambios deben ser mínimos tanto en el hardware como en el software.

### **b. Tecnologías Disponibles para Adquisición y Procesamiento de Datos**

Para la obtención del valor de la variable analógica, se ha buscado alguna manera que se ajuste a lo que se necesita implementar, ya sea por medio de circuitos u otros dispositivos para llevar a cabo el sistema de medición. Lo que se requiere es que en el dispositivo encargado del monitoreo directo o sea el transductor, que estará en contacto directo con la sustancia o material que se desea monitorear brinde cierto voltaje de salida este voltaje entre 0 - 1VDC, ya que este es el voltaje necesario en la entrada de la siguiente etapa del sistema. Para esto se puede hacer uso de transductores activos (termopares) o transductores pasivos (termistores o sensores o monolíticos). Para convertir la temperatura a un cambio de un parámetro eléctrico (corriente o voltaje). Estos transductores pueden estar acoplados a ciertos circuitos para regular su voltaje de salida para que se ajuste al nivel de voltaje que necesitemos en nuestra siguiente etapa. Este voltaje de salida del circuito con el

transductor se utilizará para alimentar un amplificador de voltaje, preferiblemente de instrumentación, el cual sería la siguiente etapa en el sistema, este debe ser capaz de manejar la salida del transductor en su entrada y proporcionar una salida estándar para procesos industriales, estas salidas pueden ser de 0 - 10VDC, 0 - 5VDC, 0 - 20mA, 4 - 20 mA. También se necesitará que el transmisor a utilizar maneje preferiblemente una entrada para estos estándares. Esto dependiendo siempre de lo que el amplificador de voltaje y/o corriente pueda brindar a la salida. Si el transmisor no pudiese tener este tipo de entrada, se necesitaría la implementación de otro dispositivo, dependiendo siempre del tipo de entrada del transmisor por ejemplo si el transmisor demandase una entrada digital se tendría la necesidad de un convertidor analógico a digital, después del amplificador para poder brindarle una señal apropiada en la entrada al transmisor.

Para la obtención del valor de la variable analógica, se utilizara primero un transductor que nos permitirá convertir un cierto valor de temperatura a un valor de voltaje determinado, para hacer el transductor vamos a mencionar los diferentes modos de poder obtener esta interfaz la cual estará en contacto directo con la sustancia o material que se desee monitorear, así tenemos:

### **Detectores de Temperatura Resistivos (RTD) [1]**

Los detectores de temperatura basados en la variación de una resistencia eléctrica se suelen designar con sus siglas en inglés RTD (Resistance Temperature Detector). Dado que el material empleado con mayor frecuencia para esta finalidad es el platino, se habla a veces de PRT (Platinum Resistance Thermometer).

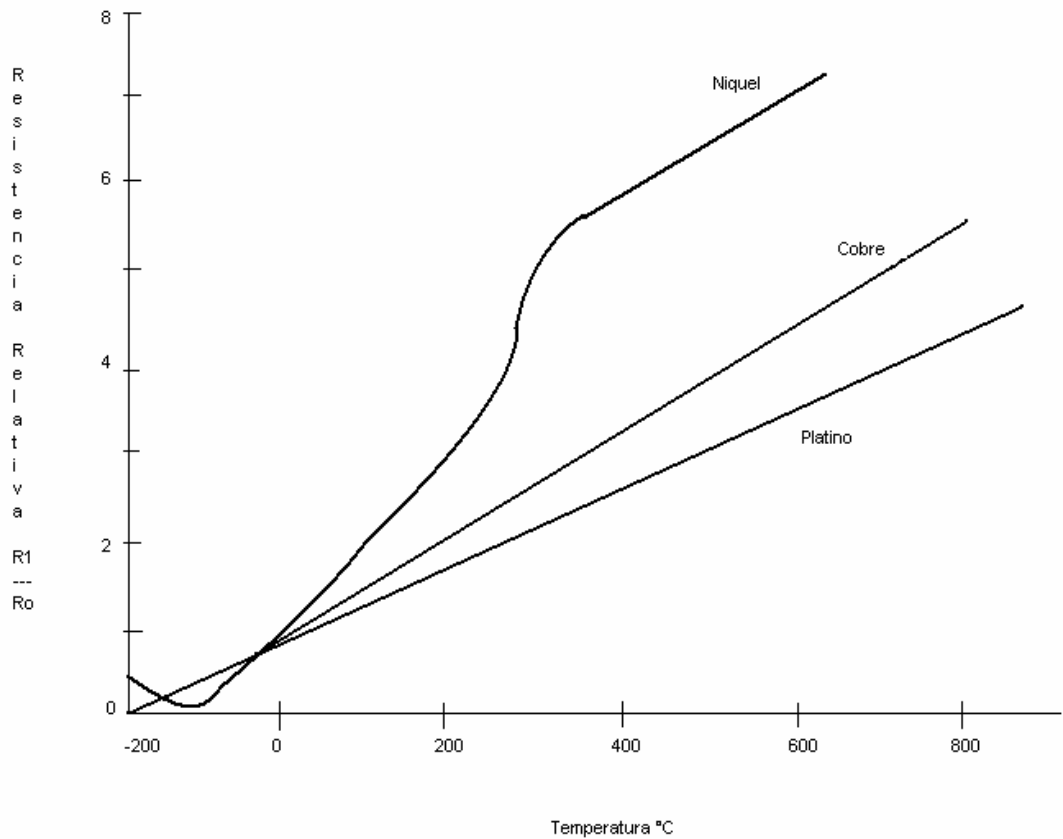
El símbolo general para estos dispositivos es el de la figura 1.1; la línea recta en diagonal sobre el resistor indica que varía de forma intrínseca lineal, y la anotación junto a dicha línea denota que la variación es debida a la temperatura y tiene coeficiente positivo.



**Figura 1. 1:** Símbolo RTD

Un termómetro de resistencia es un instrumento utilizado para medir las temperaturas aprovechando la dependencia de la resistencia eléctrica de metales, aleaciones y semiconductores (termistores) con la temperatura; tal es así que se puede utilizar esta propiedad para establecer el carácter del material como conductor, aislante o semiconductor.

El elemento consiste en un arrollamiento de hilo muy fino del conductor adecuado, bobinado entre capas de material aislante y protegido con un revestimiento de vidrio o cerámica. El material que forma el conductor, se caracteriza por el *coeficiente de temperatura de resistencia*, este se expresa en un cambio de resistencia en ohmios del conductor por grado de temperatura a una temperatura específica. Para casi todos los materiales, el coeficiente de temperatura es positivo, pero para otros muchos el coeficiente es esencialmente constante en grandes posiciones de su gama útil.



**Figura 1. 2:** Curvas para termómetros de resistencia

La figura 1.2 muestra las curvas usuales de termómetros de resistencia para alambre de platino, cobre y níquel, en donde  $R_t$  = resistencia a la temperatura  $t$  y  $R_o$  = resistencia a  $0^\circ\text{C}$ .

La relación entre estos factores, se puede ver en la expresión lineal siguiente:

$$R_t = R_o (1 + \alpha t) \quad \text{(Ecuación 1.1)}$$

Donde:

$R_t$  es la resistencia en ohmios a  $t^\circ\text{C}$

$R_o$  es la resistencia en ohmios a  $0^\circ\text{C}$ .

$\alpha$  es el coeficiente de temperatura de la resistencia.

En el caso de una resistencia fabricada con material semiconductor (termistores) la variación con la temperatura es muchísimo más grande, pero tiene el gran inconveniente de ser de tipo exponencial

$$R_t = R_o (1 - \alpha t - \beta t^2 - \delta t^3 \dots) \quad \text{(Ecuación 1.2)}$$

De las expresiones anteriores se deduce claramente que una resistencia metálica aumenta su valor con la temperatura, mientras que en los semiconductores, aumenta su valor al disminuir la temperatura. Las resistencias de tipo metálico son de uso frecuente debido a que suelen ser casi lineales durante un intervalo de temperaturas bastante elevado.

El empleo de un conductor para la medida de temperaturas, basándose en el comportamiento descrito anteriormente está sometido a varias limitaciones. En primer lugar, es obvio que no se podrán medir temperaturas próximas ni superiores a la de fusión del conductor. En segundo lugar, para poder medir una temperatura determinada con este método es necesario que el sensor esté precisamente a dicha temperatura. Habrá que evitar, pues, auto calentamientos provocados por el circuito de medida. La capacidad de disipación de calor, para un determinado sensor en un ambiente concreto, viene dada por el coeficiente de disipación, y depende del tipo de fluido y su velocidad, en el caso en que sea posible la evacuación de calor por convección.

Otra limitación a considerar es la posible presencia de deformaciones mecánicas, que provocan también un cambio en el valor de la resistencia eléctrica de un conductor. Esta situación puede darse, inadvertidamente, al medir, por ejemplo temperaturas superficiales mediante un elemento adherido a la superficie.

Características que deben poseer los materiales que forman el conductor de la resistencia:

- Alto coeficiente de temperatura de la resistencia, ya que de este modo el instrumento de medida será muy sensible.



- Alta resistividad, ya que cuanto mayor sea la resistencia a una temperatura dada, mayor será la variación por grado; mayor sensibilidad.
- Relación lineal resistencia – temperatura.
- Rigidez y ductilidad, lo que permite realizar los procesos de fabricación de estirado y arrollamiento del conductor en las bobinas de la sonda a fin de obtener tamaños pequeños (rapidez de respuesta).

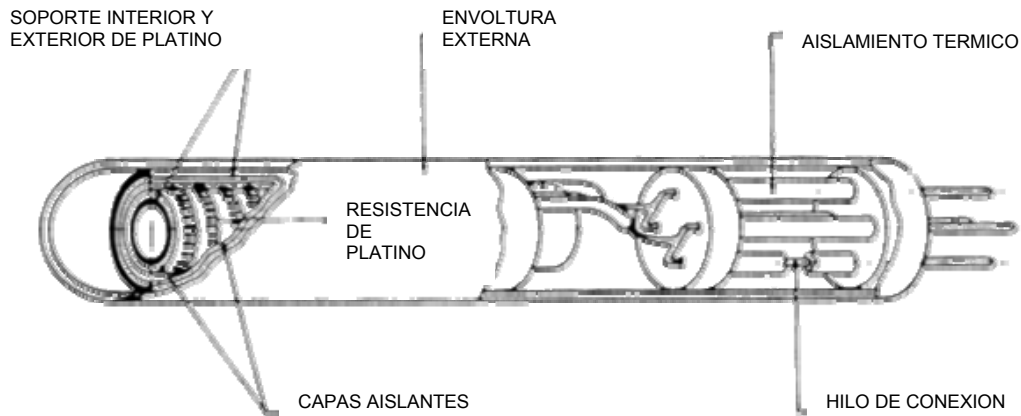
### **Materiales usados normalmente en las sondas**

#### **Platino (Pt)**

Es el material más adecuado desde el punto de vista de precisión y estabilidad, pero presenta el inconveniente de su coste. En general la sonda de resistencia de Pt utilizada en la industria tiene una resistencia de 100 ohmios a 0°C por esta razón, y por las ventajosas propiedades físicas del Pt fue elegido este termómetro como patrón para la determinación de temperaturas entre los puntos fijos desde el punto del Oxígeno (-183°C) hasta el punto de Sb (630.5°C). Con un termómetro de este tipo convenientemente graduado, se pueden hacer medidas con una exactitud de 0.01°C y cambios de temperatura de 0.001°C pueden medirse fácilmente. El valor elegido para  $R_0$  es del orden de 25.5 ohmios aproximadamente; la resistividad del platino aumenta aproximadamente 0.39% de la resistividad a 0°C por grado de elevación de temperatura. A 100°C el valor de  $R_t$  será por consiguiente 35.5 ohmios, aumento de 0.1 ohmios por grado. Para medir hasta 0.01 con un error menor que 1% habría que medirse  $R_t$  con aproximación de 0.00001 ohmios. El elemento medidor puede ser un puente de Wheatstone o un potenciómetro de precisión.

El Platino se emplea mucho en los termómetros de resistencia industriales, en escala de temperatura aproximadamente -50°C hasta 550°C. Los arrollamientos están protegidos contra desperfectos por tubos

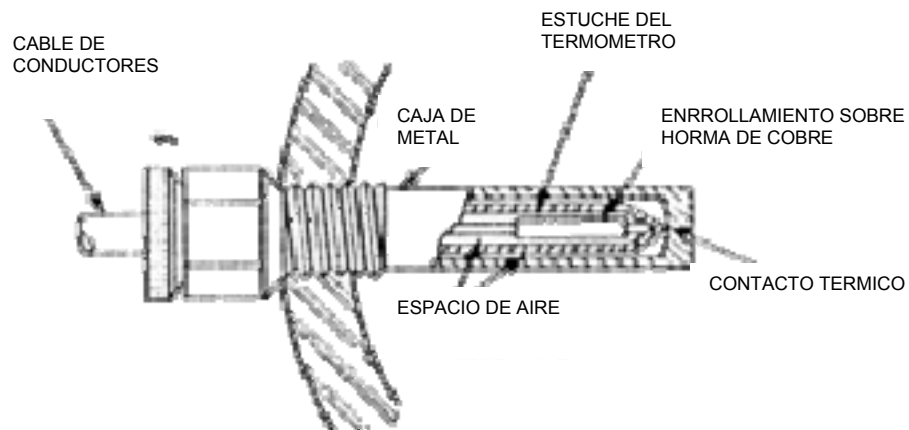
de metal y dispuestos de manera que permiten rápido intercambio de calor en el arrollamiento y el medio en que está colocado el tubo.



**Figura 1. 3: Sonda termométrica de Platino.**

## Níquel

Más barato que el Pt y posee una resistencia más elevada con una mayor variación por grado, el interés de este material lo presenta su sensibilidad; hay una falta de linealidad en su relación R-T. Efectivamente en el intervalo de temperatura de 0 a 100°C, la resistencia de Níquel aumenta en un 62% mientras que el Pt solo aumenta en un 38%. Sin embargo los problemas relativos a su oxidación u otro tipo de deterioro químico, limitan su utilización e incluso ponen en peligro la reproducibilidad de sus medidas. Otro problema añadido es la variación que experimenta su coeficiente de resistencia según los lotes fabricados.



**Figura 1. 4: Termómetro de Resistencia de Níquel**

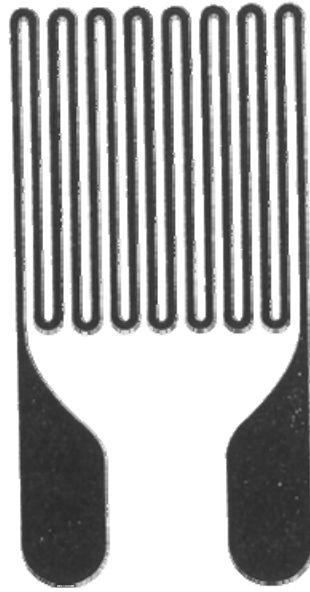
Los termómetros de resistencia de níquel se usan mucho. Su intervalo de valor de  $R_0$  es de 10 a 10000 ohmios; los valores superiores se usan para eliminar el error debido a la variación de resistencia de conductores y contactos; particularmente en los circuitos en los que solo se emplean dos conductores. En este caso el circuito medidor es un puente de Wheatstone equilibrado para una temperatura particular del termómetro. Las variaciones de temperatura desequilibran el puente y la corriente de desequilibrio mide la temperatura. Así el termómetro puede hacerse de lectura directa en el cuadrante de un amperímetro. En instalaciones industriales de precisión en las cuales se consigue el equilibrio del puente por acción manual o por medio de un registrador automático equilibrador, se usan termómetros de tres conductores.

### **Cobre**

El cobre tiene una variación de resistencia uniforme en el rango de temperatura cercano a la ambiente; es estable y barato, pero tiene el inconveniente de su baja resistividad, ya que hace que las variaciones relativas de resistencia sean menores que las de cualquier otro metal. Por otra parte sus características químicas lo hacen inutilizable por encima de los 180°C.

### **Tungsteno**

Tiene una sensibilidad térmica superior a la del platino, por encima de 100°C y se puede utilizar a temperaturas más altas, incluso con una linealidad superior. Así mismo se puede hacer hilo muy fino, de manera que se obtengan resistencias de valor elevado, pero como consecuencia de sus propiedades mecánicas su estabilidad es muy inferior a la del platino. Las técnicas actuales de fabricación de láminas delgadas por evaporación, serigrafía u otro procedimiento ligado a la microelectrónica permiten depositar en superficies muy pequeñas resistencias de los materiales indicados anteriormente



**Figura 1. 5:** Resistencia Metálica evaporada sobre una lámina aislante

### **Métodos de medida** [2]

Para la determinación del valor de la resistencia, es de tipo metálico o semiconductor, se pueden utilizar tres tipos diferentes de montaje. El procedimiento más sensible y de mayor precisión es el de tipo potenciométrico que utiliza dos fuentes de alimentación de corriente estabilizada para alimentar las dos ramas del potenciómetro. En serie con la resistencia a determinar se conecta una resistencia de precisión conocida previamente. Este método es el más preciso de los que vamos a describir porque es un método de cero, con lo que la medida realizada elimina los errores que puedan introducir los conductores de conexión al sensor de platino, pero para ello es necesario que esta resistencia tenga conectado cuatro hilos, dos para la corriente y dos para la lectura de tensión.

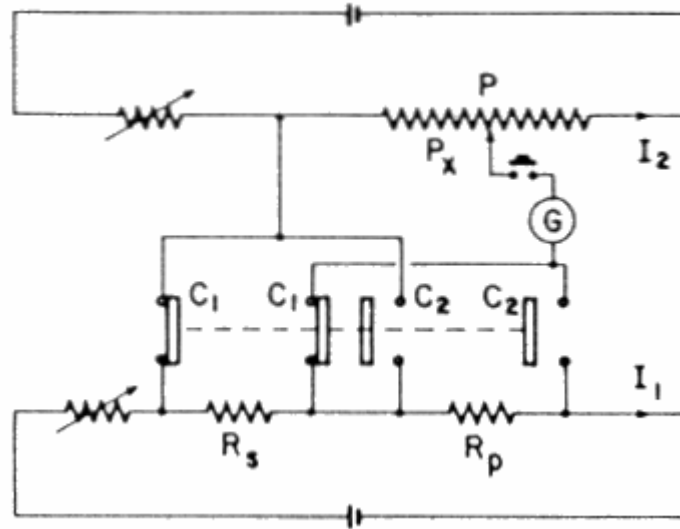
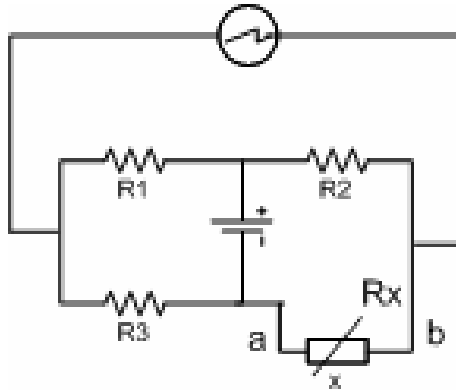


Figura 1. 6: Método Potenciométrico

El segundo procedimiento para realizar esta medida, y también el que iría en segundo lugar en cuanto a precisión, sería el método de medida de resistencias con el puente de Wheatstone, utilizando resistencias de dos, tres o cuatro hilos.

### Montaje de dos hilos

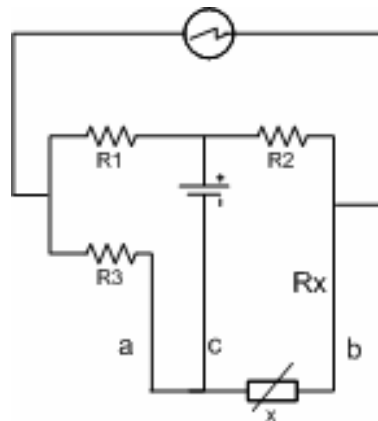
La sonda de resistencia se conecta a uno de los brazos del puente. Es el montaje más sencillo, pero presenta el inconveniente de que la resistencia de los hilos a y b de conexión de la sonda al puente varía cuando cambia la temperatura y esta variación falsea por los tanto la indicación de temperatura; aunque estos hilos sean de baja resistencia y esta sea conocida, las longitudes que puede haber en entre la sonda y el panel donde esté el instrumento receptor, añaden una gran resistencia al brazo de la sonda al tener una longitud considerable.



**Figura 1. 7:** Montaje de dos hilos

### Montaje de tres hilos

Es el más utilizado en la práctica. En este circuito la sonda está conectada mediante tres hilos al puente. De este modo, la medida no es afectada por la longitud de los conductores ni por la temperatura ya que esta influye a la vez en dos brazos adyacentes del puente, siendo la única condición que la resistencia de los hilos a y b sea exactamente la misma.

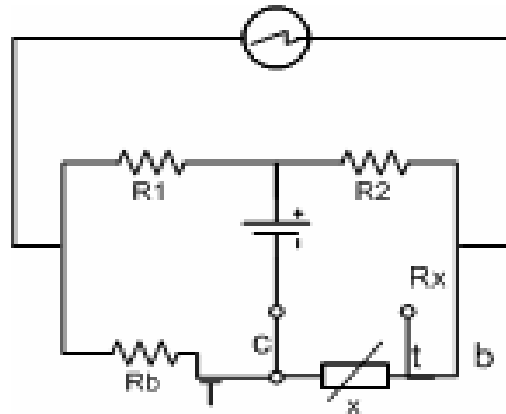


**Figura 1. 8:** Montaje de tres hilos

### Montaje de cuatro hilos

Se utiliza para obtener la mayor precisión posible en la medida como es el caso de calibración de patrones de resistencia en laboratorio. Se basa en efectuar dos mediciones de la resistencia de la sonda combinando las conexiones de modo tal que la sonda pase de un brazo del puente al

adyacente. De este modo se compensan las resistencias desiguales de los hilos de conexión.



**Figura 1. 9:** Montaje a cuatro hilos

Finalmente, el procedimiento que más se utiliza, aunque su precisión depende total y exclusivamente del aparato de medida, es la medida de la diferencia de potencial entre los extremos del sensor alimentado por una fuente de corriente constante. Este corresponde al método de las cuatro puntas de determinación de resistencias. Muchos multímetros vienen preparados para la determinación de resistencias por este método o por el de dos puntas únicamente. También cabe indicar que hay numerosos multímetros que tienen incorporado en su sistema de medida de resistencias la posibilidad de que ésta se corresponda con un termómetro de platino, y por tanto, visualiza directamente los valores de la temperatura en la escala elegida.

### **Termistores**<sub>[1]</sub>

Otro tipo de termómetros de resistencia son los termistores (palabra procedente del inglés thermistor, contracción de "thermally sensitive resistor"), formados por sustancias semiconductoras cuya conductividad eléctrica varía con la temperatura según una función del tipo:

$$R = Ae^{\left(\frac{B}{T}\right)} \quad \text{(Ecuación 1.3)}$$

siendo  $R$  la resistencia del termistor a la temperatura  $T$ , y  $A$  y  $B$  dos constantes cuyos valores típicos son del orden de  $0.05\Omega$  y  $800\Omega K$ , respectivamente.

## Termistores NTC, PTC y Termómetros de Resistencia de Silicio

### Modelos

Son resistores variables con la temperatura, pero no están basados en conductores como las RTD, sino en semiconductores. Si su coeficiente de temperatura es negativo se les llama NTC (Negative Temperature Coefficient), mientras que si es positivo, se denominan PTC. Los símbolos respectivos son los de la figura donde el trazo horizontal en el extremo de la línea inclinada indica que se trata de una variación no lineal.

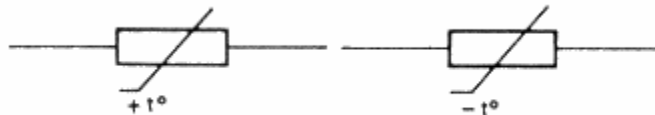


Figura 1. 10: Símbolo PTC y NTC

La principal característica de este tipo de resistencias es que tienen una sensibilidad del orden de diez veces mayor que las metálicas y aumenta su resistencia al disminuir la temperatura. Su fundamento está en la dependencia de la resistencia de los semiconductores con la temperatura, debida a la variación con ésta del número de portadores reduciéndose la resistencia, y de ahí que presenten coeficiente de temperatura negativo. Esta dependencia varía con la presencia de impurezas, y si el dopado es muy intenso, el semiconductor adquiere propiedades metálicas con coeficiente de temperatura positivo (PTC) en un margen de temperaturas limitado. Para las NTC, en un margen de temperaturas reducido ( $50\text{ }^{\circ}\text{C}$ ), la dependencia se puede considerar de tipo exponencial de la forma:



$$RT = R_0 e^{[B(\frac{1}{T} - \frac{1}{T_0})]} \quad \text{(Ecuación 1.4)}$$

Donde:  $R_0$  es la resistencia a 25°C u otra temperatura de referencia, y  $T_0$  es dicha temperatura expresada en Kelvin.

El parámetro  $B$  es la denominada temperatura característica del material, y tiene valores de 2000 K a 5000K, pero varía con la temperatura, aumentando al aumentar esta. Si por analogía con las RTD se define un coeficiente de temperatura equivalente o sensibilidad relativa.

$$\alpha = \frac{\left(\frac{dR_r}{dT}\right)}{RT} \quad \text{(Ecuación 1.5)}$$

En ese caso se obtiene a partir de esta ecuación,

$$\alpha = \frac{-B}{T^2} \quad \text{(Ecuación 1.6)}$$

Que obviamente no es constante, y por lo tanto expresa la no linealidad del comportamiento. A 25 °C y con  $B = 4000\text{K}$  resulta  $\alpha = -4.5 \% / \text{K}$ , que es más de 10 veces superior a  $\alpha$  el Pt.

El valor de  $B$  se puede encontrar midiendo la resistencia del termistor a dos temperaturas conocidas  $T_1$  y  $T_2$ . Si la resistencia respectiva es  $R_1$  y  $R_2$  tendremos:

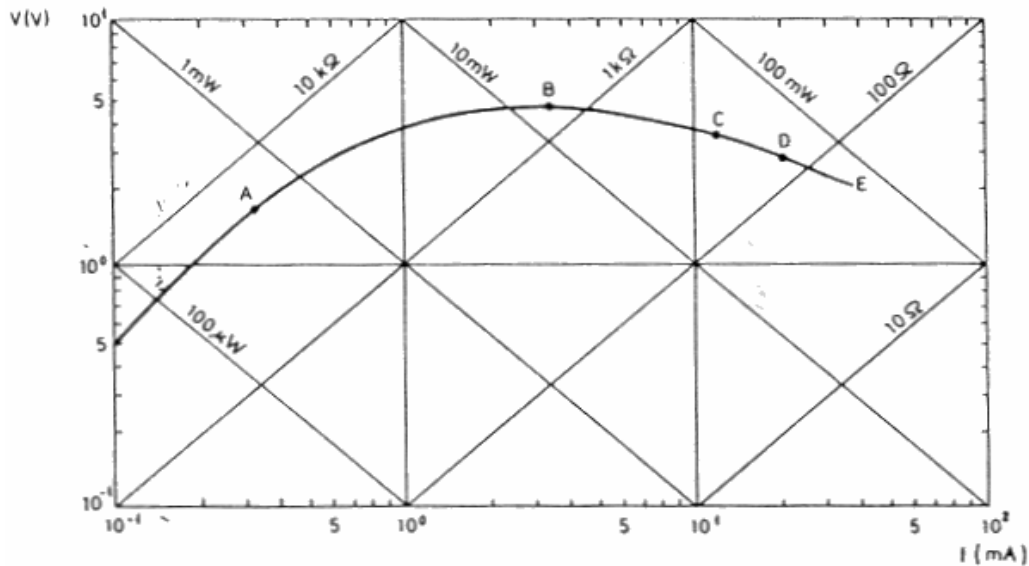
$$B = \ln \left[ \frac{R_1 R_2}{\frac{1}{T_1} - \frac{1}{T_2}} \right] \quad \text{(Ecuación 1.7)}$$

El valor de  $B$  se suele expresar como  $B_{T_1/T_2}$ . Por ejemplo  $B_{25/85^\circ}$ . Para un termistor típico, en el margen de 0 a 50°C el error cometido al emplear un

modelo de dos parámetros es del orden de + 0.3°C. Con tres parámetros se logran errores de + 0.01°C en un margen de 100°C. El modelo viene descrito en este caso mediante la ecuación empírica de Steinhart y Hart.

$$R_t = e^{\left(A+B/T+C/T^3\right)} \quad \text{(Ecuación 1.8)}$$

Con cuatro parámetros se logra un mejor ajuste entre la curva real de evolución de RT y la ecuación teórica. En este caso, se determinan los parámetros midiendo RT a cuatro temperaturas distintas conocidas. Para algunas aplicaciones de los termistores, interesan no tanto sus características resistencia-temperatura sino más bien la relación entre la tensión en los bornes del termistor y la corriente través de él.



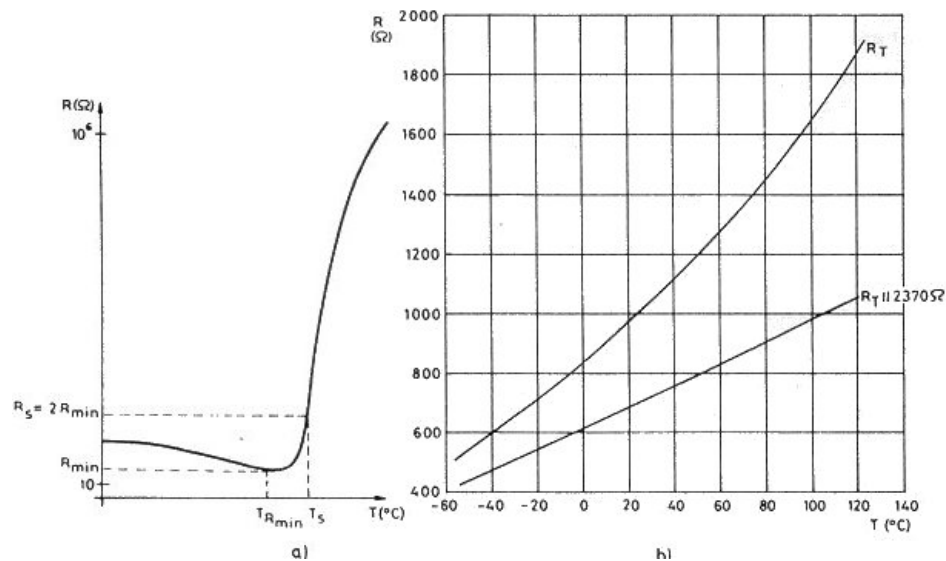
**Figura 1. 11:** Curva de Ajuste con cuatro parámetros

Para corrientes bajas, la tensión en bornes del termistor es prácticamente proporcional a la corriente porque el autocalentamiento del termistor es muy pequeño. Cuando aumenta la corriente, el termistor sufre un autocalentamiento apreciable (punto A de la curva) y alcanza una temperatura por encima de la del ambiente, reduciéndose su resistencia,

y por lo tanto, la caída de tensión a su través. La potencia disponible en el circuito determina el punto en el que se alcanza el régimen estacionario. El punto E, corresponde a la corriente máxima no peligrosa. Al aumentar la temperatura ambiente, la curva se desplaza hacia abajo.

En la zona de autocalentamiento el termistor es sensible a cualquier efecto que altera el ritmo de disipación de calor. Esto permite aplicarla a las medidas de caudal, conductividad calorífica. Si la velocidad de extracción de calor es fija, el termistor es sensible a la potencia eléctrica de entrada y entonces se puede aplicar al control de nivel de tensión o de potencia.

Para las PTC hay dos tipos de comportamiento según la composición y el dopado. Las de tipo cerámico presentan un cambio brusco de resistencia cuando se alcanza la temperatura de Currie de la forma indicada en la figura 1.12.



**Figura 1. 12:** Características Resistencia Temperatura para un posistor (a) y un silistor (b)

Se denominan a veces "positores". Su coeficiente de temperatura es positivo sólo en un margen concreto de temperaturas; fuera de él es negativo o casi nulo. La temperatura de conmutación especificada,  $T_s$ , corresponde a aquella a la que la PTC tiene una resistencia doble del

valor mínimo. Las PTC basadas en Silicio dopado presentan una variación más suave con la temperatura. A veces se comercializan ya linealizadas, con denominaciones tales como *tempsistores* o *silistores*. Este tipo de resistencia, aunque corresponde a un semiconductor, se diferencia de los termistores por las siguientes características: aumenta su resistencia con la temperatura; su variación térmica es menor, lo que permite una linealización mejor utilizando una resistencia de compensación. Debido a que la técnica de fabricación del silicio es extraordinariamente precisa, la tolerancia de la producción es excelente. El intervalo de temperaturas de utilización es, por el contrario, reducido: se extiende desde  $-50^{\circ}\text{C}$  a  $120^{\circ}\text{C}$ . Estas resistencias suelen ser de tipo N, fabricadas por técnicas planas por difusión de impurezas en una lámina de silicio monocristalino. Sus características eléctricas y térmicas dependen de su fabricación y en particular de las impurezas.

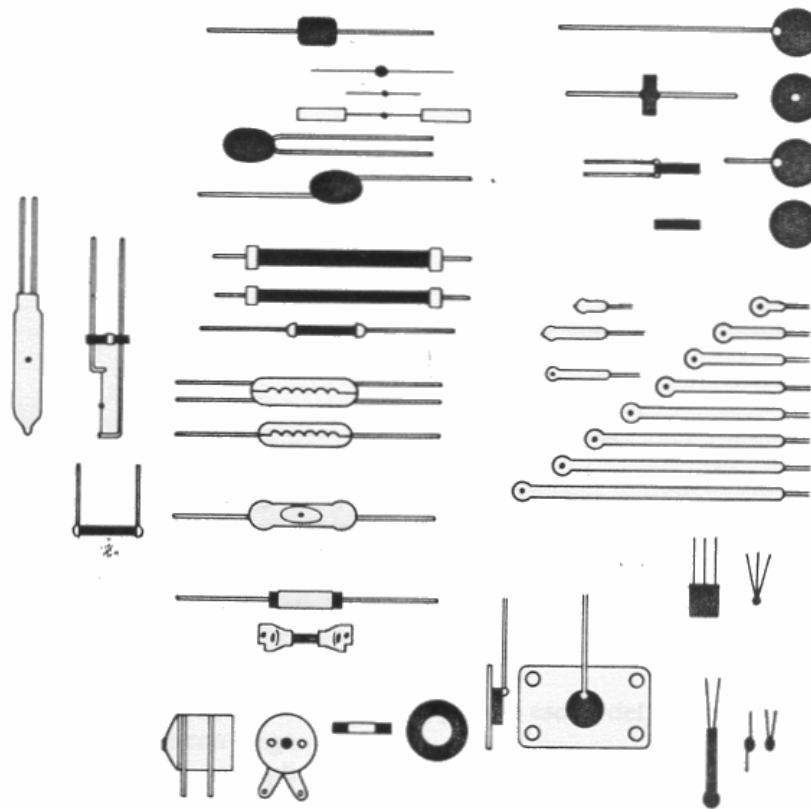
En la figura 1.12 (b) se representaba la curva característica de un silistor. Las limitaciones que cabe considerar en el empleo de los modelos anteriores al aplicar los termistores a la medida de temperatura u otras magnitudes, son similares a las expuestas para las RTD. Aquí hay también un límite impuesto por la temperatura de fusión, y hay que evitar el autocalentamiento si no lo requiere la aplicación pretendida. Mención particular merecen aquí la estabilidad del termistor con el tiempo y el medio. La primera se logra sometiéndolos a un envejecimiento artificial. La segunda se consigue recubriendo el termistor con vidrio si el medio donde va a trabajar le afecta. La intercambiabilidad es otro parámetro a considerar, pues sólo está garantizada para modelos especiales. Por ello, al sustituir un termistor en un circuito suele ser necesario reajustarlo, aunque se trate de una unidad del mismo modelo.

Por su alta sensibilidad permiten obtener alta resolución en la medida de temperatura. Dada su alta resistividad, pueden tener masa muy pequeña, lo que les confiere una velocidad de respuesta rápida y permite emplear

hilos largos para su conexión, aunque éstos vayan a estar sometidos a cambios de temperatura, porque ellos tienen mayor resistencia y coeficiente de temperatura. El costo es muy bajo.

### **Tipos y Aplicaciones**

Las NTC se fabrican a base de mezclar y sinterizar óxidos dopados de metales como el níquel, cobalto, manganeso, hierro y cobre. El proceso se realiza en una atmósfera controlada dándoles la forma y tamaño deseados. La proporción de óxidos determina la resistencia y el coeficiente de temperatura.



**Figura 1. 13:** Diferentes Tipos de NTC

La principal característica de este tipo de resistencias es que tienen una sensibilidad del orden de 10 veces mayor que las metálicas y aumentan su resistencia al disminuir la temperatura. Estos semiconductores están

constituidos por óxidos metálicos, tales como  $MgO$ ,  $MgAl_2O_4$ ,  $MnO$ ,  $FeO$ ,  $CoO$ ,  $NiO$ ,  $ZnTiO_4$ . Los termistores se fabrican por sinterización del semiconductor en polvo, lo que permite preparar resistencias del valor más adecuado y de tamaño reducido, del orden de milímetros. Este procedimiento de preparación, junto a sus características eléctricas, hace de los termistores elementos que pueden realizar la medida de temperaturas en una región muy reducida, casi puntual, y además debido a que su calor específico es pequeño, ofrecen una velocidad de respuesta muy elevada. La estabilidad de un termistor depende de su preparación y de las condiciones de utilización. El rango de utilización de los termistores se extiende hasta temperaturas cercanas a la del helio líquido pero sólo puede subir hasta unos  $300^{\circ}C$ . Hay que tener en cuenta que la temperatura de licuación de estos materiales es bastante baja en comparación con los metales. Como contrapartida, resulta necesario calibrar cada uno de los termistores, pues no son intercambiables los valores de uno a otro elemento. En general tienen una tolerancia del 10 % de su valor nominal. Para altas temperaturas ( $> 1000^{\circ}C$ ) se emplean óxidos de ltrio y circonio.

Las PTC de conmutación están basadas en titanato de bario al que se añade titanato de plomo o de circonio para determinar la temperatura de conmutación. Hay modelos entre  $-100^{\circ}C$  y  $350^{\circ}C$ . Las PTC de medida están basadas en silicio dopado. Las formas en que se presentan las NTC son múltiples, y cada una de ellas está orientada a un grupo concreto de aplicaciones. Las de "gota", "escama" y "perla" se prefieren para aplicaciones de medida de temperatura, mientras que las de "disco", "arandela" y "varilla" son aptas para la compensación y control de temperatura y para aplicaciones con autocalentamiento. En la tabla 1.1 se muestran algunas propiedades de las NTC:

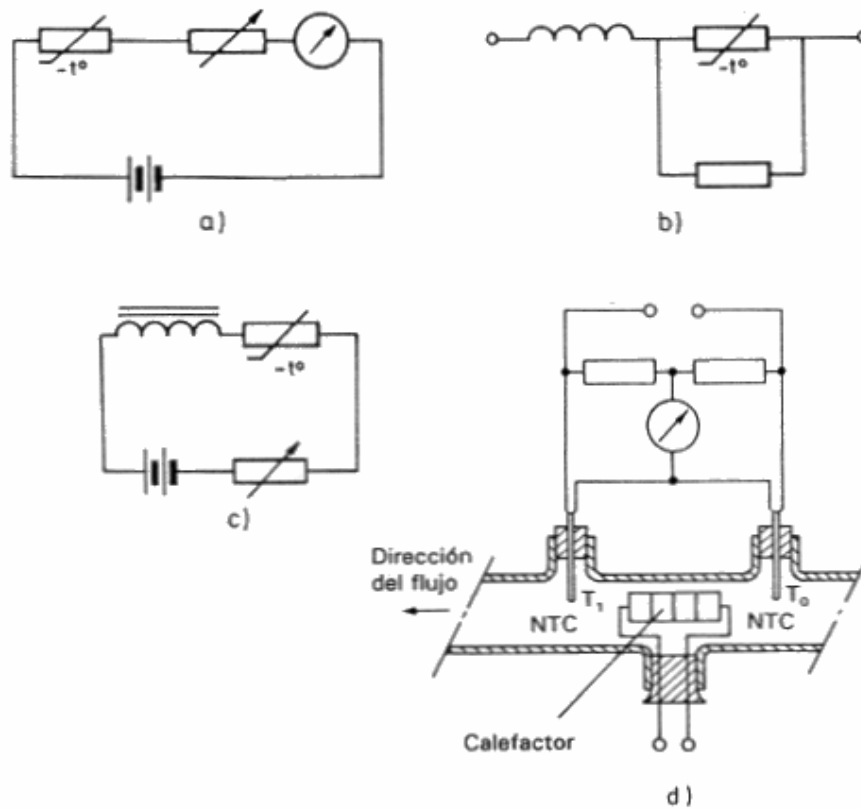
Parámetro	Rango
Margen de temperatura. Resistencia a 25°C.	-100°C a 450°C (no en un mismo modelo) 0.5Ω a 100 M Ω (±5% o ±10%). 1k Ω a 10M Ω es lo habitual.
B Temperatura máxima	2000 K a 5500 K  > 125°C  300°C habitual en régimen permanente  600°C habitual en régimen intermitente.
Coeficiente de disipación, ζ	1 mW/K en aire en reposo 8mW/K en aceite
Constante de tiempo térmica	1 ms a 22 s
Potencia disipable	1mW a 1 W.

**Tabla 1. 1:** Características generales de los modelos de NTC más frecuentes

Las aplicaciones de los termistores se pueden dividir entre las que están basadas en un calentamiento externo del termistor, y las que se basan en calentarlo mediante el propio circuito de medida. Están entre estas última las medidas de caudal, nivel y vacío y el análisis de la composición de gases, todos ellos son casos en que varía la conductividad térmica del medio alrededor del termistor, y también el control automático de volumen y potencia, la creación de retardos de tiempo y la supresión de transitorios.

El circuito de la figura 1.14(a) es indicado para medir una temperatura en un margen reducido, por ejemplo la del agua en automóviles. Consiste en una batería, una resistencia de ajuste en serie, un termistor y un microamperímetro. La corriente a través del circuito varía de forma no lineal con la temperatura debido a la variación de la resistencia del termistor, pero la escala del microamperímetro se puede graduar según convenga. La (b) es una aplicación de compensación térmica. Se trata de

compensar la sensibilidad no deseada de un galvanómetro de hilo de cobre a la temperatura. Como la variación de resistencia del cobre tiene coeficiente positivo, al añadir en serie un resistor con variación negativa puede hacerse que el conjunto presente coeficiente de temperatura casi nulo. Un control dependiendo de la temperatura se basa en el circuito de la figura 1.14(c). Cuando la temperatura ambiente supera un cierto valor, se reduce la resistencia de la NTC con lo que aumenta la corriente y se acciona el relé. El flujo de un líquido se puede medir con el circuito de la figura 1.14(d). Un calefactor inmerso en el líquido libera calor que es transportado por el líquido que fluye, de manera que se crea una diferencia de temperatura entre el líquido, aguas arriba y aguas abajo del calefactor, Esta diferencia de temperaturas depende de forma no lineal, de la velocidad del líquido, y se puede medir con dos termistores.



**Figura 1. 14:** Circuitos de Medición con Termistores



Pueden emplearse las NTC para otras muchas aplicaciones. Aunque  $R_p$  sigue sin ser lineal, su variación con la temperatura es menor que antes, por serlo, el factor que multiplica a  $d(R_p) / dt$ . El coeficiente de temperatura equivalente o sensibilidad es:

$$\frac{dR_p/dT}{R_p} = -\frac{B}{T^2} \frac{1}{1 + R_T/R} \quad \text{(Ecuación 1.9)}$$

Resulta, pues, que se ha perdido sensibilidad a costa de la linealidad ganada. En la siguiente figura se ha representado gráficamente el resultado para el caso  $R_0 = 25$  kilo ohmios,  $B = 4000$  K,  $R = 18500$  ohmios.

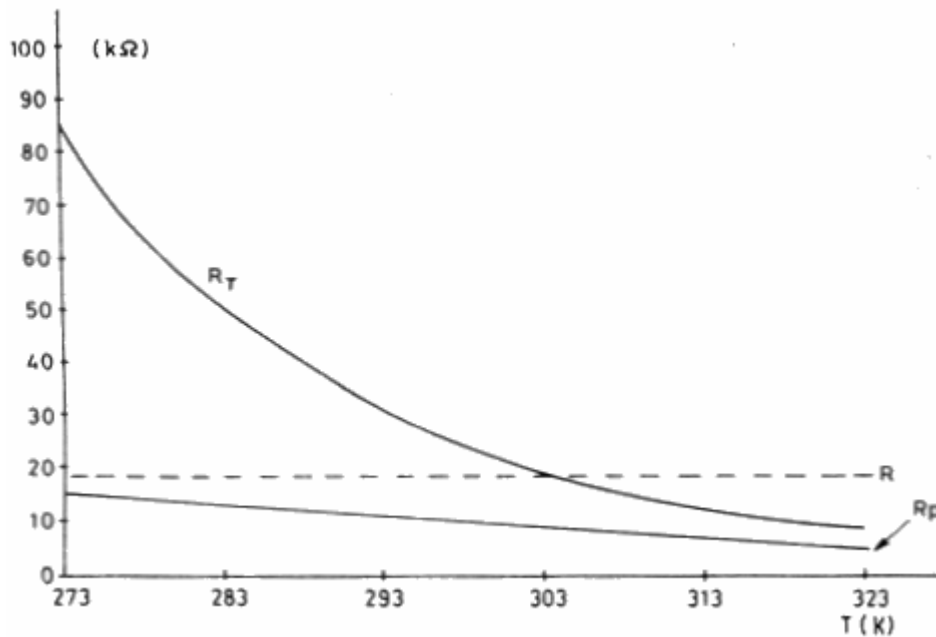


Figura 1. 15: Curva Linealizada

Otro método analítico para linealizar el sistema consiste en forzar un punto de inflexión en la curva resistencia - temperatura que esté justo en el centro del margen de medida ( $T_c$ ). Esa condición se cumple cuando  $R$  tiene un valor dado por

$$R = R_{TC} \frac{B - 2TC}{B + 2TC} \quad \text{(Ecuación 1.10)}$$

La elección de uno u otro método depende de la aplicación. En este último caso se obtiene más linealidad en el centro del margen de medida, empeorando hacia los extremos, mientras que el otro método da una mayor linealidad en las proximidades de cada punto de ajuste.

## Pares Termoeléctricos

### Introducción, principio físico [1]

Las mediciones de temperatura que utilizan termopares se basan en el descubrimiento hecho por Seebeck en 1821 de que una corriente fluye en un circuito continuo de dos alambres metálicos diferentes, si las conexiones o uniones, se encuentran a temperaturas distintas. El termopar se representa esquemáticamente como:

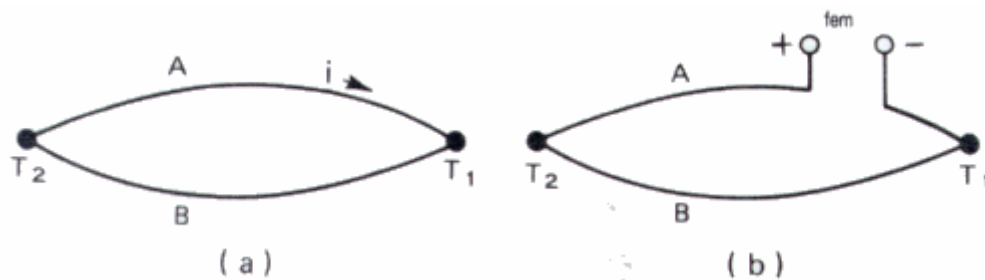


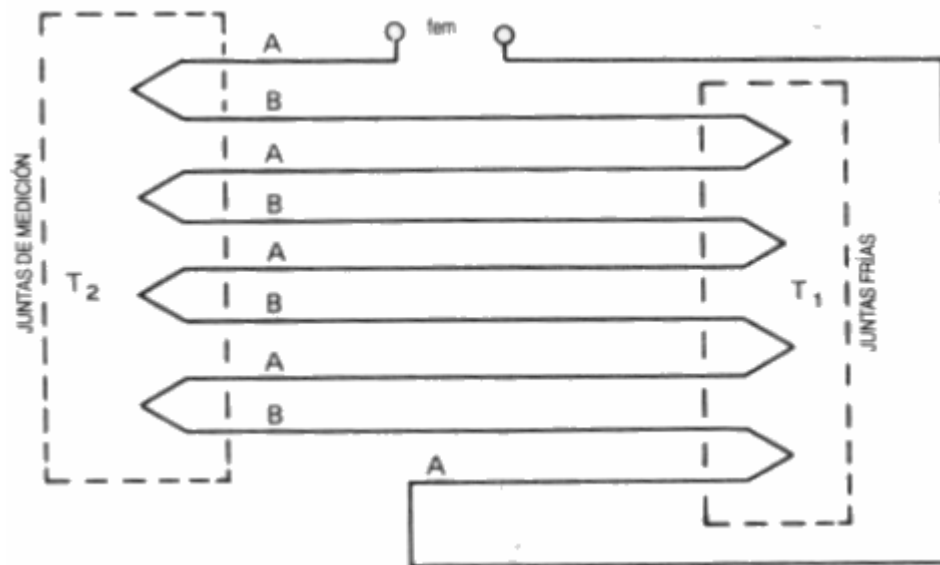
Figura 1. 16: Representación Esquemática de Termopar

A y B son los dos metales, y  $T_1$  y  $T_2$  son las temperaturas de las uniones. Si  $T_1$  es la unión más fría y la corriente termoeléctrica influye en la dirección señalada en la figura 1.16, el metal A se designa por costumbre como el punto termoelectricamente positivo en relación con el metal B.

En circuitos eléctricos la corriente depende de la fuerza electromotriz desarrollada y la resistencia del circuito. Para lograr mediciones de temperatura exactas, el instrumento de medición se construye de tal modo que se mida una fem sin corriente para eliminar los efectos de resistencia del circuito. Como se señala en la figura 1.16(b), la fem térmica es una medida de la diferencia de temperatura entre  $T_2$  y  $T_1$ . En los sistemas de control, la unión o conexión de referencia (unión fría) se

localiza casi siempre en el dispositivo de medición de la fem. La unión de referencia se puede mantener a una temperatura constante, por ejemplo, lo que se tiene en un baño de hielo o en un horno con termostato, o bien, se puede sostener a temperatura ambiente, pero con una compensación eléctrica, de tal manera que parezca conservarse a una temperatura constante.

También se pueden utilizar combinaciones en serie de varios termopares idénticos para proporcionar una señal de medición de temperatura mayor, o para promediar la temperatura en varios puntos.



**Figura 1. 17:** Arreglo de Varios Termopares

Las uniones frías deben encontrarse a la misma temperatura, en caso contrario también será necesario promediarlas. La gran ventaja de los termopares es que la determinación de la temperatura se realiza prácticamente en un punto, donde se haya colocado una de las uniones. También habría que considerar que la capacidad calorífica de un termopar puede ser muy pequeña, con lo que la respuesta a las variaciones de temperatura sería muy rápida. Como ventaja importante habría que señalar finalmente que la salida del sensor es una señal

eléctrica producida por el mismo termopar y por tanto no es necesario alimentarlo con ninguna corriente exterior, eliminando el error correspondiente al calentamiento inherente al efecto Joule. Como inconveniente habría que decir que es necesario mantener la unión de referencia a una temperatura constante y conocida pues la incertidumbre en la temperatura de referencia produce una del mismo orden en la medida.

### **Materiales usados**

Las siguientes combinaciones de materiales son las comúnmente usadas para los pares termoeléctricos:

**Platino:** soldado a una aleación de 90 de platino y 107.9 de radio. Se puede emplear a temperaturas altas hasta 1600°C y es conveniente para uso continuo a temperaturas desde 0°C hasta 1550°C. Este par termoeléctrico fue adoptado por la conferencia General de Pesos y Medidas en 1927 como medio para la definición de la escala internacional de temperaturas, desde la temperatura de fusión del antimonio hasta la temperatura de fusión del oro (630.5°C - 1063°C.). Tiene una fuerza termoeléctrica de unos 10 microvoltios por grado C y es sumamente estable y reproducible cuando se usa debidamente. No debe exponerse nunca a atmósferas reductoras a temperaturas superiores a 500°C., porque se deteriora como resultado de la absorción de gases y del depósito de metales reducidos procedentes de los óxidos del tubo protector y de los aisladores de cerámica.

**Cobre y constantán de Adam:** (aleación de 57% de cobre y 43 % de Níquel, aproximadamente, con porcentajes fraccionados de manganeso y de hierro) Este par es utilizable a temperaturas desde -250 hasta 300°C, y por períodos cortos hasta 400°C. Desde -200 hasta -100°C., la fuerza termoeléctrica es aproximadamente de 20 microvoltios por grado C., y desde 200 hasta 300°C es de unos 55 microvoltios por grado C. El termoeléctrico de *cobre-constantán* es muy usado en las medidas de

laboratorio a bajas temperaturas. No sirve a temperaturas superiores a 300°C a causa de la oxidación del cobre.

**Cromel** (90% de níquel y 10% de cromo) y **Alumel** (94% de níquel, 3% de manganeso, 2% de aluminio y 1% de silicio).

**Hierro y constantán:** Utilizable entre -190 y 870°C.

El intervalo de temperaturas en que es utilizable un termopar está limitado en la zona de bajas temperaturas por la disminución de su poder termoeléctrico y en las altas temperaturas por la posibilidad de que se funda alguno de los conductores que lo constituyen.

### **c. Selección de Tecnología para Adquisición de Datos**

Por las opciones antes descritas, y examinando características y circuitos de aplicación, se utilizará como transductor un termistor en un montaje de dos hilos ya que por propósitos prácticos no se le dará una longitud considerable a los conectores del termistor al puente de Wheatstone así que se espera no tener una variación debido a resistencias parasitas por la longitud de los conectores, como en apartados anteriores se explico.

El arreglo será de un puente de Wheatstone en el cual una de las ramas será el termistor y una de las otras será una resistencia con un potenciómetro para poder controlar nuestro voltaje en la salida del arreglo hacia el amplificador de instrumentación, en este caso se trata de un amplificador de voltaje ya que el trasmisor que se eligió para llevar acabo el proyecto demanda una entrada de voltaje de 0–10VDC, el amplificador de voltaje a utilizar será el Load Cell Amplifier M3200-ce/UR marca Montalvo [ANEXO 3], este es un amplificador de instrumentación el cual puede manejar una entrada de  $\pm 250\text{mV}$  proporcionándonos una salida de proceso ya sea de 0 – 10VDC, 4 – 20mA o 0 – 20mA, en nuestro caso se seleccionara la de 0 – 10VDC por razones mencionadas anteriormente. El amplificador de instrumentación necesita una alimentación de 115V/230V

AC la cual se suministra fácilmente, este amplificador posee una fuente de  $\pm 2.5\text{VDC}$  la cual se utilizara para alimentar el transductor en el arreglo del puente de Wheatstone.

En la tabla 1.2 se observa una comparación entre los diferentes sensores para la adquisición de la temperatura.

Característica	RTD	Termistores	Termopares
Linealidad	Muy alta	Baja	Alta
Requiere Amplificación	Sí, compleja	Sí, no critica	Sí, critica
Variedad	Limitada	Amplia	Regular
Costo	Muy Alto	Bajo	Medio
Disponibilidad en el mercado	Baja	Alta	
Sensibilidad al ruido	No	Media	Alta
Hardware Necesario	Complejo	Sencillo	Sencillo pero critico

**Tabla 1. 2** Tabla Comparativa entre los diferentes sensores

#### **d. Tecnologías Disponibles para Transmisión Inalámbrica**

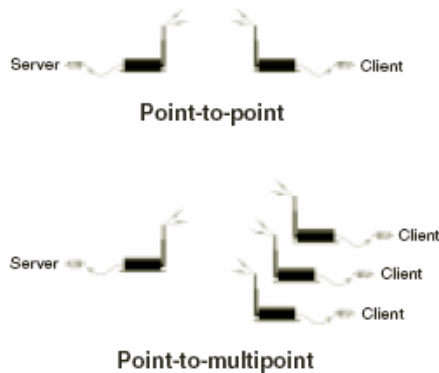
Las tecnologías que encontramos disponibles para la transmisión inalámbrica de la variable analógica (temperatura), con las características necesarias para llevar acabo nuestro monitoreo, entre estas en el transmisor tenga una entrada estandarizada en la industria ya sea de 0 - 10VDC, 0 – 20mA o 4 – 20mA, un convertidor análogo a digital de por lo menos 8 bits, resistencia a ambientes industriales, fácil de transportar. En el receptor salida de puerto serial DB-9 para conectarlo a la PC, así como transmisión entre receptor y PC Standard RS-232, compacto y practico. El alcance que pudieran tener el par transmisor - receptor no es primordial, ya que al funcionar a una corta distancia, para que funcione a una distancia mayor entre uno y otro lo único que se necesitaría fuera un par transmisor – receptor con las mismas características pero con mas potencia de transmisión, además al aumentar la distancia de transmisión

entre estos, el precio es directamente proporcional a la distancia, así que para nuestros fines se decidió buscar uno a nuestro alcance monetario, pero con las características primordiales.

Se encontró un dispositivo llamado Connexlink <sup>[3]</sup>(ver figura 1.18) de la empresa Aerocomm, el cual provee una conexión RS232 por medio de puerto serial DB-9 entre el dispositivo y una PC también hay una conexión inalámbrica entre cada dispositivo Connexlink por medio de RF a una frecuencia de banda libre ya sea de 2.4 GHz o 900 MHz dependiendo de las necesidades, a una velocidad de hasta 115.2 Kbps, cada unidad Connexlink puede ser programada como cliente o servidor, y permite la creación de una red inalámbrica entre ambos (punto a punto), además nos da la opción de tener múltiples clientes y un solo servidor (punto a multipunto) entre dispositivos (Figura 1.19). Cada unidad Connexlink posee una dirección MAC única para identificarse entre si, la transmisión de RF se da por un handshake y después un comando de protocolo con acuse de recibo y detección de errores para asegurar la transmisión de los datos, cada unidad es pequeña, portátil y robusta, indicada para ambientes industriales hostiles, se hace notar que se necesita una fuente de voltaje de alimentación para cada dispositivo. Para llevar acabo la obtención del dato por parte del cliente y transmitirlo al servidor, se tiene que hacer uso de un dispositivo extra el cual tiene que llevar la señal que le entregue un convertidor análogo a digital el cual será alimentado por la señal del amplificador de voltaje o corriente que a su vez esta ligado al transductor ya sea termopar o termistor con puente de Wheatstone, a una señal la cual el dispositivo Connexlink pueda transmitir, para esto se necesitara un adaptador llamado Null MODEM el cual puede hacer esta tarea. Esta es una de las alternativas con la cual se podría implementar la parte del sistema para la obtención y transmisión de la variable analógica hasta el servidor.



**Figura 1. 18** Aerocomm Connexlink 900MHZ Industrial



**Figura 1. 19** Configuraciones para el Connexlink

Otra posibilidad para implementar dicha parte del sistema es por medio de un par transmisor – receptor de la marca Point Six [4]. El transmisor inalámbrico Point Sensor Analog 10V (Figura 1.20) [Anexo 1], su fuente de alimentación es una batería de 3.6V, y posee un convertidor de análogo a digital de 12 bits de resolución, el cual lee la señal analógica de su entrada de 0 – 10VDC, esta entrada será la que se tomara de la salida del amplificador de voltaje o corriente (en este caso tendría que ser de voltaje de 0 – 10VDC), la cual a su vez está ligada a la salida del transductor con puente de Wheatstone. Tiene un ID único que lo identifica, al comunicarse con el receptor transmite la señal modulada a una frecuencia de 418MHz enviando su ID y el valor analógico leído por el ADC combinado con una verificación de error CRC-16 en un pequeño paquete de datos proporcionando un tiempo de encendido de solamente unos 15



milisegundos por cada transmisión, con una tasa de transmisión de 10 a 17 segundos aleatoriamente resultando en un bajo consumo de energía y larga vida de batería. Es de diseño compacto y antena interna.



**Figura 1. 20** Point Sensor Analog 10V

El receptor de RF Point View de 418MHz (figura 1.21) [Anexo 2] está diseñado para recibir y decodificar paquetes de datos de el transmisor Point Sensor, este receptor los decodifica y los traduce a secuencias ASCII que son enviados por el puerto serial RS232 a 19,200 baudios. No necesita ningún handshake o comando de protocolo para realizar la recepción de datos, todos los paquetes de datos son transmitidos serialmente a medida se van recibiendo. La mayoría de puertos RS232 proveen la potencia suficiente para operar el receptor, aunque se provee una entrada de fuente externa para puertos que no puedan suministrar la corriente necesaria. Ambos, transmisor y receptor tienen un exterior robusto en contra de humedad y condensación así como de otros agentes corrosivos, siendo ideal para aplicaciones industriales hostiles.



Figura 1. 21 Point View Receiver

#### e. Selección de Tecnología de Transmisión Inalámbrica

Después de revisar y analizar las diferentes tecnologías presentadas anteriormente, el par transmisor receptor a utilizar será el *Point Sensor Analog 10V* y el *Point View RF Receiver* respectivamente.

El transmisor presenta la ventaja de tener una entrada analógica de 0 – 10VDC, cumpliendo con el estándar industrial para amplificadores de instrumentación con este tipo de salida. Y habiendo elegido como se planteo en la selección de tecnología de adquisición de datos un amplificador de instrumentación con esas características, estos se acoplaran de excelente forma. Además al tener entrada analógica pueden conectarse a él salidas provenientes de otros dispositivos que midan otro tipo de variable analógica y no solo de uno que mida temperatura.

El receptor brinda la facilidad de tener una salida para comunicarse directamente con la computadora por medio de un puerto serie RS232 y un conector DB-9 como se explicó anteriormente, además de poder monitorear múltiples transmisores, brindando la facilidad de poder extender el sistema en un futuro para monitorear diferentes variables simultáneamente en una misma máquina ó en máquinas diferentes.

En la tabla 1.3 se muestran las características más importantes de los dos transmisores.

<b>Características</b>	<b>Connexlink</b>	<b>Point Sensor Analog</b>
Entrada	Digital	Analógica de 0 a 10V
Alcance		600 Pies
Configuraciones Posibles	Punto a Punto Punto a Multipunto	Punto a Punto Punto a Multipunto
Identificación	Dirección MAC	ID de Sensor único
Velocidad de Transmisión	115200 Baudios	19200 Baudios
Frecuencia	900 MHZ, 2.4 GHZ	418 Mhz
Requiere Handshake	Si	No
Conector a PC	DB9	DB9
Fuente de Alimentación	Externa	Interna
Tamaño	Compacto	Compacto
Ambiente	Hostil	Hostil

**Tabla 1. 3** Tabla Comparativa Transmisores Inalámbricos

#### **f. Selección del Puerto de la PC**

La selección del puerto desde donde se leerán los datos provenientes del transmisor está sujeta al tipo de receptor a utilizar y la forma en que este interactúa con el servidor. El puerto a utilizar será el puerto serial RS232 por medio de un conector DB-9, ya que es el puerto por el cual envía los datos recibidos hacia el servidor el *Point View RF Receiver*.

#### **g. Lectura de Datos desde el puerto de la Computadora.**

##### **¿Qué es RS-232?** <sup>[5]</sup>

RS-232 es un estándar de interfaz eléctrica para comunicación serial definida por la Asociación de Industrias Electrónicas (EIA). RS-232 actualmente viene en 3 diferentes modalidades (A, B y C) cada una define un rango de voltaje diferente para los niveles de encendido y apagado. La variedad más comúnmente usada es RS-232C, la cual define un bit de marca (encendido) como un voltaje entre los -3V y los -12V y un bit de espacio (apagado) como un voltaje entre +3V y +12V. Las especificaciones del RS-232C dicen que esas señales pueden viajar hasta 8 mts. (25 ft.) Antes de volverse inestables. Usualmente se pueden mandar señales un poco más lejos que esta especificación mientras la tasa de transferencia sea lo suficientemente baja.

Existen además básicamente dos tipos de conectores, el DB9 como el encontrado en las PC y el DB25 como el encontrado en los MODEM, ambos se muestran en la figura 1.22.

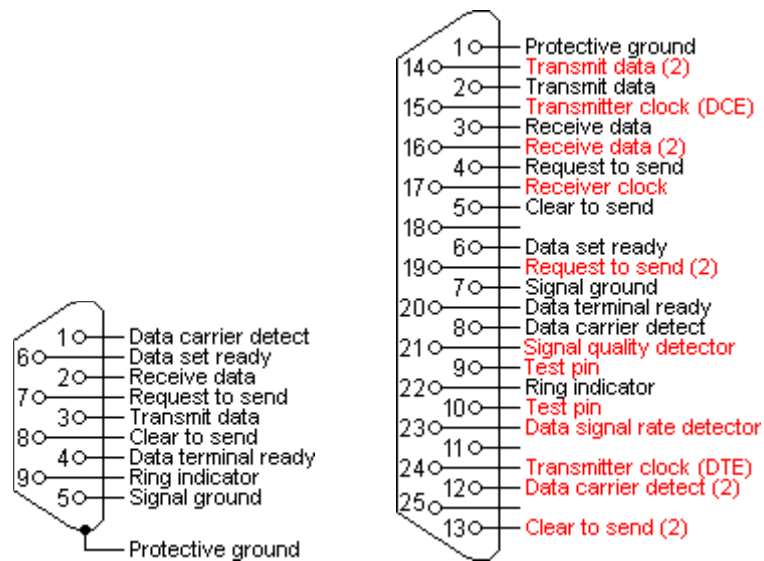


Figura 1. 22: Conectores DB9 y DB25

Para conectar tanto la PC con el receptor *Point Six Viewer* se utilizará un conector de tipo DB9.

### Comunicación asíncrona [5]

Para que la computadora entienda los datos seriales que le están llegando, necesita determinar de alguna manera donde termina un carácter y donde comienza el siguiente.

En modo asíncrono la línea de datos serial permanece en el estado de marca (1) hasta que un carácter es transmitido. Un bit de comienzo precede cada carácter y es seguido inmediatamente por cada bit en el carácter, un bit opcional de paridad, y uno o más bits de paro. El bit de inicio es siempre un espacio (0) y le dice a la computadora que nuevos datos seriales están disponibles. Los datos pueden ser enviados en cualquier momento, de ahí su nombre de asíncrono.

El bit de paridad opcional es la simple suma de los bits de datos indicando si este contiene o no un numero par de bits 1. Con paridad par, el bit de paridad es 0 si hay un número par de unos en el carácter.

Con paridad impar, el bit de paridad es 0 si hay un número impar de unos en el dato. También se escuchan los términos paridad de espacio o

paridad de marca y sin paridad. Paridad de espacio significa que el bit de paridad es siempre 0, mientras que paridad de marca el bit de paridad es siempre 1. Sin paridad significa que ningún bit de paridad es transmitido o esta presente.

Los bits restantes son llamados bits de paro. Puede haber 1, 1.5 o 2 bits de paro entre caracteres y siempre tienen el valor de 1. Los bits de paro tradicionalmente fueron usados para darle a la computadora tiempo para procesar la información del carácter anterior, pero ahora solamente sirven para sincronizar la computadora que recibe los datos con los caracteres que llegan.

Formatos de datos asíncronos son expresados usualmente como “8N1”, “7E1” y así sucesivamente. Esto significa “8 bits de datos, sin paridad, 1 bit de paro” y “7 bits de datos, paridad par, 1 bit de paro” respectivamente.

### Accesando puertos seriales <sup>[5]</sup>

Como con todos los dispositivos, Linux provee acceso a puertos seriales por medio de archivos de dispositivos. Para acceder al puerto serial simplemente se abre el archivo de dispositivo correspondiente.

### Archivos de Puerto Serial <sup>[5]</sup>

Cada puerto serial en un sistema UNIX tiene uno o mas archivos de dispositivo (archivos en el directorio */dev*) asociados con él.

Sistema	Puerto 1	Puerto 2
IRIX	/dev/ttyf1	/dev/ttyf2
HP-UX	/dev/tty1p0	/dev/tty2p0
Solaris/SunOS	/dev/ttya	/dev/ttyb
Linux	/dev/ttyS0	/dev/ttyS1
Digital Unix	/dev/tty01	/dev/tty02

**Tabla 1. 4** Nomenclatura para puerto serial en Linux

## Abriendo el Puerto Serial <sup>[5]</sup>

Ya que en GNU/Linux el puerto serial es un archivo, la función `Open()` es usada para accederlo. La única desventaja con GNU/Linux es que el archivo de dispositivo muchas veces no puede ser accedido por usuarios normales. Para poder hacerlo hay que cambiar los permisos de acceso, corriendo el programa como *root*, o haciendo el programa con la instrucción *set-userid* para que corra como el dueño del archivo del dispositivo, lo cual no es muy recomendado por obvias razones de seguridad.

Por ahora se asume que el archivo es accesible por todos los usuarios. El código para abrir el puerto serial 1 en una PC corriendo GNU/Linux es mostrado a continuación:

```
#include <stdio.h> /*Definición de Standard de Entrada/Salida */
#include <string.h> /*Definición de funciones de String */
#include <unistd.h> /*Definiciones de funciones Standard de UNIX
*/
#include <fcntl.h> /* Definición de Control de Archivos */
#include <errno.h> /* Definición de Números de Error */
#include <termios.h> /*Definiciones de Control de Terminal POSIX
*/

/*
 * `open_port() ` - Abre el Puerto Serial 1.
 * Regresa el descriptor del archivo si tuvo éxito o -1 si hubo
 * error.
 */

Int
open_port (void)
{
    int fd; /* Descriptor del archivo para el Puerto */

    fd = open ("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY);
    if (fd == -1)
    {
```

```

    /*
     * No se pudo abrir el puerto.
     */
    perror ("open_port: Unable to open /dev/ttyS0 - ");
}
else
    fcntl (fd, F_SETFL, 0);
return (fd) ;
}

```

### Opciones de Apertura <sup>[5]</sup>

Se observa que cuando se abre el archivo de dispositivo, se ocupan otras 2 banderas además del modo de lectura+escritura:

```
fd = open ("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NDELAY);
```

La bandera `O_NOCTTY` le dice a Linux que este programa no quiere ser el “controlador de Terminal” para el puerto. Si no se especifica esto entonces cualquier entrada (como la señal de abortar del teclado y demás) afectaran su proceso. Programas como *getty* usan esta función cuando se comienza el proceso de identificación; pero normalmente un programa de usuario no quiere tener este comportamiento.

La bandera `O_NDELAY` le dice a UNIX que este a este programa no le importa el estado en el que este la señal DCD – Ya sea si el otro extremo del puerto este arriba y corriendo. Si no se especifica esta bandera, su proceso será puesto a dormir hasta que la señal de línea DCD este en un voltaje de espacio (1).

### Escribiendo Datos al Puerto <sup>[5]</sup>

Para enviar o escribir datos al puerto solo se ocupa el llamado del sistema `write()`:

```
n = write (fd, "ATZ\r", 4);
```



```
if (n < 0 )
fputs ("write () of 4 bytes failed! \n", stderr);
```

La función de escritura `write` retorna el número de bytes enviados o -1 si un error ocurre. Usualmente el único error con el que se encuentra el usuario es *EIO* cuando un MODEM o enlace de datos bota la línea del detector de portadora de datos (DCD – Data Carrier Detect). Esta condición persistirá hasta que se cierre el puerto.

### **Lectura de Datos desde el Puerto** [5]

Leer datos desde un puerto puede ser un poco confuso. Cuando se opera el dato en modo de datos no depurado, cada llamado de sistema `read(2)` retorna el número de caracteres que están disponibles actualmente en el buffer de entrada serial. Si no hay caracteres disponibles, el llamado se bloquea (espera) hasta que lleguen caracteres, un temporizador de intervalo expire, o un error ocurra. La función de lectura `read` puede hacerse retornar inmediatamente haciendo lo siguiente:

```
fcntl (fd, F_SETFL, 0);
```

La opción `FNDELAY` provoca que la función de lectura retorne 0 si no hay un carácter disponible en el puerto. Para restaurar el comportamiento normal (bloqueo), se llama a `fcntl()` sin la opción `FNDELAY`:

```
fcntl (fd, F_SETFL, 0);
```

Esto también se usa después de abrir el puerto serial con la opción `O_NDELAY`.

### **Cerrando el Puerto Serial** [5]

Para cerrar el puerto serial, solo se usa el llamado de sistema `close`:

```
Close (fd);
```

Al cerrar el puerto serial usualmente se desactiva la señal DTR la que causa que la mayoría de MODEMs cuelgue el llamado.

## La interfaz de Terminal POSIX <sup>[5]</sup>

La mayoría de sistemas soportan la Interfaz de Terminal (serial) POSIX para cambiar parámetros como la tasa de baudios, la medida de los caracteres, y demás. Lo primero que hay que hacer es incluir el archivo `<termios.h>`; esto define tanto la estructura del control de Terminal como las funciones de control de POSIX.

Las dos funciones de POSIX más importantes son `tcgetattr()` y `tcsetattr()`. Estos atributos de Terminal *get* obtener y *set* configurar, respectivamente; se tiene que proveer un puntero a la estructura `termios` que contiene todo las opciones de puerto serial disponibles.

Miembro	Descripción
<code>c_cflag</code>	Control de opciones
<code>c_lflag</code>	Opciones de línea
<code>c_iflag</code>	Opciones de entrada
<code>c_oflag</code>	Opciones de salida
<code>c_cc</code>	Control de caracteres
<code>c_ispeed</code>	Entrada de baudios (nueva interface)
<code>c_ospeed</code>	Salida de baudios (nueva interface)

**Tabla 1. 5:** Miembros de la estructura de TERMIOS

### Opciones de Control

El miembro `c_cflag` controla la tasa de baudios, numero de bits de datos, paridad, bits de paro, y control de flujo de hardware. Hay constantes para todas las configuraciones soportadas.

En la tabla 1.4 se muestran algunas constantes para el miembro `c_cflag`.

Constante	Descripción
CBAUD	Mascara de bit para baud rate.
B0	0 baud (DTR abajo).
B50	50 Baud.
B75	75 Baud.
B110	110 Baud.
B134	134.5 Baud.
B150	150 Baud.
B200	200 Baud.
B300	300 Baud.
B600	600 Baud.
B1200	1,200 Baud.
B2400	2,400 Baud.
B4800	4,800 Baud.
B9600	9,600 Baud.
B19200	19,200 Baud.
B38400	38,400 Baud.
B57600	57,600 Baud.
B115200	115,200 Baud.
EXTA	Reloj de taza externo (External rate clock).
EXTB	Reloj de taza externo (External rate Clock).
CSIZE	Mascara de bit para bits de datos.
CS6	6 bits de datos.
CS7	7 bits de datos.
CS8	8 bits de datos.
CSTOPB	2 bits de paro (1 de no ser asi).
CREAD	Habilitar receptor.
PARENB	Habilitar bit de paridad.
PARODD	Usar paridad impar envez de par.
HUPCL	Colgar (botar DTR) en el último cierre.
CLOCAL	Línea local - No cambiar "propetiario" de puerto.
LOBLK	Bloquear salida de control de trabajo.
CNEW_RTSCCTS	Habilita control de flujo por hardware (no es soportado por todas las plataformas).
CRTSCCTS	

**Tabla 1. 6:** Opciones de Control de `c_flag`

El miembro `c_cflag` contiene dos opciones que siempre deberían estar habilitadas, `CLOCAL` y `CREAD`. Esto asegura que el programa no se vuelva el propietario del puerto sujeto a trabajos esporádicos de control y señales de colgar, y además que el controlador de interfaz serial leerá los bytes de datos entrantes.

Las constantes de baud rate (`CBAUD`, `B9600`, etc.) son usadas en interfaces viejas que carecen de los miembros `c_ispeed` y `c_ospeed`. Nunca se inicializa la bandera `c_cflag` (o cualquier otra bandera) directamente; siempre se debe de utilizar los operadores de bit AND, OR, y NOT para habilitar o deshabilitar bits en la bandera.

### **Configurando la taza de Baud (Baud Rate) [5]**

El Baud Rate es almacenado en varios lugares dependiendo del sistema operativo. Interfaces antiguas guardan el baud rate en el miembro `c_cflag` usando una de las constantes de baud rate en la tabla anterior, mientras que nuevas implementaciones proveen los miembros `c_ispeed` y `c_ospeed` que contienen el valor de baud rate actual.

Las funciones `cfsetospeed` y `cfsetispeed` sirven para configurar el baud rate en la estructura del `termios` sin importar la interfaz de sistema operativo que se este usando. Usualmente se usa el código siguiente para la configuración del baud rate.

```
struct termios options;
/*
 * Obtiene las opciones actuales del puerto
 */
tcgetattr (fd, &options);
/*
 * configura el baud rate a 19,200
 */
cfsetispeed (&options, B19200);
cfsetospeed (&options, B19200);
/*
```

```

*Habilita el receptor y activa el modo local
*/
Options.c_cflag |= (CLOCAL | CREAD);
/*
*configura las nuevas opciones para el puerto
*/
tcsetattr (fd, TCSANOW, &options);

```

La función `tcsetattr` llena la estructura que se provee del `termios` con la configuración actual del puerto serial. Después de configurar el baud rate y habilitar el modo local y la recepción serial de datos, se selecciona la nueva configuración utilizando `tcsetattr`. La constante `TCSANOW` especifica que todos los cambios ocurran inmediatamente sin esperar que los datos de salida terminen de enviarse o datos de entrada terminen de recibirse. Hay otras constantes para que las entradas o salidas terminen de vaciar el buffer de entrada y salida.

La mayoría de sistemas no soporta diferentes velocidades de entrada y salida, así que hay que asegurarse de configurar ambos valores iguales para máxima compatibilidad.

Constante	Descripción
TCSANOW	Hacer cambios de inmediato sin esperar que se completen los datos.
TCSADRAIN	Esperar hasta que todo haya sido transmitido.
TCSAFLUSH	Drenar buffers de entrada y salida y hacer los cambios.

**Tabla 1. 7:** Constantes para `tcsetattr`

### **Configuración del tamaño del carácter** [5]

A diferencia del baud rate, no hay función de conveniencia para configurar el tamaño del carácter. En vez de esto, se debe hacer un enmascaramiento de bits. El tamaño del carácter esta especificado en bits:

```
options.c_cflag &= ~CSIZE; /* Enmascara el tamaño del carácter de bits */
options.c_cflag |= CS8; /*Selecciona 8 bits de datos */
```

### **Configurando el chequeo de paridad** [5]

Así como el tamaño de carácter se debe configurar manualmente, también se hace lo mismo con la habilitación de paridad y el tipo de bits de paridad. Los dispositivos seriales de GNU/Linux soportan generación de bit de paridad par, impar y sin paridad.

#### **Sin paridad (8N1):**

```
options.c_cflag &= ~PARENB;
options.c_cflag &= ~CSTOPB;
options.c_cflag &= ~CSIZE;
options.c_cflag |= CS8;
```

#### **Paridad par (7E1):**

```
options.c_cflag |= PARENB;
options.c_cflag &= ~PARODD;
options.c_cflag &= ~CSTOPB;
options.c_cflag &= ~CSIZE;
options.c_cflag |= CS7;
```

#### **Paridad impar (7O1):**

```
options.c_cflag |= PARENB;
options.c_cflag |= PARODD;
options.c_cflag &= ~CSTOPB;
options.c_cflag &= ~CSIZE;
```

```
options.c_cflag |= CS7;
```

## Opciones Locales

Los modos locales del miembro `c_lflag` controlan como se manejan los caracteres de entrada por el controlador serial. En general se configura el miembro `c_lflag` para una entrada de datos canónica o corriente.

Constante	Descripción
ISIG	Habilita las señales SIGINTR, SIGSUSP, SIGDSUSP, y SIGQUIT.
ICANON	Habilita la entrada canónica (o corriente).
XCASE	Habilita mayúsculas/minúsculas (obsoleto).
ECHO	Habilita el eco en los caracteres de entrada.
ECHOE	Borra el eco en un carácter como BS-SP-BS.
ECHOK	Después de eliminar el carácter le da eco NL.
ECHONL	Eco NL.
NOFLSH	Deshabilita el drenaje del buffer de entrada después de interrumpir o quitar caracteres.
IEXTEN	Habilita las funciones extendidas.
ECHOCTL	Eco a caracteres de control como ^ y borra como ~?
ECHOPRT	Da eco al caracter borrado como caracter borrado.
ECHOKE	La línea entera BS-PS-BS en eliminación de línea.
FLUSHO	Salidas siendo drenadas.
PENDIN	Reescribe entrada pendiente en la siguiente lectura o caracter de entrada.
TOSTOP	Envía a SIGTTOU para salida de fondo.

**Tabla 1. 8** Constantes para el miembro `c_lflag`

## Escogiendo una entrada canónica <sup>[5]</sup>

La entrada canónica es orientada a línea. Los caracteres de entrada son puestos en un buffer el cual puede ser editado interactivamente por el

usuario hasta que los caracteres CR (retorno de acarreo, carriage return) o LF (alimentación de línea, line feed) son recibidos.

Cuando se selecciona este modo de entrada, normalmente se seleccionan las opciones ICANON, ECHO y ECHOE:

```
options.c_lflag |= (ICANON | ECHO | ECHOE);
```

### Escogiendo una entrada ordinaria [5]

Una entrada ordinaria no es procesada. Los caracteres de entrada pasan exactamente como son recibidos. Generalmente se quita la selección de las opciones ICANON, ECHO y ISIG cuando se usa una entrada ordinaria.

```
Options.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG);
```

Nunca se debe habilitar el eco de entrada (ECHO, ECHOE) cuando se mandan comandos a un MODEM u otra computadora que también este haciendo uso del eco en caracteres, ya que se generara un lazo de retroalimentación entre las dos interfaces seriales.

### Opciones de entrada

Los modos de entrada del miembro `c_iflag` controlan cualquier proceso de entrada que es hecho a caracteres recibidos en el puerto. Como el campo `c_cflag`, el último valor guardado en `c_iflag` es la instrucción OR de la opción deseada.

En la tabla 1.8 se muestran las opciones del miembro `c_iflag`.

Constante	Descripción
INPCK	Chequeo de paridad habilitado.
IGNPAR	Ignorar errores de paridad.
PARMRK	Marcar errores de paridad.
ISTRIP	Monitorear bits de paridad.
IXON	Habilitar control de flujo por software (salida).
IXOFF	Habilitar control de flujo por software (entrada).



Constante	Descripción
IXANY	Permitir a cualquier carácter comenzar el flujo de nuevo.
IGNBRK	Ignorar condición de quiebre
BRKINT	Enviar una SIGINT cuando una condición de quiebre es detectada.
INLCR	Mapear NL a CR.
IGNCR	Ignorar CR.
ICRNL	Mapear CR a NL.
IUCLR	Mapear mayúsculas a minúsculas.
IMAXBEL	Eco BEL si la línea de entrada es demasiado larga.

**Tabla 1. 9:** Constantes para el miembro `c_iflag`

### Configurando las opciones de entrada de paridad.

Se debería tener habilitado el chequeo de paridad en la entrada cuando se tiene habilitada la paridad en el miembro `c_cflag` (`PARENB`). Las constantes relevantes para el chequeo de paridad en la entrada son `INPCK`, `IGNPAR`, `PARMRK` y `ISTRIP`. Generalmente se selecciona `INPCK` y `ISTRIP` para habilitar el chequeo y monitoreo del bit de paridad.

```
options .c_iflag |= (INPCK | ISTRIP);
```

`IGNPAR` es una opción un poco peligrosa que le dice al controlador serial que ignore los errores de paridad y pase los datos recibidos como si no hubieran ocurrido errores. Esto puede ser útil para examinar la calidad de un vínculo de comunicación, pero en general no se utiliza para razones prácticas.

`PARMRK` hace que los errores de paridad se marquen en el flujo de entrada usando caracteres especiales. Si `IGNPAR` esta habilitado, un carácter `NULL` (000 octal) es enviado al programa antes de cualquier carácter con un error de paridad. De no ser así un carácter `DEL` (177 octal) y `NULL` son enviados con el carácter malo.

## Esquema General del Proyecto

El esquema general del proyecto se presenta en la figura 1.23 y 1.24

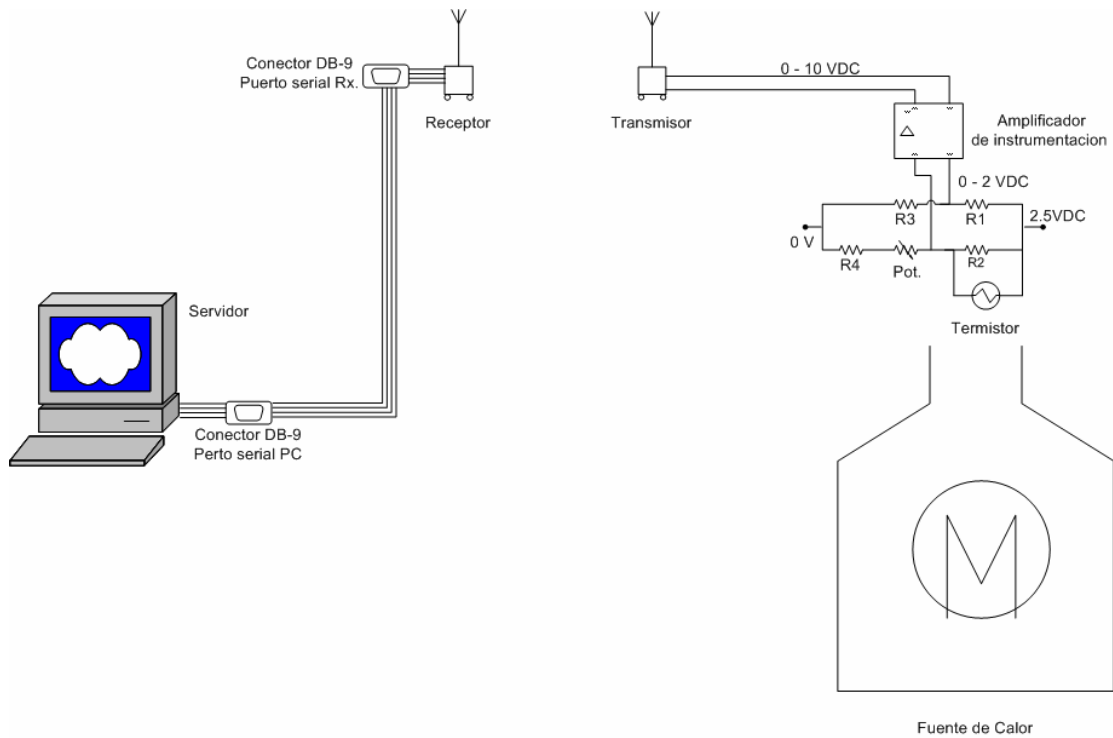
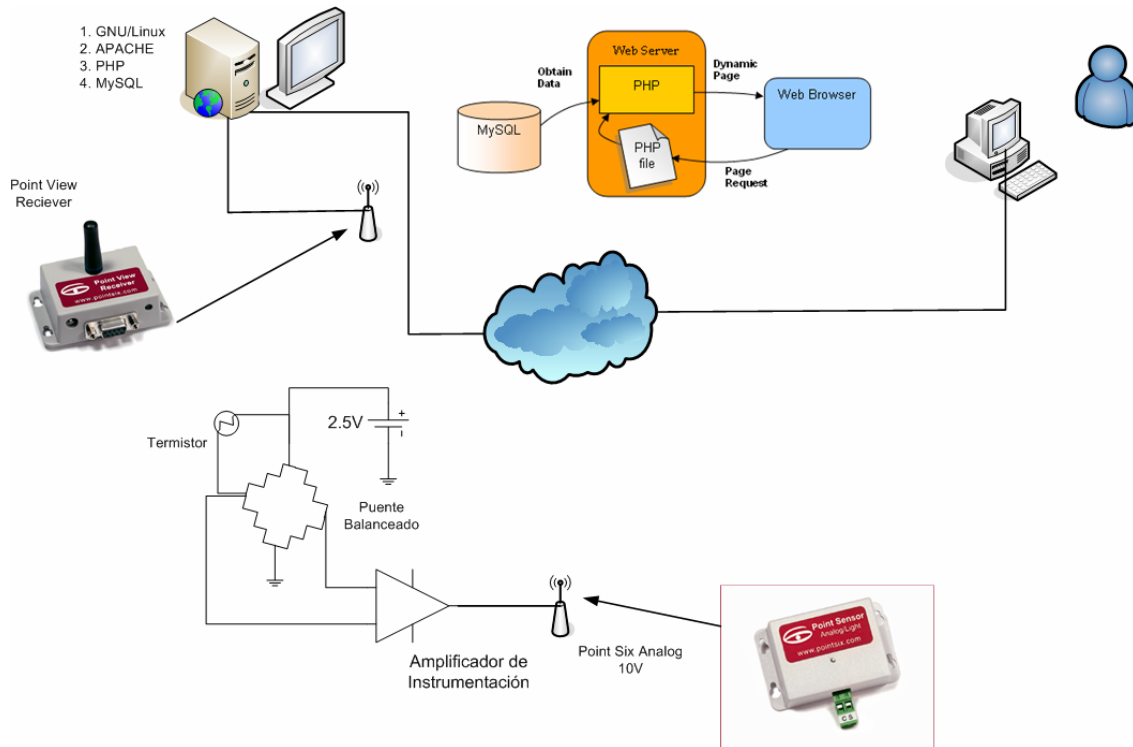


Figura 1. 23: Esquema de Conexiones



**Figura 1. 24:** Esquema de Proceso

En las figuras anteriores observamos como el sensor permanentemente toma valores de temperatura y los convierte en una señal eléctrica a través de un puente de Wheatstone. Como el valor obtenido es demasiado pequeño este debe amplificarse por medio del amplificador de instrumentación Montalvo, para que la señal sea adecuada para la entrada del transmisor inalámbrico, éste transmite en el formato que se muestra en el Anexo 2 hacia el receptor *Point View*.

Por otra parte, en el servidor un programa escrito en C leerá continuamente el puerto serial `ttys0`. Este está configurado para leer una entrada canónica o sea que debe esperar por un `<CR>` o `<LF>` para indicar que ya terminó la secuencia. Además debe estar configurarse para leer a 19200 Baudios y la transmisión será asíncrona.

En el mismo Anexo 2 podemos ver que hay un número de serie único para el transmisor, es de esta forma que en caso de tener más de un transmisor el receptor diferencia cual de todos es el que ha transmitido. También en la secuencia esta incluido el campo de datos en formato

ASCII, que es el que nos interesa, éste debe ajustarse para obtener el valor de temperatura deseado.

El programa en C++ además se encargará de insertar el dato leído y ajustado en la tabla llamada *Temperatura* de la Base de Datos *temperatura*.

En el lado del servidor, tendremos Linux Mandrake como sistema operativo, sobre este estarán los servidores *Apache* y *MySQL* que albergarán las páginas Web y la base de datos respectivamente.

Cuando un cliente solicita ver el gráfico dinámico, un script escrito en PHP leerá los datos almacenados en la base de datos y generará un gráfico que será enviado al navegador como HTML, por lo que este no debe tener instalado ningún plugin adicional para poder visualizarlo. Este gráfico se refrescará cada cierto tiempo, el cual debe ser lo suficientemente grande para permitir que el transmisor envíe el siguiente dato y sea guardado en la base de datos.

En la segunda parte del documento se discutirá el trasfondo teórico necesario para cada uno de los componentes de software que involucra el proyecto para que en la PARTE III nos centremos específicamente en el desarrollo de los scripts y la instalación del software necesario.

## PARTE II

### A. Sistema Operativo Mandrake Linux

#### a. Descripción

Mandrake Linux es una distribución *GNU/Linux* soportada por Mandrake Soft S.A. MandrakeSoft que nació en Internet en 1998 con el objetivo principal de proporcionar un sistema *GNU/Linux* amigable y fácil de usar. Los pilares para el desarrollo de esta distribución son el código abierto y el trabajo colaborativo.

Originalmente, Mandrake Linux se basó en Linux Red Hat [6] (que ha sido el líder en el sector Linux casi desde su aparición [7]), siguiendo su misma jerarquía del sistema de archivos con ciertas modificaciones que lo hacían ligeramente más amigable y con herramientas de configuración básicas.

Con el tiempo y a medida que se fue haciendo más popular entre la comunidad de usuarios se convirtió en una distribución totalmente independiente con un grupo de desarrolladores dedicados por completo a su continuo mejoramiento, más el aporte de la comunidad a través de Internet. En este momento se ha convertido en el principal sistema Operativo tipo Linux de escritorio, en gran medida por la filosofía de MandrakeSoft, de llevar Linux a cualquier usuario desde el gurú, hasta el que recién ingresa al mundo de las computadoras; siendo muy estable y robusto con una gran cantidad de herramientas de configuración para los diferentes servicios. Pero además está presente en el mundo de los servidores, siendo el sistema operativo utilizado en Francia en las instituciones de gobierno [8].

Como otras distribuciones *GNU/Linux* Mandrake Linux también se pueden descargar desde Internet [9], esta versión es gratuita y cualquiera la puede descargar y copiar sin restricción alguna. Incluye el software necesario para convertir una PC en una poderosa estación de trabajo y/o convertirla en servidor Web, de Correo, FTP entre otros. La única diferencia con los

productos comerciales de MandrakeSoft es que no incluye software comercial.

Para el desarrollo de este proyecto se utilizará la versión gratuita.

## **b. Consideraciones Previas a la Instalación**

### **Configuración del BIOS**

El **BIOS** (*Basic Input/Output System*, **Sistema Básico de Entrada/Salida**) es utilizado para poder arrancar una computadora, es la forma más primitiva de interacción con el hardware, pero necesaria para inicializarlo.

Para acceder a éste generalmente se logra presionando la tecla **Supr** pero depende del BIOS que posea por lo que se debe buscar la opción apropiada para cada caso.

Una vez que se ha ingresado se busca la opción **Boot Sequence** (Secuencia de Arranque) y se configura el CDROM como el primer dispositivo de arranque. Una vez hecho esto se guardan los cambios, generalmente presionando la tecla **F10**. Esto permitirá ejecutar la instalación de Linux desde el CDROM.

### **Hardware Soportado**

Mandrake Linux puede manejar una gran cantidad de dispositivos de hardware. Es siempre recomendable tener un detalle del hardware que se posee para evitar problemas durante la instalación y confirmar que es soportado por Mandrake Linux confirmando en el sitio Web

Por lo general no existe ningún problema con el hardware salvo que sea demasiado antiguo pero siempre es una buena práctica revisar si está soportado [10].

## c. Discos y Particiones

### **Estructura de una unidad de disco duro** <sup>[11]</sup>

Básicamente el disco está dividido en pequeños sectores, una secuencia de sectores forman una partición. Se pueden crear tantas particiones como se desee, cada una de las cuales es reconocido como un disco duro individual por el sistema operativo.

Una unidad de disco duro es una secuencia de sectores. Un sector es la unidad de datos más pequeña en un disco duro.

### **Particiones**

El uso de particiones múltiples permite crear muchas unidades de discos virtuales dentro del disco físico real. Esto tiene muchas ventajas:

- Los diferentes sistemas operativos usan sistemas de archivos diferentes (formato), al tener múltiples particiones se hace posible instalar varios sistemas operativos como *Windows* y *GNU/Linux*.
- Permite mantener separados datos personales del sistema operativo incluso si utilizan el mismo sistema de archivos, así que si se desea actualizar o reinstalar el sistema operativo no se ponen en riesgo los datos personales.
- Los errores físicos en el disco generalmente se ubican en sectores adyacentes y no están distribuidos por todo el disco. Al distribuir los archivos en las diferentes particiones se limitarán las pérdidas de datos en caso de daños físicos al disco.

### **Particiones necesarias para Instalar Linux** <sup>[12]</sup>

Para poder llevar a cabo la instalación de Linux son necesarias al menos dos particiones y suficiente espacio si es que se piensa instalar como servidor.

Realmente se pueden crear varias particiones pero al menos una para el árbol de archivos o directorio raíz y otra partición especial de intercambio de memoria del tipo *Swap*, ésta es utilizada por el sistema de archivos

como memoria virtual y se recomienda que sea el doble de la memoria física instalada.

#### d. Instalación y Desinstalación de Programas <sup>[12]</sup>

Lo que hace la diferencia entre el software libre y el software propietario es el acceso al código fuente del software libre. Esto significa que los programas se distribuyen como paquetes de archivos fuente. Esto se convierte en un problema para los principiantes porque tienen que compilar ellos mismos el software para poder utilizarlo.

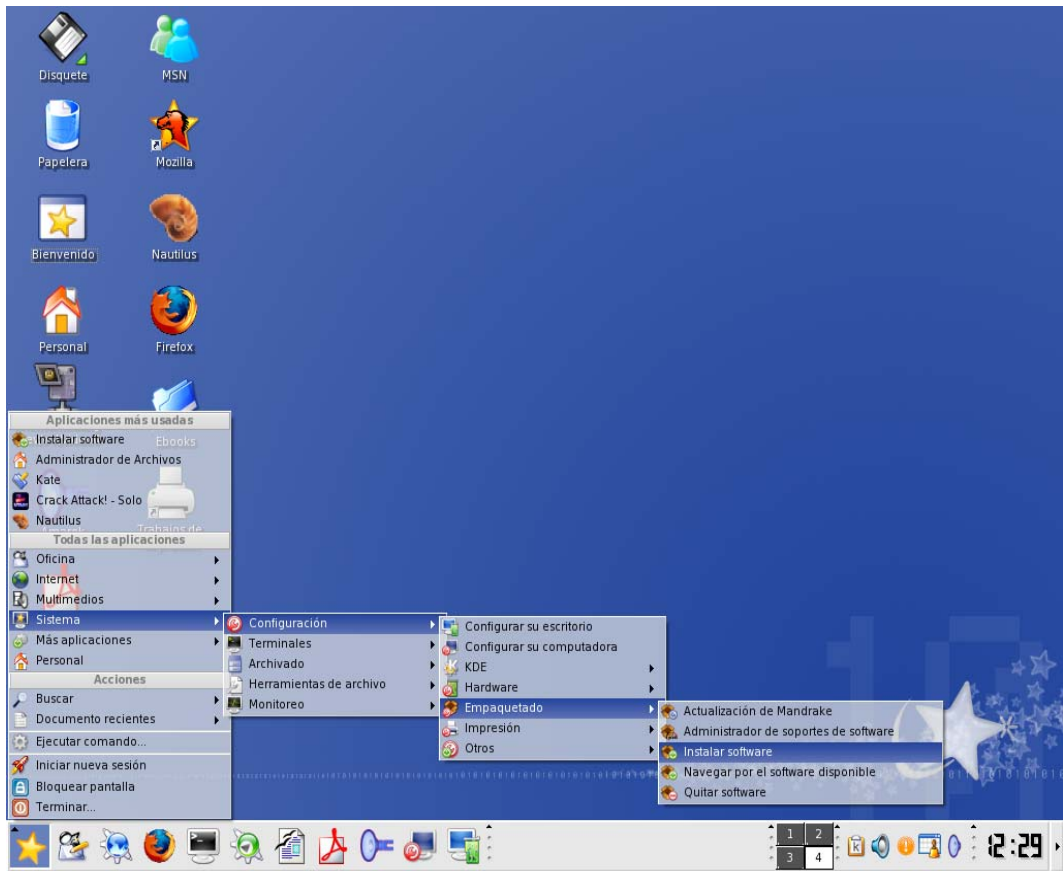
Aunque existe versiones precompiladas del software libre más popular siempre depende del sistema operativo que se este utilizando.

Afortunadamente existen varios administradores de paquetes que contienen la información necesaria para poder instalar el software con mínima intervención del usuario. El paquete de software se encarga de suministrarle la información necesaria al administrador de paquetes y este se encarga de verificar las dependencias de otros paquetes, verificación de firmas por motivos de seguridad, la ubicación donde se colocaran los archivos y todo lo referente a la instalación. Además existen paquetes preconfigurados para cada distribución Linux. Dentro de estos administradores de paquetes se pueden mencionar el administrador de paquetes de *Debian* y el más famoso de todos: **Red Hat Package Manager** o **RPM**.

*Mandrake Linux* utiliza paquetes RPM para instalar software y posee una poderosa interfaz gráfica conocida como **RpmDrake**, que brinda información sobre el paquete que esta siendo instalado, además el software esta agrupado por categorías.

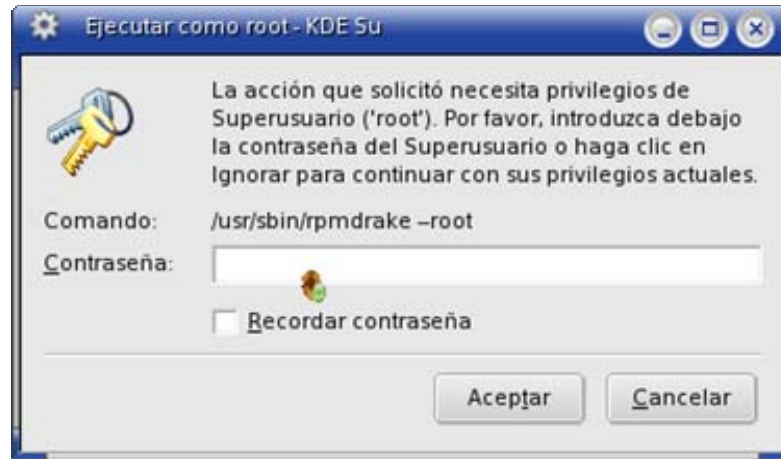
Para poder lanzar *RpmDrake* desde el escritorio ingresamos al menú de inicio, seleccionamos la categoría de *Sistema*, seleccionamos *Configuración, Empaquetado e Instalar Software*, como se muestra en la figura 2.1.





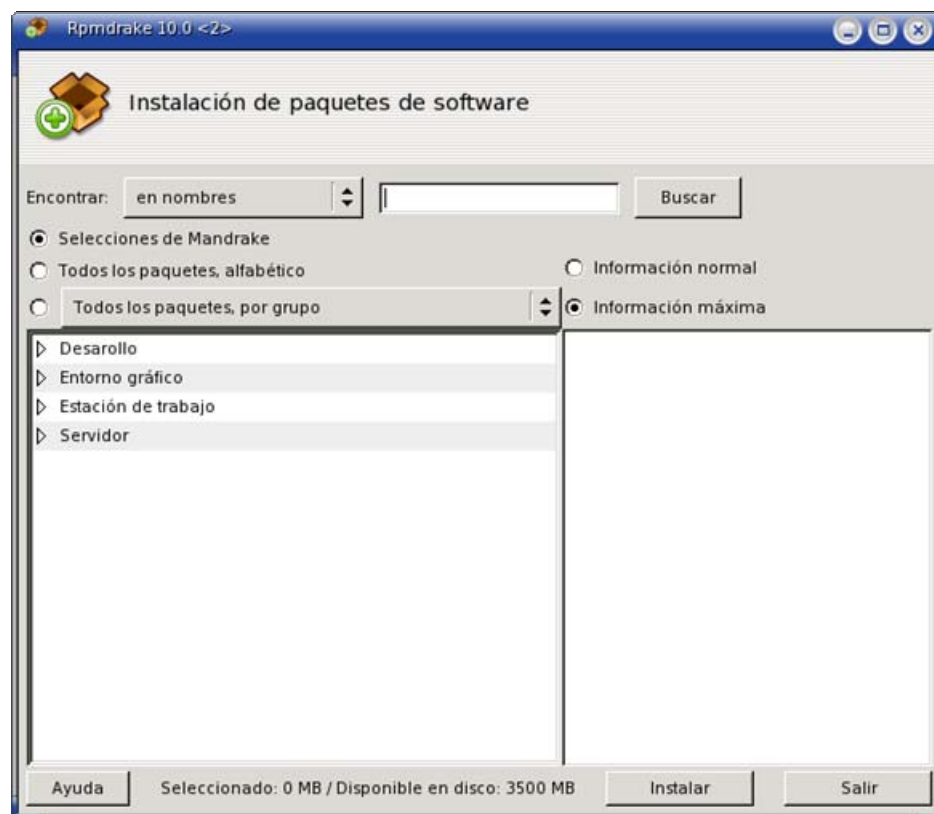
**Figura2. 1** Selección de RPM DRAKE

Una vez que seleccionamos *Instalar Software*, se aparecerá un cuadro de verificación donde ingresamos la contraseña de *root*.



**Figura2. 2:** Cuadro de Verificación

Si ingresamos correctamente la contraseña aparecerá el administrador de paquetes



**Figura2. 3:** Administrador de Paquetes RPM

En la figura 2.3 se observa que los paquetes están ordenados por categorías, pero se pueden ordenar por nombres, por descripción o todos los paquetes por orden alfabético. Se observa además que hay un campo de búsqueda donde ingresamos el nombre del paquete que deseamos instalar, también podemos navegar por el árbol de paquetes disponibles que se ve a la izquierda. Si quisiéramos instalar el paquete Apache obtendríamos una pantalla como la siguiente:

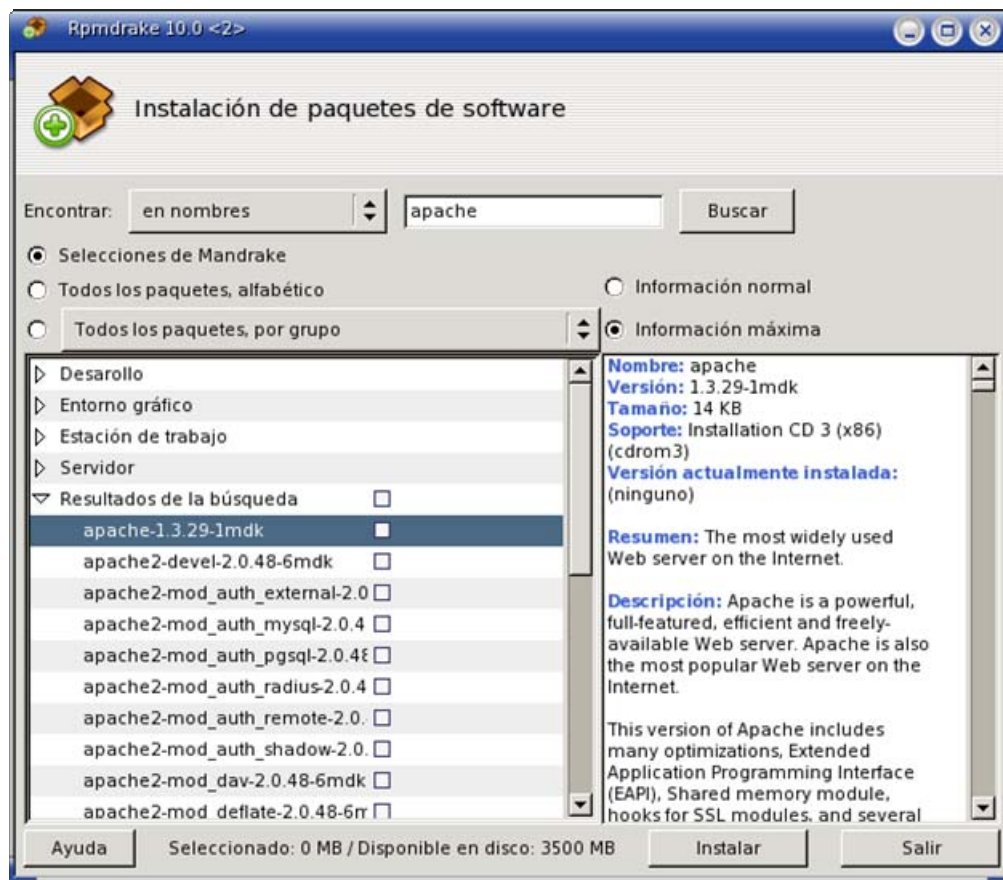
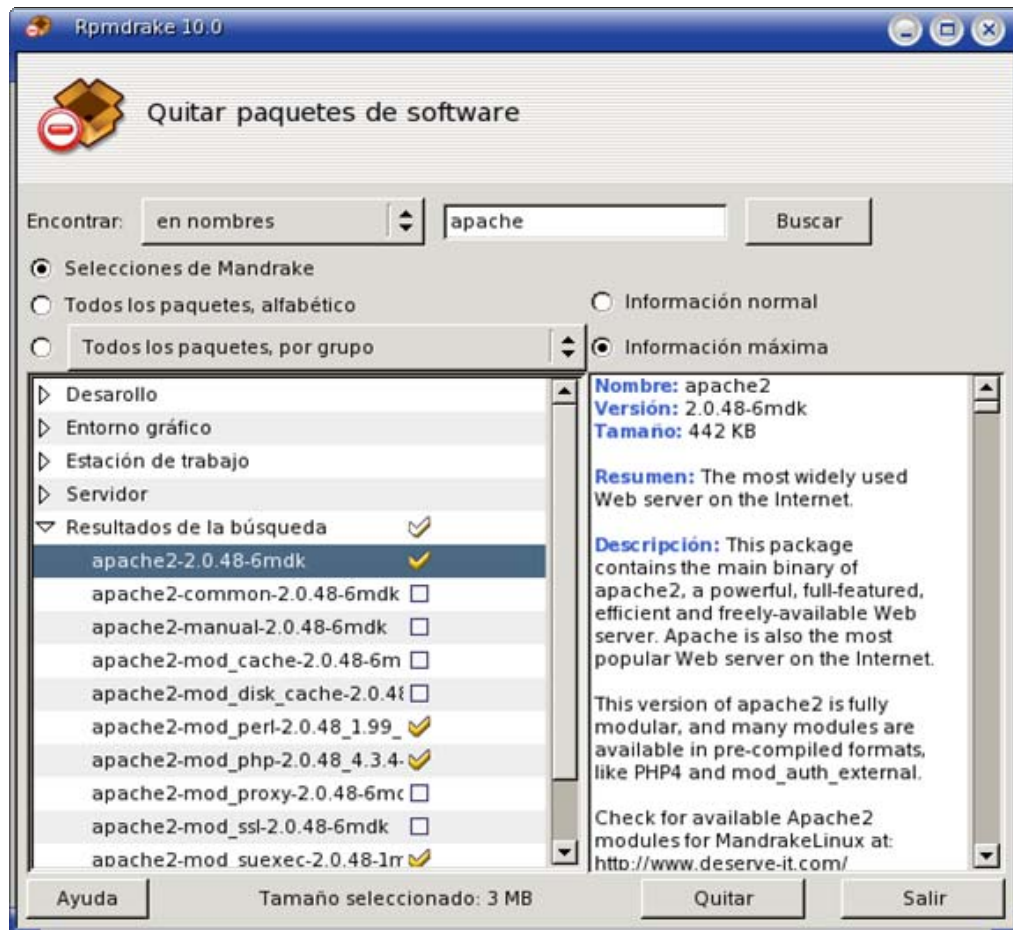


Figura2. 4: Búsqueda de Paquetes de Software

Vemos en la figura 2.4 que aparecen todos los paquetes con el nombre de la cadena de búsqueda (apache) y a la derecha de la ventana aparece una descripción del paquete seleccionado. Para instalarlo solo necesitamos chequear la casilla de verificación a la derecha del nombre del paquete y presionar el botón **Instalar**.

Para poder desinstalar programas seleccionamos la subcategoría *Desinstalar Programas*, y al igual que para instalar software hay que introducir la contraseña de *root* para ganar privilegios para realizar esta tarea.



**Figura2. 5:** Desinstalación de Programas

Observamos que cuando ingresamos el nombre de un paquete aparecen todos los paquetes con nombre similar y que si seleccionamos un paquete automáticamente se seleccionan los paquetes dependientes de éste, que como se mencionó anteriormente es de gran utilidad que el administrador de software se haga cargo de las dependencias entre paquetes.

## B. Servidor Web Apache

### a. Descripción

Apache es el servidor número uno en el mundo [14] por encima de competidores comerciales como *Microsoft* y *Netscape*, algunas de las grandes ventajas que presenta *Apache* son [13]:

- **Servidor Web altamente configurable y diseño modular**  
Es bastante fácil extender las capacidades del servidor Web *Apache*, por lo que cualquier persona con conocimientos de programación en *C* o *Perl* pueden escribir módulos para soportar alguna aplicación especial. Diseño Modular se refiere precisamente al hecho de agregar los módulos necesarios para soportar funciones especiales como *scripts* escritos en *PHP*, bases de datos *MySQL*.
- ***Apache* es una tecnología gratuita y fuente abierta:** Lo importante aquí no es el hecho de que sea gratuito sino más bien que sea software libre lo que ha provocado el desarrollo que ha tenido *Apache* y la gran cantidad de módulos escritos para soportar nuevas aplicaciones.
- ***Apache* funciona muy bien con *Perl*, *PHP* y otros lenguajes para la realización de *Scripts*:** La mayoría de las aplicaciones Web son *scripts* y *Apache* hace muy fácil trabajar con *scripts* escritos en *Perl* con su soporte para *CGI* y *mod\_perl*.
- ***Apache* corre en una gran cantidad de plataformas pero principalmente en *GNU/Linux*:** El desarrollo de *Apache* va de la mano con la introducción de *Linux* al mundo de los servidores empresariales. Además corre en otras plataformas tipo *UNIX*, como *Solaris*, *FreeBSD* y *Mac OS X*. Sin mencionar que puede instalarse sin ningún problema en servidores *Windows*.

Algunas de las características que presenta *Apache*:

- Soporte para la última versión del Protocolo HTTP 1.1
- Configuración simple basada en archivos de texto
- Soporte para CGI
- Soporte para *hosts Virtuales*
- Soporte para Autenticación
- Perl integrado para desarrollo de scripts *CGI*
- Soporte para scripts escritos en *PHP*
- Soporte para *Servlets* de *JAVA*
- Servidor *Proxy* integrado
- Registros de estado del Servidor (*Server Logs*)
- Soporte para SSL (*Secure Socket Layer, Capa de Socket Seguro*)

## **b. Instalación**

Como se menciona en la sección anterior existen varias formas de instalar software en *Linux*, y para el desarrollo de este proyecto utilizaremos los paquetes RPM incluidos en los discos de *Mandrake Linux* para la instalación del Servidor *Web Apache*, esto implica que todos los archivos y directorios serán colocados en las ubicaciones preseleccionadas por *Mandrake*.

## Requerimientos del Sistema para Instalar Apache

Recurso	¿Requerido?	Requerimientos
Espacio de Disco	Obligatorio	Se necesitan al menos 12MB para compilar e instalar <i>Apache</i> , pero ha medida que incremente los módulos adicionales que no son parte de la distribución estándar los requerimientos de espacio incrementarán.  Además debe tomar en cuenta el espacio de las páginas Web que desee albergar en su servidor. Por estas razones se recomienda dejar al menos 50MB para el servidor Web
Compilador C ANSI	Obligatorio	Para la mayoría de sistemas es recomendable el compilador de la GNU (gcc). La versión mínima para instalar Apache 2.0 es la 2.7.2. La mayoría de sistemas Linux como Mandrake ya lo incluyen
Interprete Perl	Recomendado	No es necesario tener instalado el intérprete de Perl para compilar <i>Apache</i> . Sin embargo algunos de los scripts de soporte como <code>log_server_status</code> son scripts Perl.

Tabla2. 1: Requerimientos para instalar Apache

### Requerimientos para poner en marcha el Servidor Web Apache <sup>[13]</sup>

Apache no requiere recursos masivos para operar. Corre perfectamente en sistemas Linux con 5 a 12MB de espacio libre en disco duro y 8MB de RAM. Sin embargo, el simple hecho de correr *Apache* no es lo que se tiene en mente. Es más probable que se desee correr *Apache* y servir varias páginas de contenido Web, lanzar procesos CGI y tomar ventaja de todas las aplicaciones maravillosas que ofrece la Web. Así que lo más recomendable es tener un sistema con suficientes recursos para que Apache pueda soportar la carga de servir páginas Web a diferentes clientes simultáneamente. Además se recomienda que antes de ponerlo en un ambiente de producción se realicen exhaustivas pruebas de

rendimiento para verificar que el servidor opera normalmente aún con carga pesada.

### **Ubicación de archivos y Directorios de *Apache* en *Mandrake Linux***

Los archivos y directorios están colocados de esta forma [15]:

- Los programas contenidos en el directorio `bin/` de *Apache* y que son para tareas administrativas se colocan en el directorio `/usr/sbin`.
- La documentación será colocada por completo en el directorio `/usr/share/doc/apache2-manual-version/`.
- El directorio `lib/` se colocará en el directorio `/usr/lib/apache2/`.
- Las librerías de los módulos adicionales se instalarán en `/usr/lib/apache2-extramodules/`.
- Los archivos de configuración contenidos en `conf/` se colocarán en el directorio `/etc/httpd/conf`.
- Los directorios `logs/`, `icons/` y `cgi-bin/` serán instalados en el directorio `/var/www/html/`.
- El directorio raíz de *Apache*, o sea donde deben colocarse los documentos Web estará en `/var/www/html/`.

Observamos que la ubicación de los archivos de *Apache* dentro del sistema de archivos de *Mandrake Linux* está acorde con lo que exige la normativa FHS.

### **c. Módulos del Servidor Web *Apache*** [15]

*Apache* cuenta con una gran cantidad de módulos por lo que solo se describirán los necesarios para la implementación del proyecto.

Los módulos importantes de *Apache* para la implementación de este proyecto son:



- **mod\_access:** Provee control de acceso basado en el nombre, en la dirección IP u otras características en las peticiones del cliente.
- **mod\_php:** Provee soporte para poder ejecutar *scripts* escritos en PHP.
- **mod\_auth:** Usado para la autenticación del cliente basado en archivos de texto
- **mod\_cgi:** Permite la ejecución de programas CGI.
- **mod\_so:** Carga módulos y código ejecutable al servidor al momento del arranque o de un reinicio.

## C. MySQL

### a. **Descripción** <sup>[16]</sup>

Un servidor de Bases de Datos es un programa que puede almacenar grandes cantidades de información en un formato organizado que es de fácil acceso para lenguajes de *scripts* como *PHP*. Una base de datos está compuesta de una o más tablas, relacionadas o no entre ellas.

MySQL es un servidor de Bases de Datos compatible con *SQL* (*Structured Query Language, Lenguaje Estructurado de Consultas*), que es un lenguaje estandarizado por *ANSI* e *ISO* y que es el más común para el acceso y manejo de bases de datos, y que además es software libre, disponible para diferentes plataformas incluyendo *GNU/Linux*.

Una base de Datos es una colección de datos que puede ser una sencilla lista de teléfono hasta una inmensa colección de datos en ambientes corporativos como los bancos o las compañías de servicio.

Para la implementación del proyecto deberá crearse una base de datos, la cual contendrá una tabla donde se guardarán los valores de temperatura leídos desde la ubicación remota. Esta base de datos debe estar disponible para poder realizar consultas desde la interfaz Web, para poder ya sea graficar el comportamiento de la temperatura o mostrar las coincidencias en una tabla según los parámetros de búsqueda seleccionados.

### b. **Creación de Bases de Datos** <sup>[17]</sup>

Para poder tener acceso a una base de datos primero hay que crearla y posteriormente incluir la o las tablas que deba contener. Pero lo primero que debe realizarse es conectarse a la base de datos. Para conectarnos a un servidor de base de datos lo hacemos a través de un programa cliente que se instala junto al servidor MySQL, cuando instalamos el paquete *RPM* contenido en los discos de *Mandrake Linux*.

Para acceder al cliente lo hacemos desde el Terminal de Comandos.

Para conectarnos al servidor, ingresamos la siguiente línea en:

```
mysql -h hostname -u username -p
```

donde *hostname* es el nombre o la dirección IP del cliente, *username* es el nombre del usuario de la base de datos y la opción `-p` es para decirle al programa que pida una contraseña para conectarse.

Como en nuestro caso el cliente se está ejecutando del mismo lado del servidor, entonces se puede omitir la opción `-h` o utilizar `-h localhost` o `-h 127.0.0.1`.

Cuando se instala el servidor se crea un usuario con privilegios de administrador llamado *root*, que no debe confundirse con el usuario *root* de *Linux*. Posteriormente deben crearse otros usuarios con privilegios específicos, lo cual se describirá posteriormente.

Inicialmente ninguna contraseña ha sido establecida, por lo que podemos obviar esta opción.

Entonces si ingresamos en el Terminal:

```
$mysql -u root
```

Obtendremos un *prompt* como el siguiente:

```
mysql>
```

esto indica que ya estamos conectados al servidor.

Para cerrar la conexión con el servidor ingresamos el comando `exit` o `quit`.

Esta es la línea de comandos desde la que enviaremos los comandos al servidor MySQL.

### **Ver Bases de Datos existentes** <sup>[17]</sup>

Para poder ver una lista de las bases de datos existentes utilizamos el comando `SHOW DATABASES`, como se muestra a continuación:

```
mysql>SHOW DATABASES;
```

### **Borrar una Base de Datos** [17]

Para borrar una base de datos utilizamos el comando `DROP`. Así que para borrar la base de datos `test`:

```
mysql>DROP DATABASE test;
```

### **Crear una Base de Datos** [17]

Para crear una base de datos se utiliza la siguiente sintaxis:

```
mysql>CREATE DATABASE nombre_db;
```

Donde nombre `nombre_db`, es el nombre de la base de datos que se desea crear.

Para indicarle al servidor `MySQL` que queremos usar una base de datos utilizamos el comando `use`.

```
mysql>USE nombre_db;
```

### **Crear una Tabla** [17]

El crear una base de datos resulta bastante sencillo pero como las tablas son tan flexibles la sintaxis para crearlas es un poco más complejo.

La siguiente es la sintaxis básica para la creación de tablas:

```
mysql>CREATE TABLE table_name (  
    - >nombre_columna1 tipo_columna1 detalles_columna1,  
    - >nombre_columna2 tipo_columna2 detalles_columna2,  
    - >...  
    - >);
```

Para visualizar las tablas de una base de datos utilizamos el comando

```
SHOW TABLES;
```

```
mysql>SHOW TABLES;
```

```
+-----+
| Tables in jokes |
+-----+
| cumpleaños      |
+-----+
1 row in set
```

Para borrar una tabla de una base de datos utilizamos de nuevo el comando DROP.

```
mysql> DROP nombre_tabla;
```

### **Insertar Datos en una Base de Datos** <sup>[17]</sup>

Como hemos visto hasta ahora, ya hemos discutido como crear una base de datos, como crear tablas dentro de una base de datos, pero estas tablas están vacías cuando las creamos por lo que tenemos que insertar datos para que sean útiles.

Existen dos formas básicas de insertar datos en una tabla, la primera forma y la más común es:

```
mysql>INSERT INTO nombre_tabla SET
    ->Nombre_Columna1 = "valor1",
    ->Nombre_Columna2 = "valor2",
    ->...
    ->;
```

En la segunda forma primero se lista el nombre de cada columna y posteriormente los valores correspondientes.

```
mysql>INSERT INTO nombre_tabla
    ->(Nombre_Columna1, Nombre_Columna2,...)
    ->VALUES (valor1,valor2,...);
```

Se puede utilizar cualquiera de las dos formas indistintamente.

### **Ver Datos Almacenados**

Una de las misiones de la bases de datos es poder almacenar grandes cantidades de información, pero no tendría sentido si esta información no estuviera disponible para posteriores consultas.

El comando utilizado para ver el contenido de las tablas es `SELECT`, éste puede ser utilizado de varias formas para mostrar solo el contenido deseado. La forma básica y que muestra todo el contenido de una tabla es:

```
mysql>SELECT * FROM nombre_tabla;
```

El “\*” significa todo.

Podemos hacer varias combinaciones pero el comando `SELECT` por si solo no brinda mucha flexibilidad para realizar búsquedas más detalladas. Para darle esta flexibilidad se acompaña por la cláusula `WHERE` (DONDE) que permite condicionar las búsquedas.

Para construir se utilizan comúnmente los operadores mayor que (>), menor que (<), mayor o igual que (>=), menor o igual que (<=), igual que (=). Además ciertos operadores como `AND`, `OR`, `LIKE`.

`LIKE` se utiliza para buscar cadenas de caracteres en una tabla y si utilizamos `%cadena%` no importa lo que tenga antes o después la cadena buscada.

### c. **Administración de MySQL** <sup>[17]</sup>

Como se ha mencionado anteriormente una vez que ha sido instalado el servidor MySQL se crea un usuario llamado `root`, que posee privilegios administrativos. Inicialmente este usuario conecta al servidor con únicamente ingresar `mysql -u root`, esto lo podría hacer cualquier persona que tenga acceso al servidor aún cuando no tenga la debida

autorización. Es por esta razón que lo primero que debe realizarse es establecer una contraseña para *root*, lo cual se hace de la siguiente manera:

- Conectarse al servidor `$mysql -u root`
- Establecer contraseña para *root*,

```
mysql>SET PASSWORD root@localhost=password("contraseña_deseada");
```

De esta forma cada vez que se desee ingresar al servidor debe utilizarse la opción `-p` y el servidor pedirá una contraseña para poderse conectar.

## D. PHP

### a. Descripción

*PHP* es un lenguaje de *scripts* (guiones) ejecutado en el servidor (es por esta razón que se le conoce como un lenguaje Server-Side) [18], generalmente el código escrito en este lenguaje es embedido en un contexto de HTML. Al contrario de una página HTML ordinaria, un *script* PHP no es enviado al cliente por el servidor, en lugar de eso es analizado por un binario de PHP (*parser*) o un módulo.

Los elementos HTML en un *script* PHP son dejados intactos, pero el código PHP es interpretado y ejecutado. El código PHP puede realizar consultas a bases de datos, lectura y escritura de archivos, generación de gráficos y muchas posibilidades más. La salida del código PHP es combinada con la sección HTML y enviada al cliente.

Las razones por las cuales es conveniente la utilización de PHP son variadas pero se pueden mencionar una mejor velocidad de ejecución comparada con otros lenguajes como JavaScript, el continuo desarrollo y un decremento en el tiempo de desarrollo, porque permite la separación del código PHP del HTML puro guiándonos hacia un desarrollo y diseño más flexible. Además PHP se puede ejecutar en múltiples plataformas y es soportado por muchos servidores Web, incluyendo *Apache*.

### b. Estructura del Lenguaje [19]

#### **Sintaxis y Comandos Básicos**

La sintaxis de PHP es bastante similar a la de *C*, *C++*, *Java*, *JavaScript*, *Perl* o cualquier otro lenguaje derivado de *C*. Un *script* de *PHP* consiste de varios comandos o declaraciones, cada uno de los cuales es una instrucción que el servidor Web debe seguir antes de poder proseguir con la siguiente.

Las declaraciones en *PHP* deben terminar siempre con punto y coma (;).

Esta es una declaración típica en PHP:



```
echo("Esta es una <b>Prueba</b>!");
```

Esta declaración invoca una función llamada `echo` y se le pasa una cadena de texto “Esta es una **Prueba**” como argumento. La función simplemente toma la cadena de texto y la coloca dentro del código HTML de la página Web.

Para poder insertar el código de PHP dentro del código HTML se debe indicar el inicio del código con `<?php` y el final con `?>`.

Por ejemplo consideremos el siguiente código:

```
<html>
<head>
<title> Ejemplo PHP </title>
</head>
<body>
<p>
<?php      //inicio del código PHP
echo ("Este es un ejemplo");
?>        //Fin del código PHP
</p>
</body>
</html>
```

Podemos observar que el código *PHP* se inserta en un contexto HTML como se mencionó antes. Para que este script funcione primero debe escribirse en un procesador de texto como *Vi*, *emacs*, *kate* en Linux o incluso el *Bloc de Notas* de Windows, debe guardarse con la extensión `.php` y colocarse en el directorio raíz para documentos del servidor Web. Además todo lo que aparece posterior a `//` es tomado como un comentario del autor del *script* que es de gran utilidad para el análisis de código y una buena fuente de ayuda para terceros en la comprensión o detección de errores en código escrito por otras personas.

## Variables y Operadores

Una variable es localidad en la cual se puede almacenar un valor de cualquier tipo. La diferencia de PHP con otros lenguajes es que no es necesario declarar una variable para que contenga un tipo específico de datos. Una variable en PHP puede contener datos binarios, flotantes, enteros, cadenas de texto o cualquier otro tipo de datos.

Para crear una variable seleccionamos un nombre y le antecedemos el signo de dólar (\$) para indicarle al intérprete del código PHP que en efecto es una variable.

```
$num=3;
```

La línea anterior asigna un valor de 3 a la variable \$num. No debe olvidarse terminar la línea con (;).

Este valor asignado puede ser cambiado a placer durante el desarrollo del script. Por ejemplo:

```
$num = 3;  
$num = "Tres";
```

En el ejemplo anterior \$num pasa de contener un entero a contener una cadena de texto.

El signo = es llamado operador de asignación, obviamente se utiliza para asignar valores a variables y no para realizar comparaciones.

Además se pueden utilizar operadores aritméticos para realizar alguna operación y luego guardarlos en una variable.

```
$prueba = 1+1; //asigna un valor de 2  
$prueba = 1-1; //asigna un valor de 0  
$prueba = 2*2; //asigna un valor de 4  
$prueba = 2/2; //asigna un valor de 1
```

Existe otro operador muy útil conocido como operador de concatenación (.) y que se utiliza para concatenar cadenas de texto.

```
$prueba = "Hola"."Mundo";  
// Asigna "Hola Mundo" a $prueba
```

En la tabla 2.2 vemos los operadores de comparación que podemos utilizar cuando trabajemos con PHP.

Operador	Nombre	Verdadero sí:	Ejemplo
==	Igual que	Izquierda es igual que la Derecha	<code>\$x==5</code>
!=	Diferente	Izquierda no es igual a Derecha	<code>\$x!=5</code>
===	Equivalente	Izquierda y Derecha son iguales y son del mismo tipo	<code>\$x===5</code>
>	Mayor que	Izquierda mayor que Derecha	<code>\$x&gt;5</code>
>=	Mayor o igual que	Izquierda mayor o igual que Derecha	<code>\$&gt;=5</code>
<	Menor que	Izquierda menor que Derecha	<code>\$x&lt;5</code>
<=	Menor o igual que	Izquierda menor o igual que Derecha	<code>\$x&lt;=5</code>

**Tabla2. 2** Operadores de Comparación

Los tipos de datos con los que podemos trabajar los vemos en la tabla 2.3.

Tipo	Ejemplo
Entero (Integer)	5
Doble (Double)	3.213
Cadena (String)	"hola"
Booleano (Boolean)	true
Arreglo (Array)	<code>\$arr= array(1,2,3)</code>

**Tabla2. 3:** Tipos de Datos manejados por PHP

### Operadores Lógicos

Por último podemos utilizar también los operadores lógicos ya conocidos como `and(&&)`, `or(||)`, `xor` y `not(!)`. Podemos utilizar indistintamente el nombre o el símbolo correspondiente a cada operador lógico.

## Incrementando y Decrementando una variable Entera

Cuando se escriben Scripts en PHP es muy común incrementar o decrementar una variable en una unidad o quizás en más de una unidad. O talvez queramos incrementarla o decrementarla antes o después de realizar una comparación o cualquier otra operación.

Aquí se muestran algunas de las formas más útiles de realizarlo:

```
$x +=1;    //incrementa $x en una unidad
$x+=int;  //incrementa $x en int unidades
$x++;     //incrementa $x en una unidad
$x--;     //decrementa $x en una unidad
++$x     //incrementa antes de que se realice la
         //operación
--$x     //decrementa antes de que se realice la
         //operación
```

## Arreglos (Arrays)

Un arreglo es un tipo especial de variable que contiene múltiples valores. De una forma más clara un arreglo es un vector que contiene múltiples valores indexados que pueden ser datos de cualquier tipo.

La forma más sencilla de definir un arreglo es mediante la función `array`

```
$myarray= array ("uno", 2, "tres");
```

El código anterior crea un arreglo llamado `$myarray`, que contiene dos cadenas de texto y un entero.

Para poder acceder a los elementos del arreglo se hace mediante el uso de índices, comenzando desde cero. Si creamos un arreglo de `n` elementos entonces tendremos índices desde cero hasta `(n-1)`.

Para obtener los elementos de un arreglo lo hacemos de la siguiente forma:

```
array[índice]
```

Por ejemplo:

```
echo(myarray[0]);           //muestra "uno"
echo(myarray[1]);           //muestra 2
$var1 = myarray[2];         //asigna "tres" a $var1
```

Los índices son útiles además para asignar nuevos valores y para crear nuevos elementos.

```
myarray[1] = "dos";         //Asigna un nuevo valor
myarray[3]= "cuatro";      //crea el cuarto elemento y
                             //le asigna el valor de
                             //"cuatro"
```

Además para asignar un elemento al final de un arreglo utilizamos los corchetes vacíos ([ ]).

```
myarray[]= "quinto";       //agrega el valor "quinto" al
                             //final del arreglo.
```

## Estructuras de Control

Son estructuras que nos permiten cambiar el flujo del programa que de otra forma iría en una sola dirección. En la mayoría de programas se prueban ciertas condiciones y dependiendo del resultado el flujo del programa cambia. Son estas estructuras las que nos brinda flexibilidad en la mayoría de lenguajes de programación ya que nos permiten prever múltiples casos y alternativas posibles.

### Estructuras If

**If** evalúa una expresión entre paréntesis. Si la expresión evaluada retorna *verdadero* entonces un bloque de código es ejecutado de lo contrario el bloque es ignorado por completo.

```
if (expresión) {
//Código a ejecutar si la expresión evaluada devuelve
//verdadero
```

```
}
```

### **Uso de la cláusula `else` con `if`**

Cuando se trabaja con estructuras `if` normalmente se desea definir un bloque de código alternativo a ejecutar en caso que la expresión que acompañe a `if` evalúe *falso*. La sintaxis es la siguiente:

```
if (expresión)
{
//Código a ejecutar si la expresión evaluada devuelve
//verdadero
}
else
{
//Código a ejecutar en todos los demás casos
}
```

### **Estructura `if-elseif-else`**

Cuando se necesitan evaluar múltiples expresiones entonces se utiliza una estructura `if-elseif-else`.

Pueden utilizarse múltiples cláusulas `elseif` para evaluar múltiples casos posibles.

La sintaxis de esta estructura es la siguiente:

```
if (expresión)
{
//Código a ejecutar si la condición es verdadera
}
elseif (expresión 2)
{
```

```

//Código a ejecutar si la condición 2 es verdadera
}
elseif (expresión n-esima)
{
.
.
.
}
else
{
//Código a ejecutar en todos los demás casos
}

```

## **Lazos**

Hasta aquí se han presentado diferentes estructuras de control para cambiar el flujo de un programa y ejecutar diferentes bloques de código de acuerdo a decisiones presentes en el *script*. Pero también es posible que un *script* determine cuantas veces repetir un bloque de código. Las estructuras de lazo están diseñadas para permitir tareas repetitivas.

El lazo continúa ejecutándose hasta que se alcanza alguna condición o hasta que explícitamente se decide salir del lazo.

### **Lazos while**

Ejecuta código hasta que la evaluación de la expresión devuelva *falso*.

Su sintaxis es:

```

while (expresión)
{
//Código a ejecutar
}

```

Por ejemplo

```

$x=1;
while ($x<=10)

```

```
{  
Print "El valor es: $x <br>\n";  
$x++;  
}
```

El código anterior imprimirá en pantalla un mensaje con el valor de \$x hasta que \$x llegue al valor de 11.

### **Lazos do while**

Es similar a la estructura `while`, pero el código se ejecuta al menos una vez. En la estructura `while` si la expresión devuelve falso al ser evaluada entonces no se ejecuta el código ni una vez.

La sintaxis es:

```
do{  
//código a ejecutar  
}  
while (expresión); //expresión de prueba
```

### **Lazos for**

Básicamente puede realizarse lo mismo con una estructura `while` que con una estructura `for`. Pero en la estructura `while` la variable para probar la condición se inicializa fuera del lazo y se incrementa dentro del bloque de código dentro del lazo, en un lazo `for` esto puede realizarse en una sola línea.

La sintaxis de un lazo `for` es la siguiente:

```
for(Variable; expresión de prueba; incremento)  
{  
//código a ejecutar  
}
```



## **break**

`break` es utilizado para salir de un lazo. Es una buena salvaguarda para el código escrito en *PHP* y permite salir de un lazo de acuerdo a pruebas adicionales cuando se dan condiciones extraordinarias como por ejemplo una división entre cero ó por ejemplo multiplicar dos variables en la cual una de ellas es un número y la otra una cadena de texto.

## **Anidamiento de lazos**

Como en la mayoría de lenguajes de programación *PHP* permite el anidamiento de un lazo dentro de otro. Es muy útil en la creación de tablas HTML dinámicas.

Por ejemplo si queremos enviar al navegador del cliente una tabla de multiplicación:

```
<html>
<head>
<title>Tabla de Multiplicación</title>
</head>
<body>
<?php
print "<table border=1>\n";
for ( $y=1; $y<=12; $y++ )
{
print "<tr>\n";
    for ( $x=1; $x<=12; $x++ )
    {
        print "\t<td>";
        print ($x*$y);
        print "</td>\n";
    }
    print "</tr>\n";
}
print "</table>";
```

```
?>
</body>
</html>
```

El código anterior generaría una tabla como está:

1	2	3	4	5	6	7	8	9	10	11	12
2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	27	30	33	36
4	8	12	16	20	24	28	32	36	40	44	48
5	10	15	20	25	30	35	40	45	50	55	60
6	12	18	24	30	36	42	48	54	60	66	72
7	14	21	28	35	42	49	56	63	70	77	84
8	16	24	32	40	48	56	64	72	80	88	96
9	18	27	36	45	54	63	72	81	90	99	108
10	20	30	40	50	60	70	80	90	100	110	120
11	22	33	44	55	66	77	88	99	110	121	132
12	24	36	48	60	72	84	96	108	120	132	144

**Tabla2. 4:** Tabla Dinámica generada con PHP y anidamiento de lazos

### Formularios e Interacción con el usuario

Para muchas aplicaciones de PHP la habilidad de interactuar con el usuario que ve la página Web es esencial. Otros lenguajes como *JavaScript* actúan en términos de *Manejadores de Eventos (Event Handlers)*, los cuales reaccionan directamente a acciones del usuario, por ejemplo el movimiento del puntero del ratón sobre un vínculo en la página. Los lenguajes para scripts del lado del servidor son mucho más limitados en cuanto a interacción con el usuario. Ya que el código de PHP se activa cuando una página es solicitada, la única interacción es la del cliente solicitando información al servidor, y el servidor responde enviando páginas generadas dinámicamente.

La clave para crear interacción con PHP es entender las técnicas que se pueden utilizar para enviar información sobre la interacción del usuario junto con su solicitud por una nueva página Web.

El método más simple que podemos usar para enviar información junto con la solicitud de ver una página es utilizar una *cadena de solicitud URL*. Esta técnica se observa cuando en una URL se encuentra un signo de interrogación (?) después del nombre de un archivo.

Por ejemplo si tenemos una página HTML e insertamos un enlace como este:

```
<a href= "bienvenida.php?nombre=Eduardo">Hola, soy Eduardo!</a>
```

Este es un enlace a un archivo llamado *bienvenida.php*, pero además de enlazar al archivo, enviamos una variable junto con la solicitud de ver la página, esta variable es la porción de la URL que sigue al signo de interrogación y se llama `nombre` y su valor es `Eduardo`.

Resumiendo, hemos creado un enlace que carga la página *bienvenida.php*, e informa al código PHP que `nombre` es igual a `Eduardo`.

Para entender realmente el resultado de este proceso, necesitamos ver el código contenido en el archivo *bienvenida.php*. Supongamos que el contenido es el siguiente:

```
<?php
$nombre=$_GET['nombre'];
Print "Bienvenido a nuestro sitio Web, $nombre!";
?>
```

Ahora, si utilizáramos el enlace en la página HTML para cargar esta segunda página, veríamos que la página dice "Bienvenido a nuestro sitio Web, Eduardo!" .

PHP automáticamente crea un *arreglo asociativo* llamado `$_GET` que contiene cualquier valor pasado a través de la cadena de consulta. Como vimos anteriormente podemos encontrar el valor de la variable nombre en el arreglo asociativo de esta forma, `$_GET['nombre']`. El *script* asigna el valor a una variable ordinaria llamada `$nombre` y es mostrada en pantalla por medio de la función `Print`.

Pero se puede enviar más que una sola variable en la cadena de consulta. Por ejemplo si tuviéramos el siguiente código en la página HTML:

```
<a href= "bienvenido.php?nombre=Eduardo&apellido=Castro">Hola</a>
```

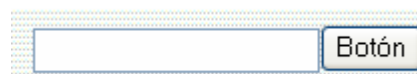
Podemos tener acceso a ambos valores de la misma forma que en el primer ejemplo:

```
<?php
$nombre=$_GET['nombre'];
$apellido=$_GET['apellido'];
Print "Bienvenido a nuestro sitio Web, $nombre
$apellido";
?>
```

Este *script* mostraría el mensaje incluyendo el nombre y apellido.

Hasta ahora hemos visto como enviar información desde una página hacia un *script* PHP, pero todavía no hay interacción con el usuario ya que enviamos siempre los mismos valores. La forma de interactuar con el usuario es por medio de formularios HTML.

Podemos escribir el valor de las variables que deseamos enviar en cajas de texto y remitirlas hacia un servidor por medio de un botón.



**Figura2. 6:** Formulario HTML

Veamos como crear un formulario HTML sencillo donde el usuario pueda ingresar su nombre y apellido:

```
<form action= "bienvenida.php" method= "GET">
Nombre: <input type= "text" name= "nombre"><br>
Apellido: <input type= "text" name= "apellido"><br>
<input type= "submit" value= "Enviar">
</form>
```

El código anterior permitiría a los diferentes usuarios ingresar su nombre y apellido, luego enviarlos por ejemplo hacia una base de datos con el botón Enviar.

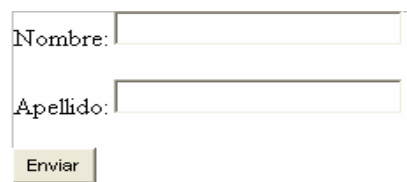
No es la intención adentrar en el lenguaje HTML pero explicaremos lo que hace el código anterior.

Primero, al pulsar el botón Enviar la información que hemos ingresado se enviará al archivo *bienvenida.php*.

El método para enviar la información será `GET` o sea que podremos acceder a la información por medio del arreglo asociativo `$_GET`.

Existirán dos cajas de texto una para el nombre y otra para el apellido, cada una se identificará por el atributo `nombre`, o sea que para obtener el valor correspondiente a nombre en el *script* bienvenida deberíamos hacerlo de la siguiente manera: `$_GET['nombre']`.

El formulario se vería así:

The image shows a rendered HTML form. It consists of two text input fields stacked vertically. The first field is labeled 'Nombre:' and the second is labeled 'Apellido:'. Below these fields is a submit button with the text 'Enviar'.

**Figura2. 7:** Formulario con dos campos vacíos

## Elementos en un formulario HTML

Existen otros elementos HTML que pueden ser incluidos en un formulario, así como las cajas de texto. Entre estos tenemos:

## Listas Desplegables (SELECT)

Estos elementos son listados con varias opciones como por ejemplo cuando se abre una cuenta de correo electrónico y hay que seleccionar el país y se abre una lista con el listado de todos los países.

La forma básica de definirla es:

```
<SELECT name= "arreglo[]">
<option>Nombre Opción 1
<option>Nombre Opción 2
<option>Nombre Opción n
</select>
```

Donde *arreglo* es el nombre del arreglo asociativo de donde se extraerá la información y `<option>` acompaña a cada opción que deseamos esté disponible.

## Casillas de Verificación (Check Boxes)

Están casillas nos permiten seleccionar una opción con solo presionar con el ratón. No es apropiado para un listado muy largo.

```
<input type= "checkbox" name="arreglo[]" value="nombre">
```

Por ejemplo otra forma para el ejemplo anterior sería:

## Funciones Especiales de PHP

### **isset**

Esta función devuelve un valor de *verdadero* si se le ha asignado un valor a una variable. Esto es muy útil cuando se trabaja con datos provistos por el usuario para determinar que opciones ha seleccionado o si ha dejado campos en blanco. Su sintaxis es:

```
isset(variable)
```

## **foreach**

Esta función es muy útil para extraer información de un arreglo. Su sintaxis es bastante sencilla:

```
foreach($arreglo as $var){  
    //código a ejecutar  
}
```

`foreach` crea un lazo y extraer cada elemento de arreglo (`$arreglo`) y lo guarda en una variable temporal (`$var`). El lazo se termina hasta que se han extraído todos los elementos del arreglo.

## **Die**

Esta función se utiliza para terminar la ejecución de un programa cuando se ha encontrado alguna condición de error y además envía un mensaje al navegador advirtiéndolo sobre esto.

```
Die(mensaje);
```

### **c. Integración de PHP con MySQL <sup>[19]</sup>**

Una de las principales características de PHP es su habilidad de trabajar con múltiples bases de datos, incluyendo *MySQL*. Trae integrada varias funciones para poder interactuar de forma satisfactoria con un servidor de bases de datos. Veremos las funciones más importantes para integrar nuestra página Web a la base de datos.

#### **Conectándose a un servidor de Bases de Datos**

PHP incluye la función `mysql_connect()` para conectarse a una base de datos. Requiere tres argumentos: un nombre de host, un nombre de usuario y una contraseña.

Si la conexión es exitosa, el servidor enviará un identificador de conexión que puede guardarse y utilizarlo para seguirse conectando al servidor.

En el siguiente código se muestra la manera adecuada de conectarse a un servidor *MySQL*:

```
$user=root;
$password = "udbadmin";
$link = mysql_connect("localhost",$user,$password);
if(!$link)
{
die ("No se pudo conectar al servidor");
}
```

Observamos que el identificador de conexión devuelto por el servidor se almacena en la variable `$link`, si no se puede establecer la conexión entonces se termina la ejecución del programa y se envía el mensaje "No se pudo conectar al servidor". Nótese la utilización de `!` como signo de negación dándole significado de "no existe" a `$link`.

### **Seleccionando una Base de Datos**

Ahora que se ha establecido la conexión al servidor, necesitamos indicarle con cual de las bases de datos trabajaremos. Esto se logra mediante la función `mysql_select_db()`, ésta acepta dos argumentos: el nombre de la base de datos y opcionalmente el identificador de conexión. Si no se incluye el identificador de conexión se utilizará cualquier conexión abierta.

Para conectarnos a la base de datos llamada muestra, lo haríamos así:

```
$db = "muestra";
mysql_select_db($db)
or die("No se pudo abrir $db".mysql_error());
```

Observamos que hemos utilizado el operador de concatenación (`.`), para agregar al mensaje los errores que puedan generarse por el servidor



*MySQL* en caso de fallar la conexión a la base de datos, estos mensajes de error se acceden mediante la función `mysql_error()`.

### **Ejecutando comandos y realizando consultas a la base de datos**

Hasta ahora hemos visto como conectarse a un servidor de bases de datos y además hemos visto como seleccionar una base de datos, pero lo más importante es poder realizar acciones sobre esas bases de datos.

Para poder ejecutar comandos y agregar, borrar o modificar datos en el servidor *MySQL*, así como para realizar consultas utilizamos la función `mysql_query()`. Esta función necesita dos argumentos: la consulta a realizar y el identificador de conexión.

Por ejemplo para ver todos los datos de la tabla `muestra` haríamos lo siguiente:

```
$consulta = "Select *.* from muestra";  
mysql_query($consulta,$link)  
or Die("no se pudo realizar consulta".mysql_error());
```

La consulta puede ser cualquier acción de las vistas en la sección referente a *MySQL*.

En el código anterior hemos desarrollado un *script* capaz de conectarse al servidor, capaz de seleccionar una base de datos y capaz de ejecutar una consulta. Una vez realizada la consulta es conveniente aunque no obligatorio cerrar la conexión, lo que se hace con la función `mysql_close()` la que necesita únicamente el identificador de conexión como vemos al final del *script*.

### **Determinando el número de Filas encontradas en un Consulta**

Podemos determinar el número de filas que devuelve una consulta cuando utilizamos `SELECT`. La función para realizarlo es `mysql_num_rows()`.

Por ejemplo

```
$consulta = "Select *.* from muestra";  
$result = mysql_query($consulta,$link);  
$num_filas = mysql_num_rows($result);
```

### **Accediendo un Resultado**

Hasta ahora solo hemos determinado el número de filas o coincidencias de una consulta realizada a una base de datos. Lo importante es poder tener acceso a la información que nos brinda este resultado.

Se puede obtener fácilmente un arreglo de los campos contenidos en cada fila encontrada en el resultado con la función `mysql_fetch_row()`. La función requiere un identificador de resultado, retornando un arreglo que contiene cada campo en determinada fila. Cuando se alcanza la última fila y el último campo de esta, la función devuelve un *false*.

La función `mysql_fetch_row` es muy útil para poder visualizar el resultado de la consulta realizada, que es al final lo que nos interesa.

## **E. Selección del Lenguaje para generación de Gráficos**

### **a. Determinación de Alternativas**

Existen múltiples alternativas para la generación de gráficos dinámicos entre las cuales tenemos:

**Java:** Presenta una excelente alternativa para graficar en tiempo real pero se tiene la desventaja que si se tiene una conexión de red muy lenta el gráfico se congelará porque el código debe ser ejecutado por el cliente. Además se debe contar un cliente llamado *Java Runtime Enviroment* instalado en el sistema, por lo que si no se cuenta con él no se podrá visualizar la página. En el lado del servidor tendrá que agregarse el soporte para *Java* instalando el llamado *Apache Tomcat*, ya que *Apache* por si solo no es capaz de manejar *scripts* escritos en *Java*. Tendría que

instalarse el *Java Development Kit* (JDK) que no es parte de los paquetes incluidos con la versión de descarga de Linux Mandrake.

**Perl:** Perl es un excelente lenguaje de programación con una sintaxis muy similar a la de C++. Para poder generar los gráficos dinámicos es necesario instalar dos módulos de *Perl*: Chart y GD. El primer módulo es el encargado de generar las gráficas. Puede generar siete tipos diferentes de gráficas entre las que tenemos líneas, barras, de pastel, de puntos, Pareto, barras apiladas y línea punteada. El segundo permite la creación de gráficas en formato GIF y PNG, pero no directamente, la creación debe realizarse a través de módulos *Perl* adicionales, en este caso Chart. Ninguna de los dos está incluido en la distribución Linux Mandrake pero pueden obtenerse en el sitio Web de Perl <sup>[20]</sup>[21].

Una vez descargados ambos módulos su instalación es muy sencilla, con la única condición es que primero debe instalarse GD y posteriormente Chart, esto debe realizarse de esta forma porque al instalarse Chart se hace una verificación de dependencias, si GD no está instalado Chart no podrá instalarse tampoco.

El procedimiento para instalar GD:

```
gunzip GD-1.18.tar.gz
tar xvf GD-1.18.tar
perl Makefile.PL
make
make test
make install
```

El procedimiento para la instalación de Chart es el siguiente:

```
gunzip chart-0.94.tar.gz
tar xvf chart-0.94.tar
cd Chart-0.94
perl Makefile.PL
make
make test
make install
```

La documentación relativa a Chart puede consultarse ejecutando el comando `perldoc Chart`.

La técnica para poder generar gráficos dinámicos consiste en escribir un *script* haciendo uso de Chart que genera un gráfico y lo guarda como una imagen en formato GIF o PNG. Este *script* se programa como un trabajo *cron* y se ejecuta cada minuto por lo que una imagen diferente es generada cada minuto.

Desde una página HTML sencilla se realiza un enlace hasta esta imagen con una etiqueta `<img src=ruta a la imagen>` y además se le coloca un `<META HTTP-EQUIV="Refresh" CONTENT="60" >` para que se cargue una nueva imagen cada minuto.

La desventaja de esto es que para monitoreo de una variable un minuto es demasiado tiempo y no pueden programarse trabajos *cron* en intervalos de tiempo menores a un minuto.

A continuación se muestra un ejemplo de esta técnica:

```
#!/usr/bin/perl -w

# Descripción: Un script Perl para generar una imagen GIF en
# intervalos de 5min. Las gráficas son seno y coseno con amplitud
# y frecuencia aleatoria. Los valores son colocados en un arreglo
# y la gráfica es generada. Las gráficas pueden ser visualizadas
# con xv o el código HTML.

use strict;
use Chart::Lines;

my($i, $graph, @sinarray, @cosarray, @radarray,
   $offset, $sinamp, $cosamp, $sinfreq, $cosfreq);

# Calcular las variables aleatorias
$offset = 5;
$sinamp = rand(5);
$cosamp = rand(5);
$sinfreq = rand(5);
$cosfreq = rand(5);

# Crear los arreglos que contendrán los datos para la gráfica
for ($i = 0; $i < 10; $i += 0.1) {
    push @sinarray, $offset + $sinamp * sin($sinfreq * $i);
    push @cosarray, $offset + $cosamp * cos($cosfreq * $i);
    push @radarray, int($i);
}

$graph = Chart::Lines->new(600,250);# declare object
```

```

# Pasar datos a la gráfica
$graph->add_dataset (@radarray);
$graph->add_dataset (@sinarray);
$graph->add_dataset (@cosarray);

# Colocar Leyendas
my($LastUpdate) = scalar localtime;
my(@XTicks) = qw(1 11 21 31 41 51 61 71 81 91);
my(@GraphLegends) = qw(Sine Cosine);
$graph->set ('title' => 'Graphing on the Fly (5 min.)');
$graph->set ('sub_title' => "Last Update $LastUpdate PST");
$graph->set ('x_label'=> 'Angle in Radians');
$graph->set ('y_label'=> 'Amplitude');
$graph->set ('legend_labels' => \@GraphLegends);
$graph->set ('stagger_x_labels' => 'false');
$graph->set ('custom_x_ticks' => \@XTicks);
$graph->set ('graph_border' => 1);
$graph->set ('grid_lines' => 'true');
$graph->gif ("/var/www/html/imagenes/Figure3.gif"); #Make graph

```

El código anterior genera un gráfico seno y otro coseno de amplitud y frecuencia aleatoria utilizando el módulo Chart, y lo guarda como Figura3.gif. El siguiente código HTML muestra la imagen generada y la refresca cada minuto. Este código debe guardarse en el directorio para documentos del servidor *Apache*.

```

<HTML>
<HEAD>
<TITLE>Gráfico Dinámico</TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="60" >
</HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000>
</a>
</BODY>
</HTML>

```

Para que la imagen se refresque cada minuto se ingresa en el Terminal de comandos `crontab -e`, y se agrega una línea como la siguiente:

```

0,5,10,15,20,25,30,35,40,45,50,55 * * * * *
/home/udb/scripts/codigo.pl /dev/null 2>&1

```

Es importante que se ingrese en una sola línea y si queremos que se refresque cada minuto entonces la primera secuencia deberá ser 0,1,2,3,...60.

Ahora cada vez que se ingresa en el navegador la dirección de la página HTML, se mostrará el último gráfico generado y se refrescara cada minuto.

**PHP:** La técnica para la generación de gráficos es muy similar ha la utilizada con *Perl*, pero existe una diferencia importante ya que en lugar de incluir la ruta hacia una imagen en la etiqueta *img*, se incluye el nombre de un *script*. Así que cada vez que se hace un refresco de la página, se ejecuta el *script* y se genera una nueva imagen por lo que ya no es necesario la programación de un trabajo *cron* que como mencionamos anteriormente tiene la limitante que no se puede programar para intervalos de tiempo menores al minuto. Así que con está técnica se puede generar una nueva gráfica en intervalos de tiempo desde un segundo en adelante. La modificación al código HTML sería:

```
<HTML>
<HEAD>
<TITLE>Gráfico Dinámico</TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="5" >
</HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000>

</BODY>
</HTML>
```

En el código anterior se realizará el refresco cada cinco segundos, la imagen que se generará será diferente cada vez.

Para la utilización de la funciones para la creación de imágenes de PHP solo es necesario instalar la librería GD incluida en los discos de Linux Mandrake.

### **Creación de Imágenes con PHP** <sup>[19]</sup>

Antes de poder trabajar con una imagen, primero debe adquirirse un identificador de imagen. Para esto se utiliza la función `imagecreate()`, la cual requiere dos argumentos, el primero es la altura de la imagen y el segundo para el ancho de la imagen. Esta función retorna un identificador de imagen que utilizaran las demás funciones para creación o modificación de imagen.

Después que se ha conseguido un identificador de imagen es posible enviar la imagen hacia el navegador Web, para esto es necesario utilizar la función `imagepng()`, la cual requiere como argumento un identificador de imagen.

Por ejemplo:

```
<?php
Header("Content-type:image/png");
$imagen=imagecreate(200,200);
Imagepng($imagen);
?>
```

Vemos que se envía un encabezado al navegador indicándole que se trata de una imagen de tipo PNG. Esto por si solo generará un cuadro negro de 200x200 píxeles.

### **Adquiriendo Color**

Para trabajar con colores es necesario adquirir un identificador de color. Esto se realiza mediante la función `imagecolorallocate()`, la cual requiere un identificador de imagen y tres enteros entre 0 y 255 representando el color rojo, verde y azul respectivamente. La función retorna un identificador de color que puede utilizarse para definir el color de las formas, rellenos y el texto.

Por ejemplo:

```
$red=imagecolorallocate($imagen,255,0,0);
$green=imagecolorallocate($imagen,0,255,0);
$blue=imagecolorallocate($imagen,0,0,255);
```

Por defecto la primera vez que se llama la función `imagecolorallocate()`, se define el color por defecto de la imagen, si agregáramos esta línea al código anterior crearíamos un cuadro rojo.

## Dibujando Líneas

Antes de dibujar una línea en una imagen es necesario determinar las coordenadas de origen y destino de los extremos de la línea.

El sistema de coordenadas utilizado es el siguiente:

- La unidad de medida es el píxel
- El origen es la esquina superior izquierda de la imagen (coordenada 0,0)
- Las coordenadas aumentan en el eje x de izquierda a derecha y en el eje Y de arriba hacia abajo.

La función utilizada es `imageline()`, la cual requiere un identificador de imagen, cuatro enteros representando las coordenadas de inicio y final, y un identificador de color.

Por ejemplo

```
imageline($imagen,0,0,0,199,$blue);
```

Si insertáramos esta línea en el *script* anterior se dibujara una línea de la esquina superior izquierda hasta la esquina inferior derecha.

## Rellenos de Imagen

Es posible rellenar un área con un color específico utilizando PHP, para ello utilizamos la función `imagefill()` que requiere un identificador de imagen, las coordenadas iniciales para aplicar el relleno y un identificador de color.

`imagefill()` cambia de color el píxel inicial definido por las coordenadas y todos los píxeles adyacentes del mismo color.

Por ejemplo:

```
imagefill($imagen,0,0,$green);
```

Esto cambiaría el color de la sección superior del cuadro rojo a verde y la inferior permanecería en rojo. La línea azul sirve como frontera y es lo que evita que todo el cuadro cambie de color.



## Dibujar Arcos y Circunferencias

Esto se realiza con la función `imagearc()`. Esta requiere de un identificador de imagen, las coordenadas del centro, un entero para el ancho, un entero para la altura, un punto de inicio y uno punto final (en grados), y un identificador de color. Los arcos se dibujan en el sentido de las agujas del reloj comenzando a las 3 en punto.

Por ejemplo:

```
imagearc($image, 99, 99, 200, 200, 0, 90, $blue);
```

La línea anterior dibuja un cuarto de círculo (desde 0 a 90 grados), centrado en (99,99).

## Dibujar rectángulos

Se utiliza la función `imagerectangle()`, la cual requiere un identificador de imagen, las coordenadas de la esquina superior izquierda, las coordenadas de la esquina inferior derecha y un identificador de color.

Ejemplo:

```
imagerectangle($imagen, 0, 0, 50, 50, $blue);
```

La línea anterior creará un rectángulo de líneas azules de 50x50 píxeles.

Después se le podría aplicar un `imagefill()` para rellenarlo por completo.

## Trabajando con Texto

Si se tienen instaladas las fuentes del tipo TrueType, pueden ser utilizadas para escribir texto en una imagen.

Puede escribirse texto con la función `imageTTFtext()`. Requiere de ocho argumentos: un identificador de imagen, un entero representando el tamaño de las letras, un ángulo de orientación, las coordenadas de origen, un identificador de color, la ruta hacia la fuente TrueType y el texto que se desea escribir.

Una forma de encontrar fuentes TrueType es mediante el comando

```
locate *.ttf.
```

Ejemplo:

```
$font="/usr/local/java/fonts/LucidaSansRegular.ttf";
```

```
imageTTFtext ($imagen, 50, 0, 20, 100, $blue, $font, "HOLA");
```

## **b. Selección de Lenguaje a Utilizar**

Además de las alternativas presentadas anteriormente existen otros lenguajes de programación tradicionales como el excelente pero complejo C, con el que se puede realizar prácticamente cualquier tarea, entre las cuales está generar código HTML pero resulta en programas muy complejos y largos.

Las tres alternativas presentadas son de excelente prestación para aplicaciones *Web*, pero en el caso de *Java*, se complicaría un poco más el desarrollo del proyecto por el software adicional que sería necesario instalar y más aún cuando no es parte del software incluido en Linux Mandrake. Por otra parte nos estaríamos saliendo un poco de los objetivos planteados y específicamente con el de no utilizar software con licencias comerciales y código cerrado.

*Perl*, por otra parte es una excelente alternativa que se integra perfectamente con el servidor *Web Apache*, y ha sido utilizado durante años por los desarrolladores para aplicaciones *Web* aunque está bastante lejos de la popularidad adquirida por *PHP*, *ASP* y *Java*.

A pesar que es necesaria la instalación de dos módulos adicionales para la generación de gráficos esto no constituye problema alguno porque su instalación es bastante sencilla. Otro punto muy importante a favor de *Perl* es que puede conectarse con el servidor *MySQL* de forma muy similar a la utilizada con *PHP*. Mencionaremos por último que *Perl* cuenta con extensa documentación y un gran número de foros de discusión.

Un punto en contra, es quizás que el módulo *Chart* genera una imagen y la guarda en disco y luego esta imagen es enlazada desde una página HTML que se refresca periódicamente pero como se explicó

anteriormente el refresco de la imagen se hace después de no menos de un minuto lo que significa mucho tiempo tratándose de monitoreo.

*PHP* por otro lado ofrece casi las mismas prestaciones que *Perl* para la generación del gráfico dinámico, pero con la diferencia que en lugar de enlazar a una imagen se enlaza a un *script*, pudiéndose de esta forma realizar el refresco del gráfico en intervalos de tiempo más pequeños que un minuto. Además como hemos visto a través del documento interactúa fácilmente con *MySQL* y funciona de forma extraordinaria con *Apache*.

En conclusión, se utilizará *PHP* para la generación del gráfico estadístico y la generación del gráfico dinámico, manteniendo de esta forma el proyecto lo más sencillo posible, sin incorporar otros lenguajes excepto *C* para la lectura del puerto serial.

## PARTE III

### a. Instalación y Configuración de Linux Mandrake <sup>[12]</sup>

Para poder instalar Linux Mandrake al igual que otras distribuciones existen dos formas hacerlo:

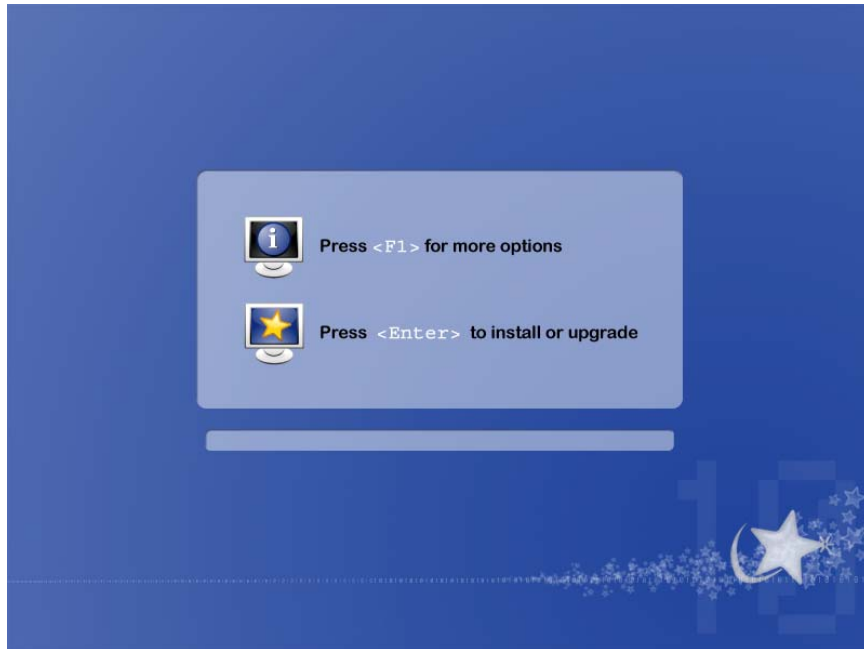
- Por medio de un asistente de instalación, la cual se realiza de forma gráfica asistida. En el caso de Linux Mandrake este asistente se llama *DrakX* que es un poderoso asistente en el que se pueden configurar de forma gráfica los aspectos más importantes.
- Por medio de una instalación en modo texto que es recomendada para expertos o en los casos que los recursos del sistema donde se esté instalando sean limitados, como cuando se intenta instalar en computadoras con tarjetas de video muy antiguas.

Nosotros utilizaremos la primera forma, utilizando *DrakX* por ser la más común y efectiva en la mayoría de los casos. En caso de que existiese algún problema sería conveniente remitirse a la documentación oficial de Mandrake Linux incluida en los discos de instalación.

*DrakX* brindará una instalación suave y sencilla ya sea que se trate de un usuario nuevo o un experto. Además antes de realizar la instalación es conveniente que todo el hardware como impresores, escáneres, dispositivos USB estén conectados para que sean detectados durante la instalación.

Para poder comenzar la instalación solo hay que configurar el **BIOS** como se indica en la segunda parte de este documento para que el primer dispositivo de arranque sea la unidad de CDROM y arrancar el sistema con el primer CD de Linux Mandrake.

Una vez que el sistema ha arrancado con el primer disco en la unidad de CDROM veremos una pantalla como la siguiente:



**Figura 3. 1** Pantalla de Instalación Inicial

Si se desea proseguir la instalación normal únicamente se debe presionar **Enter**, si se desea una instalación en modo texto o especificar otras opciones relativas al *kernel* de Linux, al hardware, la reparación de un sistema corrupto o la instalación/desinstalación del cargador de arranque se debe presionar **F1**, por nuestra parte no habrá más explicaciones referentes a esta opción.

Inmediatamente después de presionar **Enter** aparecerá una pantalla para la configuración del idioma y la región. Se observa además a la izquierda una lista de las tareas a realizar durante la instalación.

En nuestro caso se seleccionará **América Latina** como región y el español como idioma como se muestra en la figura 3.2.

La selección del idioma afectará el idioma del instalador, del software a instalar, de la documentación y del sistema en general.

Se puede instalar más de un idioma en el sistema presionando el botón **Advanced (Avanzada)** y seleccionando los idiomas deseados, esto instalará la documentación y las aplicaciones para todos los idiomas seleccionados.

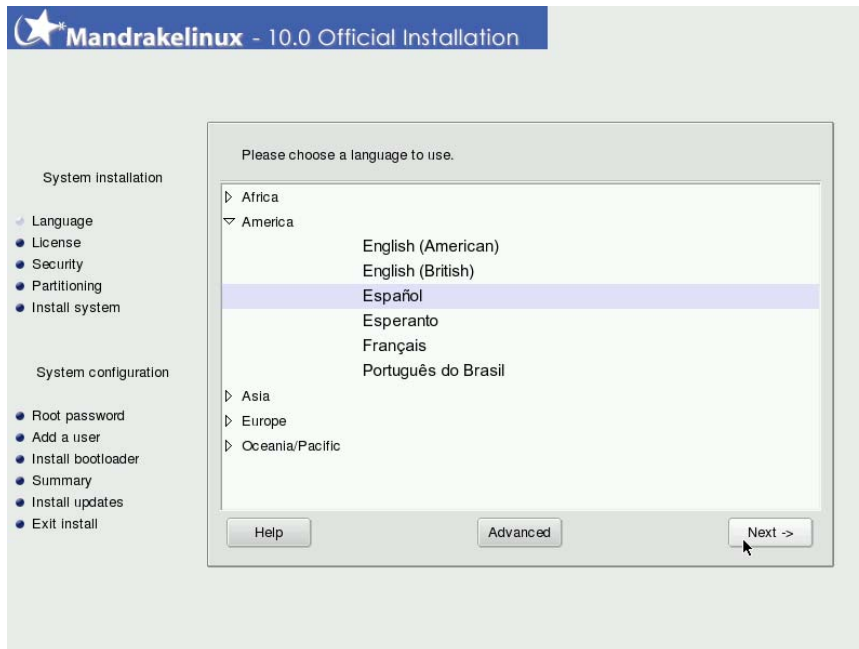


Figura 3. 2 Selección de Idioma

Posteriormente aparece una pantalla con los términos de la Licencia de Distribución en el idioma principal seleccionado, aunque la única versión válida está en idioma inglés. Si está de acuerdo presionar el botón **aceptar**.

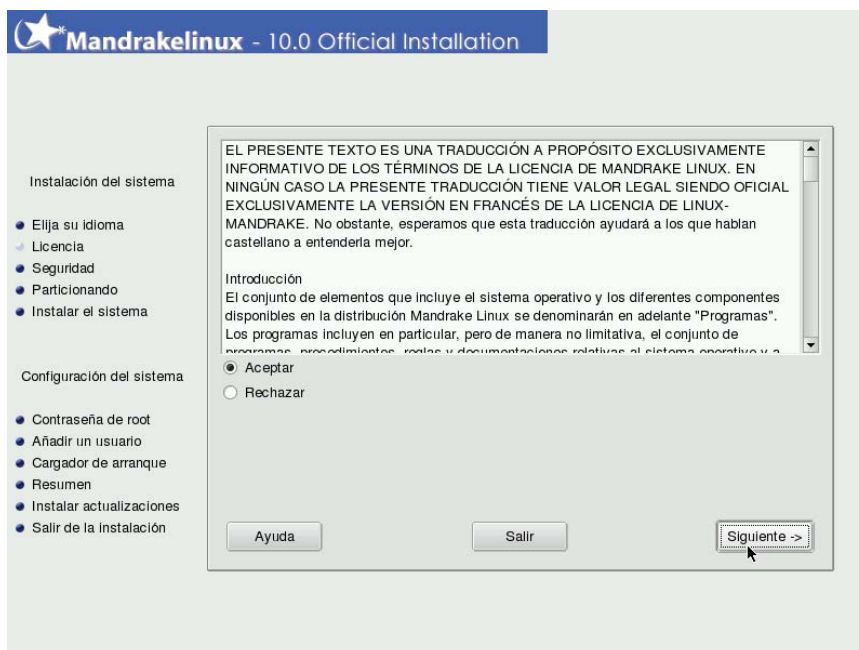


Figura 3. 3 Licencia de Distribución

A continuación aparece la ventana de configuración del ratón (figura 3.4). Por lo general, DrakX no tiene problemas en detectar la cantidad de botones de su ratón. Si no es así, asume que se tiene un ratón de dos botones y lo configurará para que emule el tercero. El tercer botón del ratón en un ratón de dos botones se puede “obtener” haciendo clic simultáneamente sobre el botón izquierdo y el derecho. *DrakX* sabrá automáticamente si su ratón utiliza una interfaz PS/2, serie o USB.

Si se elige un ratón distinto al predeterminado, se mostrará una pantalla de prueba, hay que usar los botones y la rueda para verificar que los ajustes son correctos y que el ratón está funcionando correctamente. Si el ratón no está funcionando bien, presionar la **barra espaciadora** o la tecla **Enter** para cancelar la prueba y volver a la lista de opciones.

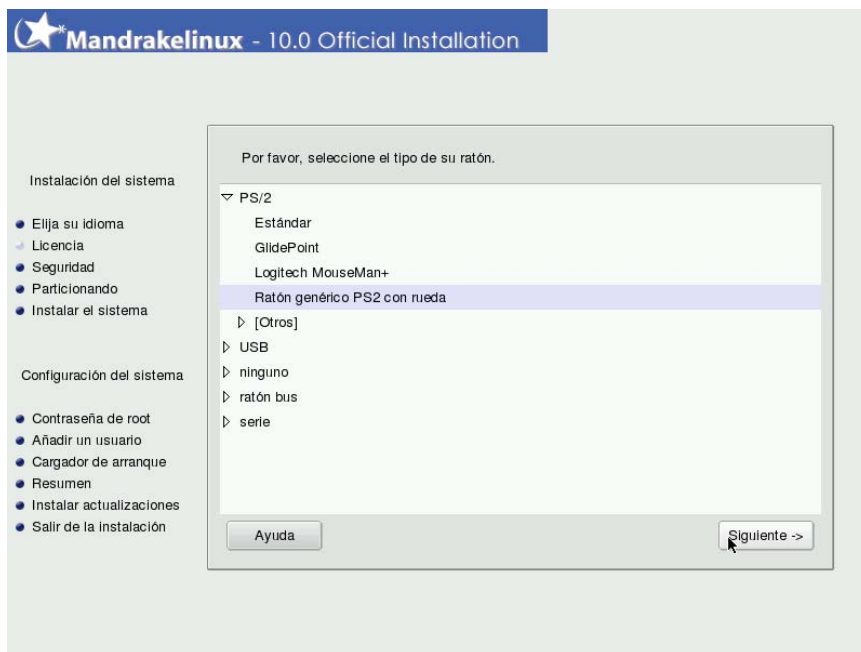
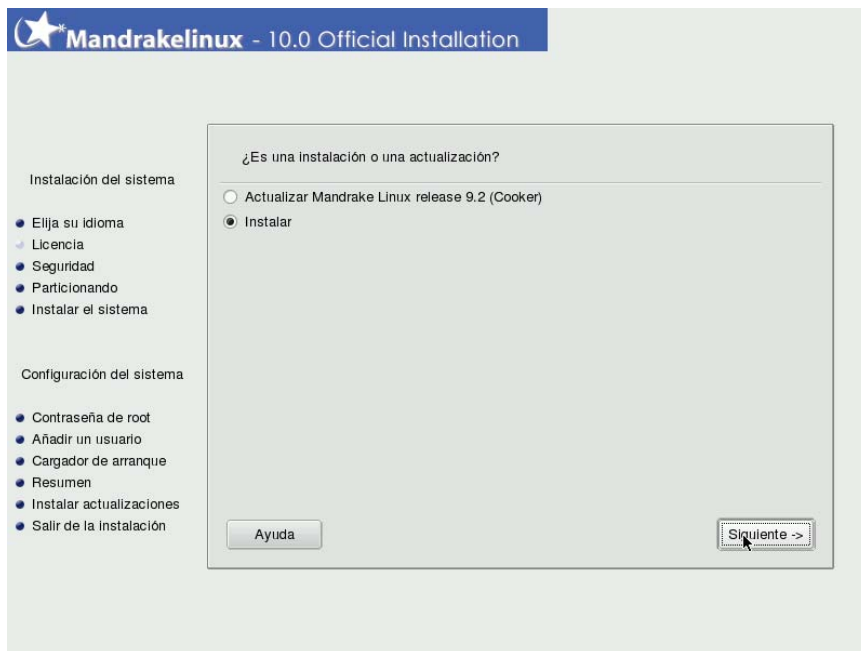


Figura 3. 4 Selección de Ratón

Una vez configurado el ratón debe seleccionarse el tipo de instalación en caso que se encuentre una partición activa de GNU/Linux (o sea que haya otra versión instalada), las dos opciones disponibles son **Instalar** o **Actualizar**. En la primera opción se hará una instalación nueva del sistema, sea que el disco duro no tenga ningún sistema operativo instalado o que exista una versión más antigua de Linux Mandrake. Esta opción borrará cualquier versión previa que se tenga instalada en ese disco.

La segunda opción revisará el software instalado y hará una actualización de los paquetes que encuentre instalados y una versión nueva de los paquetes para los que no haya sido encontrada una versión previa.

En nuestro caso seguiremos con la opción **Instalar**.



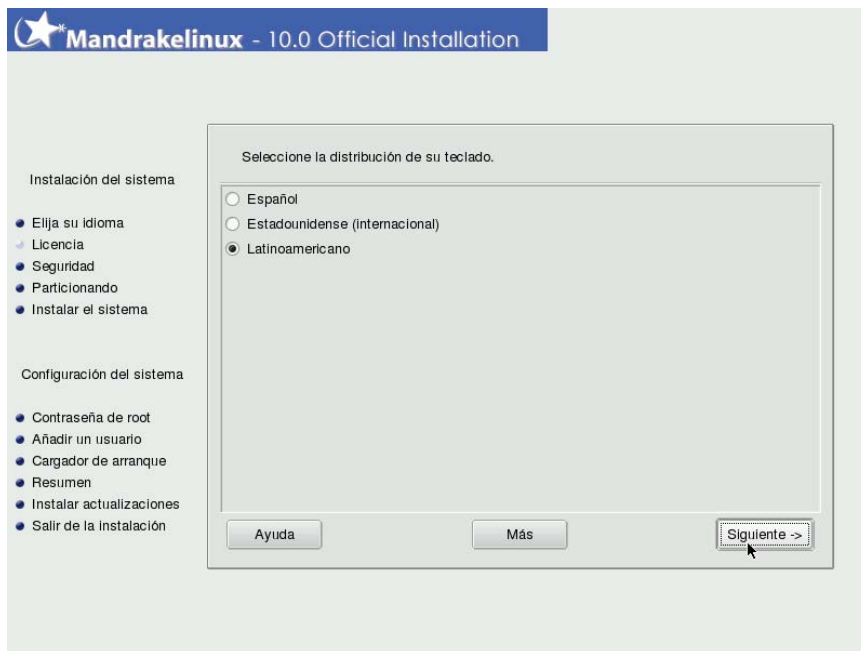
**Figura 3. 5** Selección de Tipo de Instalación

Dependiendo del idioma seleccionado, *DrakX* seleccionará automáticamente un tipo particular de configuración del teclado como se observa en la figura 3.6. Verificar que la selección satisface las necesidades o elija otra distribución de teclado.



También, puede ser que no tenga un teclado que se corresponde exactamente con su idioma: por ejemplo, si Usted es un argentino que habla inglés, puede ser que tenga un teclado latino americano. O si habla español pero está en Inglaterra puede estar en la misma situación en la cual su idioma nativo y su teclado no coinciden. En cualquier caso, este paso de instalación le permitirá elegir un teclado apropiado de una lista. Haga clic sobre el botón **Más** para que se le presente la lista completa de los teclados soportados.

Si se eligió una distribución de teclado basada en un alfabeto no latino, el próximo diálogo le permitirá elegir la combinación de teclas que cambiará la distribución del teclado entre la latina y la no latina.



**Figura 3. 6** Configuración de Teclado

Ahora es tiempo de seleccionar el nivel de seguridad de la máquina. En este punto, *DrakX* le permitirá elegir el nivel de seguridad que desea para su máquina. Como regla general, el nivel de seguridad debería ser mayor si la máquina va a contener datos cruciales, o si estará expuesta directamente a la Internet. No obstante, un nivel de seguridad más alto generalmente se obtiene a expensas de la facilidad de uso.

Si no sabe cual elegir, mantenga la opción predeterminada (ver figura 3.7). Podrá cambiarse después de la instalación con la herramienta *draksec*, que es parte del Centro de Control de Mandrakelinux.

Completar el campo Administrador de seguridad con la dirección de correo electrónico de la persona responsable de la seguridad. Los mensajes relativos a la seguridad se enviarán a esa dirección.

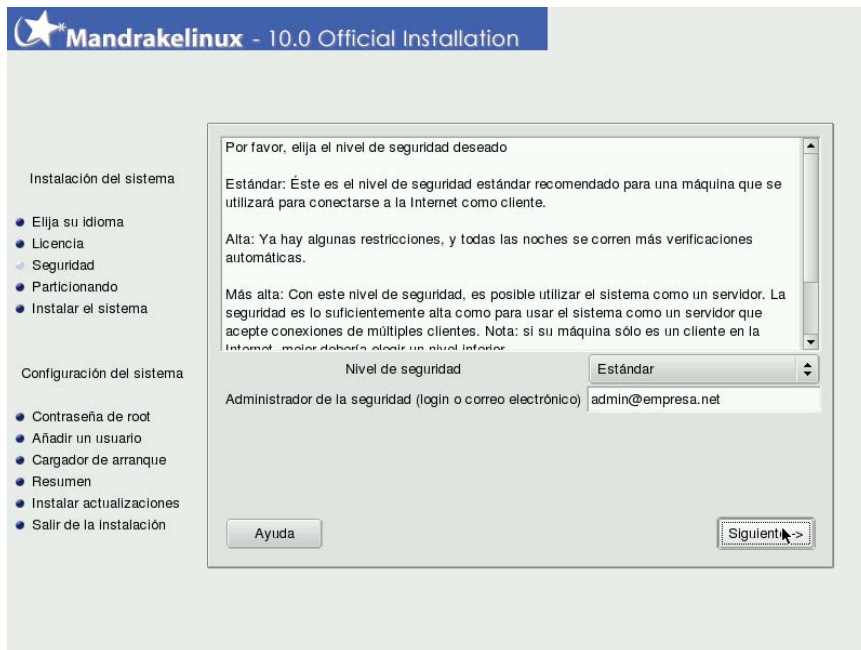


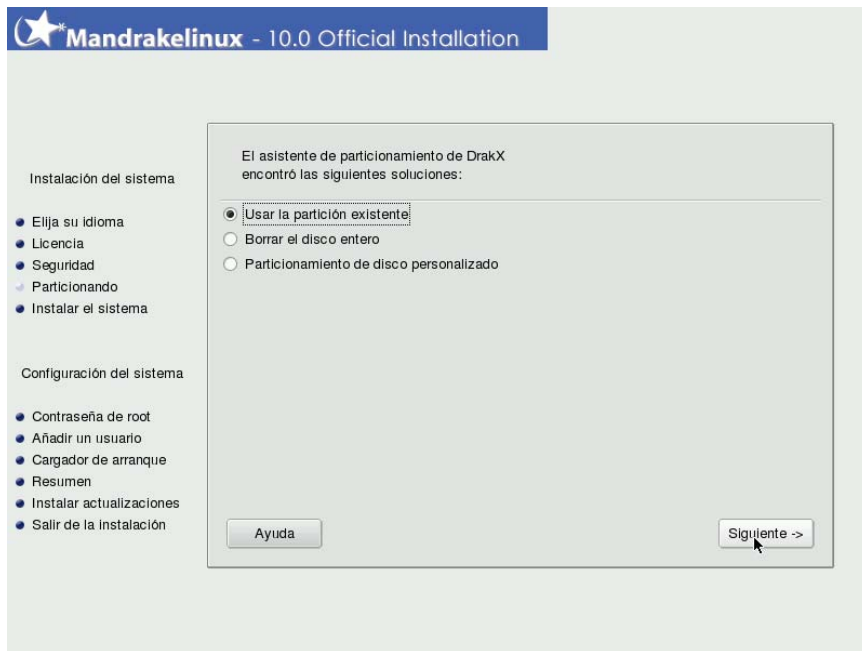
Figura 3. 7 Nivel de Seguridad

Una vez que se ha seleccionado el nivel de seguridad es tiempo de elegir los puntos de montaje para el nuevo sistema. En este punto hay bastante que contar porque es un paso muy delicado cuando exista otro sistema operativo en la máquina o el disco contiene información importante. Nosotros partiremos de la premisa que el disco está en blanco y que no habrá otro sistema operativo como **Microsoft Windows** residiendo en el disco.

Entonces solo seleccionamos la opción **Utilizar espacio libre en el disco**, con lo cual se realizará un particionamiento automático en el disco

vacío. Una vez escogido el tipo de particionamiento presionar el botón **Siguiente**.

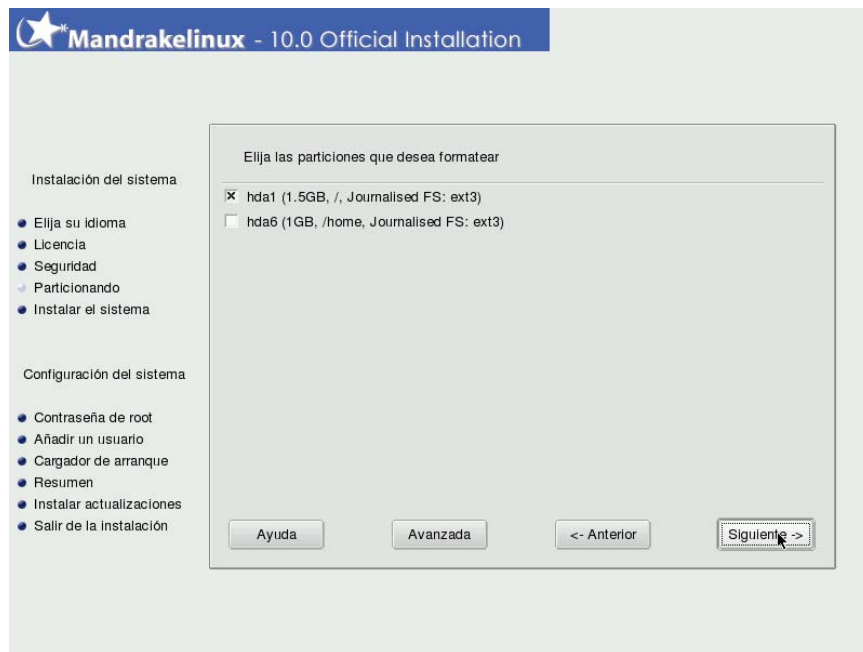
Si la persona a cargo de la instalación tiene buenos conocimientos sobre particionamiento y el sistema de archivos utilizado por Linux entonces se puede seleccionar la opción **Particionamiento de Disco Personalizado** (figura 3.8) donde el usuario puede realizar una asignación manual del espacio de disco. Por ejemplo puede seleccionar que se le asignará 2 Gigabytes a `/home`, 5 Gigabytes a `/`, 2 Gigabytes a `/usr` y 1 Gigabyte al espacio de intercambio `swap`. Esto significa que el espacio para datos de usuario (`/home`) tendrá un espacio dedicado de 2 Gigabytes, que los datos de programa en `/usr` tendrán también 2 Gigabytes dedicados, el espacio de intercambio tendrá 1 Gigabyte (por lo general es el doble de la memoria física instalada) y el resto de directorios bajo `/` tendrán 5 Gigabytes entre todos.



**Figura 3. 8** Particionamiento

Una vez realizado el particionamiento es tiempo de formatear las particiones recién creadas. En una ventana como la que se muestra en la figura 3.9 deben confirmarse las particiones que se particionarán, en el

caso de una instalación nueva aparecerán seleccionadas todas las particiones. Para proceder simplemente presionar el botón **siguiente**.



**Figura 3. 9** Formatear Particiones

Una vez que haya terminado el formateo de las particiones aparecerá una pantalla como la de la figura 3.10 para que se seleccione el software que se desea instalar. Como en Linux Mandrake hay miles de aplicaciones disponibles, estas se han agrupado en categorías para facilitar la instalación.

Linux Mandrake clasifica los grupos de paquetes en cuatro categorías. Puede mezclar y hacer coincidir aplicaciones de varias categorías, por lo que una instalación de “Estación de trabajo” puede tener instaladas aplicaciones de la categoría “Desarrollo”.

- Estación de trabajo: si planifica utilizar su máquina como una estación de trabajo, seleccione una o más aplicaciones de la categoría estación de trabajo.
- Desarrollo: si planifica utilizar su máquina para programar, elija los grupos apropiados de esa categoría.

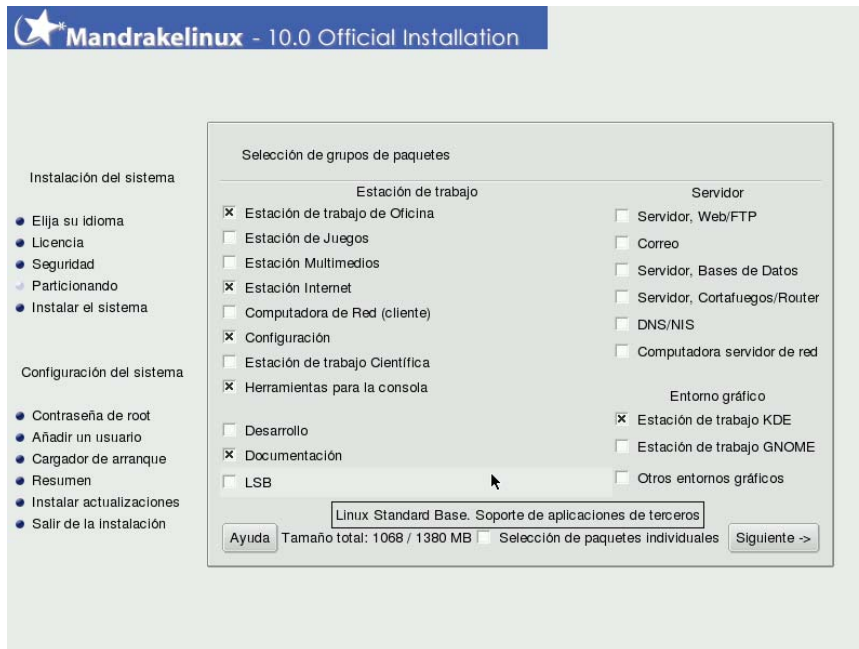
- Servidor: si pretende usar la máquina como un servidor, seleccione cuáles de los servicios más comunes desea instalar en su máquina.
- Entorno gráfico: aquí es donde seleccionará su entorno gráfico preferido. Si desea tener una estación de trabajo gráfica, debe seleccionar al menos uno.

En nuestro caso seleccionaremos los siguientes grupos:

- Estación de Trabajo de Oficina
- Configuración
- Herramientas para la consola
- Documentación
- Desarrollo
- Entorno Gráfico KDE

La instalación y configuración de los servidores puede hacerse en este paso o luego de la instalación. Nosotros la realizaremos después de la instalación en los apartados específicos para cada uno de los servidores. De forma opcional puede instalarse el otro entorno gráfico recomendado llamado GNOME.

Además si se selecciona el grupo LSB se instalará el núcleo 2.4 en lugar del 2.6 para asegurar una compatibilidad 100% con LSB del sistema. Aunque Mandresoft asegura que aunque no se instale este grupo se tendrá una compatibilidad de casi 100%.



**Figura 3. 10** Selección de Paquetes

Hecha la selección de paquetes se procede con la instalación presionando el botón **siguiente**.

Se mostrará una pantalla como la de la figura 3.12 donde se mostrará el avance de la instalación del sistema operativo y además si se han seleccionado múltiples paquetes de software *DrakX* expulsará el disco actual y pedirá el disco donde se encuentra(n) la aplicación(es) que intenta instalar. Este paso lleva su tiempo dependiendo de que tan rápida es la computadora y de la cantidad de paquetes seleccionados, pero por lo general se hace en una media de 30 minutos.

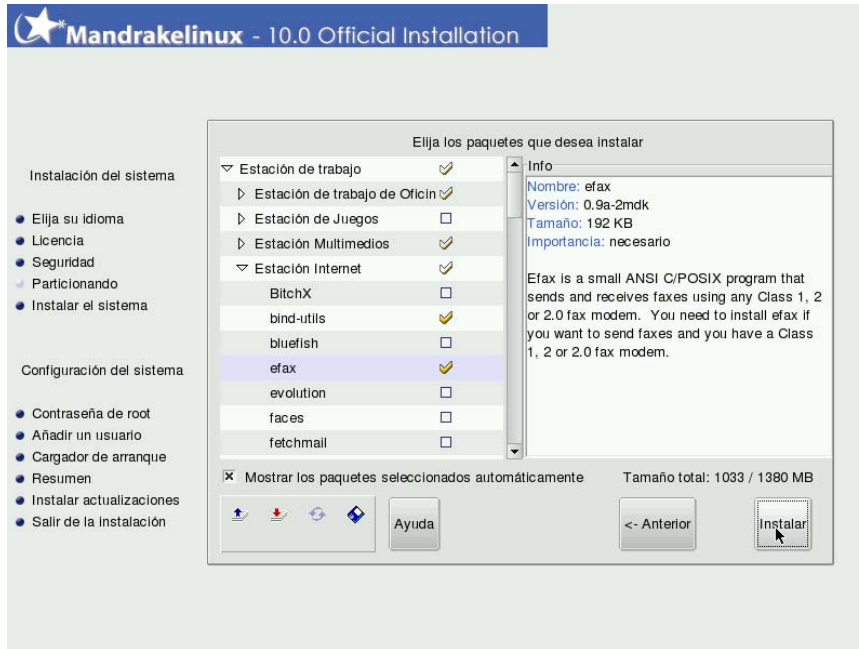


Figura 3. 11 Paquetes Individuales

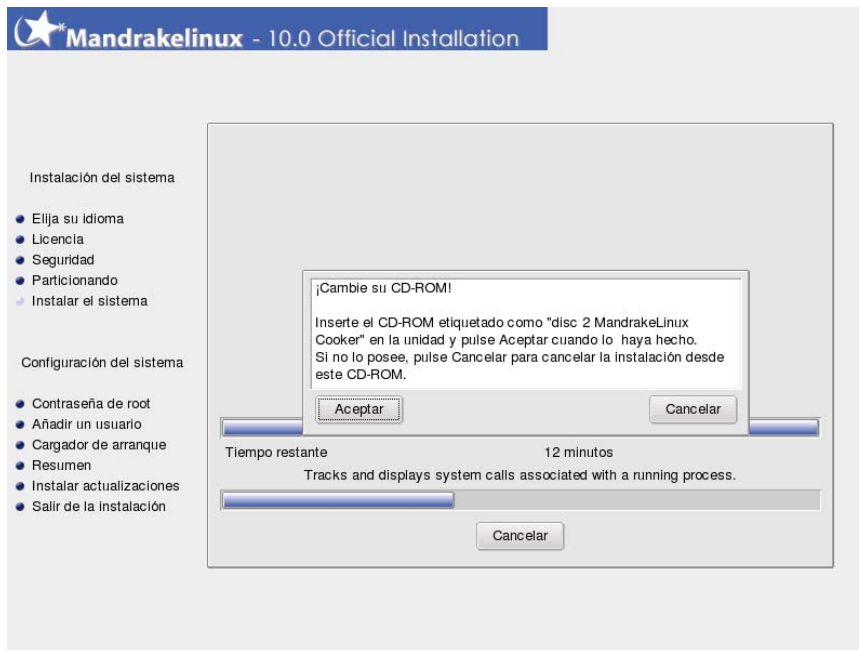


Figura 3. 12 Inicio de Instalación

Una vez concluida la instalación es tiempo de configurar el sistema, el primer paso es introducir la contraseña de root. Este es el punto de decisión más crucial para la seguridad de su sistema GNU/Linux. El usuario root es el administrador del sistema y es el único autorizado a

hacer actualizaciones, agregar usuarios, cambiar la configuración general del sistema, etc. Resumiendo, root puede hacer de todo, es por esto que deberá elegir una contraseña que sea difícil de adivinar *DrakX* le dirá si la que eligió es demasiado fácil. Como puede ver, no es forzoso ingresar una contraseña, pero se recomienda encarecidamente que se ingrese una. GNU/Linux es tan propenso a errores del operador como cualquier otro sistema operativo. Es importante que sea difícil convertirse en root debido a que root puede sobrepasar todas las limitaciones y borrar, sin intención, todos los datos que se encuentran en las particiones accediendo a las mismas sin el cuidado suficiente.

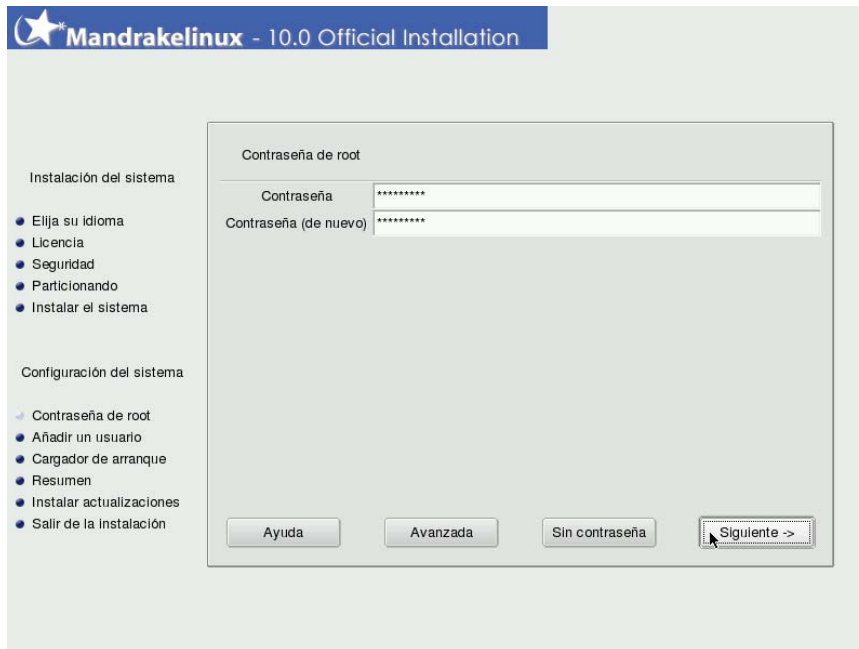
La contraseña debería ser una mezcla de caracteres alfanuméricos y tener al menos una longitud de 8 caracteres.

Sin embargo, no debería hacer la contraseña muy larga o complicada para que pueda ser recordada.

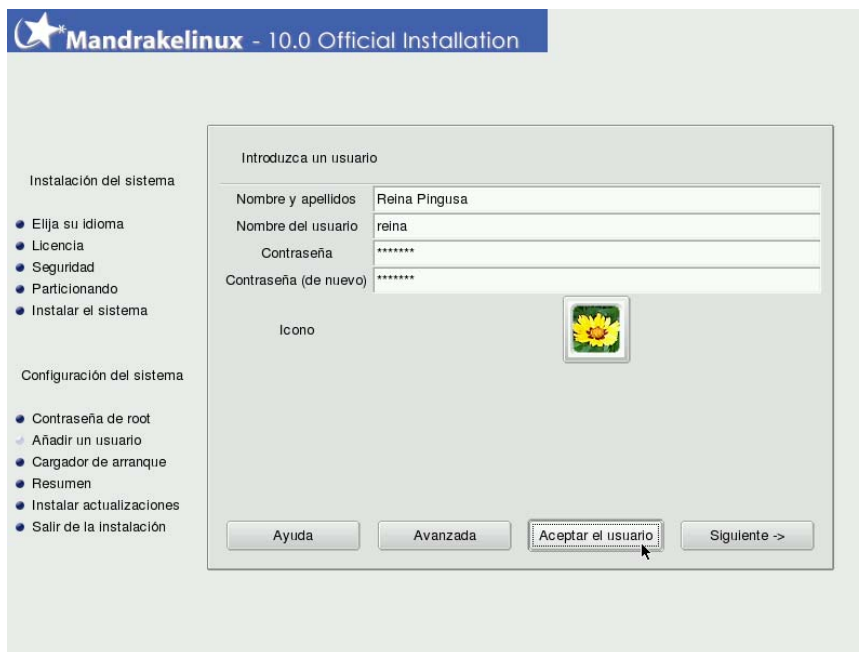
La contraseña no se mostrará en la pantalla a medida que se ingrese. Deberá teclear la contraseña dos veces para reducir la posibilidad de un error de tecleo a ciegas como se muestra en la figura 3.13.

Además de root deberá crear al menos un usuario (figura 3.14) para realizar las tareas básicas del sistema y utilizar la cuenta de root solo cuando sea indispensable. En este paso puede asociar una imagen a la cuenta. El campo **Nombre y apellido del usuario** solo es para una descripción de la cuenta. El nombre con el cual debe ingresar al sistema es el que se introduce en el campo **Nombre de Usuario**, además debe introducir la contraseña de la cuenta. Puede introducir cuantas cuentas de usuario se deseen, una vez que se ha concluido presionar el botón **siguiente**.





**Figura 3. 13** Cuenta de Administrador

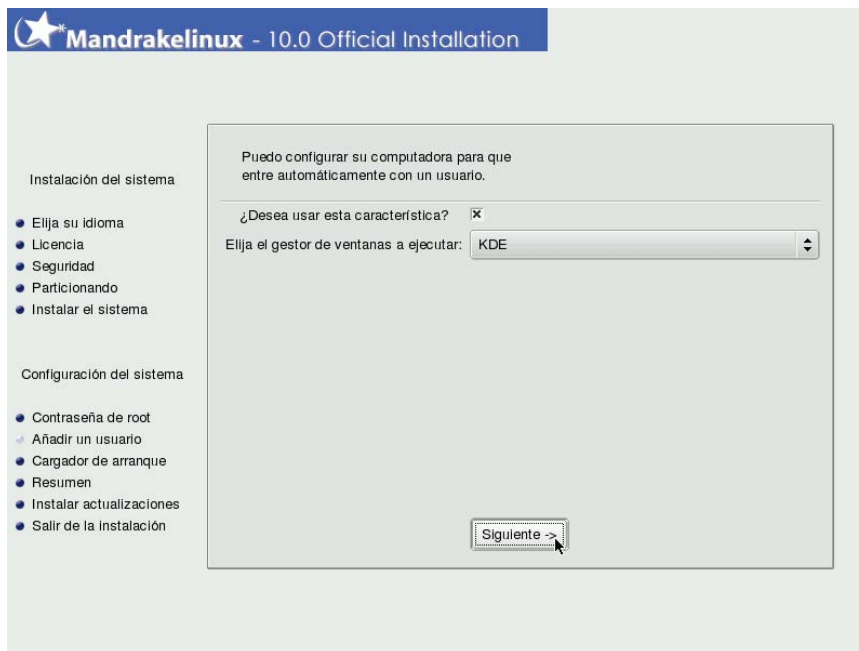


**Figura 3. 14** Creación de Usuario

Una vez creadas las cuentas de usuario, *DrakX* preguntará si deseamos arrancar en entorno gráfico o en modo consola (figura 3.14). Si seleccionamos arrancar en modo gráfico debemos especificar si queremos acceder automáticamente con una cuenta de usuario o

queremos ingresar nuestro nombre de usuario y nuestra contraseña en un cuadro de verificación.

Si seleccionamos arrancar en modo consola el usuario se verá obligado a introducir su nombre de usuario y su contraseña. Para acceder al escritorio gráfico se debe introducir el comando `startx & o startx KDE &`. El primer comando iniciará el entorno gráfico predeterminado mientras que el segundo iniciará el entorno gráfico KDE.



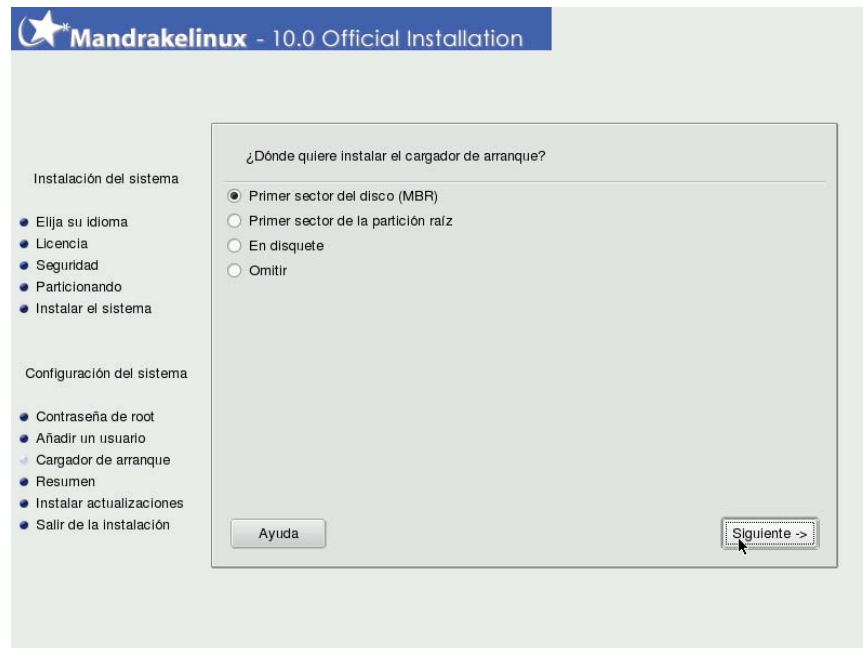
**Figura 3. 15** Selección de Escritorio

Es hora de instalar el cargador de arranque. Un cargador de arranque es un pequeño programa que la computadora inicia al momento del arranque. Es el responsable de arrancar todo el sistema. Normalmente la instalación del cargador de arranque está completamente automatizada. *DrakX* analizará el sector de arranque del disco y actuará en función de lo que encuentre allí:

- Si encuentra un sector de arranque de Windows, lo reemplazará con un sector de arranque de GRUB/LILO de forma tal que Usted

pueda cargar GNU/Linux o cualquier otro sistema operativo instalado en su máquina.

- Si encuentra un sector de arranque de GRUB o LILO, lo reemplazará con uno nuevo.
- Si no puede realizar una determinación, *DrakX* le preguntará dónde colocar el cargador de arranque. Generalmente, el Primer sector del disco (MBR) es el lugar más seguro. Si no va a instalar cargador de arranque alguno seleccione **Omitir**. Debe usarse solamente si se sabe lo que se está haciendo.



**Figura 3. 16** Instalación del cargador de Arranque

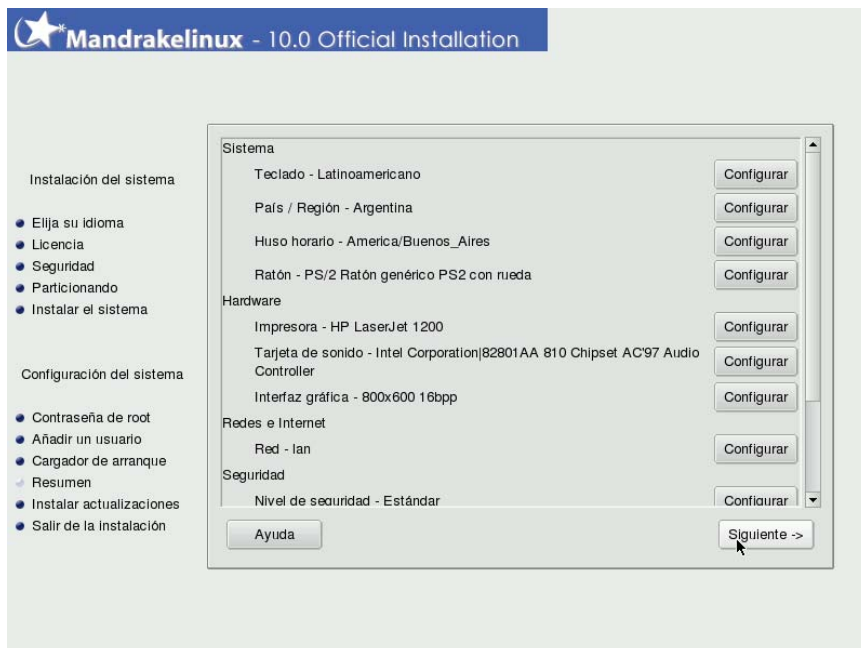
La instalación ha terminado por lo que se presenta un cuadro de resumen como el que se observa en la figura 3.17.

A manera de revisión, *DrakX* presentará un resumen de las distintas informaciones que recopiló acerca de su sistema. Dependiendo del hardware instalado en su máquina, puede tener algunas o todas las entradas siguientes. Cada entrada está compuesta del elemento a configurar, seguido de un pequeño resumen de la configuración actual.

Haga clic sobre el botón Configurar correspondiente para hacer los cambios.

- Teclado: verifique la configuración de la disposición actual del teclado y cámbiela si es necesario.
- País / Región: verificar la selección actual del país. Si no se encuentra en este país haga clic sobre el botón Configurar y seleccione otro. Si el país no se muestra en la primera lista que se muestra, pulsar sobre el botón **Más** para obtener la lista completa de países.
- Huso horario: De manera predeterminada *DrakX* deduce el huso horario basándose en el país que ha elegido. Puede pulsar sobre el botón **Configurar** si esto no es correcto.
- Ratón: verifique la configuración del ratón y pulsar el botón para cambiarla, si es necesario.
- Impresora: al pulsar sobre el botón **Configurar** se abrirá el asistente de configuración de la impresora.
- Tarjeta de sonido: si se detecta una tarjeta de sonido en su sistema, la misma se mostrará aquí. Si nota que la tarjeta de sonido no es la que está realmente presente en su sistema, puede pulsar sobre el botón y elegir un controlador diferente.
- Tarjeta de TV: si se detecta una tarjeta de TV en el sistema, la misma se muestra aquí. Si se tiene una tarjeta de TV y la misma no se detecta, haga clic sobre Configurar para intentar configurarla manualmente.
- Tarjeta RDSI: Pulsar sobre **Configurar** para cambiar los parámetros asociados a la tarjeta si no son los correctos.
- Interfaz gráfica: de manera predeterminada *DrakX* configura su interfaz gráfica en 800x600 o 1024x768 de resolución. Si eso no le satisface, pulsar **Configurar** para cambiar la configuración su interfaz gráfica.

- Red: si desea configurar ahora el acceso a la Internet o a su red local, puede hacerlo ahora.
- Proxis: permite configurar las direcciones de los proxy HTTP y FTP si la máquina sobre la que está instalando estará ubicada detrás de un servidor proxy.
- Nivel de seguridad: esta entrada ofrece volver a definir el nivel de seguridad como se ajustó anteriormente.
- Cortafuegos: si planifica conectar su máquina a la Internet, es una buena idea protegerse de las intrusiones configurando un cortafuego.
- Cargador de arranque: si se desea cambiar la configuración del cargador de arranque, pulsar sobre este botón. Esto debería estar reservado para los usuarios avanzados.
- Servicios: por medio de esta entrada podrá tener un control fino sobre qué servicios correrán en su máquina.



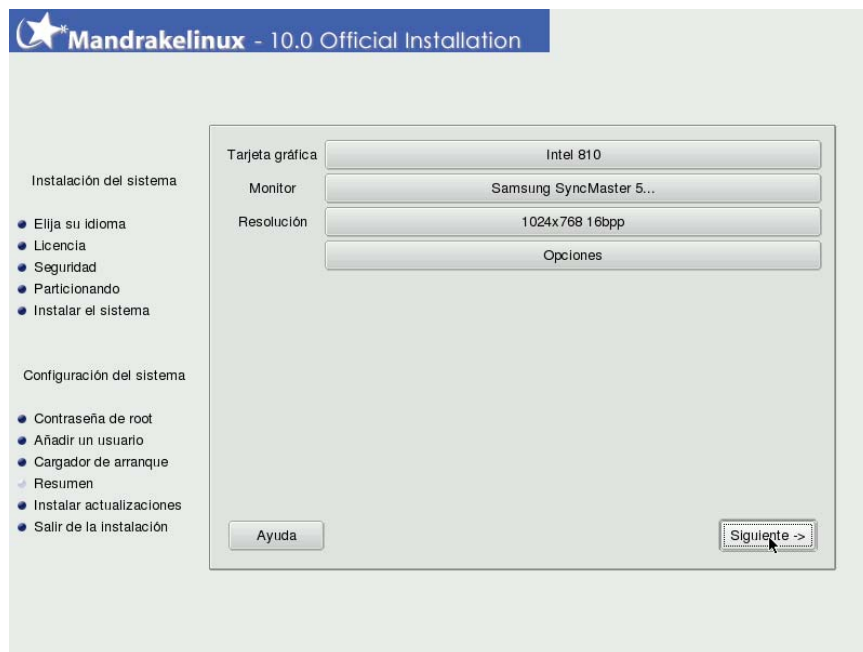
**Figura 3. 17** Configuración Final

Si deseamos utilizar una interfaz gráfica podemos configurarla y probarla seleccionando la opción **Interfaz Gráfica** como se muestra en la figura 3.18.

Aquí seleccionamos la tarjeta de video que está instalada, esto generalmente es hecho por el sistema. Si pulsamos sobre el botón **Interfaz Gráfica** aparecerá un listado de las tarjetas de video disponibles de las cuales se puede seleccionar una.

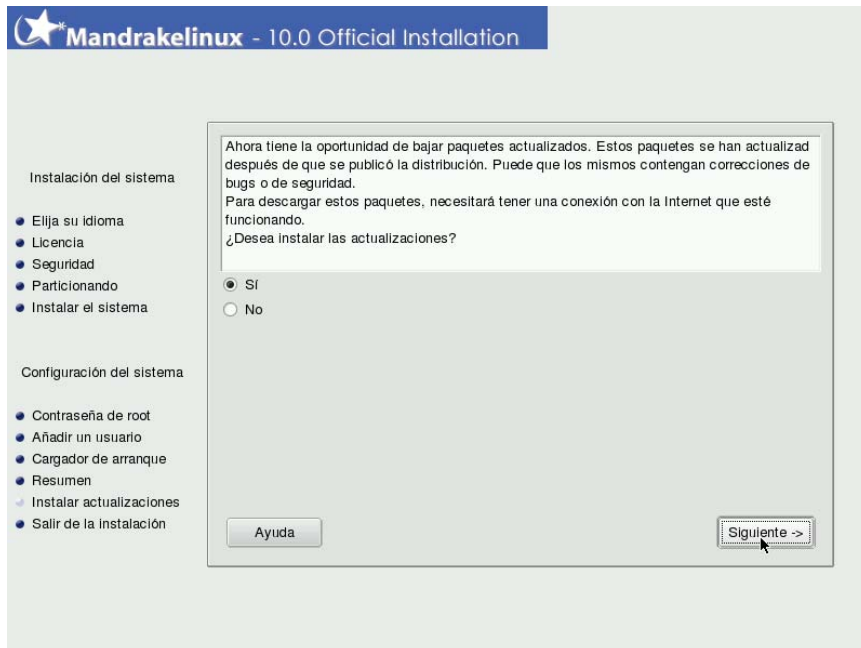
Puede seleccionarse además el monitor y la resolución, para esto es importante contar con la documentación o información del hardware porque en el caso del monitor si seleccionamos una frecuencia de refresco inadecuada podemos dañarlo permanentemente.

Una vez hechos los ajustes, pulsar el botón **probar** para verificar que la configuración del servidor gráfico es la apropiada. Aparecerá una imagen multicolor, si está bien entonces presionar el botón **aceptar**.



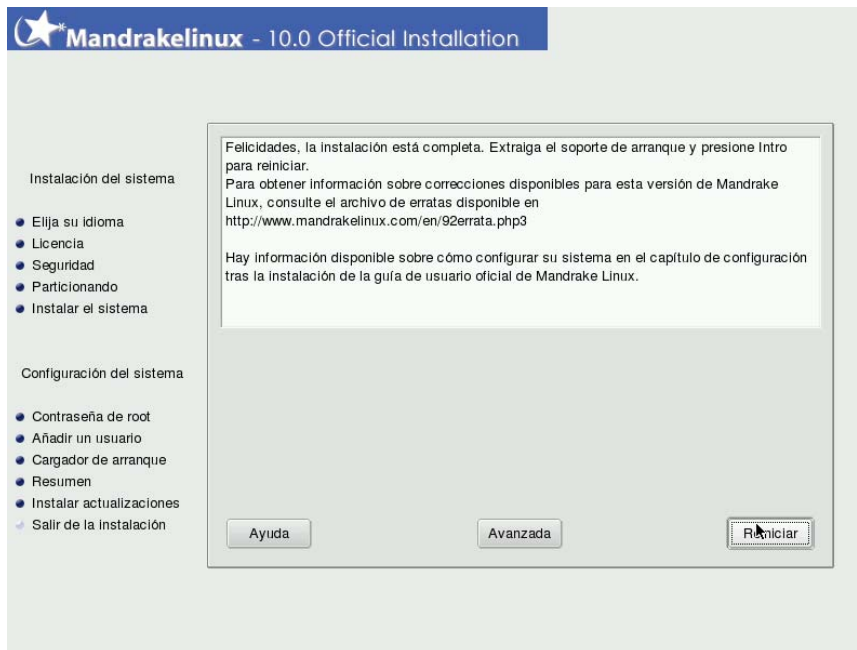
**Figura 3. 18** Configuración de la interfaz Gráfica

El último paso de la instalación es seleccionar si deseamos descargar las últimas actualizaciones del sistema operativo. Si se tiene una conexión activa a Internet puede realizarse aquí o posteriormente desde el gestor de actualizaciones de Linux Mandrake.



**Figura 3. 19** Instalación de Actualizaciones

Ahora simplemente resta presionar el botón **reiniciar** (figura 3.20) para salir de la instalación. El disco en la unidad de CDROM será expulsado y el sistema reiniciará.



**Figura 3. 20** Fin de Instalación

Después que el sistema reinicie aparecerá el cargador de arranque con varias opciones:

- Linux
- Failsafe
- Floppy

Si queremos arrancar el sistema de forma normal seleccionamos **Linux**, **Failsafe** si queremos iniciar en el modo a prueba de fallas o **Floppy** para arrancar desde un disquete.

## **b. Instalación del Servidor Apache y Módulos necesarios**

Para la instalación del servidor Web *Apache* utilizaremos los paquetes *rpm* incluidos en la distribución *Linux Mandrake*. Para esto tendremos que ejecutar la herramienta de configuración *RpmDrake* como se mostró en la **Parte I**, correspondiente a la instalación y desinstalación de programas en el formato *rpm*.

Una vez hecho esto podemos introducir la palabra clave *apache* en el campo de búsqueda y a continuación se desplegará una lista con todos



los paquetes que incluyen la palabra clave *apache*. Otra opción es navegar en el panel de la izquierda en la sección correspondiente a servidores y allí seleccionar servidores Web.

Los paquetes a instalar serán:

- `apache2-2.0.48-6mdk`
- `apache-2-mod_php`
- `apache2-mod-perl`

Se notará que al seleccionar el paquete `apache2-2.0.48-6mdk`, el administrador de paquetes indica que para satisfacer las dependencias es necesario instalar otros paquetes. Si deseamos instalar *Apache* debemos seleccionar aceptar.

Los paquetes `apache2-mod_php` y `apache2-mod_perl` son utilizados para agregar soporte para la ejecución de *scripts* escritos en *PHP* y *Perl* respectivamente.

Un paquete opcional de gran utilidad es el de documentación llamado `apache2-manual`, el cual instalará toda la información contenida en el sitio Web de *Apache* y que está en formato HTML.

Una vez seleccionados los paquetes se procede a la instalación pulsando sobre el botón **INSTALAR**, el administrador consultará si se desea proceder con la instalación para lo cual debemos confirmar, *RpmDrake* se hace cargo del proceso y únicamente nos pedirá que insertemos cualquiera de los cuatro discos de instalación. Cuando el proceso finaliza el programa hace una actualización a la base de datos de paquetes instalados y finaliza; después de esto podemos salir del programa de instalación y continuar con el siguiente paso.

### **c. Arranque y puesta en marcha de Apache**

Para arrancar el servidor Web *Apache* necesitamos ejecutar el siguiente comando desde un Terminal de comandos:

```
/etc/rc.d/init.d/httpd start
```

Después de un pequeño lapso de tiempo obtendremos el mensaje: “[OK] Service Started” en pantalla lo que indica que el servidor ha sido iniciado exitosamente. Una vez hecho esto solo es necesario colocar los documentos Web en el directorio destinado para esto.

Si en algún momento se desea realizar trabajos de mantenimiento o realizar actualizaciones el servidor se puede detener con el comando:

```
/etc/rc.d/init.d/httpd stop
```

En caso de reinicio de la máquina por cualquier motivo el servidor arrancará automáticamente.

#### **d. Pruebas de Configuración de Apache**

Una vez que Apache ha sido adecuadamente instalado y arrancado ya esta listo para alojar las páginas Web y los *scripts*. La versión en formato *tarball* incluye una página Web de prueba con información sobre *Apache*, en el caso de la versión *rpm* incluida con Linux Mandrake la página que se verá es una página de prueba con información sobre la versión de personalizada de Mandrake. Desde un navegador introducimos en la barra de dirección **localhost** ó **127.0.0.1** que es la dirección reservada para la máquina local. Si el servidor *Apache* funciona adecuadamente deberíamos ver la página de prueba como se observa en la figura 3.21.

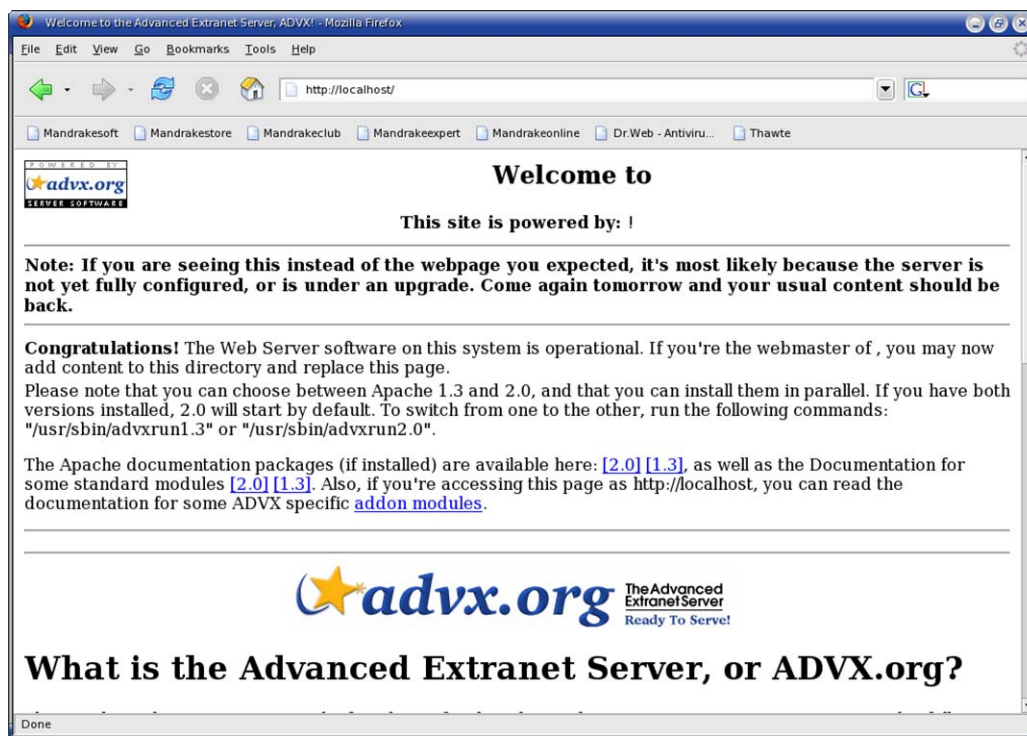


Figura 3. 21 Página de Inicio

Si fue instalado el paquete de documentación desde esta página se encuentran los enlaces a ésta. Como en los discos de *Linux Mandrake* se incluyen la versión 2 y la versión 1.3 de *Apache* aparecen enlaces a ambas documentaciones, para el caso sería la documentación para la versión 2.

El servidor está listo para ser puesto en servicio excepto que se deseen realizar cambios a la configuración. Si fuera el caso esto puede realizarse al archivo de configuración de *Apache* llamado **httpd.conf** en el directorio `/etc/httpd/conf/`.

#### e. Creación y Configuración de Autenticación en el Servidor

Una de las grandes ventajas de *Apache* es su modularidad y el uso de directivas para realizar cambios en la configuración del servidor o la implementación de características adicionales como la autenticación.

La implementación de autenticación en el servidor *Apache* significa que para poder ingresar a una página alojada en el servidor es necesario

proporcionar un nombre de usuario y una contraseña. Se puede requerir a un usuario único o a un grupo de usuarios. La autenticación es aplicable cuando un documento posee información que solo es pertinente a un grupo de usuarios.

Los módulos necesarios para implementar autenticación son:

- mod\_access
- mod\_auth

Las directivas necesarias para la implementación de autenticación son:

- AuthType
- AuthName
- Require
- AuthUserFile

Para poder realizar cambios en la configuración del servidor *Web Apache* hay dos formas posibles de realizarlo:

- Modificando el archivo de configuración `httpd.conf`.
- Por medio de archivos `.htaccess` en una base por directorios.

Cuando se utiliza la primera opción las directivas se colocan en el archivo de configuración `httpd.conf`, para esto debemos especificar el directorio en el que deseamos aplicar la(s) directiva(s).

Para la segunda opción creamos un archivo llamado `.htaccess` y lo colocamos en el directorio donde queremos aplicar la(s) directiva(s).

En nuestro caso utilizaremos las directivas para la autenticación en un archivo `.htaccess`.

Como el directorio raíz para nuestros documentos es `/var/www/html` ese será el lugar donde colocaremos el archivo `.htaccess`.

Para poder aplicar la autenticación a un directorio necesitamos crear un archivo de contraseñas por medio de la utilidad **htpasswd** y colocar este archivo en un directorio. Nosotros lo colocaremos en un directorio llamado `password` ubicado en `/var/www/html`.

Para crear el directorio, como root:

```
#mkdir /var/www/html/password
```

Para crear el archivo de contraseñas, como root ingresar el siguiente comando:

```
#/usr/sbin/htpasswd -c /var/www/html/password/contraseña administrador  
New password: admin  
Re-type new password: admin  
Adding password for user administrador
```

Ahora lo único que necesitamos es crear el archivo `.htaccess` con cualquier editor de texto y colocarlo en el directorio `/var/www/html`.

El contenido del archivo será:

```
AuthType Basic  
AuthName "Monitoreo Restringido"  
AuthUserFile /var/www/html/password/contraseña  
Require user administrador
```

La interpretación de las directivas es la siguiente:

- `AuthType` indica el tipo de autenticación
- `AuthName` el mensaje en la ventana de autenticación
- `AuthUserFile` indica la ubicación y el nombre del archivo de contraseña
- `Require` para requerir contraseña al usuario administrador

Ahora cada vez que ingresemos **127.0.0.1** en la barra de dirección del navegador Web obtendremos una ventana de verificación como la siguiente:

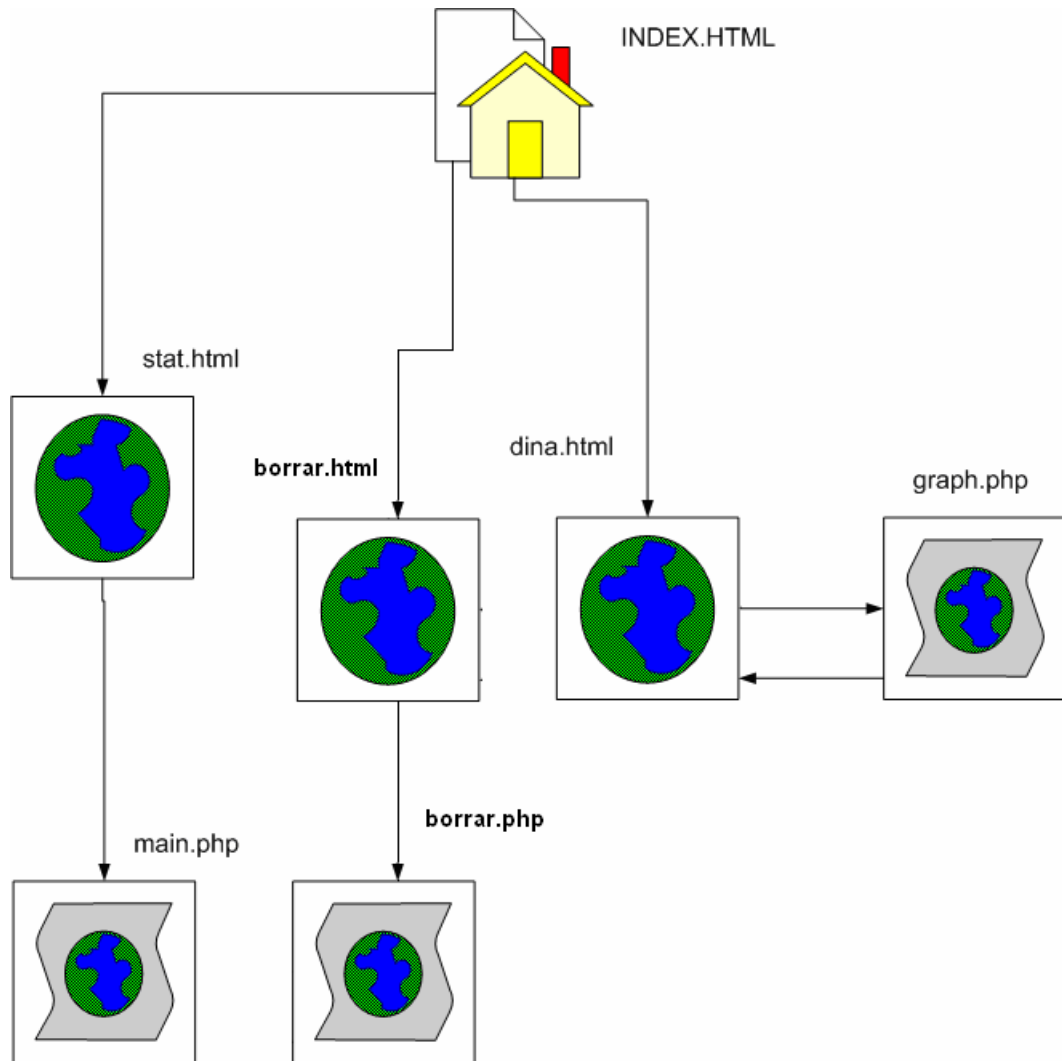


**Figura 3. 22** Cuadro de Autenticación

De no ingresar los datos correctos el navegador será redirigido a una página de error.

#### **f. Desarrollo de Interfaz Web**

Para el desarrollo de la interfaz Web se utilizarán páginas HTML sencillas desde las cuales se llamarán los *scripts* PHP. El esquema del sitio es el siguiente:



**Figura 3. 23:** Esquema del Sitio

Para acceder a la página principal bastará con ingresar la dirección IP del servidor en un navegador, por defecto se cargará la página `index.html`. Desde esta página se tendrá que escoger entre tres opciones:

- Ver gráfico Dinámico
- Ver Estadísticas
- Borrar Datos (Esta opción solo es para administradores)

Si seleccionamos la opción ver gráfico dinámico ingresaremos a una página HTML llamada `dina.html` que está enlazada a un *script* PHP llamado `graph.php` que generará el gráfico. Esta página se refrescará

cada 15 segundos y de esta forma mantener siempre actualizado el gráfico.

Si seleccionamos la opción `estat.html` ingresaremos a una página con múltiples opciones de selección para diferentes parámetros de búsqueda. Las opciones seleccionadas aquí se pasan mediante el método GET a un *script* llamado `main.php` que genera un gráfico o una tabla HTML dependiendo de las opciones y parámetros de búsqueda seleccionados. El desarrollo de ambos *scripts* PHP se detallan más adelante, por los momentos nos concentraremos en `index.html`, `dina.html` y `stat.html`.

Recordemos que `index.html` solo debe contener los hipervínculos hacia las diferentes opciones por lo que el diseño de la página es bastante sencillo.

El siguiente código es el correspondiente a esta página, únicamente lo ingresamos en cualquier editor de texto y lo guardamos como `index.html`.

```
<html>
<title>PAGINA DE MONITOREO DE LA UNIVERSIDAD DON BOSCO</title>
<META name="keywords" content="monitoreo">
<body bgcolor="#CDCDCD">
<center>
<marquee bgcolor="#006699" behavior="alternate"
direction="right">
<b><font face="arial" color="#FFFFCC" size="5">Bienvenidos a la
Página de Monitoreo de Temperatura de la UDB</font></b>
</marquee>
<hr width="100%" size="5">
<A href="http://www.udb.edu.sv"></A>
</center>
<hr width="100%" size="5">
<center>
```



```

<P><H2><FONT FACE="Arial" color="blue">SELECCIONE LA OPCION
DESEADA</FONT></H2></P>
</center>
<blockquote><blockquote><blockquote><p align="justify"><FONT
FACE="Tahoma" color="#211E60"><b>
Si desea ver el gráfico del comportamiento actual de la
temperatura entonces presione sobre el enlace GRAFICAR. Aquí se
encontrará con un gráfico dinámico que se refresca cada 5
segundos.</b></FONT></p></blockquote></blockquote></blockquote>
<blockquote><blockquote><blockquote><p align="justify"><FONT
FACE="Tahoma" color="#211E60"><b>
Si desea acceder a la Base de Datos y ver estadísticas presione
sobre el enlace ESTADÍSTICAS, esto lo llevará a una página con
diferentes patrones de búsqueda donde puede visualizar los datos
históricos en forma Gráfica o en una tabla
HTML</b></FONT></p></blockquote></blockquote></blockquote></block
quote>
<hr width=100% size=5>
<center>
<a href="dina.html">GRAFICAR</A>
<a href="stat.html">ESTADÍSTICAS</A><br><br><br>
<a href="temp/borrar.html">Borrar Datos (Solo Administrador)</A>
</center>
</body>
<html>

```

Está página funcionaría perfectamente bien si únicamente definiéramos los hipervínculos a las otras dos páginas, estos están definidos con las etiquetas `<a href>` `</a>`, pero se han incluido algunos atributos adicionales para hacer la página un poco más amigable a la vista.

No es el objetivo del documento ni del proyecto entrar en detalle con HTML pero se explicarán ligeramente algunos de los atributos:

- `<title></title>`: El texto ingresado entre estas dos etiquetas es lo que aparecerá como título de la ventana del navegador.
- `<marquee>` `</marquee>`: Con estas etiquetas definimos una marquesina de texto desplazándose continuamente de un lado a

otro de la pantalla. Los atributos que incluye son `bgcolor` que indica el color de fondo de la marquesina, `behavior="alternate"` indica que se desplazará de derecha a izquierda continuamente.

- `<hr>`: Inserta un separador horizontal, el atributo `width=100%` indica que abarque todo el ancho de la pantalla y `size` indica el grosor del separador.
- `<p> </p>`: El texto ingresado entre las dos etiquetas será ingresado como un párrafo, al que adicionalmente se le coloca el atributo `align`, para dar alineamiento al párrafo, las opciones son `right`, `left`, `center` o `justify`, dependiendo que tipo de alineación que deseamos.
- `<blockquote> </blockquote>`: el texto ingresado entre estas etiquetas se le agregará una sangría.
- `<font> </font>`: Este es para definir el tipo de letra del texto más otros atributos como `color`, `size` para el tamaño y `FACE` para definir el tipo de letra, por ejemplo Arial , Tahoma, Lucida entre otras más.

En la figura 3.24 se muestra como se verá la página `index.html`



**Figura 3. 24:** Página `index.html`

La página `dina.html` es bastante sencilla ya que únicamente es un enlace hacia el `script` `graph.php`, que será el encargado de generar el gráfico. El código HTML de `dina.html` se muestra a continuación:

```
<html>
<HEAD>
<TITLE>Graphing on the Fly</TITLE>
<META HTTP-EQUIV="Refresh" Content="15">
<META HTTP-EQUIV="Expires" Content="Wed, 11 Feb 1998 04:35:12
GMT">
<META HTTP-EQUIV="pragma" Content="no-cache">
</HEAD>
<BODY BGCOLOR=#FFFFFF TEXT=#000000>
<center><p>Gráfico Dinámico de Temperatura</p></center>
<center><p>Este Gráfico se refresca cada 5 Segundos</p></center>
<center></center>
```

```
</body>  
</html>
```

En el código anterior lo realmente importante son la etiquetas `<META>`. En primer lugar no necesitan de una etiqueta de cierre y además deben estar colocadas entre las etiquetas `<HEAD></HEAD>`.

Aquí el lugar de insertar una imagen con la etiqueta ``, apuntamos a un *script* que es el encargado de generar el gráfico. Pero esto sucederá únicamente una vez, cuando ingresemos por primera vez a la página o sea que para refrescar el gráfico el usuario tendría que recargar la página para que esta se actualice corriendo el *script* nuevamente

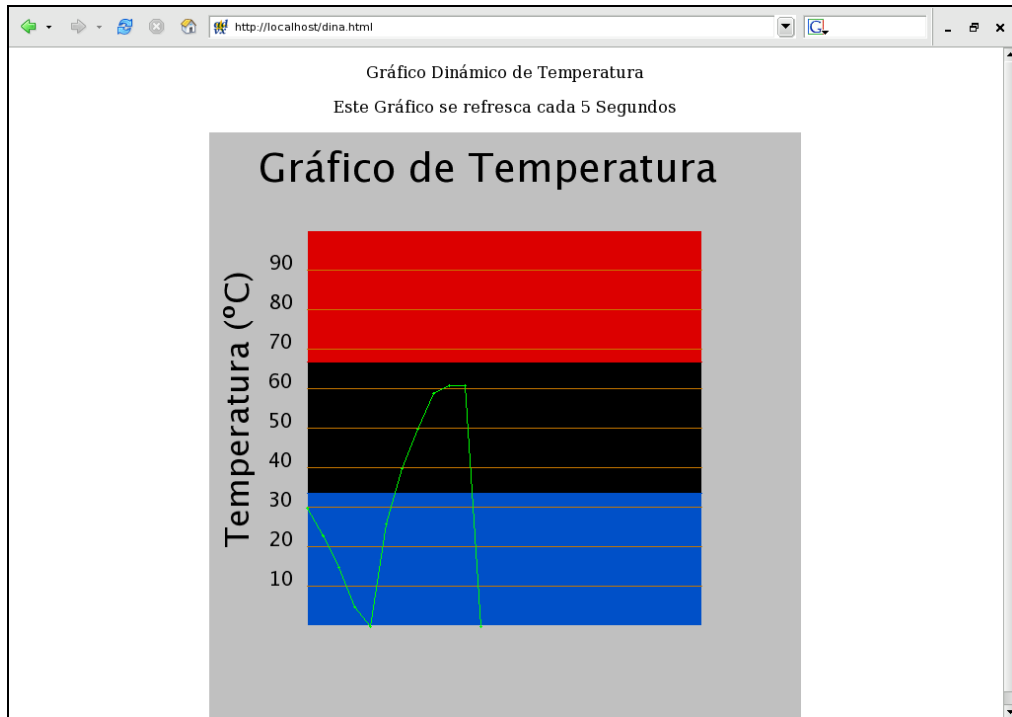
Es en este punto que entran en juego las etiquetas `<META>`:

```
<META HTTP-EQUIV="Refresh" Content="N° de segundos">
```

obliga a cargar nuevamente la página después de un número definido de segundos. Esto volvería a cargar el *script* nuevamente cargándose una nueva versión del gráfico, pero existe otro problema, si se utiliza un servidor Proxy, este enviará una copia almacenada en el cache en lugar de solicitar un refresco de la página al servidor o el propio navegador puede utilizar su cache. Esto se resuelve utilizando la etiqueta `<META HTTP-EQUIV="EXPIRES" Content="Wed, 11 Feb 1998 04:35:12 GMT">` Con esta etiqueta indicamos que la página expira en la fecha indicada, como se trata de una fecha pasada siempre solicitará una copia fresca.

El problema es que algunos navegadores pueden ignorar esta etiqueta por lo que existe un `META` más que puede ser utilizado. Se trata de la etiqueta `<META HTTP-EQUIV="pragma" CONTENT="no-cache">`, este solicita al servidor Web que le indique al navegador Web no utilizar su cache.

Finalmente la página se verá como esta:



**Figura 3. 25:** Página `dina.html`

Ahora únicamente tenemos pendiente el desarrollo de la página `stat.html`. Como se mencionó anteriormente esta página presenta varias opciones de selección para la generación de estadísticas, ya sea en forma gráfica o en tabla html.

Primero debemos seleccionar el tipo de búsqueda que queremos realizar, las opciones presentadas son:

- Por Fecha, Hora y Temperatura
- Únicamente por Fecha y Hora
- Únicamente por Rango de Temperatura

Una vez seleccionada el tipo de búsqueda se nos presentaran algunas listas desplegables para la selección de la Fecha de Inicio, Fecha Final, Temperatura Inferior y Temperatura Superior.

Por último se selecciona como queremos ver el resultado:

- Gráfico
- Tabla

La página se verá así:

http://localhost/stat.html

UNIVERSIDAD DON BOSCO  
VITAM IMPENDERE

PAGINA DE MONITOREO DE TEMPERATURA  
UNIVERSIDAD DON BOSCO

ELIJA UNA OPCION

Búsqueda por Fecha,Hora,Temperatura

Búsqueda por Fecha y Hora

Búsqueda por Rango de Temperatura

Seleccione los parámetros de búsqueda

Fecha Inicio(dd-mm-aaaa)	1	1	2005
Fecha Final(dd-mm-aaaa)	1	1	2005
Hora Inicial	00	00	00
Hora Final	00	00	00
Temperatura Inferior	0		
Temperatura Superior	0		

Gráfico  Tabla

Enviar Limpiar

Desarrollado por:  
Daniel Flores

Figura 3. 26: Página stat.html

El código necesario para está página es el siguiente:

```
<html>
<head>
<TITLE>Página de Monitoreo</TITLE>
<META NAME="keywords" CONTENT="HTML,internet">
</head>
<BODY bgcolor="#CDCDCD">
<fieldset>
<A href="http://www.udb.edu.sv"></A>
<strong><font face=Arial size=3 color=blue>PAGINA DE MONITOREO DE
TEMPERATURA<br clear>
UNIVERSIDAD DON BOSCO</font></strong>
</fieldset>
<hr width=100% size=5>
<fieldset>
<legend>
ELIJA UNA OPCION</legend>
<form action="main.php" method="GET">
```

```

<ul><input type="radio" name="reg[]" value="todo"><font
face=arial size=2>Búsqueda por
Fecha, Hora, Temperatura</font><br><br>
<input type="radio" name="reg[]" value="fecha"><font face=arial
size=2>Búsqueda por Fecha y Hora</font><br><br>
<input type="radio" name="reg[]" value="temp"><font face=arial
size=2>Búsqueda por Rango de Temperatura</font><br><br></ul>

<p>Seleccione los parámetros de búsqueda</p>

<table border="1">

<tr><td><font face=arial size=2>Fecha Inicio(dd-mm-
aaaa)</font></td><td><select name="diaini[]">
<option>1<option>2<option>3<option>4<option>5<option>6<option>7<o
ption>8<option>9<option>10<option>11<option>12<option>13<option>1
4<option>15<option>16<option>17<option>18<option>19<option>20<opti
on>21<option>22<option>23<option>24<option>25<option>26<option>2
7<option>28<option>29<option>30<option>31</select>
<select name="mesini[]">
<option>1<option>2<option>3<option>4<option>5<option>6<option>7<o
ption>8<option>9<option>10<option>11<option>12</select>
<select name="anoini[]">
<option>2005<option>2006<option>2007<option>2008<option>2009<opti
on>2010</select>
</td></tr>

<tr><td><font face=arial size=2>Fecha Final(dd-mm-
aaaa)</font></td><td><select name="diafin[]">
<option>1<option>2<option>3<option>4<option>5<option>6<option>7<o
ption>8<option>9<option>10<option>11<option>12<option>13<option>1
4<option>15<option>16<option>17<option>18<option>19<option>20<opti
on>21<option>22<option>23<option>24<option>25<option>26<option>2
7<option>28<option>29<option>30<option>31</select>
<select name="mesfin[]">
<option>1<option>2<option>3<option>4<option>5<option>6<option>7<o
ption>8<option>9<option>10<option>11<option>12</select>
<select name="anofin[]">
<option>2005<option>2006<option>2007<option>2008<option>2009<opti
on>2010</select>
</td></tr>

<tr><td><font face=arial size=2>Hora
Inicial</font></td><td><select name="horini[]">
<option>00<option>01<option>02<option>03<option>04<option>05<opti
on>06<option>07<option>08<option>09<option>10<option>11<option>12
<option>13<option>14<option>15<option>16<option>17<option>18<opti
on>19<option>20<option>21<option>22<option>23</select>

<select name="minini[]">
<option>00<option>01<option>02<option>03<option>04<option>05<opti
on>06<option>07<option>08<option>09<option>10<option>11<option>12
<option>13<option>14<option>15<option>16<option>17<option>18<opti
on>19<option>20<option>21<option>22<option>23<option>24<option>25

```





```
<option>0<option>1<option>2<option>3<option>4<option>5<option>6<option>7<option>8<option>9<option>10<option>11<option>12<option>13<option>14<option>15<option>16<option>17<option>18<option>19<option>20<option>21<option>22<option>23<option>24<option>25<option>26<option>27<option>28<option>29<option>30<option>31<option>32<option>33<option>34<option>35<option>36<option>37<option>38<option>39<option>40<option>41<option>42<option>43<option>44<option>45<option>46<option>47<option>48<option>49<option>50<option>51<option>52<option>53<option>54<option>55<option>56<option>57<option>58<option>59<option>60<option>61<option>62<option>63<option>64<option>65<option>66<option>67<option>68<option>69<option>70<option>71<option>72<option>73<option>74<option>75<option>76<option>77<option>78<option>79<option>80<option>81<option>82<option>83<option>84<option>85<option>86<option>87<option>88<option>89<option>90<option>91<option>92<option>93<option>94<option>95<option>96<option>97<option>98<option>99<option>100</select>
</td></tr>
```

```
<tr><td><font face=arial size=2>Temperatura Superior</font></td><td><select name="tf[]">
<option>0<option>1<option>2<option>3<option>4<option>5<option>6<option>7<option>8<option>9<option>10<option>11<option>12<option>13<option>14<option>15<option>16<option>17<option>18<option>19<option>20<option>21<option>22<option>23<option>24<option>25<option>26<option>27<option>28<option>29<option>30<option>31<option>32<option>33<option>34<option>35<option>36<option>37<option>38<option>39<option>40<option>41<option>42<option>43<option>44<option>45<option>46<option>47<option>48<option>49<option>50<option>51<option>52<option>53<option>54<option>55<option>56<option>57<option>58<option>59<option>60<option>61<option>62<option>63<option>64<option>65<option>66<option>67<option>68<option>69<option>70<option>71<option>72<option>73<option>74<option>75<option>76<option>77<option>78<option>79<option>80<option>81<option>82<option>83<option>84<option>85<option>86<option>87<option>88<option>89<option>90<option>91<option>92<option>93<option>94<option>95<option>96<option>97<option>98<option>99<option>100</select>
</td></tr>
```

```
<tr><td align="center"><font face=arial size=2>Gráfico</font><input type="radio" name="sel[]" value="grafico"></td>
<td align="center"><font face=arial size=2>Tabla</font><input type="radio" name="sel[]" value="tabla"></td></tr>
```

```
<tr><td colspan="2" align="center"><input type="submit" value="Enviar">
<input type="reset" value="Limpiar"></td></tr>
</fieldset>
</table>
</form>
```

```
<hr width=100% size=5>
<table border=1>
<tr>
```

```

<td><font size=1 face=Arial color=Black>Desarrollado
por:</font></td></tr>
<tr><td><A href="mailto:dflores_77@yahoo.com"><Font color="blue"
size=0 face=arial> Daniel Flores</Font></A></tr><td>
<tr><td><A href="mailto:rodrigoebianco@integra.com.sv"><Font
color="blue" size=0 face=arial> Rodrigo
Blanco</font></A></tr><td>
</table>
<A href="mailto:dflores_77@yahoo.com">Email</A>

</BODY>
</html>

```

Lo importante en este código son las variables `diaini`, `mesini`, `anoini`, `diafin`, `mesfin` y `anofin` de la forma, las cuales contienen los valores seleccionados para fecha y temperatura y que son pasadas al *script* `main.php` mediante el método `GET`. Además mediante las casillas de verificación también son enviadas las opciones:

- `todo`: Búsqueda por fecha y temperatura
- `fecha`: búsqueda solo por fecha
- `temp`: búsqueda solo por temperatura
- `grafico`: para mostrar los resultados de forma gráfica
- `tabla`: para mostrar los resultados en tabla html

Una vez seleccionadas las opciones y los valores de búsqueda se presiona el botón **Enviar** para ejecutar el script `main.php`, será este el encargado de verificar que los datos y las opciones de búsqueda sean válidos.

También en la página `index.html` se ha incluido un hipere enlace hacia la página `borrar.html`, está restringida para el administrador y sirve para seleccionar una fecha límite para borrar los datos muy antiguos de la base de datos. Todos los datos con fecha igual o anterior a la fecha seleccionada serán borrados de la base de datos.

La fecha seleccionada se envía al script `borrar.php` en el directorio que es el que se encarga de conectarse a la base de datos y borrar los datos. Ambos archivos deben ser colocados en un directorio llamado `temp`, dentro del directorio raíz de Apache.

Para restringir el acceso a la página es necesario generar otro archivo de contraseñas, tal y como se mostro anteriormente, el nombre de usuario será `mysqladmin` y la contraseña `mysql`:

```
#/usr/sbin/htpasswd -c /var/www/html/password/borrar mysqladmin
New password: mysql
Re-type new password: mysql
Adding password for user mysqladmin
```

También es necesario crear un archivo `.htaccess` y colocarlo en el directorio `/var/www/html/temp`:

```
AuthType Basic
AuthName "Acceso Restringido"
AuthUserFile /var/www/html/password/borrar
Require user mysqladmin
```

El código relacionado a la página `borrar.html` es el siguiente:

```
<html>
<title>Limpiar Base de Datos</title>
<body bgcolor="#CDCDCD">
<center>
<marquee bgcolor="#006699" behavior="alternate"
direction="right">
<b><font face="arial" color="#FFFFCC" size="5">Bienvenidos a la
Página de Monitoreo de Temperatura de la UDB</font></b>
</marquee>
</center>
<hr width=100% size=5>
<center>
<blockquote><blockquote><blockquote><p align="justify"><FONT
FACE="Tahoma" color="#211E60"><b>
```

Seleccione la fecha, todos los registros anteriores a ésta serán borrados de la base de datos y no hay forma de volverlos a recuperar.

Si no está seguro no continúe.

</b></FONT></p></blockquote></blockquote></blockquote>

<hr width=100% size=5>

</center>

<p>Seleccione los parámetros de búsqueda</p>

<table border="1">

<form action="borrar.php" method="GET">

<tr><td><font face=arial size=2>Fecha Inicio(dd-mm-aaaa)</font></td><td><select name="diaini[]">

<option>1<option>2<option>3<option>4<option>5<option>6<option>7<option>8<option>9<option>10<option>11<option>12<option>13<option>14<option>15<option>16<option>17<option>18<option>19<option>20<option>21<option>22<option>23<option>24<option>25<option>26<option>27<option>28<option>29<option>30<option>31</select>

<select name="mesini[]">

<option>1<option>2<option>3<option>4<option>5<option>6<option>7<option>8<option>9<option>10<option>11<option>12</select>

<select name="anoini[]">

<option>2005<option>2006<option>2007<option>2008<option>2009<option>2010</select>

</td></tr>

<tr><td colspan=2 align=center><input type="submit"

value="Borrar"> <input type="reset" value="Limpiar"></td></tr>

</form>

</table>

</html>

## g. Instalación del Servidor de Base de Datos MySQL

Para la instalación del servidor de Base de Datos MySQL una vez más utilizaremos la versión incluida con la distribución Linux Mandrake por lo que nuevamente acudiremos al administrador de paquetes *RpmDrake*.

Una vez abierto el administrador ingresamos la palabra clave *mysql* y seleccionamos los paquetes:

- MySQL-4.0.18-1mdk: Paquete correspondiente al servidor
- MySQL-bench-4.0.18-1mdk: Paquete correspondiente a las pruebas de rendimiento en el servidor
- MySQL-client-4.0.18-1mdk: Cliente para conectarse al servidor
- MySQL-common-4.0.18-1mdk: Librerías Comunes
- libmysql12-4.0.18-1mdk: Librería necesaria para que los lenguajes de programación tradicionales como C++ puedan conectarse al servidor de Base de Datos.
- libmysql12devel-4.0.18-1mdk: Librerías de desarrollo necesarias para la creación de clientes MySQL.

Seleccionados los paquetes se procede a instalarlos, esperamos a que termine la instalación y salimos de *RpmDrake*. El servicio es iniciado cuando es instalado así que estamos listos para conectarnos al servidor.

Desde un Terminal de comandos introducimos lo siguiente:

```
$mysql -u root
```

Si el servidor corre sin problemas deberíamos conectarnos y aparecerá en pantalla:

```
mysql>
```

## **h. Creación de Base de Datos Temperatura**

La base de datos que utilizaremos para guardar los datos leídos desde el puerto serial se llamará **temperatura** y contendrá una única tabla llamada **Temperatura** la cual contendrá los siguientes campos:

- ID: Llave primara que servirá como un índice.
- Temperatura: En este campo se almacenará el dato leído del puerto
- Fecha: Fecha en que se leyó el dato desde el puerto.
- Hora: Hora en la que se leyó el dato desde el puerto.

Desde el *prompt* de MySQL introducir la siguiente secuencia de comandos:

```
mysql>CREATE DATABASE temperatura;
mysql>USE temperatura;
mysql>CREATE TABLE Temperatura (
    ->ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ->Temperatura INT NOT NULL,
    ->Fecha DATE NOT NULL,
    ->Hora TIME NOT NULL,
    ->);
```

Con esta simple secuencia de comandos nuestra base de datos ha sido creada y está lista para ser usada para almacenar los datos leídos desde el puerto serial y para ser consultada por los *script* que generará el gráfico dinámico.

Es excelente práctica establecer una contraseña para conectarse al servidor de base de datos, esto puede realizarse al ingresar los siguientes comandos:

```
mysql>SET PASSWORD FOR root@localhost=PASSWORD("new password");
Query OK, 0 rows affected (0.00 sec)
mysql>FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

El primer comando establece la contraseña para conectarse desde la máquina donde está instalado el servidor *MySQL*. También puede establecerse una contraseña para conectarse remotamente ejecutando el siguiente comando a continuación:

```
mysql>SET PASSWORD FOR root@"%=PASSWORD("new password");
Query OK, 0 rows affected (0.00 sec)
```

Una vez hecho esto desconectarse del servidor.

Para probar la configuración, desde el Terminal de Comandos ejecutar:

```
mysqladmin -u root -p status
```

Se pedirá la contraseña recién creada, si ésta es correcta entonces se imprimirán en pantalla algunos mensajes del estado del servidor, de lo contrario aparecerá un mensaje de error.

## **i. Desarrollo de Scripts para la construcción de Gráficos Dinámicos**

El primer *script* que desarrollaremos será `graph.php` por ser más sencillo. Cabe recordar que éste es llamado desde la página `dina.html` para la generación del gráfico y que cada vez que la página se recarga se ejecuta nuevamente el *script* y se genera un nuevo gráfico.

El diagrama de flujo para este se muestra en la figura 3.27:

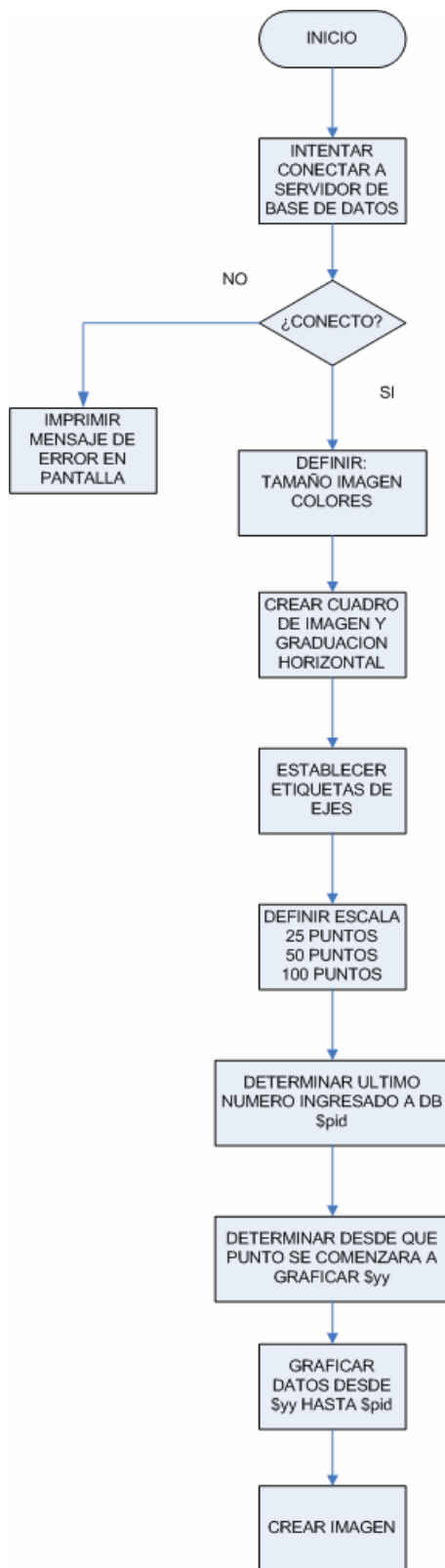


Figura 3. 27 Diagrama de Flujo para graph.php



El código de graph.php asociado a l diagrama de flujo es el siguiente:

```
<?php
$user="root";
$pass="dfloresurbina";
$db="temperatura";
$link=mysql_connect("localhost",$user,$pass);//Identificador de
Conexión
if (! $link)
die ("No se pudo conectar al Servidor MySQL"); //mensaje de
error si no conecta a la base de datos
mysql_select_db($db,$link) //selecciona la base de datos
or die("No se pudo abrir $db: ".mysql_error()); //error si no
existe la base de datos

//Graficar
header("Content-type:image/png");
$imgWidht = 600;
$imgHeight = 600;
$image = imagecreate($imgWidht,$imgHeight);
$white=imagecolorallocate($image,255,255,255);
$red=imagecolorallocate($image,220,0,0);
$green=imagecolorallocate($image,0,255,0);
$blue=imagecolorallocate($image,0,80,200);
$grey=imagecolorallocate($image,192,192,192);
$lil=imagecolorallocate($image,0,0,0);
$orange=imagecolorallocate($image,200,120,0);
//Crear Bordes Externos

imagerectangle($image,0,0,599,599,$grey);
imagerectangle($image,99,99,499,499,$grey);
//Establecimiento de Regiones
imageline($image,99,232,499,232,$red);
imageline($image,99,365,499,365,$blue);
imagefill($image,100,100,$red);
imagefill($image,100,450,$blue);
imagefill($image,100,350,$lil);
imagefill($image,1,1,$grey);
//Crear Cuadrícula
for ($i=1;$i<10;$i++){
//imageline($image,$i*50,0,$i*50,500,$orange); //lineas
verticales
imageline($image,99,(99+$i*40),499,(99+$i*40),$orange);
//lineas horizontales
}
//Leyendas del Gráfico
$font="/usr/java/jre1.5.0_01/lib/fonts/LucidaSansRegular.tt
f";
imageTTFtext($image,30,0,50,50,$lil,$font,"Gráfico de
Temperatura");
imageTTFtext($image,15,0,400,520,$lil,$font,"Tiempo
(seg)");
imageTTFtext($image,25,90,40,420,$lil,$font,"Temperatura
(°C)");
imageTTFtext($image,20,0,60,560,$lil,$font,"El último valor
registrado es:$alea °C");
//Etiquetas de Ejes
```

```

        for ($i=1;$i<10;$i++){
            imageTTFtext($image,15,0,60,(99+$i*40),$lil,$font,100-
$i*10);
        }

        $div=(int)(400/50); //Esta es la escala (50 puntos por
pantalla)

        $result=mysql_query("select Temperatura from Temperatura");
        $num_rows=mysql_num_rows($result);
        $pid=$num_rows;

        //Se graficaran hasta los 45 primeros datos de la base de
datos
        if ($pid<=45){
            while ($a_row = mysql_fetch_row($result))
            {
                foreach($a_row as $field)
                $values[]=$field;
            }

            //Lazo para graficar linea de comportamiento de la
temperatura
            for ($i=0;$i<($pid-1);$i++){

                imageline($image,99+($i)*$div,(500-
$values[$i]*4),99+($i+1)*$div,(500-$values[$i+1]*4),$green);
            }

            imagepng($image);
            imagedestroy($image);
        }
        else { //Despues de 45 datos la geafica se
ira desplazando a la izquierda
            while ($a_row = mysql_fetch_row($result))
            {
                foreach($a_row as $field)
                $values[]=$field;
            }

            //Lazo para graficar la línea de comportamiento de la
temperatura

            for ($i=($pid-45);$i<($pid-1);$i++){

                imageline($image,99+($i-($pid-45))*$div,(500-
$values[$i]*4),99+($i+1-($pid-45))*$div,(500-
$values[$i+1]*4),$green);

            }

            imagepng($image);
            imagedestroy($image);
        }
    }
}
?>

```

Ahora falta el desarrollo de `main.php` que es el script encargado de tomar las variables enviadas desde `stat.html` y mostrar ya sea de forma gráfica o en tabla html los resultados de la búsqueda.

Se presenta el diagrama de flujo en varias partes:

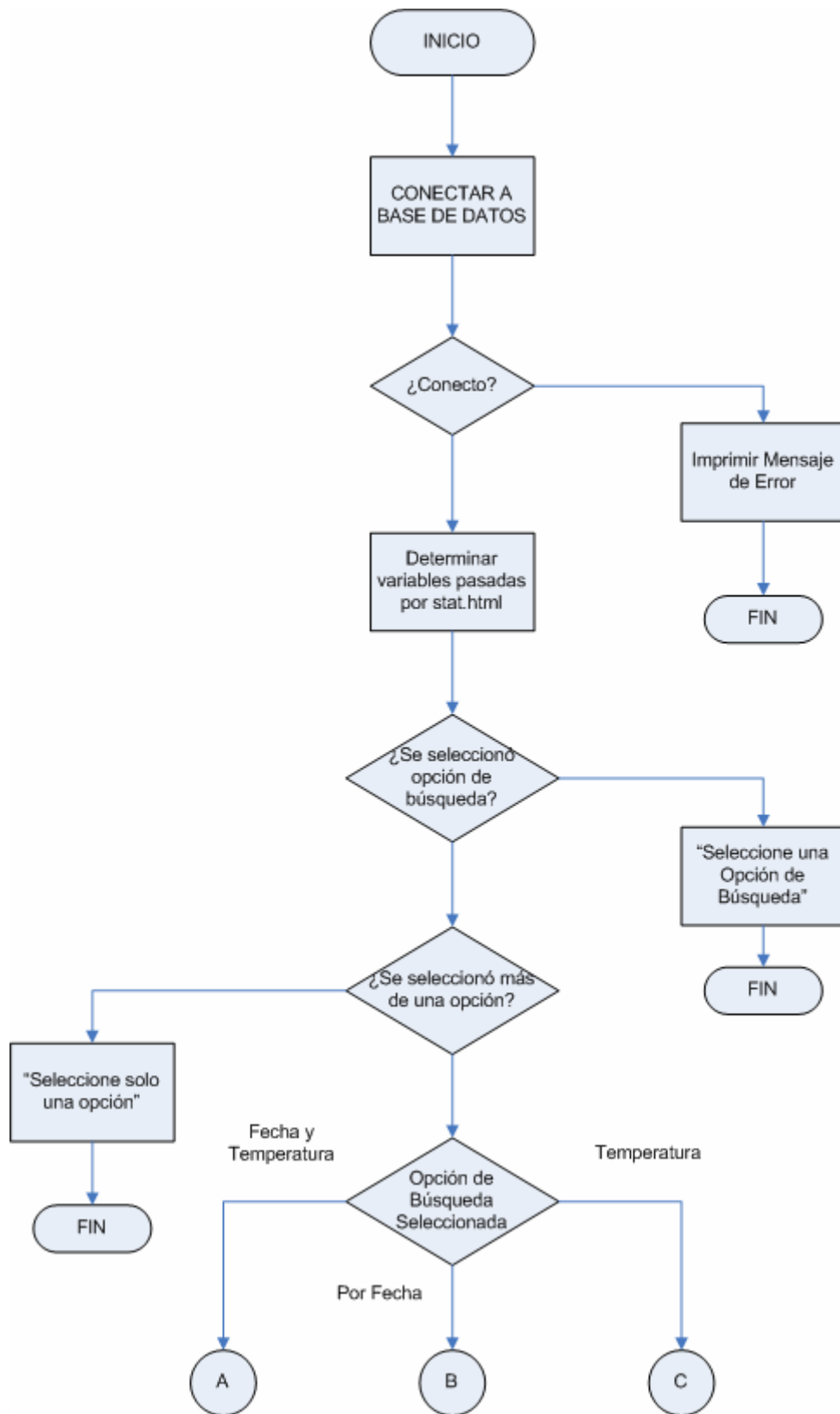


Figura 3. 28: Diagrama de Flujo main.php Parte A

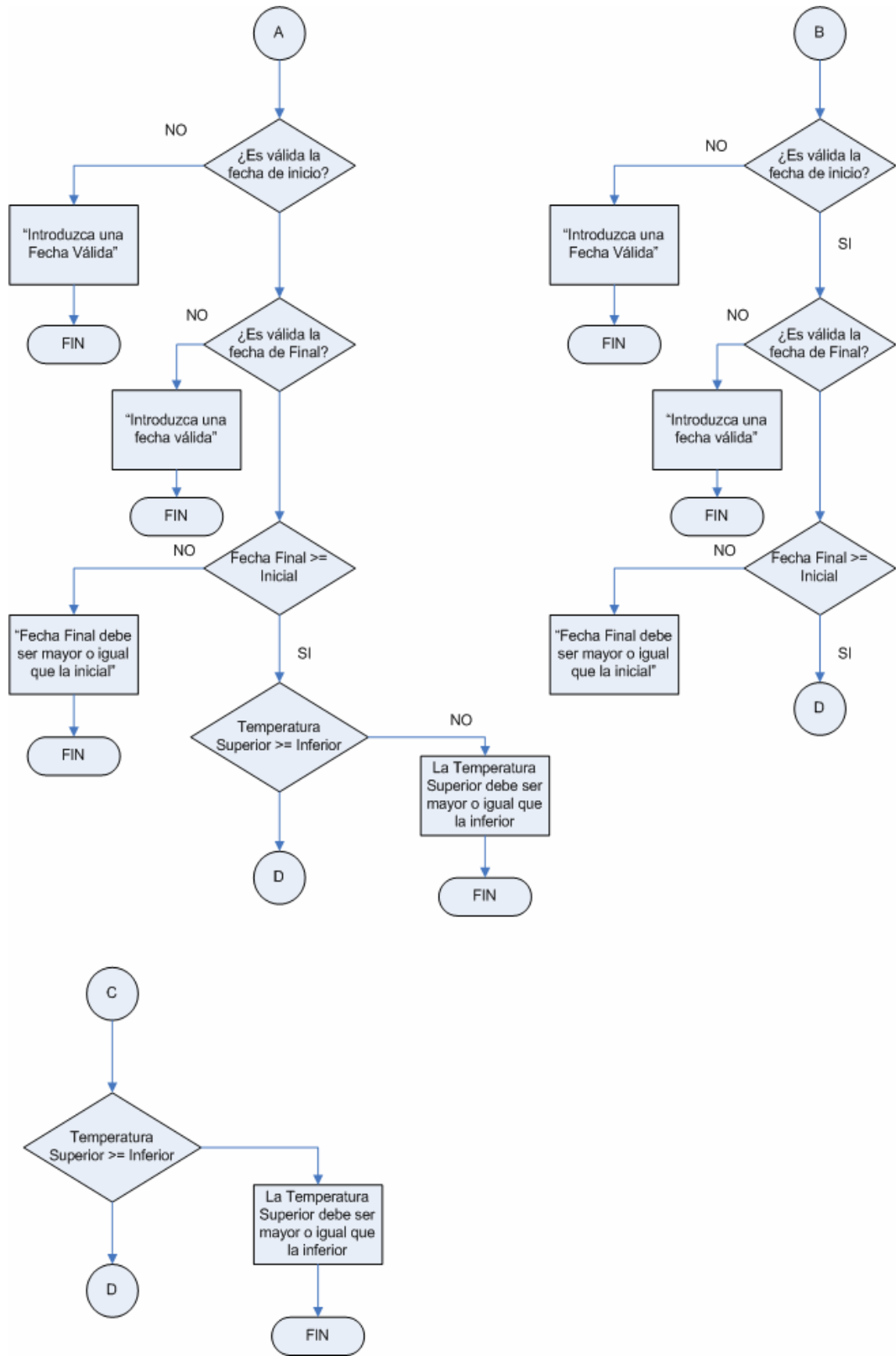


Figura 3. 29: Diagrama de Flujo Parte main.php Parte B

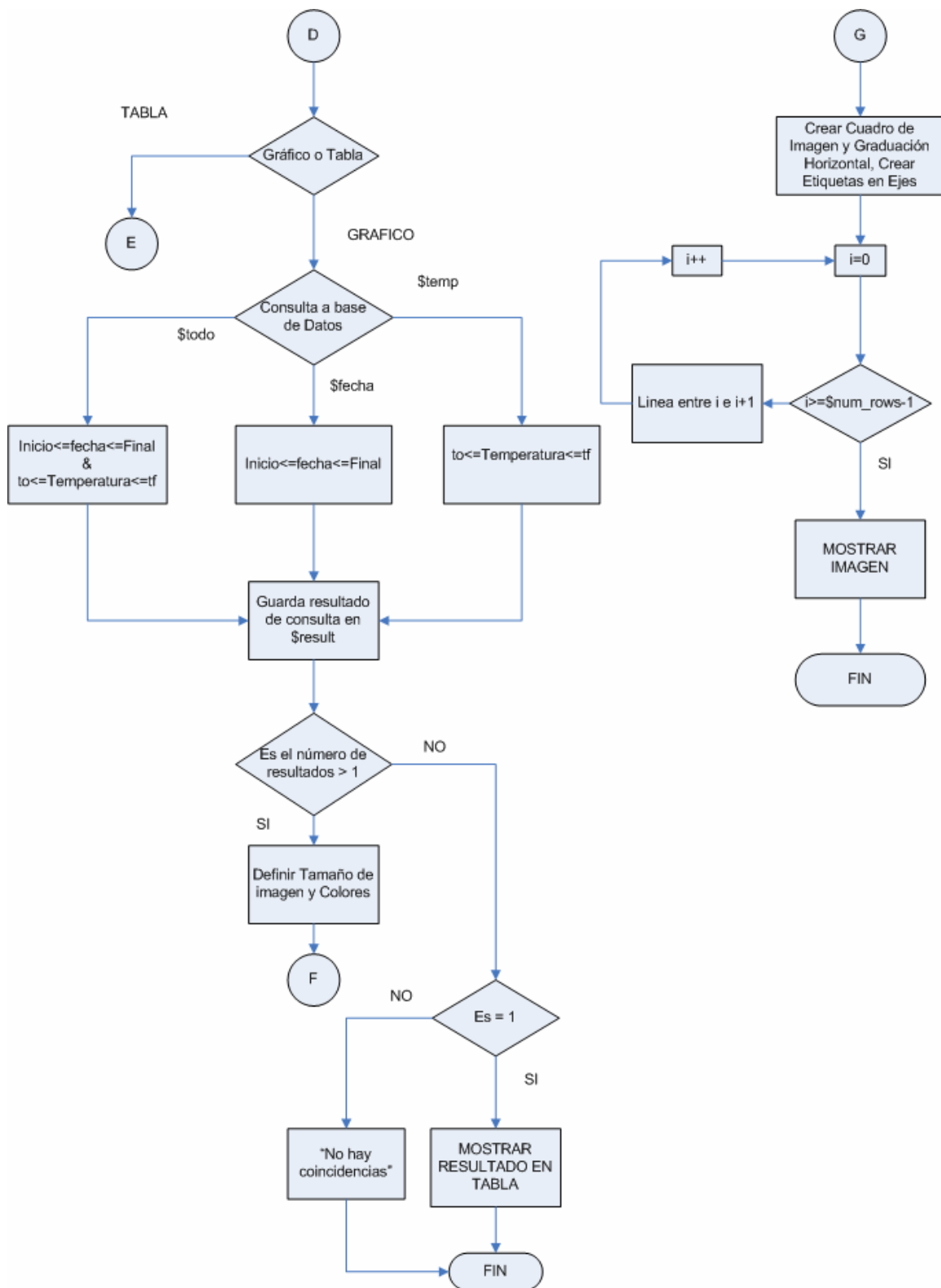


Figura 3. 30: Diagrama de Flujo main.php parte C

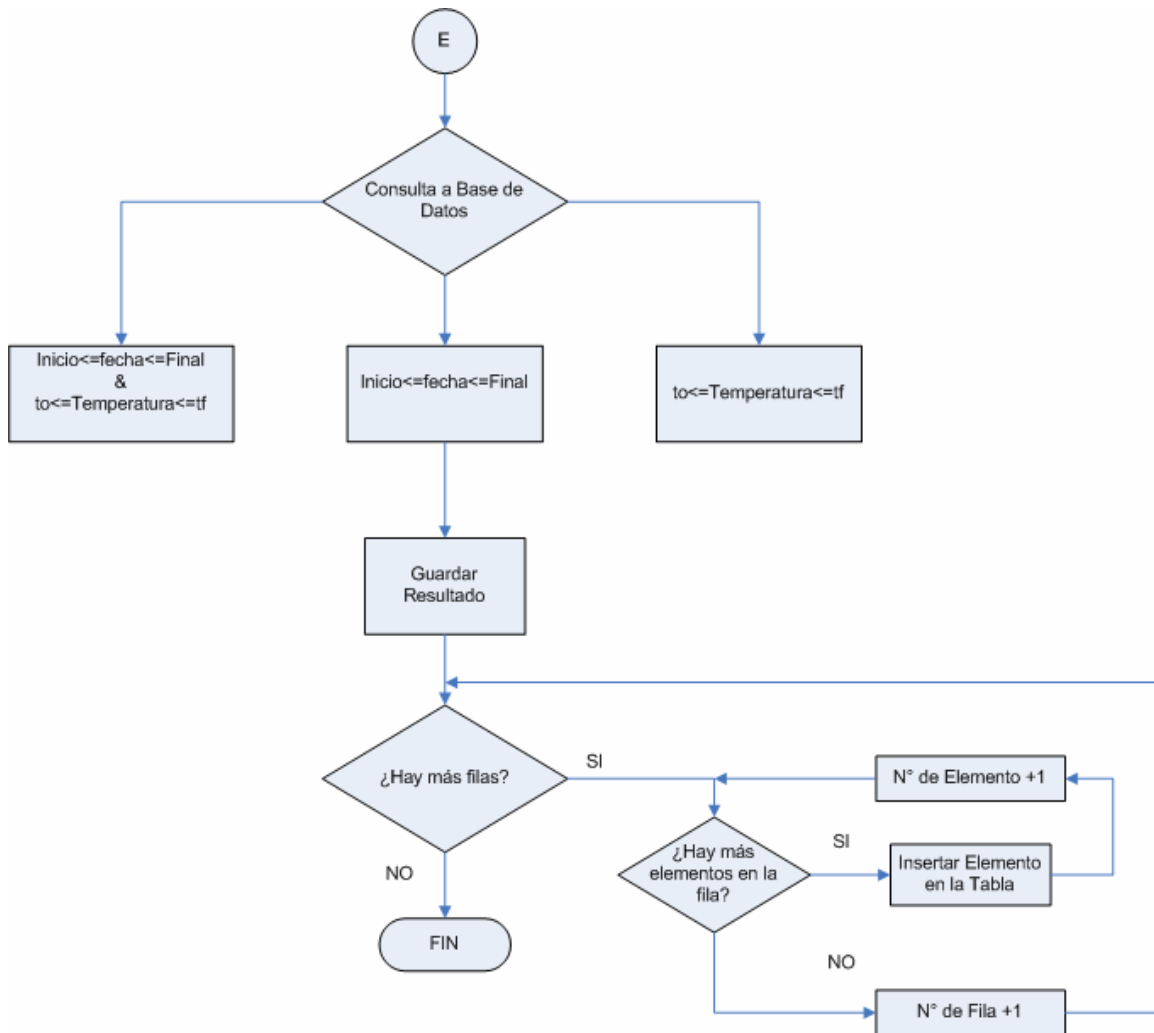


Figura 3. 31: Diagrama de flujo main.php Parte D

El código asociado a este diagrama de flujo se muestra a continuación:

```

<?php
//Conectar a la base de Datos
$user="root";
$pass="dfloresurbina";
$db="temperatura";
$link=mysql_connect("localhost",$user,$pass);//Identificador de
Conexión
if (! $link)
die ("No se pudo conectar al Servidor MySQL"); //mensaje de
error si no conecta a la base de datos
mysql_select_db($db,$link) //selecciona la base de datos
or die("No se pudo abrir $db: ".mysql_error()); //error si no
existe la base de datos

$sep="-";
$col=":";
  
```

```

$mesin=$mesini[0];
$diain=$diaini[0];
$anoin=$anoini[0];
$mesfi=$mesfin[0];
$diafi=$diafin[0];
$anofi=$anofin[0];
$to=$to[0];
$tf=$tf[0];
$horin=$horini[0];
$minin=$minini[0];
$secin=$secini[0];
$horfi=$horfin[0];
$minfi=$minfin[0];
$secfi=$secfin[0];

//Determinar opción de Búsqueda

$regi=$reg[0];

switch ($regi)
{
case "todo":
$busq="todo";
break;
case "fecha":
$busq="fecha";
break;
case "temp":
$busq="temp";
break;

default:
die ("Seleccione una opción de Búsqueda");
}

//Determinar Opción de Visualización

$seli=$sel[0];

switch ($seli)
{
case "tabla";
$esc="tabla";
break;
case "grafico";
$esc="grafico";
break;

default:
die ("Seleccione un Método de Visualización: Tabla o Gráfico");
}

//Determinar que se han llenado adecuadamente los campos
//Verificar que se han llenado Correctamente los cuatro campos
if ($regi == "todo")
{

```



```

//Validación de Fecha de Inicio
if (!checkdate($mesin,$diain,$anoin))
{
Die ("Introduzca una fecha de Inicio Válida");
}
//Validación de Fecha Final
if (!checkdate($mesfi,$diafi,$anofi))
{
Die ("Introduzca una fecha Final Válida");
}

//Verificar que la fecha final es mayor o igual que la inicial
$tsi=mktime($horin,$minin,$secin,$mesin,$diain,$anoin);
$tsf=mktime($horfi,$minfi,$secfi,$mesfi,$diafi,$anofi);
$dif=$tsf-$tsi;

if ($dif<0)
{
Die("La fecha de inicio es mayor que la final");
}
//Validación de la Temperatura
if($tf<$to){
Die("La temperatura final debe ser mayor o igual que la
inicial");
}
}

//Verificar que se han llenado correctamente los datos de la
Fecha
if ($regi =="fecha")
{
//Validación de Fecha de Inicio
if (!checkdate($mesin,$diain,$anoin))
{
Die ("Introduzca una fecha de Inicio Válida");
}
//Validación de Fecha Final
if (!checkdate($mesfi,$diafi,$anofi))
{
Die ("Introduzca una fecha Final Válida");
}

//Verificar que la fecha final es mayor o igual que la inicial
$tsi=mktime($horin,$minin,$secin,$mesin,$diain,$anoin);
$tsf=mktime($horfi,$minfi,$secfi,$mesfi,$diafi,$anofi);
$dif=$tsf-$tsi;

if ($dif<0)
{
Die("La fecha de inicio es mayor que la final");
}
}

//Validación de Temperatura
if ($busq == "temp")
{
if($tf<$to){

```

```

Die("La temperatura final debe ser mayor o igual que la
inicial");
}
}

//Creación de strings correspondiente a las fechas y horas de
inicio y finalización de búsqueda
$inicio="$anoin$sep$mesin$sep$diain";
$fin="$anofi$sep$mesfi$sep$diafi";

$stin="$horin$col$minin$colsecin";
$stfin="$horfi$col$secfi$col$secfi";

if ($esc == "grafico"){
if ($busq == "todo"){ //este lazo determina el tipo de búsqueda
que se ha seleccionado
$result = mysql_query("Select Temperatura from Temperatura
where
Fecha >='$inicio' and Fecha <='$fin' and Hora >= '$tin' and
Hora <='$tfin'
and Temperatura >='$to' and Temperatura <='$tf'");
}
elseif($busq == "fecha")
{
$result=mysql_query("select Temperatura from Temperatura
where
Fecha >='$inicio' and Fecha <='$fin' and Hora >= '$tin' and
Hora <='$tfin'");
}
else
{
$result=mysql_query("select Temperatura from Temperatura
where
Temperatura >='$to' and Temperatura <='$tf'");
}
$num_rows=mysql_num_rows($result);

if ($num_rows>1) //si hay más de una coincidencia
graficar
{
while ($a_row = mysql_fetch_row($result))
{
foreach($a_row as $field)
$values[]=$field;
}
$div=(500/($num_rows-1));

/*if ($div<1){
$div=1;
}*/
}
}

```

```

//Graficar
header("Content-type:image/png");
$imgWidht = 700;
$imgHeight = 700;
$image = imagecreate($imgWidht,$imgHeight);
$white=imagecolorallocate($image,255,255,255);
$red=imagecolorallocate($image,220,0,0);
$green=imagecolorallocate($image,0,255,0);
$blue=imagecolorallocate($image,0,80,200);
$grey=imagecolorallocate($image,192,192,192);
$lil=imagecolorallocate($image,0,0,0);
$orange=imagecolorallocate($image,200,120,0);
//Crear Bordes Externos

imagerectangle($image,0,0,699,699,$grey);
imagerectangle($image,99,99,599,599,$grey);
//Establecimiento de Regiones
imageline($image,99,249,599,249,$red);
imageline($image,99,449,599,449,$blue);
imagefill($image,100,100,$red);
imagefill($image,100,450,$blue);
imagefill($image,100,350,$lil);
imagefill($image,1,1,$grey);
//Crear Cuadrícula
for ($i=1;$i<10;$i++){
//imageline($image,$i*50,0,$i*50,500,$orange);
//lineas verticales

    imageline($image,99,(99+$i*50),599,(99+$i*50),$orange);
//lineas horizontales
}
$font="/usr/java/jre1.5.0_01/lib/fonts/LucidaSansRegular.ttf";

imageTTFtext($image,40,0,25,50,$lil,$font,"Gráfico de
Temperatura");

imageTTFtext($image,30,90,40,520,$lil,$font,"Temperatura (°C)");

imageTTFtext($image,30,0,430,640,$lil,$font,"Tiempo");
//Etiquetas de Ejes
for ($i=1;$i<10;$i++){

imageTTFtext($image,15,0,60,(99+$i*50),$lil,$font,100-$i*10);
}

//Crea línea de Comportamiento de la Temperatura
for ($i=0;$i<($num_rows-1);$i++){
imageline($image,99+$i*$div,(600-
$values[$i]*5),99+($i+1)*$div,(600-$values[$i+1]*5),$green);

}
imageTTFtext($image,30,0,100,640,$lil,$font,"$i
Registros");
imagepng($image);
imagedestroy($image);

}

```

```

elseif ($num_rows==1) //si solo hay una
coincidencia mostrar tabla
{
?>
<body>
<Strong><font color="black">Solo hay una
coincidencia</font></strong>
<?php
    if ($busq == "todo"){ //este lazo determina el tipo
de busqueda que se ha seleccionado

$result = mysql_query("Select Temperatura, Fecha, Hora from
Temperatura where
    Fecha >='$inicio' and Fecha <='$fin' and Hora >=
'$tin' and Hora <='$tfin' and
    Temperatura >='$to' and Temperatura <='$tf'");
    }
elseif($busq == "fecha")
{
$result=mysql_query("select Temperatura, Fecha, Hora
from Temperatura where
    Fecha >='$inicio' and Fecha <='$fin' Hora >= '$tin'
and Hora <='$tfin'");
    }
else
{
$result=mysql_query("select Temperatura, Fecha, Hora
from Temperatura where
    Temperatura >='$to' and Temperatura <='$tf'");
    }
Print "<center>";
Print "<table border=0 width=40%>\n";
Print "<tr>\n";
//Print "\t<td bgcolor=blue align=center><b><font
face=arial color=white size=3>ID</font></b></td>\n";
Print "\t<td bgcolor=blue align=center><b><font
face=arial color=white size=3>Temperatura</font></b></td>\n";
Print "\t<td bgcolor=blue align=center><b><font
face=arial color=white size=3>Fecha</font></b></td>\n";
Print "\t<td bgcolor=blue align=center><b><font
face=arial color=white size=3>Hora</font></b></td>\n";
Print "</tr>\n";
while ($a_row = mysql_fetch_row($result))
{
Print "<tr>\n";
foreach($a_row as $field)
Print "\t<td align=center><b><font face=arial size=0
color=#e3edff>$field</font></b></td>\n";
Print "</tr>\n";
}
Print "</table>\n";
Print "</center>";
}
else{
?>

```

```

        <body bgcolor="#CDCDCD"><strong><H2><font color="red">No
existen coincidencias en la base de Datos para esta
busqueda</font></H2></strong></body>
        <?php
        }
    }
elseif ($esc == "tabla"){          //Genera la Tabla de
Busquedas

        if ($busq == "todo"){      //este lazo determina el tipo de
busqueda que se ha seleccionado
            $result = mysql_query("Select Temperatura, Fecha, Hora
from Temperatura where
            Fecha >=' $inicio' and Fecha <=' $fin' and Hora >=
'$tin' and Hora <=' $tfin'
            and Temperatura >=' $to' and Temperatura <=' $tf'");
            }
            elseif($busq == "fecha")
            {
                $result=mysql_query("select Temperatura, Fecha, Hora
from Temperatura where
                Fecha >=' $inicio' and Fecha <=' $fin' and Hora >=
'$tin' and Hora <=' $tfin'");
            }
            else
            {
                $result=mysql_query("select Temperatura, Fecha, Hora
from Temperatura where
                Temperatura >=' $to' and Temperatura <=' $tf'");
            }

            $num_rows=mysql_num_rows($result);
            Print "<A href=index.html>Inicio</A>\n";
            Print "<A href=stat.html>Atras</A>\n<br>";

            Print "Hay $num_rows coincidencias actualmente";

            Print "<center>";
            Print "<table border=0 width=40%>\n";
            Print "<tr>\n";
            //Print "\t<td bgcolor=blue align=center><b><font
face=arial size=3 color=white>ID</font></b></td>\n";
            Print "\t<td bgcolor=blue align=center><b><font
face=arial size=3 color=white>Temperatura</font></b></td>\n";
            Print "\t<td bgcolor=blue align=center><b><font
face=arial size=3 color=white>Fecha</font></b></td>\n";
            Print "\t<td bgcolor=blue align=center><b><font
face=arial size=3 color=white>Hora</font></b></td>\n";
            Print "</tr>\n";

            $counter=1;

//Este lazo while imprime en la tabla cada una de las filas y los
datos de cada fila contenidos en $result
            while ($a_row = mysql_fetch_row($result))
            {

```

```

//Este lazo if determina el color de la fila, si $counter es par
//blanco, si es impar celeste
    $y=($counter/2);
    $z=(int)($y);
    if ($y>$z){
        $color="white";
    }
    else {
        $color="#e3edff";
    }

    $counter=$counter+1;

    Print "<tr bgcolor=$color>\n";
    foreach($a_row as $field)
        print "\t<td align=center><b><font face=arial
size=2>$field</font></b></td>\n";
    print "</tr>\n";
}
Print "</table>\n";
}
Print "</center>";

Print "<A href=index.html>Inicio</A>\n";
Print "<A href=stat.html>Atras</A>\n";
?>

```

Ambos *scripts* junto con las páginas html deben colocarse en el directorio raíz para documentos de *Apache*, que para el caso sería `/var/www/html/`.

Por último mostraremos el código del script `borrar.php` para borrar los datos antiguos de la base de datos, el código es bastante sencillo y se muestra a continuación:

```

<?php

$user="root";
$pass="dfloresurbina";
$db="temperatura";
$link=mysql_connect("localhost",$user,$pass);//Identificador de
Conexión
if (! $link)
die ("No se pudo conectar al Servidor MySQL"); //mensaje de
error si no conecta a la base de datos
mysql_select_db($db,$link) //selecciona la base de datos

```

```

or die("No se pudo abrir $db: ".mysql_error()); //error si no
existe la base de datos
//Se obtiene la fecha seleccionada
$mesin=$mesini[0];
$anoin=$anoini[0];
$diain=$diaini[0];

$sep="-";
//Se verifica que sea válida
if (!checkdate($mesin,$diain,$anoin))
{
Die ("Introduzca una fecha Válida");
}

$Fec="$anoin$sep$mesin$sep$diain";
//Se envía el comando para borrar de la base de datos
$query="Delete from Temperatura where Fecha <= '$Fec'";
mysql_query($query,$link)
or die("No se pudo realizar la acción solicitada");

$afecto=mysql_affected_rows($link);
Print "Se borraron $afecto registros\n<br>";

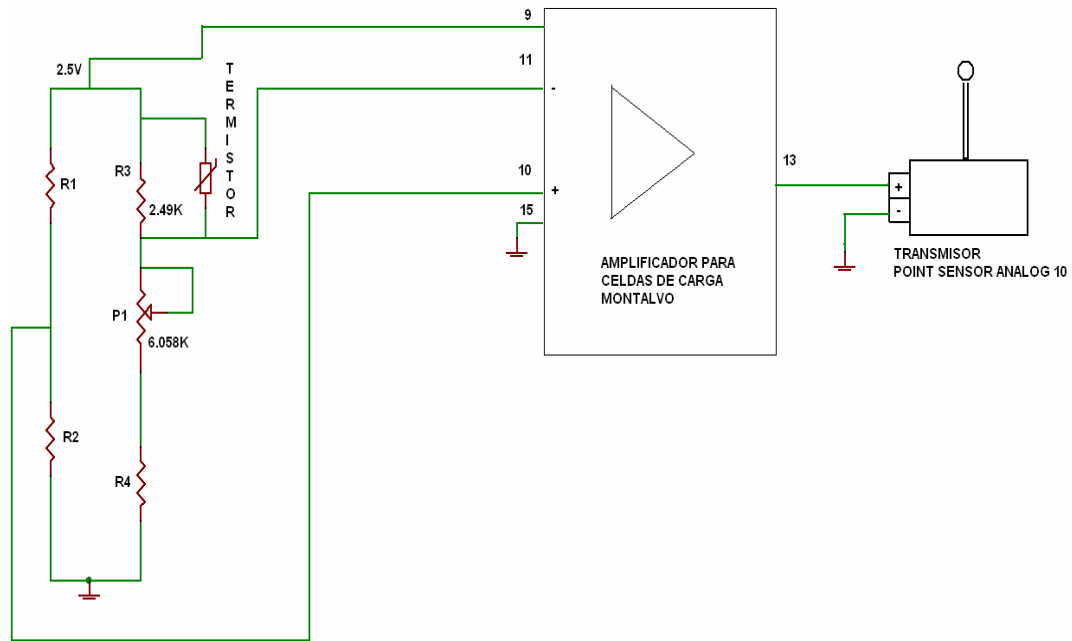
Print "<A href=/index.html>Inicio</A>\n";
Print "<A href=borrar.html>Atras</A>\n";
?>

```

## PARTE IV

### a. Implementación del Sistema de Adquisición y Procesamiento de la variable analógica

El circuito a utilizar para adquirir y amplificar la temperatura será el que se presenta en la figura 4.1.



**Figura 4. 1:** Circuito para adquisición de la variable analógica

El circuito es bastante sencillo al igual que su funcionamiento. El puente de resistencias se ha ajustado para que su salida varíe linealmente de 0 a 200mV. Como se explicó en la PARTE I, el termistor no es un dispositivo lineal, así que para poder linealizarlo es necesario colocar en paralelo una resistencia y colocar en serie a este arreglo un potenciómetro para un mejor ajuste. La salida del puente de resistencias es enviada a la entrada diferencial del Amplificador Montalvo<sub>[Anexo 3]</sub>, la salida de éste varía linealmente con la entrada. Además la salida se ajustó para que brinde un



voltaje entre 0 y 10V ya que debe estar en el rango de operación de la entrada del transmisor inalámbrico.

Se logró ajustar la salida para que varíe 0.1V por cada grado centígrado. O sea que si se tienen 2.0V en la salida del amplificador esto indicará que la temperatura es 20°C.

El transmisor envía un paquete de datos [Anexo 1] al receptor, este a su vez lo envía al puerto serie de la computadora por medio de una interfaz RS232.

Es a partir de la descripción del paquete de datos que se escribió un programa para leer el dato de interés del puerto serie de la PC. El programa como se mencionó anteriormente está escrito en C++, y leerá de forma periódica el buffer del puerto serie, extraerá del paquete de datos solo el campo correspondiente al valor hexadecimal de la temperatura representado con 12 bits de precisión. Pero como este valor no representa el valor real, el programa lo convertirá a un valor decimal de acuerdo a una sencilla regla de tres.

El valor máximo que puede tomar a la entrada el transmisor es 9.99V o FFF en hexadecimal, si convertimos este valor a decimal obtendremos 4095. Así que para obtener el valor de temperatura aplicamos la siguiente fórmula:

$$Temperatura = \frac{(Dato)(9.99)}{4095} \quad \text{(Ecuación 4.1)}$$

Donde:

Dato es el valor leído del puerto expresado como un número decimal.

El programa además obtendrá la fecha y hora de la lectura, y escribirá los tres valores en la tabla Temperatura de la base de datos creada anteriormente.

## b. Desarrollo del Programa para lectura del puerto

A continuación se presenta el código fuente del programa escrito en C++, este debe ingresarse en cualquier editor de texto y guardarse con la extensión .cpp. En nuestro caso el archivo se llama puerto.cpp

```
#ifndef HAVE_CONFIG_H
#include <config.h>
#endif

#include <iostream>      /*Definición de libreria de entrada/salida */
#include <stdio.h>       /*Definición de Standard de Entrada/Salida */
#include <string.h>      /*Definición de funciones de Cadenas de
caracteres */

#include <unistd.h>      /*Definiciones de funciones Standard de UNIX */
#include <fcntl.h>       /* Definición de Control de Archivos */
#include <errno.h>       /* Definición de Números de Error */
#include <termios.h>     /*Definiciones de Control de Terminal POSIX */
#include <iomanip.h>     /**/
#include <ctype.h>       /**/
#include <mysql/mysql.h> /*Definiciones para control de base de datos*/
#include <sys/signal.h> /*Definiciones para librería de puerto*/
#include <sys/types.h>   /*Definición para librería de puerto*/

#define BAUDRATE B19200 /*Definicion de constante para velocidad de
transferencia */

#define MODEMDEVICE "/dev/ttyS0" /*Definición de constate para
utilizacion de puerto */

#define _POSIX_SOURCE 1 /*Definición de constante para
fuente de compilante POSIX */

#define FALSE 0 /*Definición de constante para
utilizacion del puerto*/
#define TRUE 1 /*Definición de constante para
utilizacion del puerto*/

using namespace std;

//declaracionde variables
volatile int STOP=FALSE; /* Definición de variable como entero
volatil*/
int wait_flag=TRUE; /* Verdadero mientras no haya señal recibida*/
MYSQL mysql; /*Conexion estatica con Mysql*/

//declaración de funciones
unsigned long hextodec(const char* aNumber);/*Convierte la cadena de
datos hexadecimales a un numero decimal */
float interpretar(string str);
void signal_handler_IO (int status); /* Definicion de funcion de
manejador de señal para la entrada */
```

```

void exitUsage(const char* command);          /*Definicion de funcion */
void connectDB();                            /*Pasa a subrutina para conectarse a base
de datos*/
void closeConnection();                      /*Pasa a subrutina para Cerrar la
coneccion*/
int atohd(const char* aNumber);              /*Se convierte la cadena de
caracteres a un numero */

/*Convierte la cadena de datos hexadecimales a un numero decimal */
unsigned long hextodec(const char* aNumber) {
    return (int)strtoul(aNumber, (char**)NULL, 16);
}

//TOMA EL DATO QUE DEVUELVE READ Y LO PARSEA
/**
str recibe el string que se lee del puerto y lo convierte a decimal
o sea devuelve la temperatura
*/
float interpretar(string str){
    float temp=0.0;//declaro float
    int length=str.length();                //longitud del parametro de entrada
    unsigned long aValue;                   //declaro long sin signo
    const char* buf = "000";               //declaro string
    string val="";                          //declaro string

    val=str.substr(19, 3);//Se extrae la parte de datos de la cadena
HEX
    cout << "Val (hex): " << val << endl; //imprime valor en
hexadecimal
    buf=val.c_str();                        //son cierto string de hex a char*
para usar la funcion de conversion a decimal

    aValue = hextodec(buf);                 //convierto el valor HEX a DEC
    cout << "Val (dec): " << aValue << endl; //imprimir el valor
decimal leido
    temp=aValue*9.9*10/4095;               /*Operacion para obtener el valor
de temperatura*/
    printf ("Temp: %.1f ", temp); /*Despliega el valor de temperatura
en pantalla*/
    return temp;
}

/*Define la funcion que se insertara en base de datos*/
int insertarRegistro(MYSQL *mysql,float temp)
{
    char my_string[100]; //declaro string que tendra el query para
insertar en BD
    sprintf (my_string, "INSERT INTO Temperatura (Temperatura, Fecha,
Hora) VALUES(%.1f, SYSDATE(), SYSDATE()) ", temp); /*Se inserta en la
base de datos la temperatura, fecha y hora*/
    return mysql_real_query(mysql,my_string,strlen(my_string));
//funcion de la libreria de MySql que ejecuta el query
}

```

```

//ABRIR Y LEER PUERTO
void openPort2(){
    int fd,c, res;                /*Definicion de variables*/
    struct termios oldtio,newtio;  /*Definicion de
variables para configuracion de puerto*/
    struct sigaction saio;        /*Definicion de la señal de
accion*/
    char buf[255];                /*Largo de buffer de
caracteres*/
    fd_set readfs;

    /*Abre el dispositivo para que este sin bloqueo osea que la
lectura se obtiene inmediatamente*/
    fd = open(MODEMDEVICE, O_RDWR | O_NOCTTY | O_NONBLOCK); /*Abre el
puerto*/
    if (fd <0) {perror(MODEMDEVICE); exit(-1); }          /*Retorna -
1 en fd si la coneccion al puerto tiene exito*/

    /*Instala el manejador de señal entrada/salida antes de hacer
el dispositivo asincrono*/
    saio.sa_handler = signal_handler_IO;    /*Define el manejador
de señal entrada/salida*/
    saio.sa_flags = 0;                    /*Configura banderas a cero*/
    saio.sa_restorer = NULL;              /*Configura restaurador*/
    sigaction(SIGIO,&saio,NULL);          /*Configura accion de
manejador de señal SIGIO*/

    fcntl(fd, F_SETOWN, getpid());        /* Permite al proceso recibir
el manejador de señal SIGIO*/
    fcntl(fd, F_SETFL, FASYNC);          /*Hace el descriptor de
archivo asincrono*/

    tcgetattr(fd,&oldtio); /* Guarda la configuracion de puerto
actual */

    /*AJUSTA LA NUEVA CONFIGURACION DEL PUERTO PARA PROCESO DE
ENTRADA CANONICA*/
    newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
/*Configuracion de banderas del puerto*/
    newtio.c_iflag = IGNPAR | ICRNL;
/*Configuracion de banderas del puerto*/
    newtio.c_oflag = 0;
/*Configuracion de banderas del puerto*/
    newtio.c_lflag = ICANON;
/*Configuración de banderas del puerto*/
    newtio.c_cc[VMIN]=1;
/*Configuración de banderas del puerto*/
    newtio.c_cc[VTIME]=0;
/*Configuración de banderas del puerto*/
    tcflush(fd, TCIFLUSH);
/*Drena la entrada antes de volver a leer*/
    tcsetattr(fd,TCSANOW,&newtio);
/*Guarda atributos*/

    //ABRE CONECCION A BASE DE DATOS
connectDB();

```

```

        res=select(fd, &readfs, NULL, NULL, NULL);
/*Espera hasta que el puerto este disponible*/
        printf("Hizo el select: %d\n", res);
/*Imprime confirmacion*/
        while (res=select(fd, &readfs, NULL, NULL, NULL)!=0) {
/*Lazo mientras espera la entrada, para no consumir procesador*/
        printf(".\n");
        usleep(100000); //duerme
        if (wait_flag==FALSE) { /*Despues
de recibir SIGIO, wait_flag=FALSE, La entrada esta disponible y se
puede leer*/
                res = read(fd,buf,255);
                buf[res]=0;
                printf("Buf:%sRes:%d\n", buf, res);
//Imprime el string completo que devuelve el puerto: buf

                if(insertarRegistro(&mysql,interpretar(buf))==0)
/*Inserta registro en base de datos*/
                printf( "Registro agregado\n");
/*Confirmacion de exito en la escritura de la base de datos*/
                else
                printf( "Fallo al tratar de agregar registro: Error: %s\n",
mysql_error(&mysql)); /*Confirmacion de fallo en la escritura en la
base de datos*/

                if (res==1) STOP=TRUE; /* Para el lazo solo si CR estaba en
la entrada*/
                wait_flag = TRUE; /* Espera una nueva entrada */
        }
        }
        tcsetattr(fd,TCSANOW,&oldtio); /*Restaura configuraciones
anteriores del puerto */
        closeConnection(); /*Cierra la coneccion*/
    }

/*****
****
* Manejador de señal, ajusta bandera de espera wait_flag a falso
para
*
* indicar que en el lazo de arriba se han recibido caracteres.
*
****
*****/

void signal_handler_IO (int status)
{
    printf("Señal SIGIO recibida.\n"); /*Confirmacion de
recepcion de senal de manejador SIGIO*/
    wait_flag = FALSE;
}

//conecta a la BD
void connectDB(){
    char record[1000]; /*Buffer de caracteres*/

```

```

    string sql;                                /*Define cadena de datos*/

    if(mysql_init(&mysql)==NULL)                /*Abre la coneccion a la base
de datos*/
    {
        printf("\nFallo al iniciar coneccion a MySQL"); /*Mensaje
de fallo en inicializar coneccion a base de datos*/
        exit(1);
    }

    /*Ahora se puede llamar cualquier funcion de MySQL*/

    if
(!mysql_real_connect(&mysql,"localhost","root","dfloresurbina","tempera
tura",0,NULL,0)) /*Se conecta a la base de datos*/
    {
        printf( "Fallo al tratar de conectarse a MySQL: Error:
%s\n", mysql_error(&mysql)); /*Confirmacion de fallo en la
coneccion a la base de datos*/
        exit(1);
    }

    printf("Coneccion a base de datos exitosa\n"); /*Confirmacion de
coneccion a base de datos exitosa*/
    if(mysql_select_db(&mysql,"temperatura" )==0) /*Se selecciona
base de datos*/
        printf( "Base de datos seleccionada\n");
/*Confirmacion de base de datos seleccionada*/
    else
        printf( "Fallo en la coneccion a la base de datos: Error:
%s\n", mysql_error(&mysql)); /*Confirmacion de fallo en seleccionar
base de datos*/

    }
void closeConnection(){
    mysql_close(&mysql); /*Cierra la coneccion a la base de
datos*/
}

int main(int argc, char *argv[])
{

    openPort2(); /*Abre el puerto de nuevo*/
    return EXIT_SUCCESS;
}

```

### c. Compilación del Código fuente

El código fuente presentado anteriormente debe compilarse para obtener el ejecutable, así que una vez que tenemos el archivo puerto.cpp lo

guardamos en un directorio y como `root` ingresamos a un Terminal y nos ubicamos en el directorio donde está el archivo. Luego ingresamos el siguiente comando en una sola línea:

```
[rodri@servidor UDB]$su
Password:
[root@servidor UDB]#g++ puerto.cpp -o puerto -I /usr/include -L
/usr/lib/mysql -lmysqlclient -lz -Wno-deprecated
```

Esto generará un ejecutable llamado `puerto`.

- `g++`: Compilador de C++ de la GNU
- `puerto.cpp`: archivo con código fuente
- `-o puerto`: archivo ejecutable generado después de la compilación
- `-I /usr/include`: indica la ruta a las librerías de C++
- `-L /usr/lib/mysql`: indica la ruta a la librería `libmysql`
- `-l mysqlclient`: para poder conectarse como cliente al servidor MySQL
- `-Wno-deprecated`: No advertir sobre opciones no aprobadas.

El programa puede ejecutarse ingresando en un Terminal:

```
[root@servidor UDB]#./puerto&
```

Con esto el proyecto está listo para utilizarse. Ahora cuando se ingrese desde un navegador Web, se graficará el valor de temperatura.

Pero lo más conveniente es colocar este archivo junto con los otros ejecutables en el directorio `/usr/bin`; al hacer esto bastará con ingresar `puerto&` en el Terminal, ya que este directorio se encuentra definido en

la variable de entorno `$PATH`, como uno de los directorios que contienen ejecutables de programas de usuario.



## PARTE V

### Resumen de Costos del Proyecto

Elemento	Cantidad	Precio Unitario (US\$)	Total (US\$)
Transmisor Six Point Analog 10	1	149.55	149.55
Receptor Point View	1	156.20	156.20
Sistema Operativo Mandrake Linux 10.0	1	0.00	0.00
Servidor Web Apache	1	0.00	0.00
Servidor Base da Datos MySQL	1	0.00	0.00
PHP	1	0.00	0.00
Resistencias 10K	3	0.35	1.05
Termistor	1	3.00	3.00
Potenciómetros 10K	2	2.5	5.00
Otros (Gastos Envío)	1	69.17	69.17
<b>TOTAL</b>			<b>383.92</b>

**Tabla 5. 1:** Resumen de Costos del Proyecto

## Conclusiones

- Es posible la implementación de proyectos utilizando herramientas de software libre consiguiendo un buen desempeño y alcanzando en algunos casos un producto final igual o mejor que si se hubiesen utilizado las alternativas con licencia comercial.
- El par transmisor receptor utilizado es de gran funcionabilidad y precisión, sí se quisiera una mayor resolución de temperatura este es capaz de entregarla pero para nuestro caso se redondeo la temperatura a un decimal, además al ser probado en ambiente industrial no hubo interferencia alguna con su transmisión siempre y cuando estuviese ubicado el par transmisor receptor en el rango especificado por el fabricante.
- El uso de una base de datos de mediana capacidad como Mysql no presento ninguna dificultad en el almacenamiento y búsqueda de parámetros deseados por el usuario, ya que para nuestro fin la base datos no se extendio demasiado y pudo manejarse muy fácilmente.
- Se logro una muy buena eficiencia con la utilización de C++ como lenguaje para la realización del programa de lectura del puerto y enlace con la base de datos ya que su variedad de comandos para intercambiar datos con Mysql permitió una completa compatibilidad entre ambos, así como la gran capacidad de manejo de puertos por parte del lenguaje de programación facilitó la elaboración del mismo.
- Es evidente que la precisión del proyecto depende en gran medida de la precisión del sensor a utilizar, por lo que si se desea tener una buena medición es conveniente la utilización de un transductor de mejor calidad.
- La relación COSTO-BENEFICIO se verá mejorada al aumentar el número de máquinas a monitorear, ya que el receptor es capaz de manejar hasta 100 transmisores identificándolos por un ID único.
- Es posible reducir el tiempo de transmisión utilizando otros transmisores de tipo industrial disponibles en el mercado, pero ello implicaría un

incremento considerable en el costo del proyecto, además si se reduce el tiempo de refresco de la página Web y si muchos clientes se conectan simultáneamente al servidor Web, este podría saturarse por lo que es posible que haya que invertir en un servidor más potente.

## Recomendaciones

- La distancia de 600 Pies cubierta por el transmisor es suficiente para la mayoría de aplicaciones, pero si se necesita ampliar la cobertura entonces puede utilizarse un repetidor de los propuestos por Point Six.
- Para una mejora en el transductor se recomienda la utilización de elementos de precisión, de esta manera se obtendrá un mejor balance en el puente de resistencias y a su vez una mejor respuesta.
- Si el proyecto llegase a ser implementado es recomendable la utilización de cable con blindaje para la conexión desde el puente balanceado hasta la entrada del transmisor.
- Pueden implementarse políticas de seguridad más estrictas en el lado del servidor cuando sea una aplicación muy crítica y que no se han implementado por estar fuera del alcance de este trabajo de graduación.
- No se recomienda bajo ninguna circunstancia utilizar el monitoreo presentado en este trabajo de graduación para monitoreo de señales vitales.
- En GNU/Linux existen siete niveles de ejecución en los cuales se inician o terminan servicios ya sea en el arranque, en el reinicio o en la detención total del sistema. Cuando el sistema arranca, el kernel ha configurado todo y ha montado la raíz del sistema de archivos arranca el programa `/sbin/init`, que es el padre de todos los procesos del sistema y el encargado de llevar al nivel de ejecución deseado.

Todos los archivos relacionados al arranque están ubicados en `/etc/rc.d`, el contenido del directorio es:

```
$ls /etc/rc.d
init.d/  rc0.d/  rc2.d/  rc4.d/  rc6.d/  rc.local*  rc.sysinit*
rc*      rc1.d/  rc3.d/  rc5.d/  rc.modules*
```

Los scripts para el inicio y finalización de un servicio específico como `httpd` (Apache) se encuentran en `init.d`, en los directorios `rcx.d/` (donde

x representa cada uno de los niveles de ejecución), solo se encuentran enlaces simbólicos a estos scripts. Los niveles de ejecución definidos son:

- 0: Detención total de la máquina
- 1: modo monousuario, puede ser utilizado cuando ocurren serios problemas o para la recuperación del sistema
- 2: modo multiusuario sin soporte para redes.
- 3: modo multiusuario con soporte para redes. El sistema está completamente operativo.
- 4: No utilizado, puede ser definido por el usuario.
- 5: Igual al 3 pero con el servidor gráfico operativo.
- 6: Reinicio del sistema

En base a lo anterior se recomienda correr el programa `puerto` como un servicio en el nivel de ejecución 3, porque para que el conjunto funcione adecuadamente la red debe estar operativa y el sistema debe estar en modo multiusuario. Así que debe colocarse el script para el arranque de `puerto` en el directorio `rc3.d`

- Se ha utilizado el control de acceso más básico que viene incluido con el módulo de autenticación de Apache, lo cual es funcional pero bastante limitado. Al utilizar un lenguaje como PHP y un servidor de Bases de Datos como MySQL, puede implementarse un control de usuarios con diferentes privilegios, por ejemplo varios usuarios con privilegios de administrador, otros solo para ver el comportamiento de temperatura pero sin acceso estadísticas.
- Para mantener la base de datos en un tamaño adecuado se ha implementado el borrado desde la interfaz Web pero únicamente puede ser realizado por un usuario administrador, por lo que este debe tener plena seguridad que los datos que borren ya no son de utilidad, además una vez que confirme que realmente desea borrar los datos estos no pueden ser recuperados. También es recomendable que se hagan regularmente copias de respaldo de la base de datos a un medio

extraíble como una cinta magnética o CD para lo cual será necesario avocarse a la documentación de MySQL<sub>[16]</sub>.

- La configuración por defecto del servidor Web Apache es para que atienda las peticiones de los clientes utilizando múltiples procesos, es decir que es multiproceso y es la que ha sido utilizada para la implementación del proyecto; esta configuración funciona perfectamente pero puede utilizarse el módulo MPM (Módulo de Multiprocesamiento) para crear un servidor híbrido multiproceso que atienda mayor número de peticiones y utilice menos recursos del sistema. Este módulo funciona de la siguiente manera:

Un solo proceso de control (el padre) es el responsable de crear los procesos hijo. Cada proceso hijo crea un número fijo de hebras del servidor, así como una hebra de escucha que escuchará si se producen peticiones y las pasará a una hebra del servidor para que la procese.

Apache siempre intenta mantener en reserva cierto número de hebras *de sobra* o en espera, que están preparadas para servir peticiones en el momento en que lleguen. Así, los clientes no tienen que esperar a que se creen nuevas hebras o procesos para que sean atendidas sus peticiones. Después durante el funcionamiento del servidor, Apache calcula el número total de hebras en espera entre todos los procesos, y crea o elimina procesos para mantener ese número dentro de los límites.

- Se recomienda el uso de una placa de circuito impreso para una mejor presentación y una mejor fijación de los componentes. Se presenta una alternativa en el anexo 5. Además se incluyen en el CD de documentación los archivos generados con el programa Eagle, tanto para el esquemático como para la tarjeta de circuito impreso.

## Referencias Bibliográficas

- [1] Instrumentation for Process Measurement and Control  
Norman A. Andersen, Third Edition
- [2] Análisis Instrumental  
Skoog/Leary Cuarta Edición
- [3] <http://www.aerocomm.com/Devices/link.htm>  
Última Visita: 11/07/2005
- [4] URL:<http://www.pointsix.com>  
Última Visita: 11/07/2005
- [5] Serial Programing HOWTO  
URL:<http://www.faqs.org/docs/Linux-HOWTO/Serial-Programming-HOWTO.html>  
Última Visita: 11/07/2005
- [6] URL:<http://distrowatch.com/dwres.php?resource=major>  
Última Visita:11/07/2005
- [7] URL:<http://news.com.com/2100-1001-245203.html?legacy=cnet>  
Última Visita:11/07/2005
- [8] URL:<http://www.noticiasdot.com/publicaciones/2004/0604/1906/>  
Última Visita: 30/06/2005
- [9] URL: <http://www.mandrakelinux.com>  
Última Visita: 20/03/2005
- [10] URL:<http://www.mandrakelinux.com/es/hardware.php3>  
Última Visita: 20/03/2005
- [11] Red Hat Linux 8.0: The oficial Red Hat Custumization Guide  
Red Hat 2002
- [12] Mandrake Linux 8.2 Guía de Instalación y Usuario  
Mandrakesoft Marzo 2002
- [13] Apache Server 2 Bible  
Mohammed J. Kabir Hungry Minds 2002

Página 3, Capítulo 1

- [14] URL:<http://www.netcraft.co.uk/Survey>  
Última Visita: 11/07/2005
- [15] Mandrake 8.2 Manual de Referencia  
MandrakeSoft Marzo 2002
- [16] MySQL Reference Manual for Version 4.0.18  
URL:<http://www.mysql.com/doc/>  
Última Visita: 11/07/2005
- [17] Build Your Own Database-Driven Website Using PHP & MySQL  
Kevin Yank, Sitepoint 2003
- [18] PHP Official Documentation  
URL:<http://www.php.net/docs>  
Última Visita: 11/07/2005
- [19] SAMS Teach Yourself in 24 Hours  
SAMS Publishing Junio 2000
- [20] URL:<http://www.perl.org/CPAN/modules/by-module/GD>  
Última Visita: 11/07/2005
- [21] URL:<http://www.perl.org/CPAN/modules/by-module/Chart/>  
Última Visita: 11/07/2005

### **Otras Fuentes:**

1. Documentación Oficial de Apache  
URL:<http://www.httpd.apache.org/docs-project/>  
Última Visita: 11/07/2005
2. Serial Programming Guide for POSIX Operating Systems  
URL:<http://www.easysw.com/~mike/serial/serial.html>  
Última Visita: 11/07/2005



## Glosario

**Amplificador:** Dispositivo utilizado para aumentar el nivel operativo de una señal de entrada.

**Analógico:** Señal que varía con el tiempo y que toma valores diferentes en cada instante de tiempo.

**Arranque, cargador de:** Es un programa que inicia el sistema operativo. Además existen algunos como GRUB y LILO que permiten seleccionar el sistema operativo que se desea iniciar (cuando se tiene instalado más de uno).

**APACHE:** Potente servidor Web multiplataforma y de diseño modular, desarrollado por Apache Software Foundation que se ha convertido en el más utilizado en la mayoría de sitios Web y es el principal representante del Software Libre.

**ASCII (American Standard Code for Information Interchange):** Código estándar que se utiliza para almacenar caracteres, incluyendo a los caracteres de control en una computadora.

**ASP (Active Server Pages):** Lenguaje de Script bastante potente cuyo código es ejecutado del lado de servidor y enviado al cliente como HTML, es la contraparte comercial a PHP.

**Buffer:** Porción de memoria de tamaño fijo, que puede ser asociada a un archive de modo de bloques (disco duro), una tabla de sistema, proceso.

En electrónica es un dispositivo que mantiene en su salida el último dato hasta que se recibe uno nuevo.

**Cliente:** Programa o computadora que esporádicamente, y por un tiempo dado, se conecta a otro programa u otra computadora para darle órdenes o solicitar información.

**DHCP (Dynamic Host Configuration Protocol):** Protocolo diseñado para que las máquinas de una red local obtengan una dirección IP dinámicamente.

**DNS (Domain Name System):** Mecanismo de direcciones/nombres distribuido que se utiliza en Internet. Este mecanismo permite mapear un nombre de

dominio a una dirección IP, que es lo que permite buscar un sitio por el nombre de dominio sin conocer la dirección IP de dicho sitio.

**GNU (GNU's not UNIX):** Proyecto iniciado por Richard Stallman al comienzo de los años ochenta y tiene como objetivo el desarrollo de un sistema operativo libre (no significa gratis, sino más bien libertad de modificarlo). Linux toma prestados de GNU el compilador GCC y la licencia GPL.

**GPL (General Public License):** La licencia del núcleo de GNU/Linux, va en la dirección contraria a todas las licencias propietarias en el sentido de que no se pone restricción alguna a la copia, modificación y redistribución del software, con la condición que el código fuente esté disponible. La única restricción, si se puede determinar así, es que a las personas a las que se redistribuye el software también se deben beneficiar con los mismos derechos.

**GUI (Graphical User Interface):** Un programa que utiliza menús, botones, colores y fuentes diferentes para parecer más fácil de usar a primera vista.

**HTML (HyperText Markup Language):** Lenguaje utilizado para crear documentos Web.

**HTTP (Hyper Text Transfer Protocol):** Protocolo utilizado para conectarse a sitios Web y recuperar documentos HTML.

**IP, dirección:** Es una dirección numérica que consiste de cuatro partes que identifica una computadora en Internet o en cualquier otra red basada en TCP/IP. Las direcciones IP están estructuradas de forma jerárquica, con los dominios de nivel superior y los dominios nacionales, los dominios, los subdominios y la dirección personal de cada máquina. Estas direcciones pueden ser estáticas o dinámicas.

**LAN (Local Area Network):** Nombre de una red con máquinas conectadas al mismo cable físico o mismo segmento de red.

**LINUX:** Sistema operativo tipo UNIX que corre en una variedad de plataformas y que cualquiera es libre de utilizarlo y modificarlo, aunque los cambios y modificaciones realizados al núcleo oficial son aprobados solo por los programadores encargados del proyecto del núcleo. El primer núcleo fue escrito por Linus Torvalds.

**PHP (PHP Hypertext Preprocessor):** Poderoso lenguaje para ejecución de scripts del lado del servidor. Una de sus principales ventajas es la facilidad para conectarse a servidores de Bases de Datos por lo que es muy útil para el desarrollo de aplicaciones Web interactivas.

**PNG (Portable Network Graphics):** Formato de archivo de imagen creado principalmente para su uso en la Web, ha sido diseñado como un reemplazo de GIF libre de patentes y además de otras características adicionales.

**Receptor:** Dispositivo electrónico que recibe las señales radiadas en el espacio por medio de una antena y extrae la información importante para su posterior procesamiento.

**root:** Es el superusuario de cualquier sistema Unix. Típicamente root es la persona responsable de mantener y supervisar al sistema Unix, tiene acceso completo a cualquier tarea.

**RPM (Red Hat Package Manager):** Un formato de empaquetado desarrollado por Red Hat para crear paquetes de software, este formato es utilizado por varias distribuciones.

**Script:** Un script es una secuencia de comandos a ejecutar.

**Sensor:** Dispositivo sensible que utiliza un fenómeno físico o químico dependiente de la naturaleza y el valor de la magnitud físico química a medir, lo cual permite la transducción del estímulo a una señal utilizada directa o indirectamente como medida.

**Servidor:** Programa o computadora que propone una característica o presta un servicio y espera las conexiones de los clientes para ejecutar las ordenes que estos solicitan. Ejemplos típicos son los servidores FTP, HTTP, NFS.

**Terminal:** Es la interfaz básica al núcleo del sistema operativo y es quien proporciona la línea de comandos donde el usuario ingresa comandos para ejecutar programas y comandos del sistema.

**Termistor:** Resistor que cambia su resistencia con la variación de la temperatura. Dada esta dependencia de la resistencia con respecto a la temperatura éste dispositivo se utiliza como un sensor de temperatura.

**Transmisor:** Equipo que genera una señal de radio ya modulada y enviarla a una antena para su radiación al espacio en forma de energía electromagnética.

## **Anexos**

Anexo 1: Transmisor Point Sensor Analog 10V

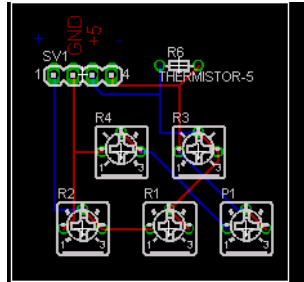
## Anexo 2: Point View Reciever

## Anexo 3: Load Cell Amplifier Montalvo M3200-ce/UR

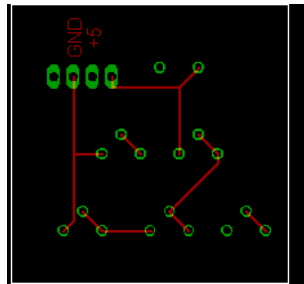
## Anexo 4: Licencia Pública General (GPL)



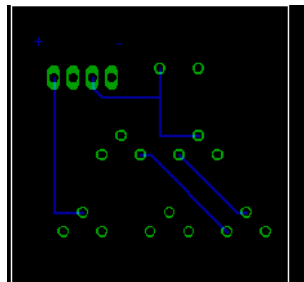
## Anexo 5: Placa de circuito impresso



**Completo**



**Capa Superior**



**Capa Inferior**

