

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

**SISTEMA INFORMÁTICO PARA MONITOREO DEL RITMO
CARDIACO DE PACIENTES, VÍA INTERNET**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
MENCIÓN REDES DE INFORMACIÓN**

**CARMEN RUTH CÉLLERI POTOSÍ
NARDO PATRICIO PADRÓN CORREA**

DIRECTOR: Ing. Edgar Torres

Quito, Octubre 2004

DECLARACIÓN

Nosotros, Carmen Céleri P. y Patricio Padrón C., declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Carmen Céleri P.

Patricio Padrón C.

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado Carmen Céleri P. y Patricio Padrón C., bajo mi supervisión.

Ing. Edgar Torres
DIRECTOR DE PROYECTO

DEDICATORIA

A nuestros Padres, por su constante ejemplo de vida.

A nuestros sobrinos por ser nuestra alegría.

A María Inés y Diego Andrés por enseñarnos lo hermoso de vivir.

AGRADECIMIENTO

A Dios por ser nuestra Fuerza.
A nuestra familia, por su gran apoyo.
A P.P. y C.C por su gran Amor.

CONTENIDO

CAPITULO 1.....	9
INTRODUCCIÓN.....	9
1.1 LOS SISTEMAS DE MONITOREO DEL RITMO CARDIACO EN PACIENTES	9
1.2 IMPORTANCIA Y OBJETIVIDAD DEL SISTEMA	11
1.3 DESCRIPCIÓN DEL SISTEMA	12
CAPITULO 2.....	15
SISTEMAS DE MEDICIÓN REMOTA (TELEMETRÍA).....	15
2.1 EVOLUCIÓN.....	15
2.2 QUE ES TELEMETRÍA	16
2.3 FORMAS DE COMUNICACIÓN REMOTA	16
2.4 DISPOSITIVOS DE TELEMETRÍA.....	18
CAPITULO 3.....	21
PROTOCOLOS DE COMUNICACIÓN	21
3.1 COMUNICACIÓN SERIAL CON EL S. O. LINUX	21
3.2 CONEXIÓN DE PERIFÉRICOS AL S. O. LINUX	30
3.3 UTILIZACIÓN DEL PROTOCOLO TCP/IP PARA SISTEMAS DE TELEMETRÍA VÍA INTERNET.....	31
3.4 COMUNICACIÓN DEL DISPOSITIVO DE MONITOREO CARDIACO CON EL S.O. LINUX	33
CAPITULO 4.....	34
DESARROLLO DEL SISTEMA.....	34
4.1 REQUISITOS DE HARDWARE Y SOFTWARE PARA EL SISTEMA.....	34
4.2 ANÁLISIS DEL SISTEMA	35
4.3 DISEÑO DEL SISTEMA	37
4.4 CODIFICACIÓN DE MÓDULOS DEL SISTEMA.....	54
4.5 CONSTRUCCIÓN DEL PROTOTIPO BASADO EN LOS MÓDULOS DEL SISTEMA DESARROLLADOS.....	76
CAPITULO 5.....	81
PRUEBAS	81
5.1 PRUEBAS DE INSTALACIÓN Y FUNCIONAMIENTO	81
5.2 RESULTADOS DE LAS PRUEBAS.....	86
CAPITULO 6.....	89
CONCLUSIONES Y RECOMENDACIONES	89
6.1. CONCLUSIONES	89
6.2. RECOMENDACIONES.....	90
BIBLIOGRAFÍA	92
ANEXOS	94
A. MANUAL DE USUARIO.....	94
B. MANUAL TÉCNICO.....	102
C. MANUAL DE INSTALACIÓN	104
D. COOKIES	107
E. LISTA DE TABLAS Y FIGURAS.....	109

RESUMEN

En este proyecto hemos presentado el sistema de monitoreo del ritmo cardiaco vía Internet de pacientes que se compone de módulos, el proyecto proporciona un mecanismo de ayuda a los médicos que realizan monitoreo del ritmo cardiaco en sus pacientes ya que usa la innovación de la tecnología orientada a una optimización de recursos del médico, materiales y económicos.

El uso del módulo del medidor del ritmo cardiaco, permite obtener un seguimiento del estado del corazón en cuanto a sus pulsaciones por minuto.

El uso del módulo de Adquisición y Almacenamiento de datos, permite recopilar y guardar en una base de datos la información básica de los pacientes y el monitoreo de su ritmo cardiaco para luego brindar remotamente dicha información a los médicos por medio del acceso al Servidor Web.

El uso del módulo del Servidor Web, permite crear sitios web dinámicos, para brindar al médico el acceso a la información de sus pacientes para su respectivo monitoreo.

Finalmente, el uso del módulo del Cliente, permite acceder a los sitios web del servidor y mostrar de forma gráfica los resultados del monitoreo del ritmo cardiaco vía Internet de los pacientes.

PRESENTACIÓN

El proyecto desarrollado cubre las necesidades que tienen los médicos de monitorear el funcionamiento de un órgano muy importante de las personas que es el corazón, dicho monitoreo lo hace de forma remota, de manera que optimiza el trabajo para el médico y representa un beneficio para sus pacientes.

CAPITULO 1

INTRODUCCIÓN

1.1 LOS SISTEMAS DE MONITOREO DEL RITMO CARDIACO EN PACIENTES

El control del ritmo cardiaco y su monitoreo constituyen un desafío frecuente en la práctica médica. La historia clínica de pacientes, el control del ritmo cardiaco, su estado físico, son herramientas suficientes para llegar a un correcto diagnóstico causal.

Los médicos que realizan el monitoreo del ritmo cardiaco en sus pacientes deben atenderlos en el sitio donde se encuentre el paciente, en clínicas u hospitales, para el médico esto es un gasto de recursos como tiempo y movilización, en el caso de trasladarse al sitio donde lo necesiten, de igual manera para el paciente.

Las cifras de hospitalización son elevadas y para los médicos resulta un poco incómodo manejar una administración de la información de pacientes que deseen y/o requieren ser atendidos.¹

“La enfermedad cardiaca y en particular las fallas cardiacas son las condiciones más importantes que contribuyen a limitar las actividades diarias, deteriorando la calidad de vida”¹, es por ello que para los médicos manejar una información del monitoreo cardiaco, de sus pacientes delicados que necesitan reposo, ayuda a controlar su ritmo de vida.

¹ Juan Páez, Holger Salazar, Tulane, Joffre Lara, Priscila Sánchez, “Falla cardiaca y modificaciones neurohormonales”, <http://scielo-co.bvs.br/scielo.php>, Diciembre 2001.

El corazón es una bomba que controla la circulación sanguínea en el organismo, variando esta circulación por cualquier actividad que realice la persona, siendo mucho mayor cuando se realiza tareas que requieran esfuerzo o cuando se tiene emociones fuertes.²

El funcionamiento normal del corazón³ está formado por dos movimientos principales llamados:

- El de diástole.
- El de sístole.

En el primer movimiento el corazón se llena de sangre y en el segundo la expulsa hacia las arterias. Este movimiento se refleja en algunas partes del cuerpo como un golpe pequeño llamado “de Korotkoff”, siendo posible medir este golpe en diferentes partes del cuerpo como el cuello, la muñeca, los dedos, etc. En el caso del medidor usado lo hace en un dedo.

El dispositivo de medida de ritmo cardiaco⁴ recibe la señal del sensor de pulsaciones cardiacas conectado a un dedo del paciente. Este sensor obtiene (por medio de un haz de luz infrarroja que pasa a través del dedo) la variación de volumen de los conductos sanguíneos la cual es expresada en pulsaciones por minuto y llamada frecuencia cardiaca.

En un adulto normal la frecuencia cardiaca oscila entre el rango de 70 -100 latidos/minuto, la cual puede disminuir durante el sueño (aprox. 10-20 latidos/minuto) o aumentar durante el ejercicio o la excitación emocional.

Generalmente se conoce por Bradicardia si se encuentra por abajo de este rango y Taquicardia por encima.

² Methodist Health Care System, “Las Enfermedades Cardiovasculares”, <http://www.methodisthealth.com/spanish/cardio/electro.htm>, 2004.

³ Marco A. Salgado, “Manual de Electrocardiografía”, Editorial Universitaria, 1976, págs.135-148

⁴ El dispositivo de medida de ritmo cardiaco se adquirió en una casa comercial con su respectivo manual de funcionamiento.

1.2 IMPORTANCIA Y OBJETIVIDAD DEL SISTEMA

Tanto al paciente como al médico se les dificulta la movilización hacia el sitio de atención médica debido al estado delicado de salud del paciente o a la falta de disponibilidad de tiempo del médico.⁵

De esta forma se puede decir que los medios tecnológicos que ayuden a estas personas a desempeñarse mejor, tanto en su vida cotidiana como en el sector productivo, podría ser una alternativa eficaz, aunque no exenta de limitaciones sobre todo económicas.

Por tanto, el objetivo del " Sistema informático para monitoreo del ritmo cardiaco de pacientes, vía Internet" se orienta a ayudar:

- A médicos
- A pacientes

A los médicos, para monitorear el ritmo cardiaco de sus pacientes delicados de una manera óptima y eficaz ya que podrán realizar el monitoreo y manejar la información de forma remota desde su consultorio o desde su casa, de esta manera, se facilitará enormemente la administración de la información periódica de sus pacientes.

Ayudará a optimizar el requerimiento de varios médicos para varios pacientes ya que un solo médico podrá realizar varios monitoreos a la vez y almacenar sus resultados para luego posteriormente analizarlos.

A los pacientes delicados que necesitan reposo, ya que no tendrán que ir al consultorio a la clínica o al hospital para monitorear o medir su ritmo cardiaco.

⁵ Se necesita encontrar una vía que permita al médico como al paciente no utilizar recursos de movilización.

1.3 DESCRIPCIÓN DEL SISTEMA

El Sistema informático para monitoreo del ritmo cardíaco de pacientes vía Internet consiste en:

- Un servidor de páginas Web.
- Un programa CGI.⁶
- Un driver de enlace con el puerto serial al dispositivo sensor.
- Un dispositivo hardware de medición del ritmo cardíaco.

El sistema servidor, esta instalado en una plataforma Linux por ser un sistema gratuito y además no requiere de hardware avanzado, facilitando la instalación en equipos antiguos (desde Intel Pentium MMX).

El servidor almacenará los datos suministrados por el hardware de medición del ritmo cardíaco del paciente, en una base de datos. Los datos que se registra son:

- La hora, minutos y segundos en el que se realiza la medición del ritmo cardíaco.
- El valor numérico de la medida de ritmo cardíaco.

En el servidor se ejecuta un programa que se encarga de grabar en la base de datos los datos de monitoreo con un intervalo de 10 segundos, además se encarga de leer el puerto de comunicaciones RS232.⁷

Este programa realiza las funciones de driver con el puerto serial y almacenamiento de datos de monitoreo.

⁶ CGI: Common Gateway Interface (Interfaz de Pasarela Común).

⁷ Para más información, Ver Capítulo 3.

El servidor utiliza un programa CGI, para presentar al médico un gráfico de ejes de coordenadas (pulsos/min. Vs. tiempo), el gráfico muestra la curva de variación del ritmo cardiaco con un muestreo de 10 segundos, según los valores almacenados en la base de datos, en un intervalo de una hora.

El médico puede revisar los datos del monitoreo del paciente tanto en tiempo histórico como en tiempo real sin tomar en cuenta la demora que tengan los paquetes de datos de información que existe desde el servidor de monitoreo (paciente) hasta el cliente (doctor), para el caso se denominará solo tiempo real.

Además, el médico puede cambiar la información de los datos del paciente que se encuentran almacenados en la base de datos, como son:

- Datos del paciente:
 - o Cédula
 - o Nombre
 - o Edad
 - o Sexo
 - o Nacimiento
 - o Dirección
 - o Teléfono
- Estado del paciente:
 - o (Paciente monitoreado = activo)
 - o (Paciente no monitoreado = inactivo)

Un dispositivo hardware de medición del ritmo cardiaco consiste en 4 sensores detectores de pulsación cardiaca y una tarjeta de adquisición de datos con salida RS232. Este dispositivo está conectado al computador por el puerto serie para lo cual se utiliza un programa "driver" que sirve de enlace entre el CGI y el hardware de medida.

El médico utiliza un navegador Web desde cualquier sistema operativo que admita el protocolo http (interfaz gráfica) y tenga una conexión a Internet o a la red que está el servidor de monitoreo para obtener la información del paciente que está conectado al dispositivo de medida del ritmo cardiaco. (Figura 1.1)

El sistema aprovecha hardware existente como son:

- La computadora del médico.
- La computadora del paciente.
- La infraestructura de Internet.

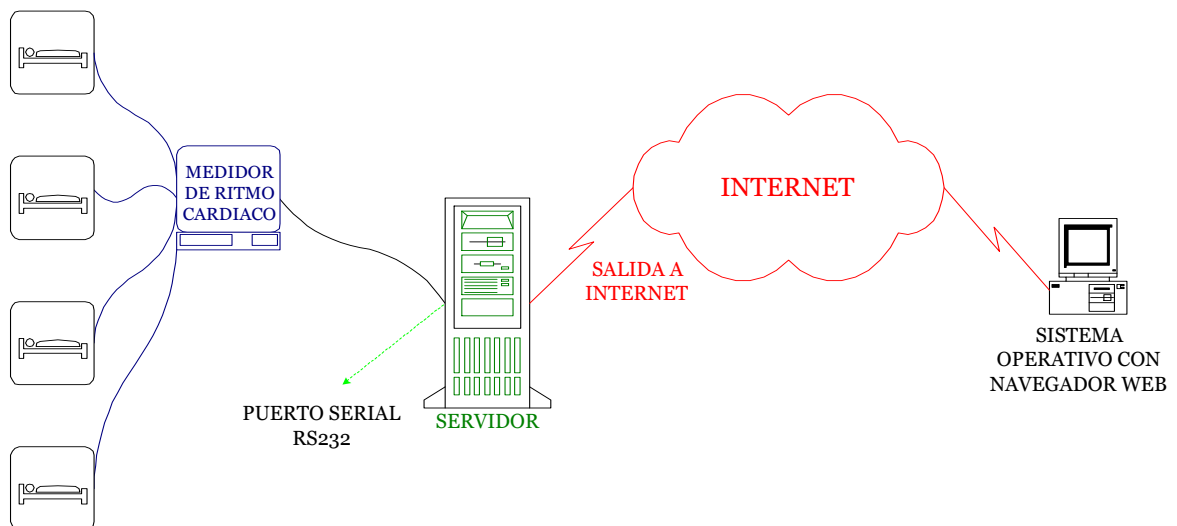


Figura 1.1 Diagrama a bloques del monitoreo de ritmo cardiaco vía Internet. (Fuente de Autores)

CAPITULO 2

SISTEMAS DE MEDICIÓN REMOTA (TELEMETRÍA)

2.1 EVOLUCIÓN

Desde sus orígenes la motivación principal para el uso de la telemetría ha sido la de facilitar el acceso a la información de medida de parámetros eléctricos desde lugares remotos y aislados.

Con el uso comercial de la comunicación telefónica avanzó mucho el campo de la telemetría al igual que con el uso de las ondas de radio.

La tecnología de comunicaciones actual aplicada a la atención médica nos permite una flexibilidad y una movilidad de la monitorización de los pacientes utilizando radiocomunicaciones, cables eléctricos, etc., que suponen una mejora de la calidad y una reducción del coste de la atención del paciente.

Al igual que las radiocomunicaciones y el tendido de cables eléctricos de grandes amplitudes requieren una inversión y un tiempo de instalación, aunque las radiocomunicaciones es un medio muy efectivo para la telemetría existe una preocupación creciente por la seguridad de los datos y protección electromagnética en aplicaciones de telemetría en medicina debido a la amplia difusión de sistemas de monitorización de pacientes que utilizan transmisores y receptores de radiofrecuencia.

Un medio de transmisión mucho más óptimo que ha evolucionado en los últimos años y es accesible para la mayoría de pacientes y hospitales es el Internet, puesto que este medio ya se encuentra instalado y se transmite por líneas telefónicas, radiocomunicaciones, líneas dedicadas, cables eléctricos, etc., podría parecer que tiene los mismos defectos de seguridad de datos, pero al transportarse la información se rige a los protocolos de Internet, por lo tanto, a la

seguridad de información de la misma, para la cual se desarrolla cada día muchos avances en hardware y software.¹

2.2 QUE ES TELEMETRÍA

Tradicionalmente, las mediciones de variables críticas se realizan manualmente casi en su totalidad, pues cada cierto tiempo debe acudir una persona a recoger los datos almacenados por los distintos sensores.

La telemetría es utilizada cuando se necesita medir una magnitud física en un lugar inaccesible para los aparatos convencionales o en lugares de acceso peligrosos como son los volcanes (este sistema de medición se utiliza mucho en nuestro país).

Se registra parámetros físicos mediante sensores y mediante cualquier medio de transporte se llega remotamente al receptor o colector de datos, el cual almacena la información, la procesa o la visualiza.

2.3 FORMAS DE COMUNICACIÓN REMOTA

Como antes se mencionó, para comunicarse remotamente uno de los medios más comunes son las ondas electromagnéticas, que pueden ser las de la telefonía celular, radio módem, comunicación satelital, o medios físicos como lo son los cables eléctricos, por medio de estas formas de comunicación se puede encontrar las más diversas medidas de parámetros, desde la medida de temperatura en un invernadero hasta la medida de temperatura en el planeta Marte.

¹ En cuanto a la seguridad de datos, puede ser la comunicación bajo SSL (Secure Socket Layer).

A continuación en la Figura 2.1 se puede ver un sistema básico de comunicación remota vía ondas electromagnéticas.²

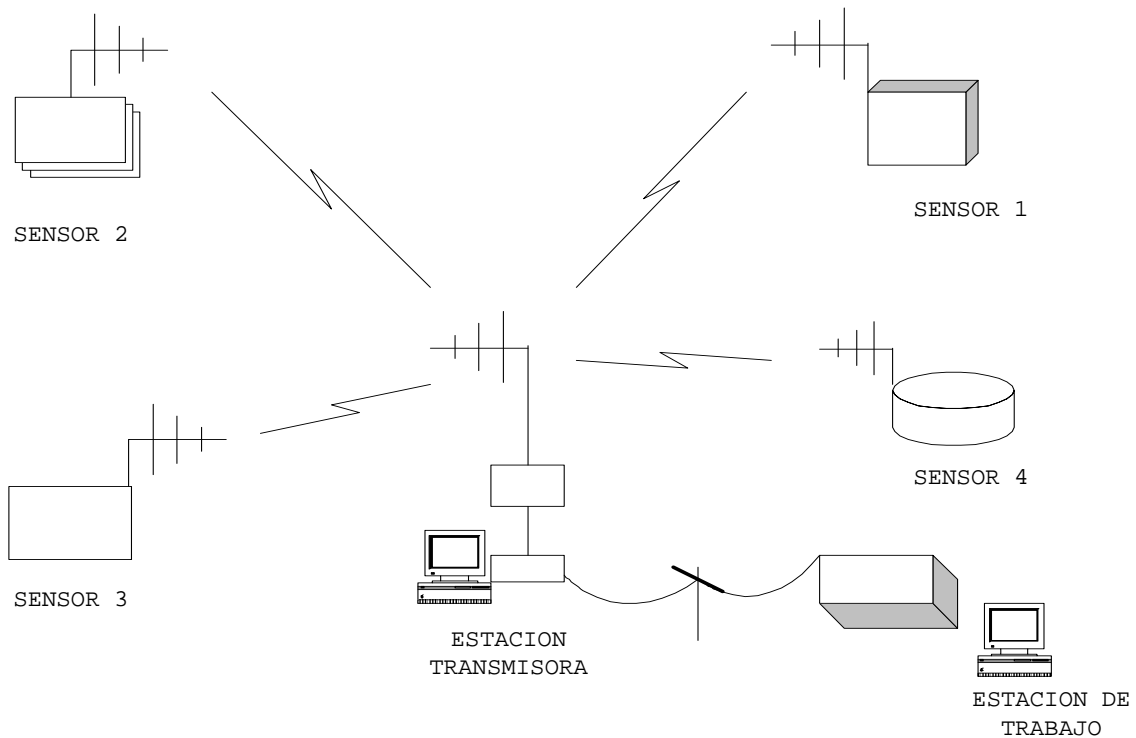


Figura 2.1 Ejemplo de Comunicación remota vía ondas electromagnéticas. (Fuente de Autores)

En este ejemplo, la información de cada sensor es transmitida por una unidad por separado mediante ondas electromagnéticas hacia una central de recepción la cual transmite hacia una estación de trabajo por medio de línea telefónica, esta estación es la encargada de procesar la información.

Con el ejemplo antes mencionado podemos ver la utilización mixta de medios de transporte de datos.

² Calibraciones Industriales, "Telemetría, Aplicaciones", <http://www.ci-sa.com/telemetry.htm>.

En la Figura 2.2 se puede ver el mismo sistema de telemetría pero por medio de Internet utilizando el protocolo TCP/IP.

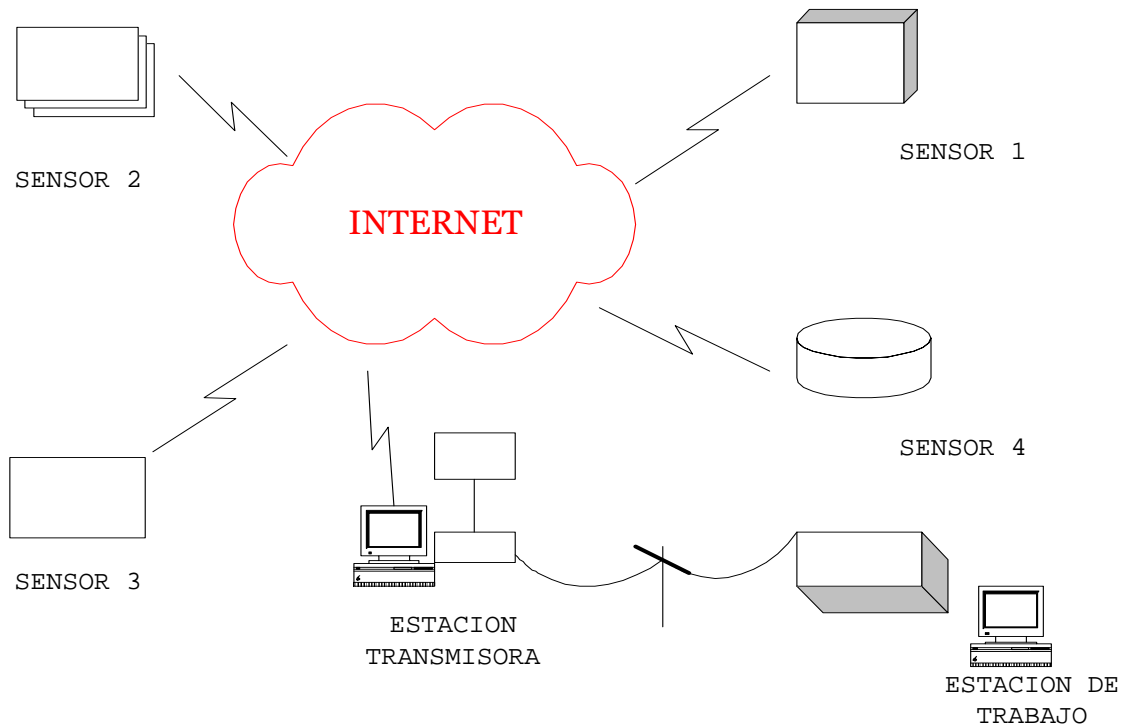


Figura 2.2 Ejemplo de Sistema de telemetría por medio de Internet. (Fuente de Autores)

2.4 DISPOSITIVOS DE TELEMETRÍA

El propósito del sistema de telemetría es proporcionar la capacidad para adquirir, procesar y distribuir los datos de telemetría.

Para adquirir el parámetro físico se utiliza sensores, que pueden ser tan simples como un contacto eléctrico a tan complejo como una tomografía.

Existen dispositivos comerciales como los de la casa comercial x@natel.com los cuales proporcionan una medición al instante de una forma sencilla e ideal para ubicaciones remotas.

Estos dispositivos utilizan la comunicación celular GSM para transportar la información por lo cual requieren pagar una tarifa mensual muy alta en nuestro país.

Para el campo médico existen dispositivos que envían remotamente datos críticos de los pacientes hacia los hospitales, como se puede ver un ejemplo en la Figura 2.3.



Figura 2.3 Dispositivo portátil de monitoreo de pacientes.³

Este dispositivo fabricado por la casa comercial DR Systems envía datos de todas las modalidades de imágenes, incluyendo rayos x estándar, Ultrasonido y Medicina nuclear, y le permite a las imágenes que sean distribuidas electrónicamente e interpretadas en estaciones de computadoras remotas.

Este dispositivo también utiliza la telefonía celular para el transporte de información.

³ Hospital de la Universidad George Washington, "Sistema de telemetría MicroPaq", <https://espanol.gwhospital.com/p4475.html>.

Un medio más óptimo de transporte es el Internet, en la Figura 2.4 se observa un módulo para telemetría vía Internet.



Figura 2.4 Módulo para telemetría vía Internet.⁴

Este módulo es programable y puede funcionar como servidor web, tiene un puerto 10/100Base-T Ethernet integrado, permite una conectividad en tiempo real de modo local o desde cualquier otro lugar del mundo.

Un sistema como el anterior puede ser reemplazado con un computador personal (si tenemos espacio para ubicarlo en el lugar de los sensores) con una conexión a Internet.

⁴ Matriz Electrónica S.I., “Nuevo módulo 10/100Base-T de Rabbit”, <http://www.matrix.es/Noticias/verNoticia.asp>.

CAPITULO 3

PROTOCOLOS DE COMUNICACIÓN

3.1 COMUNICACIÓN SERIAL CON EL S. O. LINUX

3.1.1 EL MODELO DE ENTRADA / SALIDA DE UNIX/LINUX

"En Unix, los periféricos se encuentran identificados como ficheros del sistema de archivos. De modo, que si se quiere enviar datos a un dispositivo como la pantalla o disco duro, etc., es suficiente con abrir el fichero asociado a este dispositivo y escribir o leer con las funciones comunes de entrada / salida". ¹

Por la razón de tratar a los dispositivos como ficheros, simplificando notablemente la programación pues el programador no necesita aprender funciones específicas para cada dispositivo.

El fichero del dispositivo usa las funciones open(), read(), write() y close() para abrir, leer, escribir y cerrar respectivamente.

Configurar un puerto serial. ²

```
struct confio;
confio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
confio.c_iflag = IGNPAR | ICRNL;
confio.c_oflag = 0;
confio.c_lflag = ICANON;
confio.c_cc[VMIN]=1;
confio.c_cc[VTIME]=0;
tcflush(fd, TCIFLUSH);
tcsetattr(fd,TCSANOW,&confio);
```

¹ Sergio Costas Rodrigue, "Programación a bajo nivel del puerto serie", http://ultimaorbita.com/raciel/x-ezine/x2/2x007-Serial_low_level_programming.html.

² Peter Baumann, "Configuración del Puerto", <http://linux-cd.com.ar/manuales/howtos/programacion-serie/Programacion-Serie-Como-2.html>, 1997.

Abrir un puerto serial

Para abrir un puerto serial en Linux, se procede como si se tratara de un fichero normal:

```
Int fd;
fd = open("/dev/ttyS0", O_RDWR | O_NOCTTY | O_NONBLOCK);
    if (fd < 0) { perror("Error en Puerto serial"); }
```

Esta instrucción abre el puerto serial ttyS0 para lectura y escritura en modo no bloqueo (para que no se quede bloqueado con read, sino que esté atento a la señal). Es el primer puerto serial (COM1 en DOS y Windows). Si en caso retorna menor a cero da error.

Escribir por un puerto serial

La escritura se realiza de la siguiente manera:

```
n = write(fd, "Prueba", 6);
if (n < 0)
{ perror("Error en Puerto serial"); }
```

Esta instrucción escribe por el puerto serial ttyS0 la palabra "Prueba" y si retorna menor a cero da error.

Manejador de señales

Se puede esperar a que lleguen los datos al puerto serial ya sea leyendo el puerto con la instrucción "read" (el programa queda bloqueado hasta que exista datos) o primero atendiendo a una señal programada para que atienda al puerto serial.

Por ejemplo en el siguiente código se observa la manera de instalar el manejador de señales.³

```
struct sigaction pio;
//se instala el manejador de señales
pio.sa_handler = atiente_PUERTO; //función que atiende la señal
pio.sa_mask = 0;
pio.sa_flags = 0;
pio.sa_restorer = NULL;
sigaction(SIGIO,&pio,NULL);
//permite al proceso recibir la señal SIGIO
fcntl(fd, F_SETOWN, getpid());
//se hace asíncrono al descriptor de archivo
fcntl(fd, F_SETFL, FASYNC);
//se guarda la configuración del Puerto
tcgetattr(fd,&pio);
...
...
...
// manipulador de la señal SIGIO
void atiente_PUERTO (int estado)
{
    printf("señal SIGIO.\n");
    espera = 0;
}
```

³ Peter Baumann, “Configuración del Puerto”, <http://linux-cd.com.ar/manuales/howtos/programacion-serie/Programacion-Serie-Como-3.html>, 1997.

Leer de un puerto serial

```
if (espera ==0) //cuando ya ha sido atendida la señal
{
    respuesta = read(fd,buf,255);
    bufer[respuesta]=0;
    printf(":%s:%d\n", bufer, respuesta);
    espera = TRUE; // espera para una nueva entrada de datos
}
```

Cerrar un puerto serial

Para cerrar el puerto serial se lo hace con la función:

```
close(fd);
```

//fd es el descriptor del fichero, en este caso el puerto ttyS0

Si se quiere enviar parámetros más específicos al dispositivo se deberán usar funciones especiales como la función IOCTL.

La función IOCTL

Todo driver de dispositivo y programa de conexión al puerto serie soporta una las funciones básicas de: apertura, lectura, escritura y cierre. Estas funciones son las que permiten acceder a los dispositivos como si fuesen ficheros normales. Sin embargo, existe otra función que nos permite enviar comandos específicos para cada periférico, esta es la función de control de entrada / salida IOCTL (Input/Output ConTroL).⁴

Esta función recibe tres parámetros y devuelve uno:

```
#include <sys/ioctl.h>
int ioctl(int id, int cm, void *pt);
```

⁴ Sergio Costas Rodrigue, "Programación a bajo nivel del puerto serie", http://ultimaorbita.com/raciel/x-ezine/x2/2x007-Serial_low_level_programming.html.

El primer parámetro `id` es un identificador de dispositivo.

El segundo parámetro `cm` especifica el comando que se ha de enviar al dispositivo. Estos comandos dependen del periférico con el que se está trabajando.

El tercer parámetro `pt` es un puntero al parámetro o a los parámetros que se le han de pasar al manejador del dispositivo o driver, o donde ha de retornar éste los resultados. El tipo de dato del puntero depende del comando que se envíe, pudiendo ser un puntero a `int`, a `char`, o incluso a una estructura.

El valor de retorno de `ioctl()` es 0 si la función se ha ejecutado con éxito, o -1 si ha ocurrido algún problema. En este caso, el tipo de error lo podrá ver en la variable global `errno`.

Funciones IOCTL disponibles para el puerto serie

Las funciones `ioctl`⁵ para el control de las líneas del puerto serie son: `TIOCMGET`, `TIOCMBSIS`, `TIOCMBSIC`, `TIOCMSET`, `TIOCSBRK` y `TIOCCBRK`.

Para disponer de estas definiciones es necesario colocar la siguiente cabecera en el programa:

```
#include <asm/ioctls.h>
```

La función `TIOCMGET` precisa que se le pase un puntero a `int` como argumento de la función `ioctl()`:

```
retorno=ioctl(fichero,TIOCMGET,&argumento);
```

En `argumento` almacenará el estado de las distintas entradas y salidas del puerto serie.

⁵ Todas las funciones IOCTL son tomadas de la Ref.: Michael R. Sweet, "Serial Programming Guide for POSIX Operating Systems", <http://www.easysw.com/~mike/serial/serial.html>, 1994-2003.

La función TICMBIS activará aquellas salidas que le indiquemos, dejando inalteradas el resto. Precisa de un puntero a int como argumento de la función ioctl():

```
retorno=ioctl(fichero,TIOCMBIS,&argumento);
```

La función TICMBIC hace lo contrario a la anterior: desactiva (pone a cero) las salidas que se le indica, sin modificar el resto:

```
retorno=ioctl(fichero,TIOCMBIC,&argumento);
```

La función TIOCMSET activa las salidas que se le indica y desactiva el resto.

Precisa también un puntero a int como argumento de la función ioctl():

```
retorno=ioctl(fichero,TIOCMSET,&argumento);
```

Activará las salidas que se le indica en función del patrón de bits que contenga argumento.

El bit que corresponde a cada entrada y salida viene definido también en el fichero ioctl-types.h. Este fichero se incluye automáticamente con ioctl.h.

Las definiciones son las siguientes:

```
TIOCM_DTR  DTR  0x002
TIOCM_RTS  RTS  0x004
TIOCM_CTS  CTS  0x020
TIOCM_CAR  DCD  0x040
TIOCM_CD   DCD  0x040
TIOCM_RNG  RI   0x080
TIOCM_RI   RI   0x080
TIOCM_DSR  DSR  0x100
```

Por último, las funciones TIOCSBRK y TIOCCBRK permiten poner a cero (+12 voltios) o devolver al estado normal, respectivamente, el pin TXD. Como dicho estado de reposo es un uno, estas funciones permiten controlar a voluntad dicha salida.

```
retorno=ioctl(fichero,TIOCSBRK,&argumento);  
retorno=ioctl(fichero,TIOCCBRK,&argumento);
```

El puntero &argumento no es modificado, por lo que se puede dar cualquiera.

3.1.2 CONFIGURACIÓN DE ENTRADA PARA DISPOSITIVOS SERIE

Hay 2 tipos diferentes de proceso de entrada. La configuración adecuada se tiene que escoger de acuerdo a la aplicación a la que lo queremos destinar.

Siempre que sea posible no se debe usar un bucle para leer un sólo carácter a fin de obtener una cadena completa. Para hacerlo de forma adecuada se utiliza la instrucción read para leer toda la cadena.

Proceso de Entrada Canónico

Este el modo de proceso de entrada que realizan los terminales de Unix pero puede ser útil para comunicaciones con otros dispositivos serie.

Toda la entrada es procesada en unidades de líneas, lo que significa que al ejecutarse la instrucción read sólo devolverá una línea completa de entrada. Una línea está, finalizada con un NL(ASCII LF), y fin de fichero, o un carácter fin de línea, pudiendo ser un CR (el fin de línea por defecto de DOS/Windows).

El proceso de entrada canónica puede, también, manejar los caracteres borrado, borrado de palabra, reimprimir carácter, traducir CR a NL, o cualquier carácter de control.

Proceso de Entrada No Canónico

En esta configuración se manejará un conjunto fijo de caracteres por lectura, y permite un carácter temporizador. Este modo se debería usar si la aplicación siempre lee un número fijo de caracteres, o si el dispositivo conectado envía ráfagas de caracteres.

Este es el método utilizado en el proyecto de tesis.

Los formas de configuración descritas anteriormente se pueden usar en:

- Modo síncrono.
- Modo asíncrono.

El modo síncrono viene por defecto, donde la sentencia read se bloqueará hasta que la lectura esté completa. En modo asíncrono la sentencia read devolverá inmediatamente y enviará una señal al programa llamador cuando esté completa. Esta señal puede ser recibida por un manejador de señales.

Permisos de acceso a los puertos serie

Para poder acceder a un fichero es preciso que se tenga los permisos de acceso adecuados, de la misma manera se comportan los dispositivos.

El dueño de los ficheros de dispositivo es el usuario root, por lo tanto es necesario hacer el cambio entrando como él.

Para permitir que cualquiera pueda leer y escribir en cualquier puerto serie, se debe hacer (siempre como root):

```
chmod 666 /dev/ttyS*
```

Ejemplo:

Si se quiere permitir que cualquiera acceda al primer puerto serie (COM1 en DOS o Windows), se ejecutará:

```
chmod 666 /dev/ttyS0
```

3.1.3 DESCRIPCIÓN FÍSICA DEL PUERTO SERIE

En la Tabla 3.1. se muestra la configuración física de un puerto serie.

Nombre	Sentido	Conector 25 pines	Conector 9 pines
TXD	salida	2	3
RXD	entrada	3	2
RTS	salida	4	7
CTS	entrada	5	8
DTR	salida	20	4
DSR	entrada	6	6
DCD	entrada	8	1
RI	entrada	22	9
GND	masa	7	5

Tabla 3.1 Descripción del puerto serie. ⁶

TXD es la salida y RXD es la entrada de datos serie.

Estos datos se envían en el siguiente formato:

Por defecto, TXD se encuentra a -12 voltios (nivel alto)

Cuando se quiere enviar un byte, la línea se pone a 12 voltios (nivel bajo) durante el tiempo de un bit. Es el bit de inicio, que marca el comienzo de una transmisión. A continuación se envían los bits del dato, empezando por el menos significativo.

⁶ José A. Carballar, "El Libro de la Comunicaciones del PC", Computec ra-ma, 1989, págs. 32-33.

El cero (nivel bajo) se representa con 12 voltios, y el uno (nivel alto) se representa con -12 voltios.

Por ultimo se envían uno o dos bits de parada a -12 voltios (nivel alto) antes de iniciar la transmisión del siguiente.

En estas 2 líneas el 0 son 12 voltios y el 1 son -12 voltios. En el resto, el cero son -12 voltios y el uno son 12.

Debido a que las líneas TXD y RXD son controladas por el propio puerto serie y no por el procesador, en principio no resulta posible indicar directamente un valor de tensión en TXD o leer el que se encuentre en RXD. Todo lo que podemos hacer es enviar un byte al puerto para que este lo envíe por sí mismo en el formato indicado. Sin embargo, en la practica, las UARTs disponen de un bit que, al activarlo, fuerzan un cero (+12 voltios) en dicho pin.

3.2 CONEXIÓN DE PERIFÉRICOS AL S. O. LINUX

Los puertos serie tienen asociados una serie de ficheros en el directorio /dev.

En versiones antiguas de Linux estos eran los ficheros /dev/cua seguidos de un número que especificaba el puerto de cada uno. Siendo /dev/cua0 el primer puerto serie (COM1 de DOS o Windows)

En las versiones más recientes del núcleo (versiones 2.x) los puertos serie tienen asociados los ficheros /dev/ttyS seguidos de un número. Siendo /dev/cua0 el primer puerto serie (COM1 de DOS o Windows)

Por razones de compatibilidad con programas antiguos, se siguen soportando los viejos ficheros... /cua, pero no se recomienda su uso en programas nuevos.

Puertos seriales en diferentes sistemas derivados de Unix

Ejemplo de los nombres que dan los diferentes sistemas operativos a sus puertos seriales:⁷

IRIX® /dev/ttyf1

HP-UX /dev/tty1p0

Solaris®/SunOS® /dev/ttya

Linux® /dev/ttyS0

Digital UNIX® /dev/tty01

3.3 UTILIZACIÓN DEL PROTOCOLO TCP/IP PARA SISTEMAS DE TELEMETRÍA VÍA INTERNET

La mayoría de los protocolos que fueron especialmente diseñados o creados para monitoreo o control de dispositivos han optimizado sus tramas de forma que el espacio útil para datos de las aplicaciones fuera el máximo.

Por ejemplo, para leer el dato de un sensor que indique abierto o cerrado basta con una orden codificada en unos cuantos bytes. Por lo tanto, se trata de minimizar los campos de control (dirección origen, dirección destino, CRCs, etc.) que el protocolo necesita para transferir estos bytes al dispositivo destino.

El protocolo TCP/IP, que fue diseñado para transferir grandes cantidades de datos entre dos máquinas, cuando se usa para transferir 2 o 3 bytes de información el coste en ancho de banda es muy alto. Por ejemplo, entrando directamente a nivel IP, el paquete mínimo necesita 20 octetos sólo de campos de control. Si entramos al nivel de TCP son 40 octetos de control como mínimo.⁸

⁷ Michael R. Sweet,, “Serial Programming Guide for POSIX Operating Systems”, <http://www.easysw.com/~mike/serial/serial.html>, 1994-2003.

⁸ Kris Jamás/Ken COPE, “Programación en Internet”, McGRAW-HILL, 1996, págs.105-122.

Por este motivo, el protocolo TCP/IP no está optimizado para su uso en monitoreo y telemetría, pero a pesar de las razones técnicas que se acaban de mencionar TCP/IP está siendo usado en la mayoría de computadores y sistemas operativos de forma que ha conseguido estandarizarse llegando a ser la herramienta ideal para asegurar la ínter conectividad total entre máquinas en cualquier parte del mundo.

Una cualidad que hace a TCP/IP inmejorable a pesar de su recarga de datos, es la capacidad de enrutamiento, de manera que no importa en que lugar del mundo estemos monitoreando, siendo la única condición de que tenga una conexión a Internet. (La Figura 3.1, muestra un dispositivo de monitoreo y la comunicación con el computador remoto a través de Internet).

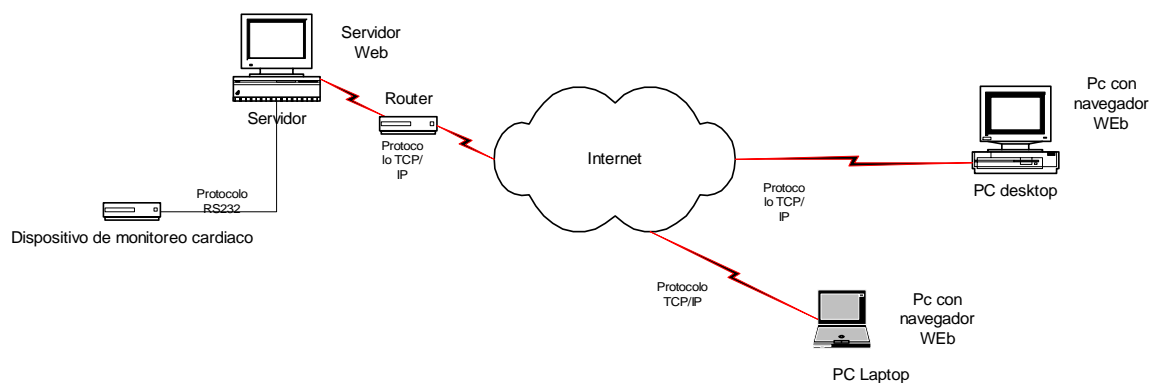


Figura 3.1 Acceso al dispositivo de monitoreo por medio de Internet. (Fuente de Autores).

3.4 COMUNICACIÓN DEL DISPOSITIVO DE MONITOREO CARDIACO CON EL S.O. LINUX

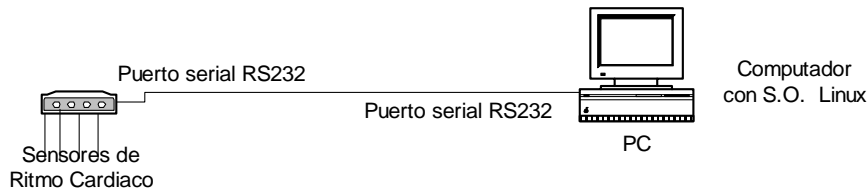


Figura 3.2 Diagrama a bloques de conexión al computador del dispositivo de monitoreo. (Fuente de Autores).

El computador y el dispositivo de medición del ritmo cardíaco se comunican a una velocidad de 9600 bps , 8 bits, sin paridad y 1 bit de parada.

El dispositivo de medición del ritmo cardíaco multiplexa la medida de los 4 sensores y envía al puerto serial RS232 del computador. (Figura 3.2).

En el computador corre un proceso que está a la espera de datos por el puerto serial.

El dispositivo de monitoreo⁹ entrega al computador un paquete de datos en el que están las 4 medidas de los 4 sensores. (Figura 3.3).

Sensor 1	Sensor 2	Sensor 3	Sensor 4
----------	----------	----------	----------

Figura 3.3 Trama de datos entregada por el dispositivo de medición. (Fuente de Autores).

⁹ En el capítulo 4 se analiza con más detalle la comunicación del dispositivo de medición del ritmo cardíaco con el computador.

CAPITULO 4

DESARROLLO DEL SISTEMA

4.1 REQUISITOS DE HARDWARE Y SOFTWARE PARA EL SISTEMA

Como se mencionó en el anterior capítulo el sistema basa su funcionamiento en señales de datos enviadas por un dispositivo de medida del pulso cardiaco hacia el puerto serial de un computador.

Computador Servidor

Para que este computador sea apto para recibir la información de forma correcta y almacenarla debe tener las siguientes características de hardware:

- Puerto serial RS232 a una velocidad de 9600 bits por segundo.
- Microprocesador Pentium MMX 233 Mhz.
- Disco duro de 1 Gbyte.
- Memoria RAM de 64Mbyte.
- Monitor SVGA (indispensable solo para instalación).
- CDROM (indispensable solo para instalación).
- Tarjeta de red 10/100 o conexión a Internet vía MODEM.

Para procesar y almacenar los datos de monitoreo es necesario las siguientes características de software:

- Linux Red Hat 7.2 o compatible con los siguientes paquetes:
 - o Base de datos MySql.
 - o Servidor Web Apache 1.3
 - o Php 4.0
- Conexión a Internet con dirección IP fija. ¹

¹ La dirección IP debe ser fija porque de otra manera no se podría acceder remotamente, a no ser que, nos informen que dirección IP se tiene en ese momento, lo cual no es muy práctico.

Se aprovecha el servidor Web Apache por ser uno de los más sólidos en su funcionamiento y estabilidad, además fácil de configurar.

Se escoge la base de datos MySQL por la rapidez de acceso a sus datos y la mejor integración con PHP.

PHP se usa porque es un lenguaje muy apto y de fácil integración con HTML, lo cual nos facilita enormemente la programación logrando que sea fácil de actualizar y de ampliar sus características de funcionamiento en el futuro.

Computador Cliente

El único requisito para este computador será que tenga un navegador Web compatible con gráficos como Netscape, Internet Explorer, y que tenga una conexión a Internet.

4.2 ANÁLISIS DEL SISTEMA

El sistema lo podemos dividir en 2 partes:

- parte local.
- parte remota.

La parte local del sistema es la encargada de medir los pulsos cardiacos por minuto de una persona que en adelante lo denominaremos "ritmo cardiaco".

La manera de medir puede ser por cualquier puerto físico, dependerá del dispositivo a usar que en este caso es serial.

En un sistema operativo tipo Unix como es el caso de Linux ² el acceso al puerto serial se lo puede realizar de diferentes maneras:

- aplicando funciones básicas de archivos como son: `open()`, `read()`, `write()`.
- redirigiendo la entrada serial a un archivo.
- si el caso lo requiere (ej: driver a nivel de kernel) se lo puede manejar al puerto a bajo nivel por medio de direcciones hardware y peticiones de interrupción.

En el proyecto se aplicará las funciones básicas de archivos (`open`, `read`, `close`) ya que el flujo de los datos no amerita mayor complejidad y tenemos un control sobre los datos.

El dispositivo de entrada que en este caso es el medidor de ritmo cardiaco puede ser cualquiera que envíe sus datos de manera serial con la diferencia que se tendrá que cambiar la interpretación del flujo de datos de entrada y la velocidad de transmisión en el programa de acceso al puerto (si en caso sea diferente de 9600 bits/seg).

Luego de la lectura del puerto serial se procesa esta información e interpreta para luego almacenarla ya sea en un archivo o en una base de datos.

En el caso del proyecto se utiliza el acceso a la base de datos aunque la forma en que están almacenados los datos es muy sencilla, pero de esta manera tenemos un control total de forma modular consiguiendo con esto una estructura fácil de ampliar y mejorar sus características de almacenamiento en el futuro.

La información almacenada puede ser accesada de manera local o remota.

² sistema operativo que trata a todos los dispositivos como archivos.

Esta información almacenada en la base de datos no es muy útil sin un programa que la extraiga, procese y visualice, para esto necesitamos de alguna manera de presentar los datos a la persona interesada (en este caso el médico) de forma remota.

Para que el médico acceda a la información guardada mediante un computador tendrá que acceder mediante algún programa cliente (en este caso un navegador Web).

El sistema no entrega un diagnóstico del paciente, solamente se limita a monitorear su ritmo cardiaco.

Para que el cliente pueda acceder a la información necesitamos un enlace hacia la base de datos y un programa servidor que reciba las peticiones del médico y entregue la información según sus requerimientos.

Este servidor estará en la parte local del sistema siendo la manera más óptima utilizar un servidor de páginas Web.

4.3 DISEÑO DEL SISTEMA

Es un sistema cliente / servidor que nos permite monitorear el ritmo cardiaco de pacientes vía Internet o mediante una red TCP/IP.

4.3.1 MÓDULOS DEL SISTEMA

Para describir el diseño del sistema se divide en las siguientes partes:

- Módulo de medida del pulso cardiaco.
- Servidor.
- Cliente.

En la Figura 4.1 se puede ver un diagrama de bloques del sistema Cliente Servidor de monitoreo cardíaco.

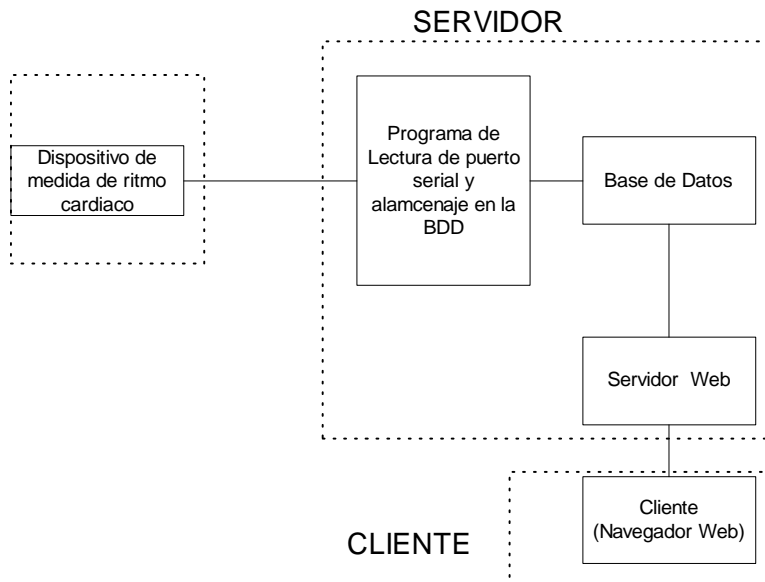


Figura 4.1 Diagrama de bloques del Sistema. (Fuente de Autores).

Módulo de medida del pulso cardíaco

La medida del pulso cardíaco se lo hace de diferentes maneras, pudiendo ser por la medida de señales eléctricas por medio del electrocardiógrafo (ECG) o midiendo la variación de volumen de los conductos sanguíneos, el dispositivo utilizado tiene este principio de funcionamiento.

Al obtener las pulsaciones por minuto el módulo la convierte en señal digital RS232 para que el computador pueda interpretarla.

La información es enviada por el módulo en forma de una cadena de caracteres conteniendo las siguientes partes:³

- Dato sensor 1.
- Dato sensor 2.
- Dato sensor 3.
- Dato sensor 4.

El tamaño del paquete de datos es de 12 caracteres siendo cada 3 bytes para cada medidor. (Figura 4.2).

Sensor 1 = 3 caracteres	Sensor 2 = 3 caracteres	Sensor 3 = 3 caracteres	Sensor 4 = 3 caracteres
----------------------------	----------------------------	----------------------------	----------------------------

Figura 4.2 Trama de datos de 12 caracteres entregada por el dispositivo de medición. (Fuente de Autores).

En la Figura 4.3 se puede observar un ejemplo de un paquete de datos.

- sensor 1= "048"
- sensor 2= "076"
- sensor 3= "120"
- sensor 4= "087"

048	076	120	087
-----	-----	-----	-----

Figura 4.3 Medida con valores reales. (Fuente de Autores).

Los caracteres son leídos por el programa del servidor y convertidos a números enteros de 3 dígitos cada medida.

³ Manual de referencia del dispositivo de medición de ritmo cardiaco.

Los números indican las pulsaciones por minuto de la persona conectada al sensor.

La velocidad de envío de datos es de 9600 bits por segundo con 1 bit de parada, no paridad y 8 bits de tamaño del dato. Esta velocidad debe estar configurada en el computador para poder leer los datos de forma correcta.

Servidor

El servidor lee los datos desde el puerto serial mediante un programa que se encuentra monitoreando y los guarda en una base de datos, los cuales son extraídos desde el computador cliente (navegador Web).

El programa que lee el puerto serial esta programado en lenguaje C. Tiene el nombre "lserial".

Este programa monitorea de manera continua dicho puerto (en este caso el puerto serial 1 o ttyS0), luego interpreta los datos leídos y los envía a otro programa para que grabe en la base de datos. (Ver Capítulo 4.4 para código completo).

La rutina encargada de leer el puerto serial es la siguiente:

```
while (STOP==FALSE) {
    respuesta = read(fd,buf,255);
    buf[res]=0;
    //..... utilización de la cadena respuesta
}
```


Entra en un bucle infinito y entrega resultados en la variable (tipo cadena de caracteres) "respuesta" cuando el buffer se llena con 12 caracteres, valor que se define anteriormente mediante la línea de instrucción:

```
newtio.c_cc[VMIN] = 12;
```

Esta cadena de caracteres es dividida en 4 secciones, cada una para la medida de cada sensor de la siguiente manera:

```
sprintf(c1,"%c%c%c",buf[0],buf[1],buf[2]) ;  
sprintf(c2,"%c%c%c",buf[3],buf[4],buf[5]) ;  
sprintf(c3,"%c%c%c",buf[6],buf[7],buf[8]) ;  
sprintf(c4,"%c%c%c",buf[9],buf[10],buf[11]) ;
```

Con estas instrucciones obtenemos las cuatro medidas en las variables c1, c2, c3, c4, las cuales las enviamos a grabar en la base de datos mediante un programa externo llamado "gbase" (también programado en lenguaje C) de la siguiente manera:

```
sprintf(comando,"./gbase %d %d %d %d",atoi(c1),atoi(c2),atoi(c3),atoi(c4));  
system(comando);
```

El programa gbase recibe en línea de comando las cuatro medidas y luego las convierte a entero mediante la función atoi() y envía a la función de grabar en la base de datos Mysql:

```
int main(int argc, char *argv[])  
{  
    conecta_db(); // se conecta a la base de datos monitoreo  
    grabar_db(atoi(argv[1]),atoi(argv[2]),atoi(argv[3]),atoi(argv[4]));  
}
```

Antes de grabar estos datos el programa se conecta a la base y comprueba posibles errores.⁴

La información guardada no serviría de nada si no se la puede observar de manera remota, para esto se instala los programas CGI (páginas Web) en el servidor Web Apache.

Se crea el directorio /monitoreo (como usuario root) en el directorio de publicación de páginas Web:

En el directorio:

```
/var/www/html/
```

ejecutar

```
mkdir monitoreo
```

Algo muy importante es los permisos que se tienen que dar al directorio y a los archivos los cuales tienen que ser de ejecución para todos los usuarios y no de escritura, solamente se podrán leer y escribir por el usuario administrador o root.

Además del servidor Web Apache se debe instalar el módulo de PHP con librería gráfica GD que por lo general viene activado en la instalación común de Linux.

Cliente

Este módulo no se lo diseña ya que se considera al Navegador Web como módulo Cliente.

⁴ Ver Capítulo 4.4 para código completo.

4.3.2 DISEÑO DE LAS INTERFASES WEB

El computador del médico, que puede ser Mac, PC, y tener cualquier sistema operativo que admita el protocolo http y utilice un navegador Web (cliente con interfaz gráfica), y una conexión a Internet o a la red en la que está el servidor de monitoreo, esto le permite ingresar a las páginas Web desarrolladas para observar la información del paciente que está conectado al dispositivo de medida del ritmo cardiaco, además a una página para actualizar los datos básicos de sus pacientes.

Las páginas Web están desarrolladas en:

- Código PHP
- Código HTML
- Código Java Script
- Herramienta de desarrollo Web Macromedia Dreamweaver MX 2004.

El código PHP se utiliza para validar los datos ingresados en las páginas con la base de datos almacenada y autorizar el ingreso al sistema.

El código Java Script se utiliza para redireccionar páginas Web.

El código de HTML se utiliza para realizar la presentación al usuario e ingreso de datos.

Página inicial index.html

La página index.html está desarrollada en lenguaje HTML, se muestra la fecha actual del sistema, una presentación al sistema de Monitoreo de ritmo cardíaco de Pacientes, y las opciones para ingresar al monitoreo de los pacientes tanto Histórico como en tiempo real o para ingresar a la página de actualización de datos de pacientes. (Figura 4.4).

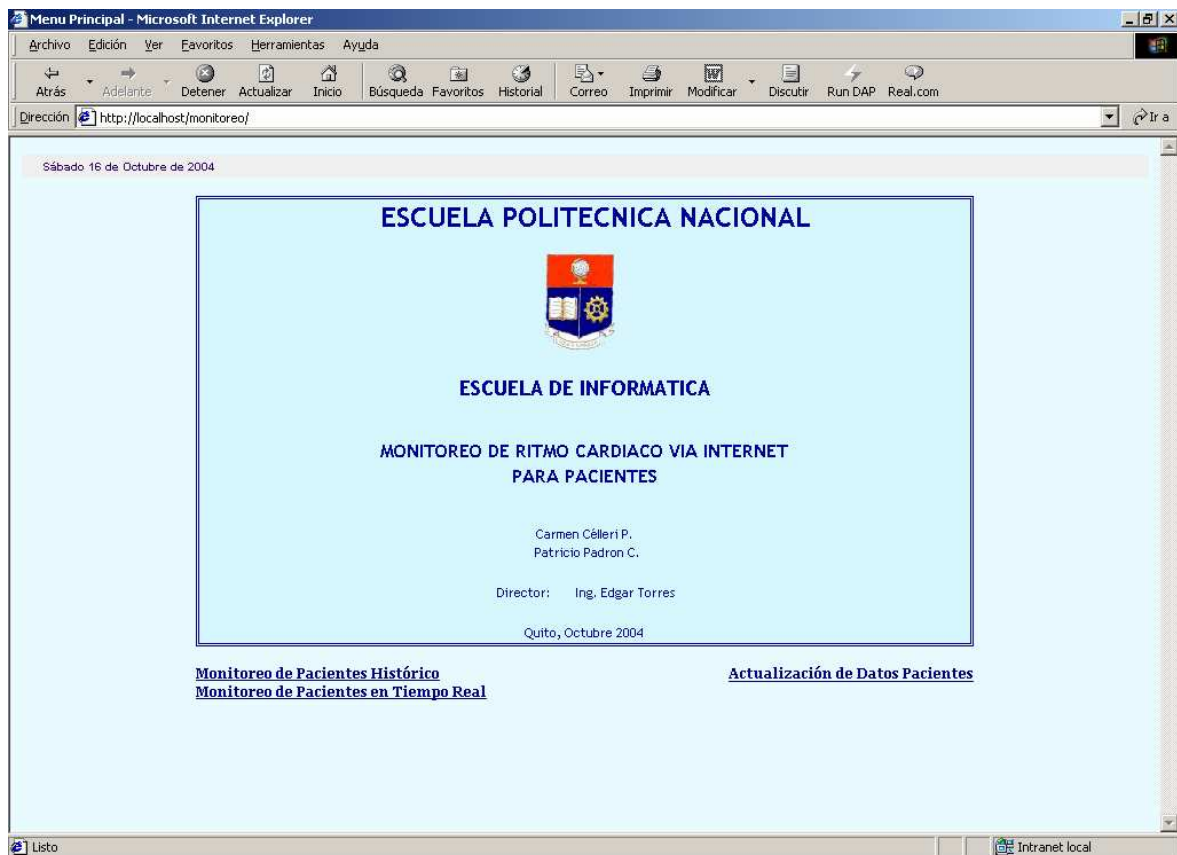


Figura 4.4 Pantalla index.html. (Fuente de Autores).

Página de Validación de Usuarios

En esta página se realiza la conexión a la base de datos para la validar el usuario y su password.

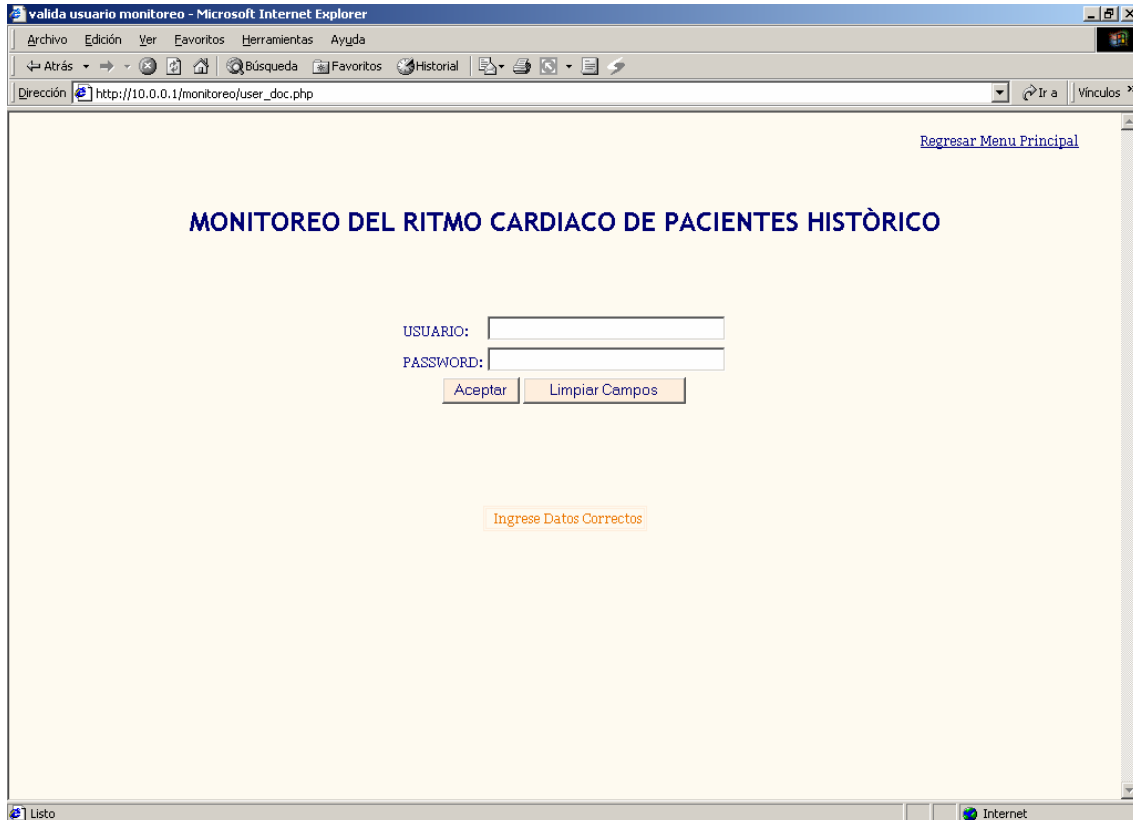


Figura 4.5 Pantalla de login de monitoreo histórico. (Fuente de Autores).

A continuación se muestra parte del código de consulta para la verificación de usuario y password en la base de datos que contiene los nombres de los doctores autorizados para realizar el monitoreo:

```
mysql_select_db($database_validadoc, $validadoc);  
$query_ingresa = sprintf("SELECT * FROM validadoc WHERE  
validadoc.USUARIO='%s'", $mitexto_ingresa);  
$ingresa = mysql_query($query_ingresa, $validadoc) or die(mysql_error());  
$row_ingresa = mysql_fetch_assoc($ingresa);  
$totalRows_ingresa = mysql_num_rows($ingresa);
```

Mientras el usuario y el password ingresados NO sean válidos, se indica el mensaje "Ingrese Datos Correctos" ya se para monitoreo de pacientes en tiempo real o histórico.(Figura 4.5).

Si el usuario es válido quiere decir que SI existe en la base de datos.

Suponiendo que el usuario exista en la base de datos, también se comprueba que el password sea correcto, para ello se hace una simple comparación de igual manera con la base de datos, si es correcto se continúa, de lo contrario sigue indicando el mensaje "Ingrese Datos Correctos".

Si el usuario y el password ingresados son correctos, entonces, se crea automáticamente "cookies" ⁵ de usuario y password (Los cuales asignamos en el programa con las variables "15041972"/"24061972" y contienen el usuario y password respectivamente) en la máquina del usuario, en la página se indica el mensaje "Usuario Autorizado" e inmediatamente se lo envía a la página de monitoreo de pacientes.

La cookie creada permite que el médico posea autorización para el manejo de la Página de Monitoreo de Pacientes, mientras la cookie no caduque el médico podrá realizar en dicha página cualquier consulta que lo desee.

Caso contrario, si la página se encuentra inactiva por un lapso de una hora (tiempo de vida de la cookie), cualquier consulta que se desee realizar ya no se la podrá ejecutar y en su defecto mostrará el mensaje "Usuario no autorizado para entrar a Monitoreo de Pacientes".

Las cookies se graban en la máquina del usuario y se las puede editar como un archivo de texto.

⁵ Ver Anexo D.

El password ingresado es encriptado mediante la función MD5⁶ y luego es grabado directamente en la base de datos, de la misma manera se realiza la consulta de verificación password.

A continuación la utilización de la función de encriptación MD5 y la creación de cookies:

```
$codif=md5($pass); //utilización de función MD5
if($row_ingresa !=0 and $row_ingresa['PASSWD']==$codif ){
//crea cookies de autorización
setcookie("15041972",$nombre,time()+120);
setcookie("24061976",md5($pass),time()+120);
$usrvalido = "USUARIO AUTORIZADO ";
```

Página de Monitoreo de Pacientes Histórico

La página de monitoreo de pacientes histórico, presenta las siguientes opciones:

- Selección de Fecha de Monitoreo.
- Selección de Hora de Monitoreo.
- Limpiar Gráfico.

Para realizar la consulta de monitoreo del paciente, se selecciona la Fecha (en formato “dd-mm-aaaa”) y la hora de Monitoreo en la que se desea ver los datos de monitoreo del paciente.

Una vez ingresado la Fecha y la Hora de Monitoreo, se selecciona el nombre del paciente, con esto, se logra realizar la consulta en la base de datos de monitoreo del paciente especificado mostrando los resultados en el gráfico de coordenadas (pulsos/min. Vs. tiempo), (Figura 4.6).

⁶ Ashish Wilfred, “Proyectos Profesionales PHP”, ANAYA Multimedia, 2002, págs.457-471

Caso contrario el gráfico no mostrará ningún resultado.

A continuación se muestra una parte del código para la consulta de datos con parámetros de fecha y tiempo:

```
mysql_select_db($database_dbdatos, $dbdatos);  
$ver_datos = sprintf("SELECT * FROM datos where id='%s' and fecha='%s' and  
tiempo >='%s' and tiempo < '%s'", $id, $fecha, $tini, $tfin);
```

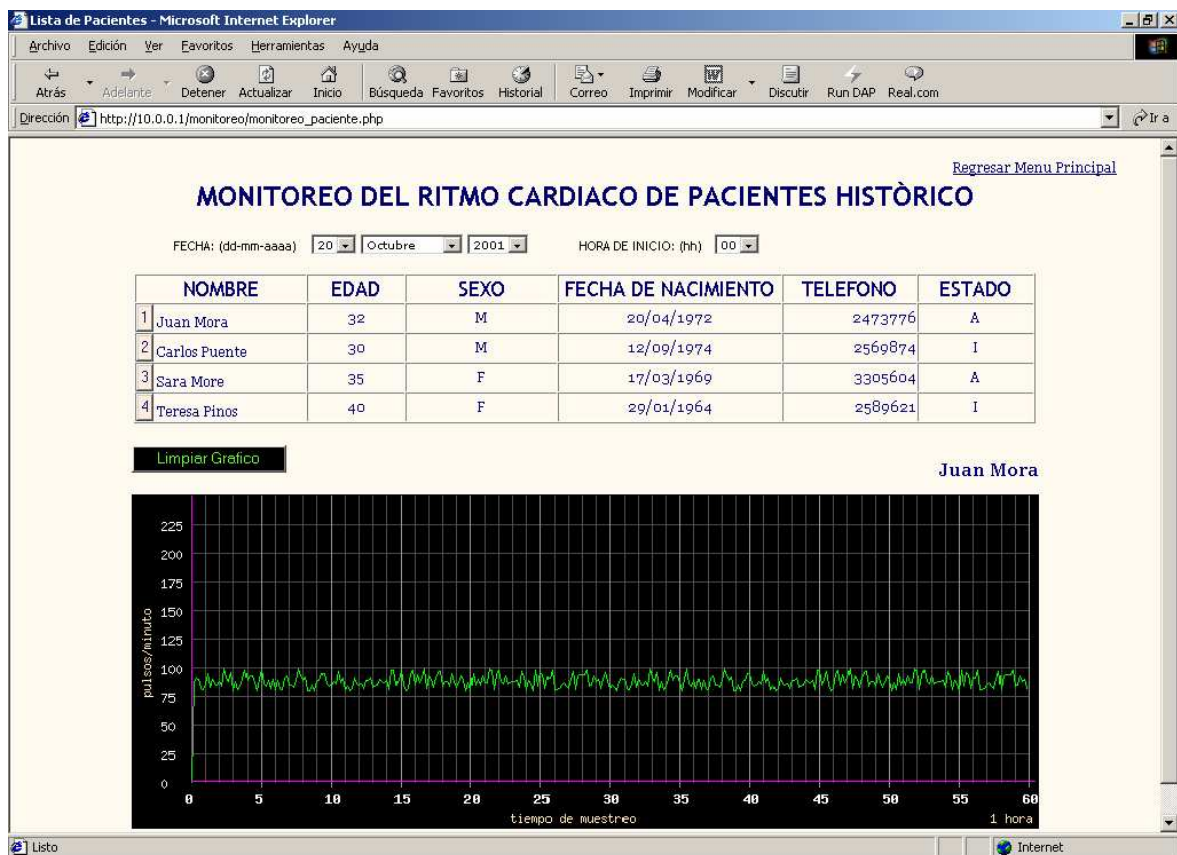


Figura 4.6 Pantalla de monitoreo del ritmo cardiaco histórico. (Fuente de Autores).

En el gráfico, la escala máxima en los pulsos por minutos es de 250 pulsos, mientras que en la escala de tiempo de muestreo es de una hora.

La opción "Limpiar Gráfico", permite actualizar los datos consultados.

Caso contrario, si desea consultar el monitoreo para más de un paciente en una misma fecha u hora, basta con ingresar una vez la fecha y la hora de monitoreo, lo que debería solamente seleccionar es el paciente que desea monitorear.

Así mismo, si desea cambiar cualquiera de las opciones de fecha u hora, basta con seleccionarla y luego seleccionar al paciente para mostrar los resultados en el gráfico.

La opción “Regresar Menú Principal”, permite ir a la página de inicio donde nuevamente tiene las opciones para ingresar a la página Web que desee.

Página de Monitoreo de Pacientes en Tiempo Real



Figura 4.7 Pantalla de monitoreo del ritmo cardiaco en tiempo real. (Fuente de Autores).

La página de monitoreo de pacientes en tiempo real, a diferencia del monitoreo de pacientes histórico no presenta la opción de escoger la hora y fecha, solamente permite seleccionar el paciente al cual monitorear.

Para obtener los datos de cada paciente se accede a la base de datos y se obtiene los datos de la última hora de monitoreo, es decir, que los datos visualizados son en tiempo real, que como se indicó en el *capítulo 1.3*, sin tomar en cuenta la demora que tengan los paquetes de datos de información que existe desde el servidor de monitoreo (paciente) hasta el cliente (doctor) ni la interferencia que exista en la transmisión, la página se refresca cada 3 segundos.

Página de Validación de Usuario para de Actualización de datos

El funcionamiento de la página de validación de usuario para actualización de datos es similar al funcionamiento de validación de monitoreo de pacientes, la diferencia está, en que a ésta página Web pueden tener acceso otro tipo de usuarios que también puede ser el médico.

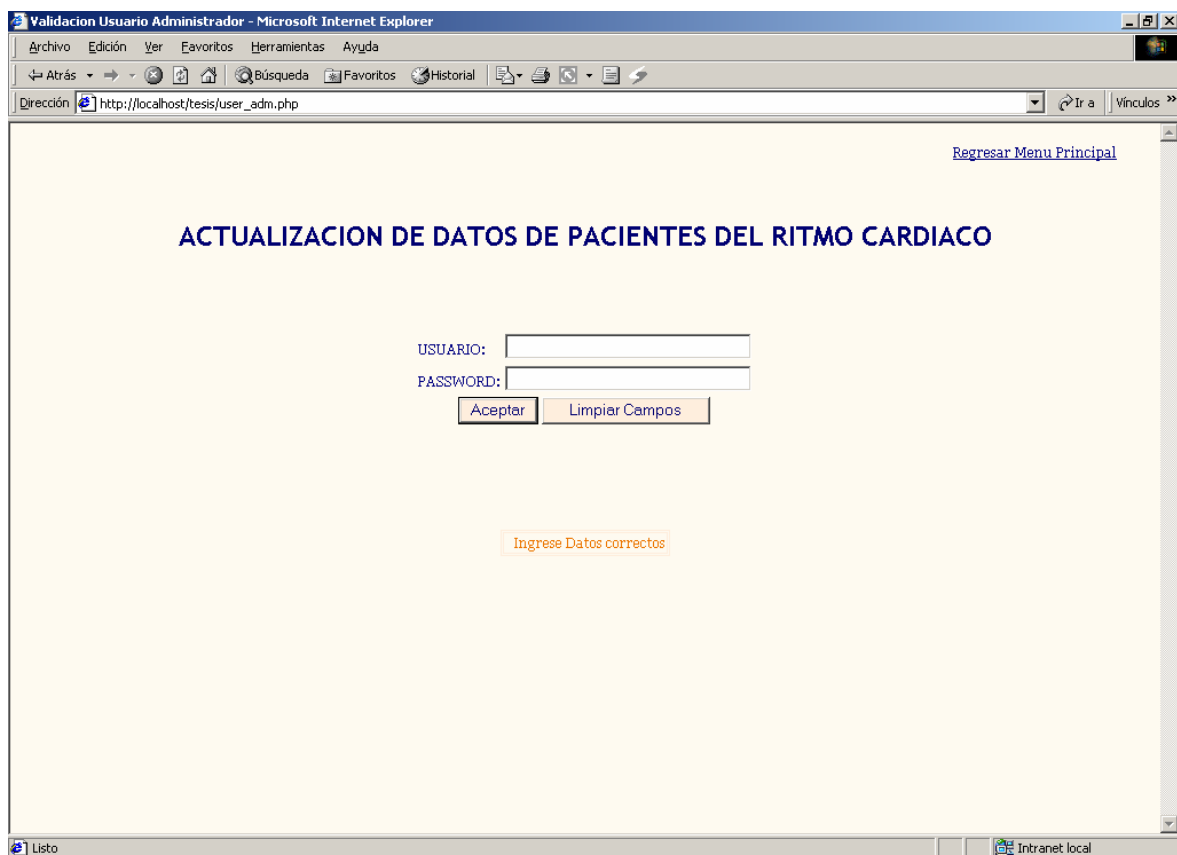


Figura 4.8 Pantalla de login para actualización de datos. (Fuente de Autores).

Mientras el usuario y el password ingresados NO sean válidos, se indica el mensaje "Ingrese Datos Correctos". (Figura 4.8).

Si el usuario es válido quiere decir que SI existe en la base de datos.

Suponiendo que el usuario exista en la base de datos, también se comprueba que el password sea correcto, para ello se hace una simple comparación de igual manera con la base de datos, si es correcto se continúa, de lo contrario sigue indicando el mensaje "Ingrese Datos Correctos".

Página de Actualización de Datos de Pacientes

La página de actualización de datos pacientes, permite cambiar los siguientes datos del mismo:

- CEDULA
- NOMBRE
- EDAD
- SEXO
- NACIMIENTO
- DIRECCIÓN
- TELÉFONO
- ESTADO (sí el paciente se encuentra en monitoreo o no)

Esta página permite realizar cambios de los datos de pacientes y actualizarlos directamente en la base de datos.

Se presenta una barra de desplazamiento para mostrar los datos de cada uno de los pacientes. (Figura 4.9)

La cookie creada para actualización de datos de pacientes, permite que el médico o la persona autorizada posea autorización para el manejo de esta página,

mientras la cookie no caduque el usuario podrá realizar en dicha página cualquier actualización que desee.

Caso contrario, si la página se encuentra inactiva por un lapso de quince minutos (tiempo de vida de la cookie), cualquier actualización que se desee realizar ya no se la podrá ejecutar y en su defecto mostrará el mensaje "Usuario no autorizado para entrar a Actualización de Datos de Pacientes"

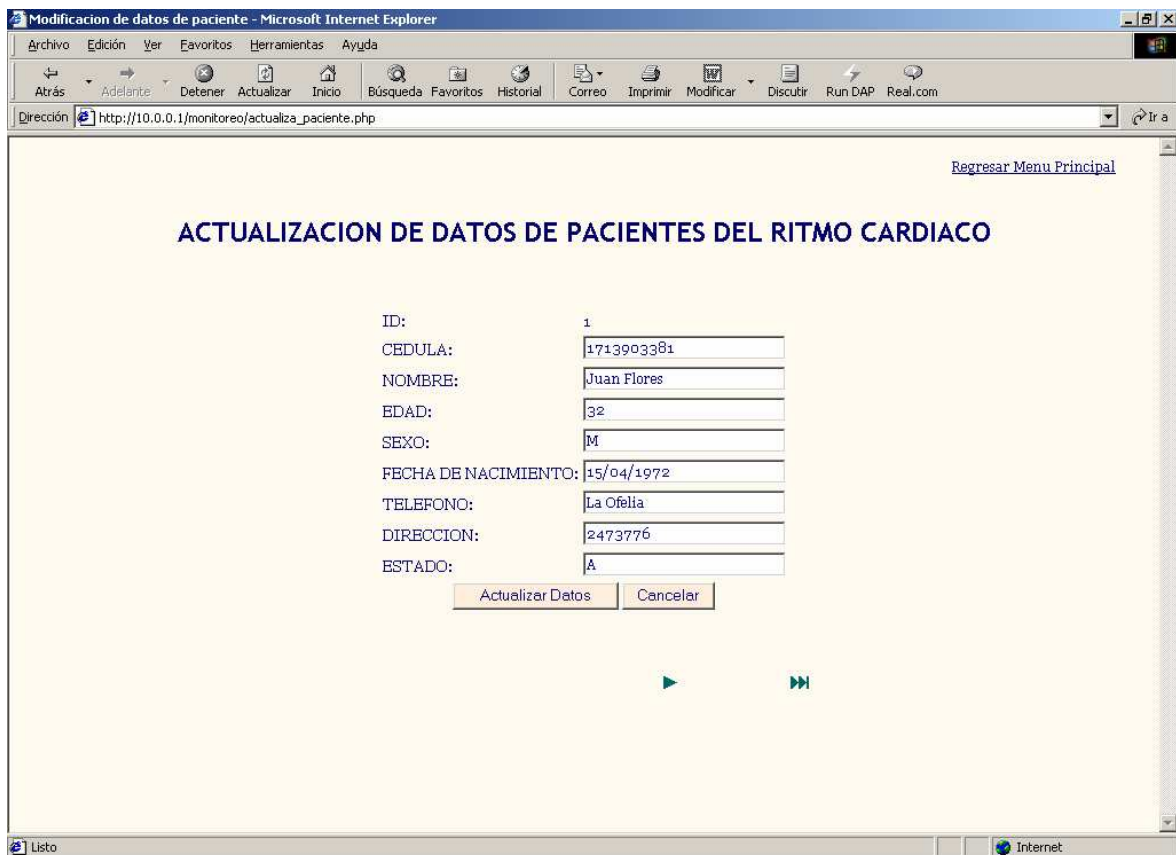


Figura 4.9 Pantalla de Actualización de Datos de Pacientes. (Fuente de Autores).

La opción "Regresar Menú Principal", permite ir a la página de inicio donde nuevamente tiene las opciones para ingresar a la página que desee.

Para la validación del acceso a las páginas Web se diseñan las siguientes páginas:

- entrar.php.
- entrar_admin.php.

Estas páginas comprueban la presencia de “cookies” de autorización, si las encuentran, verifican los datos de usuario y password en la base de datos y actualizan dichas “cookies”, caso contrario las destruyen.

A continuación las instrucciones para destruir las “cookies”:

```
//Destruimos las cookies.
```

```
setcookie("15041972","x",time()-120);
```

```
setcookie("24061976","x",time()-120);
```

4.4 CODIFICACIÓN DE MÓDULOS DEL SISTEMA

A continuación se muestra el código más importante del sistema.

Código para lectura del puerto serial (lserial.c)

```
#include <mysql/mysql.h>
#include <mysql/errmsg.h>
#include <mysql/mysql_error.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>

#define VELOCIDAD B9600
#define PUERTO "/dev/ttyS0"
#define FALSE 0
#define TRUE 1
```

```
volatile int STOP=FALSE;
long respu;
unsigned int i;
```

```
main(){
int fd,c, res;
struct termios oldtio,newtio;
struct tm *tr;
char buf[255];
int r;
char hora[8];
char fecha[10];
char comando[22];
char recibe[12];
```

```
char c1[3];
char c2[3];
char c3[3];
char c4[3];
time_t t;
time(&t);
tr=localtime(&t);
memset(fecha,0,10);
printf("\n_____ \n");
printf("\nt INICIO DE MONITOREO\n");
sprintf(hora,"%d:%d:%d",tr->tm_hour,tr->tm_min,tr->tm_sec);
sprintf(fecha,"%d-%d-%d",tr->tm_mday,tr->tm_mon,tr->tm_year+1900);
//en time.h indica que se debe sumar 1900
printf(" [hora: %s ][fecha: %s ]",hora,fecha);
printf("\n_____ \n");
//Abre el puerto serial para lectura y escritura
fd = open(PUERTO, O_RDWR | O_NOCTTY);
if (fd <0){
    perror(PUERTO); exit(-1);
}
//guarda la configuracion actual del puerto
tcgetattr(fd,&oldtio);
newtio.c_cflag = VELOCIDAD | CS8 | CLOCAL | CREAD;
newtio.c_iflag = IGNPAR;
newtio.c_oflag = 0;
//pone el modo entrada (no-canónico, sin eco,...) */
newtio.c_lflag = 0;
newtio.c_cc[VTIME] = 0;
newtio.c_cc[VMIN] = 12; // bloquea lectura hasta recibir 12 caracteres
tcflush(fd, TCIFLUSH);
tcsetattr(fd,TCSANOW,&newtio);
while (STOP==FALSE) {
    res = read(fd,buf,255);
    buf[res]=0; // envio de fin de cadena, a fin de poder imprimir printf
    time(&t);
    tr=localtime(&t);
    memset(fecha,0,10);memset(hora,0,8);
    sprintf(hora,"%d:%d:%d",tr->tm_hour,tr->tm_min,tr->tm_sec);
    sprintf(fecha,"%d-%d-%d",tr->tm_mday,tr->tm_mon,tr->tm_year+1900);
    //en time.h indica que se debe sumar 1900
```

```
fflush(0);
printf("[%s]", fecha);
printf("[%s]\n", hora);
//convierte la cadena de caracteres de 12 a 4 cadenas de 3 caracteres
sprintf(c1, "%c%c%c", buf[0], buf[1], buf[2]) ;
sprintf(c2, "%c%c%c", buf[3], buf[4], buf[5]) ;
sprintf(c3, "%c%c%c", buf[6], buf[7], buf[8]) ;
sprintf(c4, "%c%c%c", buf[9], buf[10], buf[11]) ;
// graba en base de datos dirigiendo los datos al programa gbase
sprintf(comando, ".gbase %d %d %d %d", atoi(c1), atoi(c2), atoi(c3), atoi(c4));
graba_archivo();
system(comando);
}
tcsetattr(fd, TCSANOW, &oldtio);
}
```


Código para almacenamiento en la BDD Mysql (gbasel.c)

```
#include <mysql/mysql.h>
#include <mysql/errmsg.h>
#include <mysql/mysql_errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <termios.h>
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>

#define FALSE 0
#define TRUE 1
#define nombre_host "localhost"
#define nombre_user "root"
#define pass "mysql"
#define nombre_base "monitoreo"

volatile int STOP=FALSE;
MYSQL *conexcion;
long respu;
MYSQL_RES *res;
MYSQL_ROW row;
unsigned int i;
int conecta_db(){
conexcion=mysql_init (NULL);
if (conexcion== NULL ){
fprintf(stderr, "\nmysql_init() ha fallado.\n");
exit(1);
}
if (mysql_real_connect(
conexcion, nombre_host, nombre_user, pass, nombre_base, 0, NULL, 0) == NULL){
fprintf(stderr, "\nmysql_real_connect() ha fallado :\n Error %u
(%s)\n",mysql_errno(conexcion),mysql_error(conexcion));
```

```
        printf("\n\n");
        mysql_close(coneccion);
        exit(0);
    }
}
int grabar_db(int v1,int v2, int v3 , int v4){
char sql[80];
struct tm *tr;
char buf[255];
char hora[8];
char fecha[10];
int i,c1,c2,c3,c4,comprueba;
time_t t;
time(&t);
tr=localtime(&t);
memset(fecha,0,10);memset(hora,0,8);
sprintf(hora,"%d:%d:%d",tr->tm_hour,tr->tm_min,tr->tm_sec);
sprintf(fecha,"%d-%d-%d",tr->tm_year+1900,tr->tm_mon,tr->tm_mday);
fflush(0);
c1=0;c2=0;c3=0;c4=0;

sprintf(sql,"INSERT INTO datos SET ID='%d' ,TIEMPO='%s', VALOR=%d,
FECHA='%s'",1,hora,v1,fecha);

if(mysql_query(coneccion,sql)!=0){
    printf("\nFallo la entrada de datos de ID=1.\n");c1=1;}
sprintf(sql,"INSERT INTO datos SET ID='%d' ,TIEMPO='%s', VALOR=%d,
FECHA='%s'",2,hora,v2,fecha);

if(mysql_query(coneccion,sql)!=0){
    printf("\nFallo la entrada de datos de ID=2.\n");c2=2;}
sprintf(sql,"INSERT INTO datos SET ID='%d' ,TIEMPO='%s', VALOR=%d,
FECHA='%s'",3,hora,v3,fecha);

if(mysql_query(coneccion,sql)!=0){
    printf("\nFallo la entrada de datos de ID=3.\n");c3=4;}
sprintf(sql,"INSERT INTO datos SET ID='%d' ,TIEMPO='%s', VALOR=%d,
FECHA='%s'",4,hora,v4,fecha);
```

```
if(mysql_query(conexion,sql)!=0){
    printf("\nFallo la entrada de datos de ID=4.\n");c4=8;}
comprueba=c1+c2+c3+c4;
if(comprueba==0)
    printf(".Si");
    else
        printf(".No");
}
int main(int argc, char *argv[]){
    conecta_db();
    grabar_db(atoi(argv[1]),atoi(argv[2]),atoi(argv[3]),atoi(argv[4]));
}
```

Archivo MAKE para crear los ejecutables

#Archivo de creación de ejecutables

```
gbase:    gbase.o
          gcc -o gbase gbase.o -L /usr/lib/mysql -lmysqlclient
gbase.o:  gbase.c
          gcc -c gbase.c
lserial : lserial.c
          gcc -o lserial lserial.c
clean:
          rm gbase.o
```

Archivo de instalación

```
make
make lserial
make clean
```

Código fuente de página de validación de usuario (user_doc.php)

```
<?php
require_once('Connections/validadoc.php');
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue = "")
{
    $theValue = (!get_magic_quotes_gpc()) ? addslashes($theValue) : $theValue;
//para guardar la variable ingresada
    switch ($theType) {
        case "text":
            $theValue = ($theValue != "") ? $theValue : "NULL";
            break;
        case "long":
        case "int":
            $theValue = ($theValue != "") ? intval($theValue) : "NULL";
            break;
        case "double":
            $theValue = ($theValue != "") ? "" . doubleval($theValue) . "" : "NULL";
            break;
        case "date":
            $theValue = ($theValue != "") ? "" . $theValue . "" : "NULL";
            break;
        case "defined":
            $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
            break;
    }
    return $theValue;
}

$editFormAction = $HTTP_SERVER_VARS['PHP_SELF'];
if (isset($HTTP_SERVER_VARS['QUERY_STRING'])) {
    $editFormAction .= "?" . $HTTP_SERVER_VARS['QUERY_STRING'];
}

if ((isset($HTTP_POST_VARS["verifica"]) && ($HTTP_POST_VARS["verifica"] == "form1")) {
    $nombre=GetSQLValueString($HTTP_POST_VARS['USUARIO'], "text");
    $pass=GetSQLValueString($HTTP_POST_VARS['PASSWD'], "text");
}

$mitexto_ingresa = $nombre;
$mitexto_ingresa1 = $pass;
mysql_select_db($database_validadoc, $validadoc);
```

```
$query_ingresa = sprintf("SELECT * FROM validadoc WHERE
validadoc.USUARIO='%s'", $mitexto_ingresa);

$ingresa = mysql_query($query_ingresa, $validadoc) or die(mysql_error());
$row_ingresa = mysql_fetch_assoc($ingresa);
$totalRows_ingresa = mysql_num_rows($ingresa);
//función para encriptar el password
$codif=md5($pass);

if($row_ingresa !=0 and $row_ingresa['PASSWORD']==$codif ){
//crea cookies de autorización
setcookie("15041972",$nombre,time()+120);
setcookie("24061976",md5($pass),time()+120);
$usrvalido = "USUARIO AUTORIZADO ";
printf ("<SCRIPT LANGUAGE='javascript'> location.href =
'monitoreo_paciente.php'; </SCRIPT> ");
}
if($row_ingresa !=0 and $row_ingresa['PASSWORD']!=$codif )
    //echo "<br class='style11'>Ingrese Datos correctos",$row_ingresa;
$mensaje = "Ingrese Datos Correctos";
if($row_ingresa ==0 and $row_ingresa['PASSWORD']!=$codif )
    //printf("<br class='style11'>Ingrese Datos
correctos",$row_ingresa);
$mensaje = "Ingrese Datos Correctos";
?>
<html>
<head>
<title>valida usuario monitoreo</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--

@import url("style.css");
@import url("style1.css");
.style11 {
    color: #000080;
    font-size: 12px;
    font-family: Georgia, "Times New Roman", Times, serif;}
```

```
.style12 {
    color: #EC7600;
    font-size: 12px;
    font-family: Georgia, "Times New Roman", Times, serif;}
.style13 {color: #000080;
    font-size: 12px;
    font-family: Georgia, "Times New Roman", Times, serif; }
-->
</style>
</head>

<BODY bgcolor="floralwhite" vLink=#333366 aLink=#333366 link=#333366>

<table width="95%" border="0" align="center" >
  <tr><td><div align="right"><a href="index.htm" class="style11">Regresar Menu
Principal</a></div></td>
  </tr></table><br><br><br><div align="center"><strong><font color="#000071" size="5">
MONITOREO DEL RITMO CARDIACO DE PACIENTES</font></strong> </div>
<br><br><br><br>

<form method="post" name="form1">
  <table align="center"><tr valign="baseline">
    <td align="right" nowrap><div align="justify" class="style13"> USUARIO:</div></td>
    <td><input type="text" name="USUARIO" value="" size="32"></td></tr>
    <tr valign="baseline"><td nowrap align="right"><div align="justify"
class="style13">PASSWORD:</div></td><td>

<input type="password" name="PASSWD" value="" size="32"></td></tr>
  <tr valign="baseline"><td colspan="2" align="right" nowrap><div align="center">
  <input style=" background-color:#FFFEEDD; color:#000071" title="Aceptar
  usuario autorizado" type="submit" value="Aceptar">

  <input name="Reset" style=" background-color:#FFFEEDD; color:#000071"
  title="Limpiar los Campos del formulario" type="reset" value="Limpiar Campos">
  </div></td></tr></table>

  <input type="hidden" name="verifica" value="form1">
</form><p>&nbsp;</p><p>&nbsp;</p>
```

```
<table width="15%" border="1" align="center" cellspacing="1"
bordercolor="#FFEEDD" bgcolor="floralwhite"><tr><td class="style12">
    <div align="right"><?php echo $mensaje; ?>
</div></td></tr></table>
```

```
<table width="15%" border="0" align="center" cellspacing="0"
bordercolor="#FFEEDD" bgcolor="floralwhite"><tr>
    <td class="style12"><div align="right"><?php echo $usrvalido; ?>
</div></td></tr></table>
</body>
</html>
```

```
<?php
mysql_free_result($ingresa);
?>
```


Código fuente de página de validación de actualización (user_doc.php)

```
<?php
require_once('Connections/dbuseradm.php');
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
    $theNotDefinedValue = "")
{
    $theValue = (!get_magic_quotes_gpc()) ? addslashes($theValue) : $theValue;
    switch ($theType) {
        case "text":
            $theValue = ($theValue != "") ? $theValue : "NULL";
            break;
        case "long":
        case "int":
            $theValue = ($theValue != "") ? intval($theValue) : "NULL"; break;
        case "double":
            $theValue = ($theValue != "") ? "" . doubleval($theValue) . "" : "NULL"; break;
        case "date":
            $theValue = ($theValue != "") ? "" . $theValue . "" : "NULL"; break;
        case "defined":
            $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue; break;
    }
    return $theValue;
}

$editFormAction = $HTTP_SERVER_VARS['PHP_SELF'];
if (isset($HTTP_SERVER_VARS['QUERY_STRING'])) {
    $editFormAction .= "?" . $HTTP_SERVER_VARS['QUERY_STRING'];
}

if ((isset($HTTP_POST_VARS["MM_insert"])) &&
($HTTP_POST_VARS["MM_insert"] == "form1"))
{
    $nombre=GetSQLValueString($HTTP_POST_VARS['usuario_adm'], "text");
    $pass=GetSQLValueString($HTTP_POST_VARS['pass_adm'], "text");
}

$mitexto_ingresa = $nombre;
$mitexto_ingresa1 = $pass;

mysql_select_db($database_dbuseradm, $dbuseradm);
$query_rsuseradm = sprintf("SELECT * FROM useradmin WHERE
useradmin.usuario_adm='%s'", $mitexto_ingresa);
$rsuseradm = mysql_query($query_rsuseradm, $dbuseradm) or die(mysql_error());
```

```
$row_rsuseradm = mysql_fetch_assoc($rsuseradm);
$totalRows_rsuseradm = mysql_num_rows($rsuseradm);
//función para encriptar el password
$codif=md5($pass);
if($row_rsuseradm !=0 and $row_rsuseradm['pass_adm']==$codif ){
//crea cookies de autorización
setcookie("15041972",$mitexto_ingresa,time()+120);
setcookie("24061976",md5($pass),time()+120);
$usrvalido = "USUARIO AUTORIZADO ";
printf ("<SCRIPT LANGUAGE='javascript'> location.href =
'actualiza_paciente.php'; </SCRIPT> ");
}
if($row_rsuseradm !=0 and $row_rsuseradm['pass_adm']!=$codif )
$mensaje = "Ingrese Datos correctos";
if($row_rsuseradm ==0 and $row_rsuseradm['pass_adm']!=$codif )
$mensaje = "Ingrese Datos correctos";
?>
<html>
<head>
<title>Validacion Usuario Administrador</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
@import url("style.css");
@import url("style1.css");
.style11 {
    color: #000080;
    font-size: 12px;
    font-family: Georgia, "Times New Roman", Times, serif;}
.style12 {
    color: #EC7600;
    font-size: 12px;
    font-family: Georgia, "Times New Roman", Times, serif;}
.style13 {color: #000080;
    font-size: 12px;
    font-family: Georgia, "Times New Roman", Times, serif; }
-->
</style>
</head>
```

```
<BODY bgcolor="floralwhite" vLink=#333366 aLink=#333366 link=#333366>
<table width="95%" border="0" align="center" ><tr><td>
  <div align="right"><a href="index.htm" class="style11">Regresar Menu Principal</a></div></td>
</tr></table><br><br><br><div align="center"><strong><font color="#000071" size="5">
<strong><font color="#000071" size="5">ACTUALIZACION DE DATOS
  DE PACIENTES DEL RITMO CARDIACO</font></strong></font></strong>
</div><br><br><br><br>

<form method="post" name="form1" action="user_adm.php">
<table align="center">
  <tr valign="baseline"><td align="right" nowrap><div align="justify"
class="style13">USUARIO:</div></td>
  <td><input type="text" name="usuario_adm" value="" size="32"></td></tr>
  <tr valign="baseline"><td nowrap align="right"><div align="justify"
class="style13">PASSWORD:</div></td>
  <td><input type="password" name="pass_adm" value="" size="32"></td></tr>
  <tr valign="baseline"><td colspan="2" align="right" nowrap><div align="center">

<input style=" background-color:#FFEEDD; color:#000071" title="Aceptar usuario autorizado"
type="submit" value="Aceptar">

<input style=" background-color:#FFEEDD; color:#000071" title="Limpia los Campos del
formulario" type="submit" value="Limpiar Campos"></div></td></tr></table>
  <input type="hidden" name="MM_insert" value="form1">
</form>
<p>&nbsp;</p><p>&nbsp;</p>
<table width="15%" border="1" align="center" cellspacing="1" bordercolor="#FFEEDD"
bgcolor="floralwhite">
  <tr><td class="style12">
    <div align="right"><?php echo $mensaje; ?></div> </td></tr></table>
<table width="15%" border="0" align="center" cellspacing="0"
bordercolor="#FFEEDD" bgcolor="floralwhite">
  <tr><td class="style12"><div align="right"><?php echo $usrvalido; ?></div></td>
  </tr></table>
</body>
</html>
<?php
mysql_free_result($rsuseradm);
?>
```

Código fuente de página para graficar resultados (monitoreo_paciente.php)

```
<?php
require_once('Connections/lista_pacientes.php');
require_once('Connections/dbdatos.php');
include("entrar.php");
if($doctor) {
mysql_select_db($database_lista_pacientes, $lista_pacientes);
$query_rslista = "SELECT ID, NOMBRE, TELEFONO, ESTADO FROM paciente";
$rslista = mysql_query($query_rslista, $lista_pacientes) or die(mysql_error());
$row_rslista = mysql_fetch_assoc($rslista);
$totalRows_rslista = mysql_num_rows($rslista);
}
else
{
printf("<body bgcolor='floralwhite'>");
printf("<br><br><br><br><br><br><br><br><br><br><br>");
printf("<font color = '#FF0000'><div align='center' class='style11'>Usuario no
autorizado para entrar a Monitoreo de Pacientes.</div></font>");
printf("<br><br><br><br><br><br><br><br><br><br><br><br><br><br>");
printf("<div align='right'><a href='index.htm' class='style11'>Regresar Menu
Principal</a></div>");
}
?>
<?php
if($doctor) {
?>
<html>
<head>
<title>Lista de Pacientes</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
@import url("style.css");
@import url("style1.css");
.style11 {
    color: #000080;
    font-size: 12px;
    font-family: Georgia, "Times New Roman", Times, serif;}

```

```
.style12 {
    color: #000080;
    font-size: 16px;
    font-family: Georgia, "Times New Roman", Times, serif;
    font-weight: bold;}
.style13 {color: #000080;
    font-size: 12px;
    font-family: Georgia, "Times New Roman", Times, serif; }
.style14 {
    font-family: verdana;
    color: #330066;
    font-size: 14px;
    font-weight: bold;}

-->
</style>
</head>
<body bgcolor="floralwhite" link=#333366 vLink=#333366 aLink=#333366>
<table width="95%" border="0" align="center" >
  <tr><td><div align="right"><a href="index.htm" class="style11">Regresar Menu
Principal</a></div></td></tr></table>
<div align="center"><strong><font color="#000071" size="5">
MONITOREO DEL RITMO CARDIACO DE PACIENTES</font></strong> </div>
<FORM ACTION="monitoreo_paciente.php" method="post" name="ingfecha">
<table width="75%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr valign="baseline">
    <td width="10" align="left">&nbsp;</td>
    <td width="119" align="left"><span class="style9">FECHA: (dd-mm-aaaa)</span></td>
      <td width="231" ><div align="left">
        <?php
          $fecha= sprintf("%s-%s-%s",$nanio, $nmes, $ndia);
          $hora= $nhora;
          $n_ani=substr($fecha,0,4);
          $n_mes=substr($fecha,5,2);
          $n_dia=substr($fecha,8,2);
          $n_hora=substr($hora,0,2);
        ?>
        <select name=ndia style='font-family:verdana;font-size: 10px' class="style9">
          <option value='-' selected></option>
          <option value=<?php echo $n_dia; ?> selected>
          <?php echo $n_dia; ?></option>
```

```
<option value='01' >01 </option>
```

...

Opciones de los días 2 hasta 30

...

```
<option value='31' >31 </option>
```

```
</select>
```

```
<select name=nmes style='font-family:verdana;font-size: 10px'  
class="style9">
```

```
<option value='--' selected></option>
```

```
<option value= <?php echo $n_mes;  
if ($n_mes == "01") $n_mes="Enero";
```

...

Se asigna el nombre del mes al valor correspondiente .

...

```
if ($n_mes == "12") $n_mes="Diciembre";
```

```
?>
```

```
selected<?php echo $n_mes;?></option>
```

```
<option value='01'>Enero </option>
```

...

Opciones de los meses 2 hasta 11

...

```
<option value='12'>Diciembre</option>
```

```
</select>
```

```
<select name=nanio style='font-family:verdana;font-size: 10px'  
class="style9">
```

```
<option value='--' selected></option>
```

```
<option value=<?php echo $n_ani; ?> selected<?php echo $n_ani;  
?></option>
```

```
<option value='2000'>2000 </option>
```

...

Opciones de los años 2001 hasta 2014

...

```
<option value='2015'>2015 </option>
```

```
</select>
```

```
</font>
```

```
</div></td>
```

```
<td align="left" width="117"><span class="style9">HORA DE INICIO: (hh)
```

```
</span></td>
```

```
<td width="253">
```

```
<select name=nhora style='font-family:verdana;font-size: 10px'  
class="style9">  
  <option value=<?php  
    echo $n_hora;  
    if (substr($n_hora,0,2)=="se" || substr($n_hora,0,2)=="")  
      $n_hora="00";  
  ?>selected><?php echo $n_hora;></option>  
  <option value='00'>00</option>  
  ...
```

Opciones de las horas 1 hasta 22

```
  ...  
  <option value='23'>23 </option>  
</select>  
</td></tr></table><br>  
<table width="75%" border="1" align="center" cellpadding="0" cellspacing="0">  
  <tr><td width="42%"><div align="justify"><font color="#000071"  
size="4">NOMBRE</font></div></td>  
  <td width="26%"><div align="center"><font color="#000071"  
size="4">ESTADO</font></div></td>  
  <td width="32%"><div align="right"><font color="#000071"  
size="4">TELEFONO</font></div></td></tr>  
<?php  
$nom = array();  
$conta=0;  
do {  
  $conta= $conta+1;  
?>  
<tr><td>  
<input name="id" type="submit" style=" background-color:#FFFEEDD;  
color:#000071" title="MOSTRAR GRAFICO" value=<?php echo $conta;?>>  
  <?php $nom[$conta]=$row_rslista['NOMBRE']; ?>  
  <span class="style13"> <?php echo $row_rslista['NOMBRE']; ?>  
</span></td>  
  <td><div align="center" class="style13">  
  <?php echo $row_rslista['ESTADO']; ?></div></td>  
  <td><div align="right" class="style13">  
  <?php echo $row_rslista['TELEFONO']; ?>
```

```
</div></td></tr>
<?php } while ($row_rlista = mysql_fetch_assoc($rlista)); ?>
</table></form>
<table border="0" align="center" cellpadding="0" cellspacing="0">
<tr><td>
<FORM ACTION="monitoreo_paciente.php" method="post" name="limpiar">
    <input name="id" type="hidden" id="id" value="0">
    <input style=" background-color:#000000; color:#66FF33 " type="submit" title="Limpiar
Grafico" value="Limpiar Grafico">
</form></td>
<td align="right" class="style12"><?php echo $nom[$id]; ?></td></tr>
<tr><td colspan="2">
    
</td></tr></table>
</body>
</html>
<?php
mysql_free_result($rlista);
?>
<?php
$pulsos = array();
mysql_select_db($database_dbdatos, $dbdatos);
$nueva_hora=substr($hora,0,2);
$tini = sprintf ("%s:00:00",$nueva_hora);
$tfm = sprintf ("%s:00:00",$nueva_hora+1);
$ver_datos = sprintf("SELECT * FROM datos where id='%s' and fecha='%s' and
tiempo >='%s' and tiempo < '%s'", $id,$fecha,$tini,$tfm);
$result = mysql_query($ver_datos, $dbdatos) or die(mysql_error());
$cuenta_dato = 0;
while ($row =mysql_fetch_row ($result)){
    $cuenta_dato = $cuenta_dato + 1;
    //grabar en un array para luego graficarlo
    $pulsos[$cuenta_dato]=$row[2];
    //printf ("%d <br>", $pulsos[$cuenta_dato]);
}
printf ("<center><font color='#000071'> %d Muestras de 360 -- tomadas durante
una Hora </font></center>",$cuenta_dato);
$inx = 50;
```



```
//grafico eje de coordenadas
```

```
$img=imagecreate(728+$inx,290); //tamaño imagen  
$fondo=imagecolorallocate($img,0,0,0);  
$marco=imagecolorallocate($img,255,0,255);  
$cuadrícula=imagecolorallocate($img,78,78,78);  
$cuadro eje=imagecolorallocate($img,150,150,150);  
$colorpix = imagecolorallocate($img,0,255,0); //color del grafico  
$letras=imagecolorallocate($img,255,255,255);  
$letrasejes=imagecolorallocate($img,255,228,181);  
imageline($img,$inx,0,$inx,250,$marco); // eje y  
imageline($img,$inx,249,750+25,249,$marco); // eje x
```

```
//grafica de la cuadrícula horizontales
```

```
imageline($img,$inx,25,720+$inx,25,$cuadrícula);  
imageline($img,$inx,50,720+$inx,50,$cuadrícula);  
imageline($img,$inx,75,720+$inx,75,$cuadrícula);  
imageline($img,$inx,100,720+$inx,100,$cuadrícula);  
imageline($img,$inx,125,720+$inx,125,$cuadrícula);  
imageline($img,$inx,150,720+$inx,150,$cuadrícula);  
imageline($img,$inx,175,720+$inx,175,$cuadrícula);  
imageline($img,$inx,200,720+$inx,200,$cuadrícula);  
imageline($img,$inx,225,720+$inx,225,$cuadrícula);
```

```
//verticales
```

```
for ($j=1; $j<60;$j++){  
    imageline($img,12*$j +$inx,0,12*$j + $inx,250,$cuadrícula);  
}
```

```
//numeros eje y "estilo,x,y"
```

```
imagestring($img, 2, 25, 20, "225", $letras );  
imagestring($img, 2, 25, 45, "200", $letras );  
imagestring($img, 2, 25, 70, "175", $letras );  
imagestring($img, 2, 25, 95, "150", $letras );  
imagestring($img, 2, 25, 120, "125", $letras );  
imagestring($img, 2, 25, 145, "100", $letras );  
imagestring($img, 2, 25, 170, "75", $letras );  
imagestring($img, 2, 25, 195, "50", $letras );  
imagestring($img, 2, 25, 220, "25", $letras );  
imagestring($img, 2, 25, 245, "0", $letras );  
imagestringup($img, 2, 5, 175, "pulsos/minuto", $letrasejes );
```

```
//numeros eje x
for ($k=0;$k<13;$k++){
    imagestring($img, 3, $inx +60*$k-5, 258,$k*5,$letras );
    if ($k>0) imageline($img,60*$k +$inx,0,60*$k + $inx,254,$cuadroje);
}
    imagestring($img, 2, 325, 275,"tiempo de muestreo",$letrasejes );
    imagestring($img, 2, 738, 275,"1 hora",$letrasejes );

//para titulo en la imagen
//imagestring($img, 1, 8, 15," 1",$letras );imagestring($img, 1, 8, 25," 2",$letras );
//cargar los valores de la revision de los datos de la hora de monitoreo
//grafico
for ($i=0;$i<360;$i++)
{
    $pulsos[$i]=0;
}

    $x1 = 0;
    $y1 = 250 - $pulsos[0];
    for ($i=1;$i<360;$i++)
    {
        $segundos=$i;
        $x2 = $i * 2;
        $y2 = 250 - $pulsos[$i];
// Grafica lineas en los ejes
        imageline($img,$x1+$inx,$y1,$x2+$inx,$y2,$colorpix);
//guarda los valores iniciales de la coordenada inicial
        $x1 = $x2;
        $y1 = $y2;
    }
    imageline($img,$inx,249,750,249,$marco); // eje x
    imagepng($img,"grafico_xy.png");
    imagedestroy($img); //destruye la imagen de memoria
mysql_free_result($result);
?>
<?php
}
?>
```

Código fuente de autorización de página de monitoreo (entrar.php)

```
<?php
require_once('Connections/validadoc.php');
$doctor = false;
//variables para almacenar usuario y password respectivamente: "15041972" ;
"24061976"
if(isset($_HTTP_COOKIE_VARS["15041972"]) &&
isset($_HTTP_COOKIE_VARS["24061976"]))
{
mysql_select_db($database_validadoc, $validadoc);
$consulta=sprintf("SELECT * FROM validadoc WHERE
USUARIO='".$_HTTP_COOKIE_VARS["15041972"]."' AND
PASSWD='".$_HTTP_COOKIE_VARS["24061976"]."'");
$ingresa = mysql_query($consulta);
if($row = mysql_fetch_array($ingresa)){
setcookie("15041972",$_HTTP_COOKIE_VARS["15041972"],time()+3600);
setcookie("24061976",$_HTTP_COOKIE_VARS["24061976"],time()+3600);
$doctor = true;
}
else
{
//Destruimos las cookies.
setcookie("15041972","x",time()-120);
setcookie("24061976","x",time()-120);
}
mysql_free_result($ingresa);
}
?>
```

4.5 CONSTRUCCIÓN DEL PROTOTIPO BASADO EN LOS MÓDULOS DEL SISTEMA DESARROLLADOS

Se ha diseñado los módulos del sistema de modo que desempeñe las siguientes actividades principales:

- Medir el ritmo cardiaco
- Adquirir y almacenar datos del ritmo cardiaco
- Ofrecer información de monitoreo del ritmo cardiaco
- Acceder a la información de monitoreo de ritmo cardiaco remotamente.

Medir el ritmo cardiaco

El dispositivo de medida del ritmo cardiaco tiene la facultad de medir el pulso y entregar dicha medida al computador que contiene la base de datos (MySQL/Linux), la cual almacena el ID ⁷, tiempo, valor y fecha durante el seguimiento del ritmo cardiaco.

Dicho dispositivo usa un cable serial que está conectado al computador en el puerto serie /dev/ttyS0.⁸

Una vez realizada el establecimiento de la comunicación del dispositivo de medida con el computador, ésta es de manera continua, y es interrumpida solamente si falla el cable serial o existen problemas en el proceso de lectura del puerto que se encuentra ejecutándose en el computador.

⁷ Identificador del paciente.

⁸ En DOS y Windows este puerto se lo nombra COM1.

Adquirir y almacenar datos del ritmo cardiaco

Cada dato leído por el puerto serial es procesado y almacenado en la tabla (monitoreo) de la base de datos, el dispositivo transmite cada 10 segundos una medida de los sensores del ritmo cardiaco por lo que se grabarán con el mismo intervalo de tiempo.

Ofrecer información de monitoreo del ritmo cardiaco

El prototipo dispone de un módulo de 11 páginas Web diseñadas para un servidor Web que dará servicio al Cliente (médico) y a través de ellas obtener información de monitoreo del ritmo cardiaco de sus pacientes.

Se ha diseñado páginas Web de control de acceso (user_doc.php, user_doc_real.php, user_adm.php) para permitir el ingreso a otras páginas Web diseñadas tanto para el monitoreo de pacientes (monitoreo_paciente.php, monitoreo_paciente_real.php) como para la actualización de datos (actualiza_paciente.php) del mismo.

Las páginas Web de control de acceso verifican la autenticidad del usuario, al ser correcto, activa una cookie (con duración de 1 hora) en el computador del usuario (médico) y podrá ingresar a la(s) página(s) Web que desee.

Mientras utiliza dichas páginas Web, éstas estarán activas, sin embargo si la página Web tanto de monitoreo de pacientes como la página Web de actualización de datos de los mismos no detecta ninguna actividad por 1 hora, la(s) cookie encaminará al programa hacia la desactivación y muestra al usuario el mensaje de que ya no dispone de autorización, el mensaje generado obliga al usuario (médico) volver a autenticarse.

La página Web de monitoreo de paciente, está diseñada para mostrar los resultados de monitoreo en una grafico de coordenadas (pulsos/min.vs.tiempo).

En la página de monitoreo de pacientes histórico permite seleccionar la fecha, hora de monitoreo y nombre del paciente para verificar cada uno en la base de datos y así presentar los resultados.

Este procedimiento se repite cuantas consultas desee el médico realizar a sus pacientes en la fecha u hora indicada.

En la página de monitoreo de pacientes en tiempo real permite seleccionar el nombre del paciente para verificarlo en la base de datos y así presentar los resultados de la ultima hora de monitoreo, por tanto, los últimos datos monitoreados.

Los datos son extraídos y mostrados en el gráfico cada 10 segundos mediante la recarga de la página.

El funcionamiento de la página Web de actualización de pacientes, permite tener la capacidad para autorizar al usuario (médico) actualizar los datos de los pacientes que se encuentra monitoreando.

Acceder a la información de monitoreo de ritmo cardiaco remotamente.

El médico tiene a su disposición las siguientes páginas Web desarrolladas en el servidor Web:

- Página de inicio.
- Página de control de acceso.
- Página de monitoreo del ritmo cardiaco.
- Página de actualización de datos.

Página de inicio

Muestra información de bienvenida al prototipo y las opciones para ingreso a las páginas de actualización y monitoreo de pacientes.

Página de control de acceso

Cuando el médico ingresa su usuario y su password para obtener autorización al ingreso del resto de páginas, muestra un mensaje de “usuario autorizado”, si los datos ingresados son correctos, o “ingrese datos correctos”, si los datos no son válidos.

Página de monitoreo de pacientes

El médico ingresa los datos de fecha y hora de monitoreo para los pacientes que desee realizar la consulta de su ritmo cardiaco y visualizar los resultados en el grafico de coordenadas.

Página de actualización de datos

Muestra al médico o a una persona autorizada los datos de los pacientes que son monitoreados y permite actualizarlos.

A continuación (Figura 4.10) se presenta el modelo del prototipo diseñado.

PROTOTIPO DEL SISTEMA

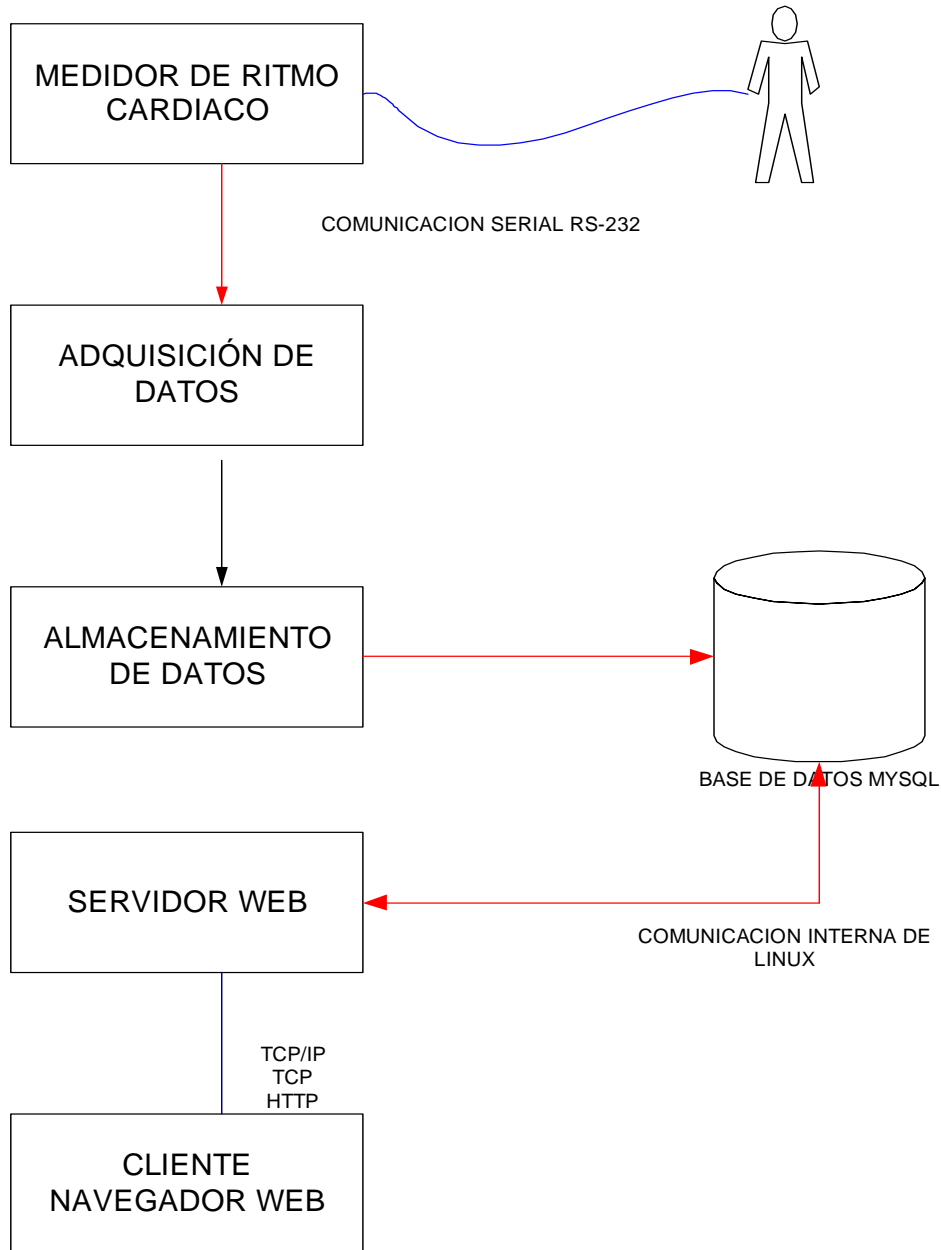


Figura 4.10. Prototipo del sistema de Monitoreo del ritmo cardíaco. (Fuente de Autores).

CAPITULO 5

PRUEBAS

5.1 PRUEBAS DE INSTALACIÓN Y FUNCIONAMIENTO

Lo primero que se realizó para la instalación fue verificar que el servidor Web Apache instalado en el servidor Linux este iniciado:

Se subió el servicio mediante el siguiente comando:

```
service httpd start
```

Se verificó que el modulo PHP esté instalado, para esto se creó una página con el siguiente código:

```
<?php  
echo phpinfo();  
?>
```

y se lo grabó en el directorio /var/www/html/ con el nombre de prueba.php, luego mediante el navegador Web se ejecutó:

```
http://localhost/prueba.php
```

Se desplegó la información de todas las características y módulos de PHP, con esto nos indicó que la instalación estuvo correcta.

El siguiente paso de la instalación fue revisar la conexión del módulo medidor de ritmo cardiaco al paciente, el cual se lo conectó al dedo índice de la mano del paciente.

Una vez que la conexión entre el dispositivo de monitoreo y el computador funcionó correctamente se inició el programa “Iserial” que se encarga de monitorear el puerto serial y grabar en la base de datos. . (Figura 5.9).

Luego de que la prueba instalación tuvo éxito se procede a realizar las pruebas para el monitoreo del paciente.

Estas pruebas se realizaron desde un cliente plataforma Windows por medio de un navegador Web.

Se ejecutó la dirección <http://10.0.0.1/monitoreo/index.html> en el navegador Web se obtuvo los siguientes resultados:

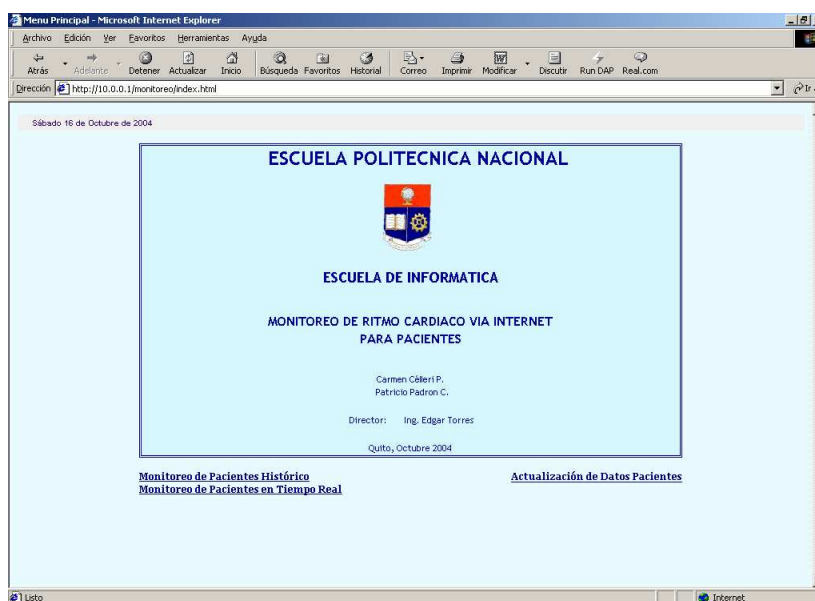


Figura 5.1 Pantalla de bienvenida. (Fuente de Autores).

Se seleccionó la opción de “Monitoreo de Paciente Histórico” solicita el ingreso de usuario y password (Figura 5.1).

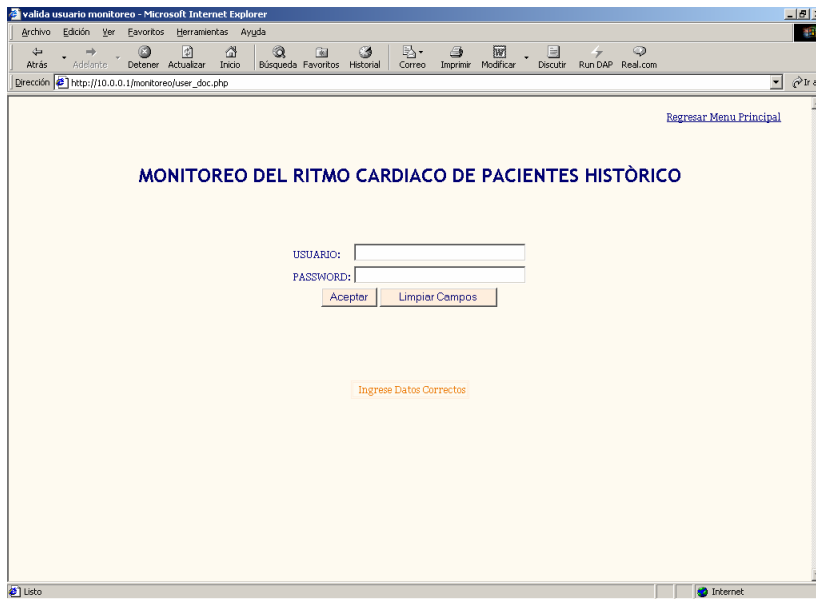


Figura 5.2 Pantalla de login para ingreso a monitoreo histórico. (Fuente de Autores).

Al ingresar datos de usuario y password erróneos el sistema mostró la misma pantalla de monitoreo de pacientes de la Figura 5.2.

Al ingresar los datos correctamente permitió acceder a la página de monitoreo de pacientes histórico (Figura 5.3).



Figura 5.3 Pantalla de monitoreo del ritmo cardiaco histórico. (Fuente de Autores).

Se ingresó los siguientes datos:

- Fecha de monitoreo = 20 de octubre del 2001
- hora de monitoreo = 00
- seleccionar paciente = 1
-

Se presentó como resultado el gráfico mostrado en la Figura 5.4



Figura 5.4 Pantalla de resultado de monitoreo del ritmo cardiaco histórico del paciente 1. (Fuente de Autores).

Se regresó al menú principal para ingresar a la opción de “Monitoreo de Pacientes en Tiempo Real”, se ingreso de usuario y password de la misma manera que al ingresar a “Monitoreo de Pacientes Histórico” (Figura 5.2).

Al ingresar los datos correctamente permitió acceder a la página de monitoreo de pacientes en tiempo real (Figura 5.5).



Figura 5.5 Pantalla de monitoreo del ritmo cardiaco en tiempo real. (Fuente de Autores).

Se seleccionó al paciente 1 mostrando como resultado el gráfico de la Figura 5.6

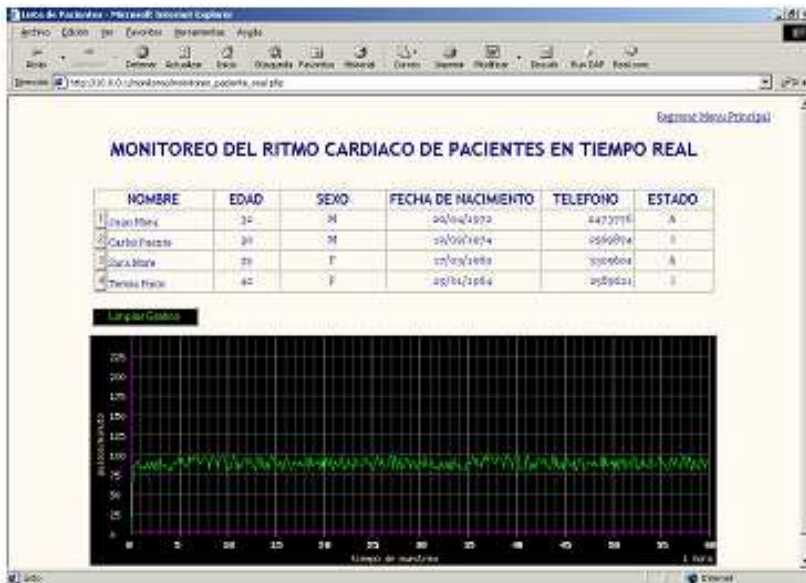
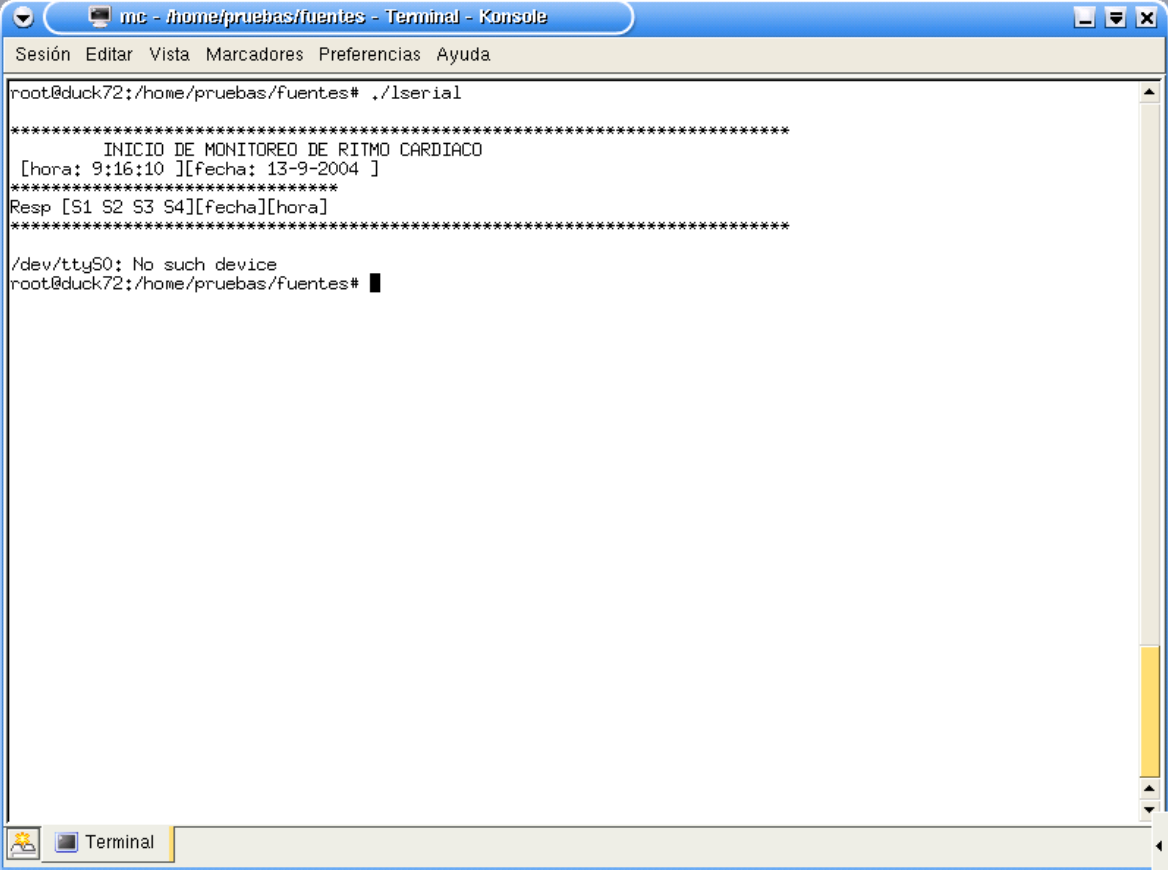


Figura 5.6 Pantalla de resultado de monitoreo de ritmo cardiaco en tiempo real del paciente 1. (Fuente de Autores).

5.2 RESULTADOS DE LAS PRUEBAS

Pruebas realizadas con resultados erróneos

Al ejecutar el programa lserial en el servidor linux por primera vez se visualizó la ejecución del programa de monitoreo cardiaco, en el servidor Linux. (Figura 5.7).

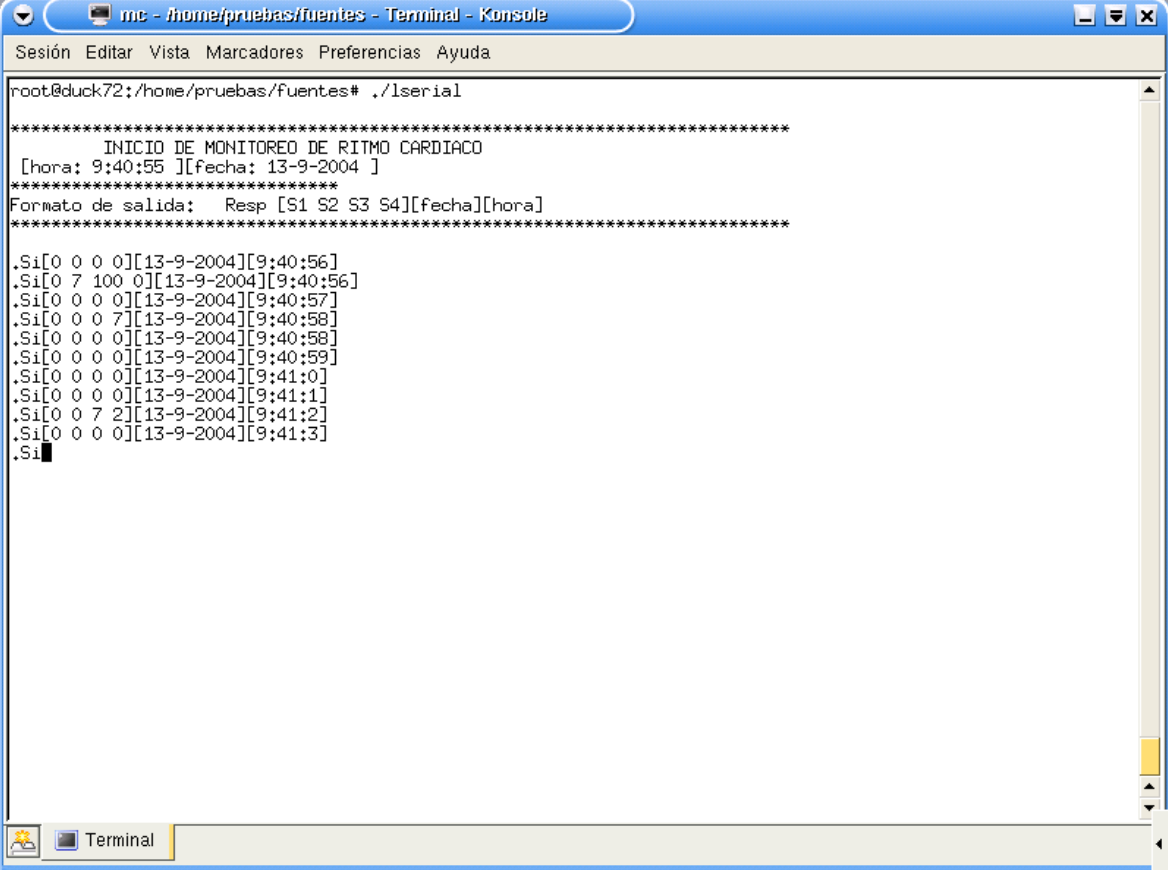


```
mc - /home/pruebas/fuentes - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
root@duck72:/home/pruebas/fuentes# ./lserial
*****
          INICIO DE MONITOREO DE RITMO CARDIACO
[hora: 9:16:10 ][fecha: 13-9-2004 ]
*****
Resp [S1 S2 S3 S4][fecha][hora]
*****
/dev/ttyS0: No such device
root@duck72:/home/pruebas/fuentes#
```

Figura 5.7 Pantalla de ejecución del programa “lserial” con error. (Fuente de Autores).

El error fue ocasionado por que el cable serial se habia desconectado del dispositivo de medicion cardiaca.

Al ejecutar el programa lserial en el servidor linux se visualizó la ejecución del programa de monitoreo cardiaco, en el servidor Linux. (Figura 5.8).



```
mc - home/pruebas/fuentes - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
root@duck72:/home/pruebas/fuentes# ./lserial
*****
          INICIO DE MONITOREO DE RITMO CARDIACO
[hora: 9:40:55 ][fecha: 13-9-2004 ]
*****
Formato de salida:  Resp [S1 S2 S3 S4][fecha][hora]
*****

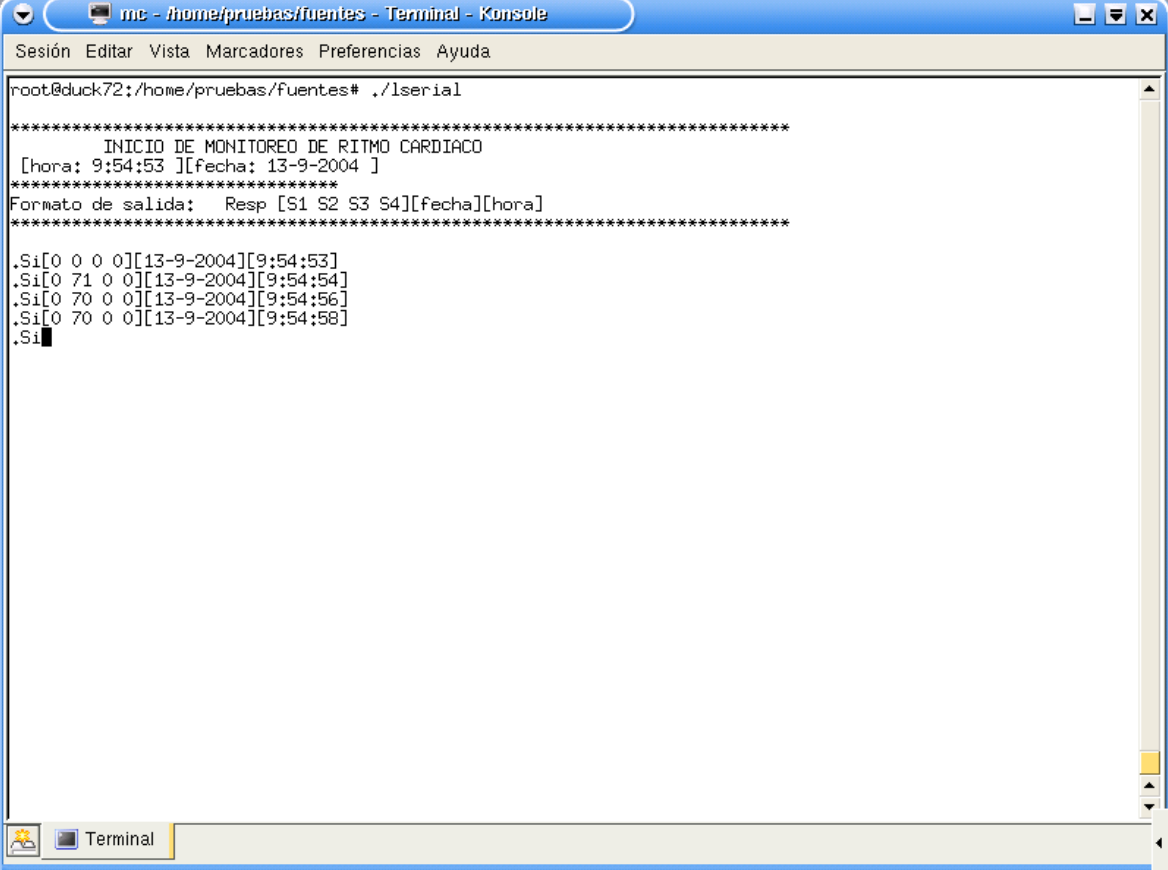
.Si[0 0 0 0][13-9-2004][9:40:56]
.Si[0 7 100 0][13-9-2004][9:40:56]
.Si[0 0 0 0][13-9-2004][9:40:57]
.Si[0 0 0 7][13-9-2004][9:40:58]
.Si[0 0 0 0][13-9-2004][9:40:58]
.Si[0 0 0 0][13-9-2004][9:40:59]
.Si[0 0 0 0][13-9-2004][9:41:0]
.Si[0 0 0 0][13-9-2004][9:41:1]
.Si[0 0 7 2][13-9-2004][9:41:2]
.Si[0 0 0 0][13-9-2004][9:41:3]
.Si█
```

Figura 5.8 Pantalla de ejecución del programa “lserial” con error de lectura. (Fuente de Autores).

El error fue detectado y se debio a un tamaño incorrecto del buffer en el programa lserial.c, fue corregido.

Pruebas realizadas con resultados exitosos

Se visualizó la ejecución del programa de monitoreo cardiaco con un paciente, sin errores, en el servidor Linux. (Figura 5.9).



```
mc - /home/pruebas/fuentes - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
root@duck72:/home/pruebas/fuentes# ./lserial
*****
          INICIO DE MONITOREO DE RITMO CARDIACO
[hora: 9:54:53 ][fecha: 13-9-2004 ]
*****
Formato de salida: Resp [S1 S2 S3 S4][fecha][hora]
*****
.Si[0 0 0 0][13-9-2004][9:54:53]
.Si[0 71 0 0][13-9-2004][9:54:54]
.Si[0 70 0 0][13-9-2004][9:54:56]
.Si[0 70 0 0][13-9-2004][9:54:58]
.Si█
```

Figura 5.9 Pantalla de ejecución del programa “lserial” sin errores. (Fuente de Autores).

Luego de las pruebas realizadas exitosamente se procedio a permitir a un medico efectuar un test al Sistema de Monitoreo Cardiaco via Internet, dando la siguiente crítica y opinión:

“El sistema de monitoreo cardíaco que he probado tiene un alcance muy grande en la práctica de medicina real debido a que da una gran ayuda a los doctores debido a que se puede medir este parámetro físico a varios pacientes a la vez sin movernos del consultorio. La información básica que entrega es de mucha ayuda en el seguimiento del estado de un paciente.

Realmente la operación del sistema requiere de muy pocos conocimientos de Internet.”

Dr. Hugo Correa. (Médico General)

CAPITULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- El proyecto “Monitoreo de Ritmo cardiaco vía Internet para pacientes”, pone de manifiesto la capacidad de innovar tecnología en nuestro país, y a pesar de que el proyecto se inspiró en un sistema que data de algún tiempo atrás, el mismo es totalmente original y difiere de aquel, tanto en su estructura física como en los métodos empleados para establecer la comunicación. Quizá lo único que compartan sea el uso del Internet como medio de transporte de datos.

- La utilización de telemetría a través de Internet constituye un método simple y fiable para realizar medidas de eventos a distancia, debido a que los medios para su elaboración no representa mayor dificultad de puesta en marcha, esto es, porque al tener Internet ya se cuenta con el medio físico de transporte y al usar el protocolo TCP/IP que es un protocolo fiable ya que al utilizar TCP está garantiza la entrega de datos.

- La primicia de la tecnología no sería del todo factible si no se encamina hacia una optimización de los recursos humanos, materiales y económicos. De este modo, el proyecto de “Monitoreo de ritmo cardiaco vía Internet para pacientes” puede ser utilizado como una herramienta importante para ayudar en el trabajo de los médicos al momento de realizar un diagnostico del estado del corazón en sus pacientes.

- Los elementos de los que está constituido el sistema son de uso común en el área informática con la única excepción del dispositivo de medidor de pulso cardiaco, el cual, se lo puede conseguir a precios de alrededor de \$300 en cualquier casa comercial de equipo médico en el país.
- Para enlazar el sistema de monitoreo de pulsos cardiacos, que puede utilizarse para una sola persona o hasta cuatro personas; solo es necesario que esté conectado a una computadora y ésta a su vez a Internet para que pueda realizarse el enlace, en este sentido, se elimina las barreras de distancia y costo que impiden o encarecen la transmisión móvil de datos.
- El sistema de monitoreo de ritmo cardiaco en pacientes presenta solamente resultados de dicho monitoreo más no de diagnóstico.

6.2. RECOMENDACIONES

Al diseñar específicamente un prototipo del sistema de Monitoreo, no resultaba del todo práctico el implementar algunas opciones, que aparecieron en el transcurso del desarrollo del mismo, que si bien no incrementan su rendimiento, si podían ser mejoras para su utilización. A continuación se detallan algunas de estas opciones.

- Una opción en el servicio que pudiera prestar a futuro el prototipo, seria la de utilizar como dispositivo de medida del ritmo cardiaco un ECG (electrocardiograma) para obtener una respuesta fiel del estado del corazón del paciente. Dicha implementación estaría encaminada a disponer del dispositivo ECG y cambiar los parámetros de lectura del puerto serial.

- Una característica que se podría agregar al prototipo sería el de diagnosticar problemas cardiacos, de tal manera que el médico obtuviera un estado médico del corazón del paciente durante y/o al final del monitoreo. La implementación de esta utilidad implicaría que exista una tabla en la base de datos con las características de los problemas cardiacos y los parámetros bajo los cuales suceden.
- Se puede implementar al prototipo las opciones de medir por ejemplo: temperatura, presión en los pacientes.

BIBLIOGRAFÍA

- Rich Bowen/Ken Coar,"Servidor Apache al descubierto", Prentice Hall, 2000.
- Ashish Wilfred/Meeta Gupta, "Proyectos Profesionales PHP", Anaya Multimedia, 2002.
- Thomas Schenk,"Administración de Red Hat Linux al Descubierto", Prentice Hall, 2001.
- John Ray, "Edición Especial TCP/IP", Prentice Hall, 1999.
- Marco A. Salgado B., "Manual de Electrocardiografía", Editorial Universitaria, 1975
- Sean Walton, "Programación de Socket Linux", Prentice Hall, 2001
- Kris Jamsa/ Ken Cope, "Programación en Internet", McGRAW-HILL, 1996
- Cricket Liu/ Jerry Peek, "Administración de Servicios de Información en Internet", 1997.
- Alvaro Del Castillo, "Revista Linux Actual, Php para desarrollo de aplicaciones Web", Prensa Técnica, 1998.
- Paul Dubois, "Edición Especial MySQL", Prentice Hall, 2001
- Kurt Wall, "Programación en Linux Al descubierto", Prentice Hall, 2001.

- José M. Canosa, "Programación avanzada en Unix", McGRAW-HILL, 2000.
- Rick Stout, "World Wide Web Manual de Referencia", McGraw-Hill, 1996.
- Carlos Gallús Lahoz, "Utilización de PHP en la creación de cookies", <http://www.webestilo.com/php/php13c.phtml>

ANEXOS

A. MANUAL DE USUARIO

En la página de inicio se tiene la opción Monitoreo de pacientes tanto histórico como en tiempo real para ingresar al monitoreo de los pacientes y la opción Actualización de datos de pacientes. (Figura A.1).

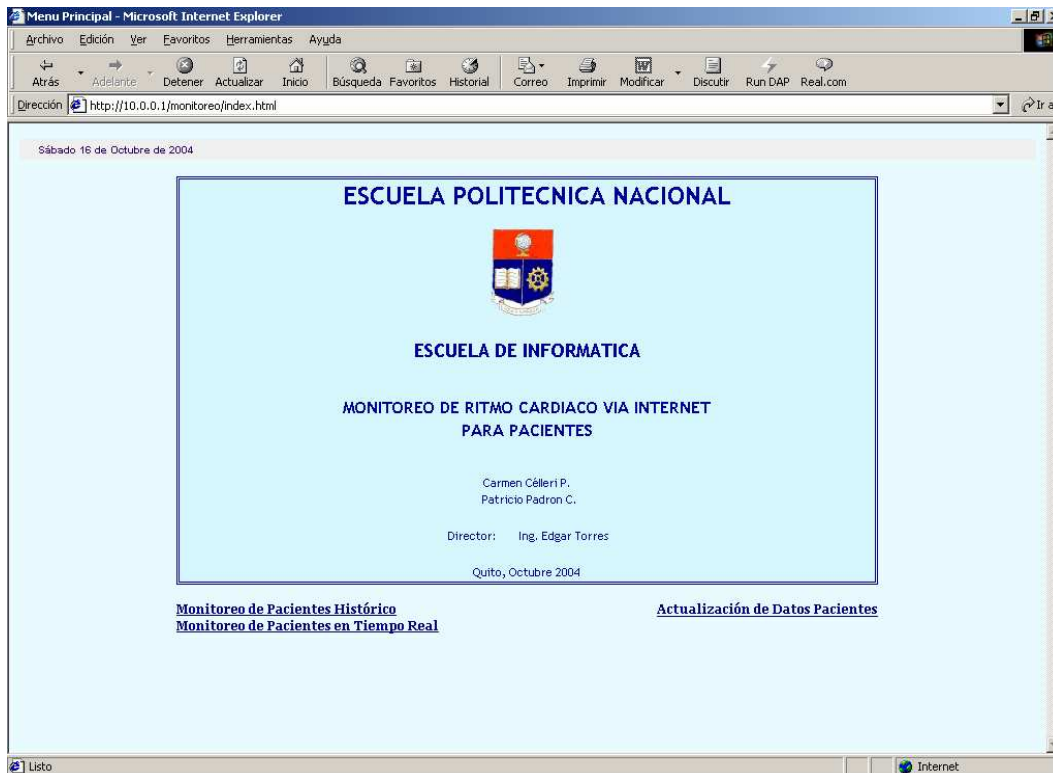


Figura A.1 Pantalla de inicio de sesión que presenta el Menú Principal. (Fuente de Autores).

Ingreso a la Página de Monitoreo de Pacientes Histórico

Al ingresar a la opción de “monitoreo de pacientes histórico” se presenta la página de validación de usuario (Figura A.2), esta página permite tener acceso al usuario a las siguientes páginas.

Ingrese el usuario y el password para su validación.

Si el usuario y el password ingresados NO son válidos, se indica el mensaje “Ingrese Datos Correctos”.

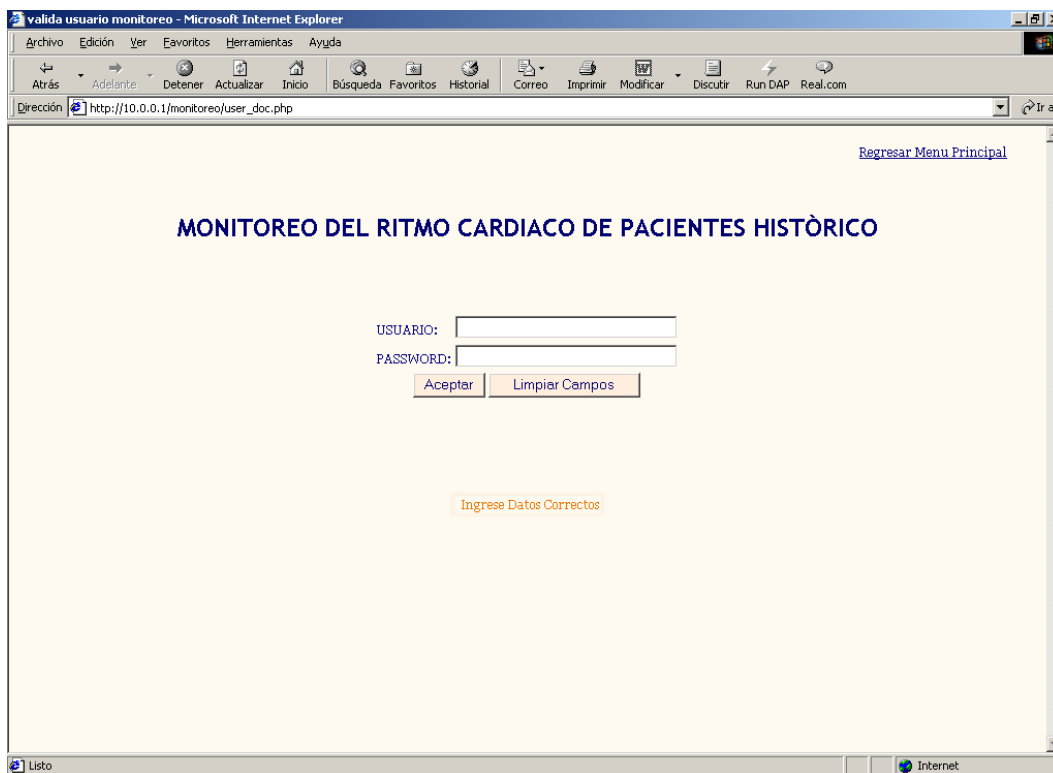


Figura A.2 Pantalla para validar usuario y password de monitoreo histórico. (Fuente de Autores).

Si el usuario y el password son correctos, entonces, se indica el mensaje “Usuario Autorizado” y automáticamente se lo envía a la página de monitoreo de pacientes.

Caso contrario si el usuario no está autorizado, se muestra el mensaje que el usuario no es autorizado para dicha página. (Figura A.3).

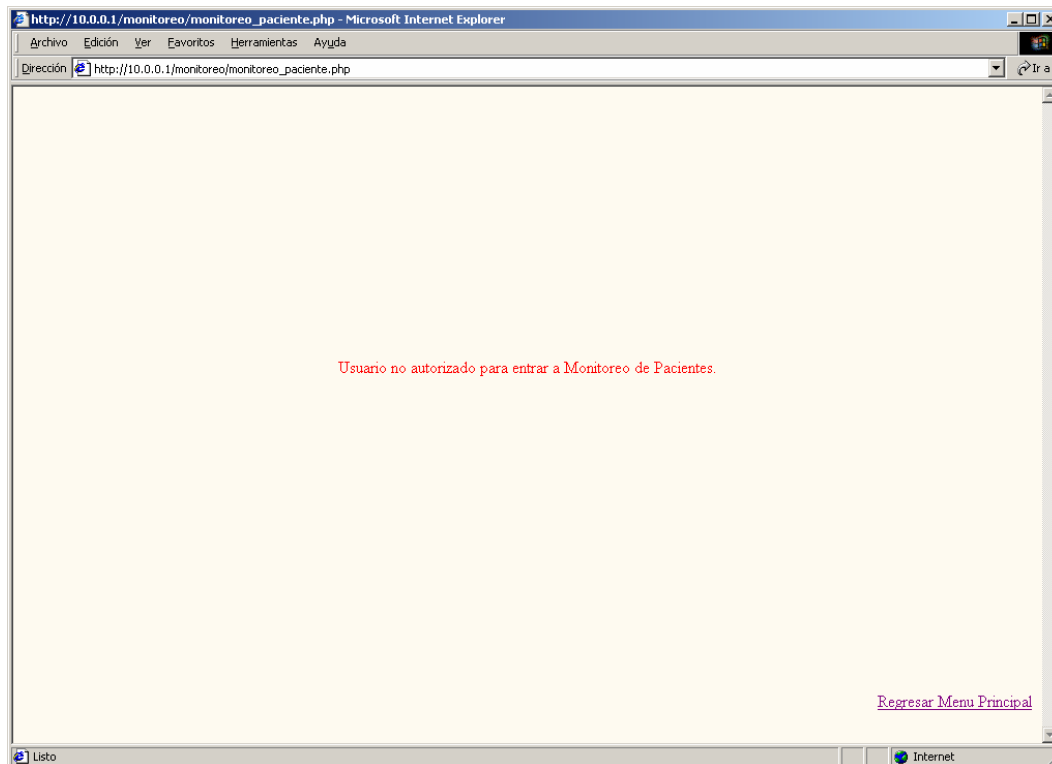


Figura A.3 Pantalla mensaje de usuario no autorizado al monitoreo. (Fuente de Autores).

Al momento de ingresar a la página de monitoreo de pacientes histórico, se presentan las siguientes opciones: (Figura A.4).

- Selección de Fecha de Monitoreo.
- Selección de Hora de Monitoreo.
- Limpiar Gráfico.

Para realizar la consulta, debe seleccionar la Fecha (en formato “dd/mm/aaaa”) y la hora de Monitoreo en la que desea ver los datos de monitoreo del paciente.

Luego de ingresar la Fecha y la Hora de Monitoreo, se selecciona el nombre del paciente, los resultados se muestran en el gráfico de coordenadas (pulsos/min.Vs.tiempo),

Si el gráfico no muestra resultados, quiere decir que el paciente no esta conectado para el monitoreo y no se ha registrado ningún dato para éste.

El gráfico muestra una escala máxima en los pulsos por minutos de hasta 250 pulsos, y en la escala de tiempo de muestreo de una hora.

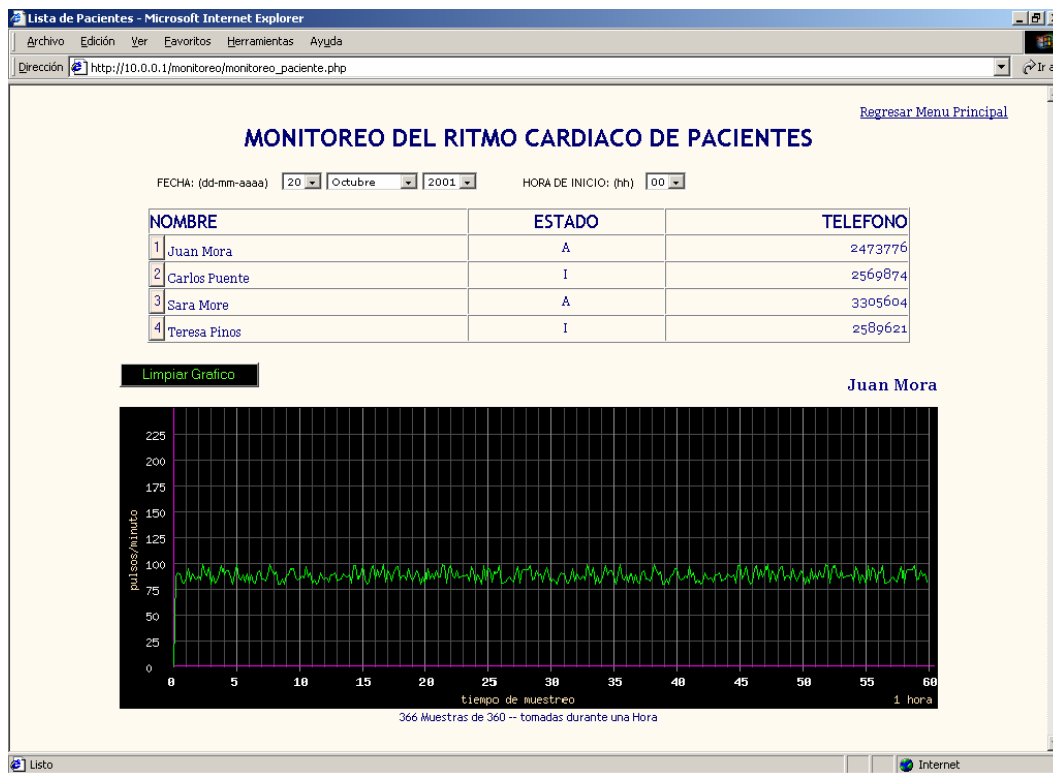


Figura A.4 Pantalla de monitoreo del ritmo cardiaco histórico. (Fuente de Autores).

La opción “Limpiar Gráfico”, permite actualizar los resultados.

Si desea consultar el monitoreo para más de un paciente en una misma fecha u hora, basta con ingresar una vez la fecha y la hora de monitoreo, y solamente seleccionar el paciente que desea observar.

De la misma manera, si desea cambiar las otras opciones de fecha u hora, basta con seleccionarla y luego ir al paciente para mostrar los resultados en el gráfico.

Si la página se encuentra inactiva por un lapso de Una Hora, se inhabilitará mostrando el mensaje "Usuario no autorizado para entrar a Monitoreo de Pacientes" y tendrá que ingresar nuevamente a la sesión de inicio para conseguir autorización.

La opción "Regresar Menú Principal", permite ir a la página de inicio de sesión donde nuevamente tiene las opciones para ingresar a la página que desee.

Ingreso a la Página de Monitoreo de Pacientes en Tiempo Real

Al ingresar a la página de monitoreo de pacientes en tiempo real, de igual manera que al ingresar a la página de monitoreo de pacientes histórico, primero se valida el usuario con su respectivo password.

Ingrese correctamente su usuario y password, si sus datos son válidos, ingresa a la página para monitorear a los pacientes en tiempo real. (Figura A.5)



Figura A.5 Pantalla de monitoreo del ritmo cardiaco en tiempo real. (Fuente de Autores).

Ingreso a la Página de Actualización de Datos de Pacientes

Como se dijo anteriormente, en la página de inicio, se selecciona la opción de Actualización de datos de pacientes.

Entonces, se redirigirá a la página de validación de usuario (Figura A.6) que da la autorización a la página de actualización de datos.

Ingrese un usuario y un password para comprobar si son autorizados o no a realizar dichas actualizaciones.

Si el usuario y el password ingresados NO son válidos, se indica el mensaje “Ingrese Datos Correctos”.

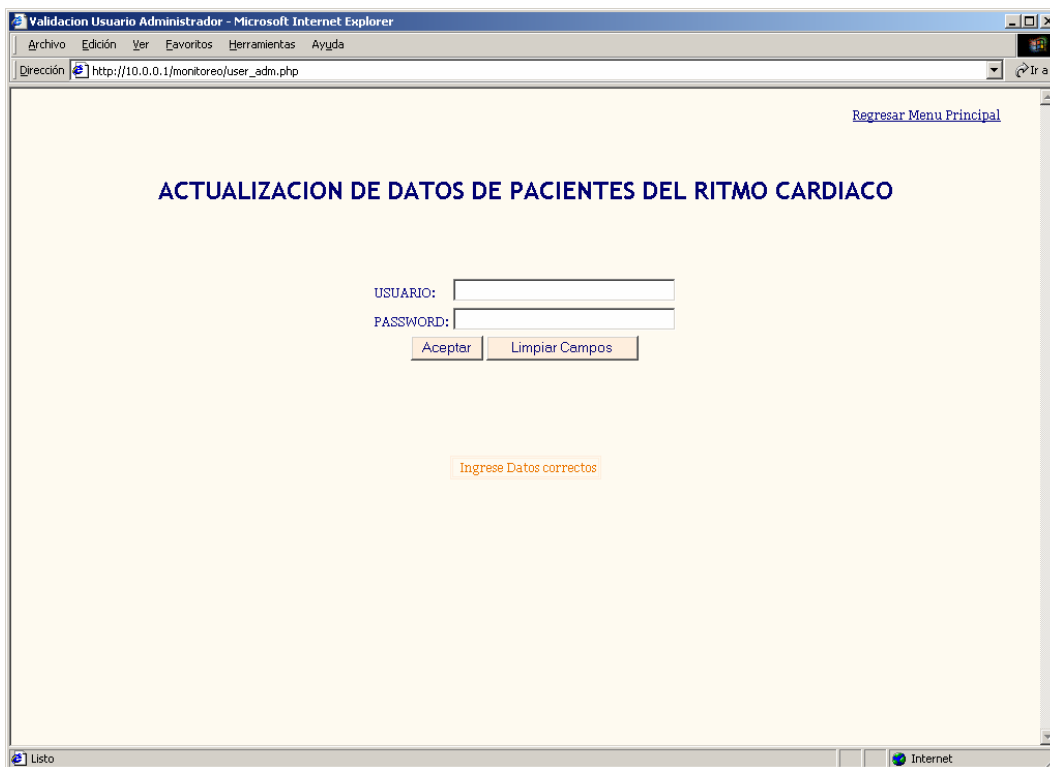


Figura A.6 Pantalla para validar usuario y password de actualización de datos de pacientes. (Fuente de Autores).

Si el usuario y el password son correctos, entonces, se indica el mensaje “Usuario Autorizado” y automáticamente se lo envía a la página de Actualización de datos de pacientes.

Caso contrario si el usuario no está autorizado, se muestra el mensaje que el usuario no es autorizado para dicha página. (Figura A.7).

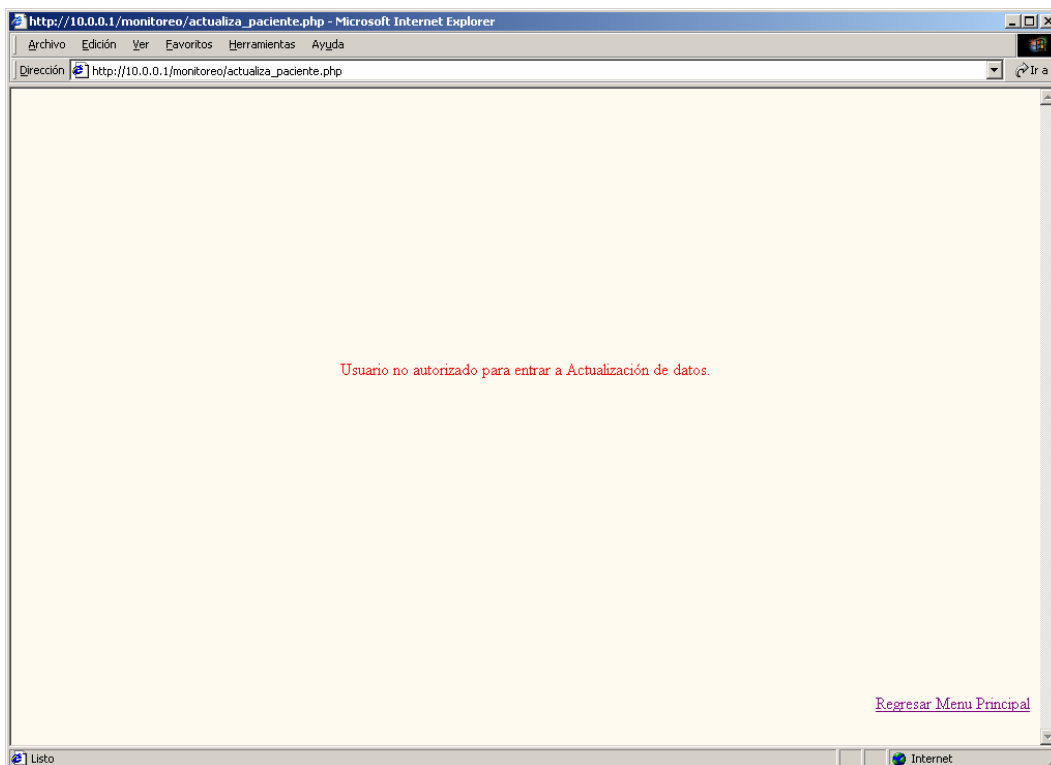


Figura A.7 Pantalla de mensaje usuario no autorizado a la actualización. (Fuente de Autores).

En la página de actualización de datos pacientes (Figura A.8), le permite cambiar los siguientes datos:

- CÉDULA
- NOMBRE
- EDAD
- SEXO
- FECHA DE NACIMIENTO
- TELÉFONO
- DIRECCIÓN
- ESTADO (sí el paciente se encuentra en monitoreo o no.)

Tiene la opción de “Actualizar Datos“, que le indicará que la actualización ha sido realizada, o “Cancelar” si no desea ejecutar la actualización y si existió un error de escritura, dejar los campos sin actualizar.

Use la barra de desplazamiento para ver los datos de los pacientes.

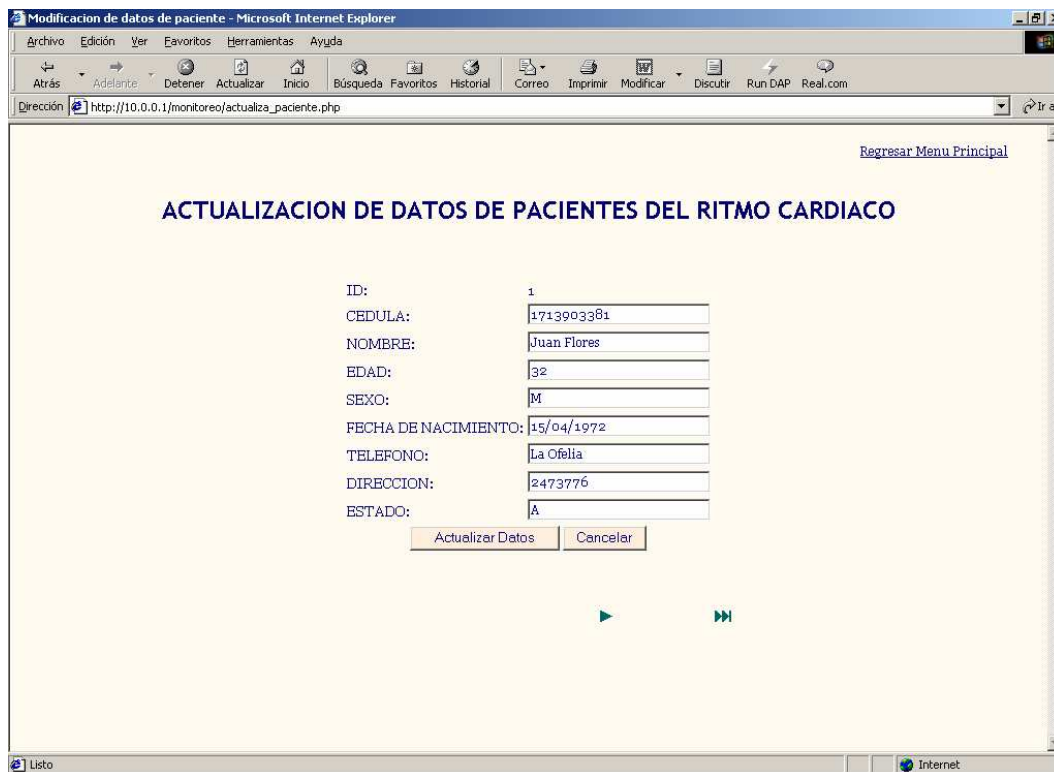


Figura A.8 Pantalla de Actualización de Datos de pacientes. (Fuente de Autores).

La opción “Regresar Menú Principal”, para ir a la página de inicio de sesión donde nuevamente tiene las opciones para ingresar a la página que desee.

Si la página se encuentra inactiva por un lapso de quince minutos, se inhabilitará mostrando el mensaje “Usuario no autorizado para entrar a Actualización de Datos de Pacientes” y tendrá que ingresar nuevamente a la sesión de inicio para conseguir autorización.

B. MANUAL TÉCNICO

Descripción de la base de datos “monitoreo” utilizada para el almacenamiento de datos de monitoreo y datos básicos del paciente.

```
# Base de datos: `monitoreo`
# -----
# Estructura de tabla para la tabla `datos`
#
# Creación: 11-10-2004 a las 14:18:37
# Última actualización: 11-10-2004 a las 22:09:57
#
CREATE TABLE `datos` (
  `ID` char(1) NOT NULL default '0',
  `TIEMPO` time default NULL,
  `VALOR` int(3) default NULL,
  `FECHA` date default NULL
)
#
# Estructura de tabla para la tabla `paciente`
#
# Creación: 08-10-2004 a las 19:52:11
# Última actualización: 11-10-2004 a las 13:45:32
#
CREATE TABLE `paciente` (
  `ID` char(1) NOT NULL default "",
  `CEDULA` varchar(10) NOT NULL default '0',
  `NOMBRE` varchar(25) default NULL,
  `EDAD` char(3) NOT NULL default "",
  `SEXO` char(1) NOT NULL default "",
  `NACIMIENTO` varchar(10) NOT NULL default "",
  `DIRECCION` varchar(30) default NULL,
  `TELEFONO` varchar(9) default NULL,
  `ESTADO` varchar(10) NOT NULL default "",
  PRIMARY KEY (`CEDULA`)
) TYPE=MyISAM;
```

Descripción de la base de datos “doctores” utilizada para el almacenamiento de usuario y password de acceso para monitoreo y actualización de datos del paciente.

```
# Estructura de tabla para la tabla `useradmin`
#
# Creación: 09-10-2004 a las 11:07:10
# Última actualización: 09-10-2004 a las 11:07:13
CREATE TABLE `useradmin` (
  `usuario_adm` varchar(20) NOT NULL default "",
  `pass_adm` varchar(32) NOT NULL default "",
  PRIMARY KEY (`usuario_adm`)
)
# Volcar la base de datos para la tabla `useradmin`
INSERT INTO `useradmin` VALUES ('alucia', '200a4fa5c42fae86ecbd47b366292a8b');
INSERT INTO `useradmin` VALUES ('ccecilia', 'aabc6b0498c7c8ef4636f36c4972bc66');
INSERT INTO `useradmin` VALUES ('mines', '1b62fd62b993e4b2bc19564796495bdc');
# -----
# Estructura de tabla para la tabla `validadoc`
#
# Creación: 09-10-2004 a las 11:07:10
# Última actualización: 09-10-2004 a las 11:07:13
CREATE TABLE `validadoc` (
  `USUARIO` varchar(20) NOT NULL default "",
  `PASSWD` varchar(32) NOT NULL default "",
  PRIMARY KEY (`USUARIO`)
) TYPE=MyISAM;
# Volcar la base de datos para la tabla `validadoc`
INSERT INTO `validadoc` VALUES ('charly', 'a6d4ef4dd38b1bb016d250c16a680470');
INSERT INTO `validadoc` VALUES ('linda', 'eaf450085c15c3b880c66d0b78f2c041');
INSERT INTO `validadoc` VALUES ('carmin', '47bea507e2df5ce7be5a41c3d57f7da7');
INSERT INTO `validadoc` VALUES ('pato', '05167f1ebabab1cc5a3cce7080c1b8b3');
```

C. MANUAL DE INSTALACIÓN

Los servicios mínimos a instalar son:

- Servidor de páginas Web Apache.
- Módulo PHP para Apache.

El servidor Web Apache viene instalado por defecto en cualquier distribución de Linux por medio del daemon httpd, comprobamos con el comando siguiente (como usuario root):

```
# service httpd status
```

la respuesta a este comando es el estado del servidor, si en caso esta detenido lo activamos con la orden:

```
# service httpd start
```

Para comprobar que el módulo PHP este instalado creamos una página Web con el nombre “prueba.php” con el siguiente contenido:

```
<?
echo phpinfo();
?>
```

y la copiamos en el directorio de publicación html:

```
/var/www/html
```

al ejecutarla desde el navegador Web debe presentar la información de PHP y sus módulos.

Para probar e instalar la base de datos “monitoreo” y “doctores” la instalación de MySql ejecutamos los siguientes comandos:

```
# mysql
```

obtenemos:

```
mysql >
```

ejecutamos:

```
mysql > create table monitoreo;
```

```
mysql > create table doctores;
```

```
mysql > quit
```

Luego creamos 2 archivos con las ordenes de creación de sus respectivas bases de datos:

- Doctores.sql
- Monitoreo.sql

Y ejecutamos:

```
# mysql doctores < doctores.sql
```

```
# mysql monitoreo < monitoreo.sql
```

En este momento ya están creadas las bases de datos.

Cuando todos estos servicios estén activados y en funcionamiento creamos un usuario normal:

```
# createuser monitoreo
```

```
# passwd monitoreo
```

New password:

Retype password:

Con las anteriores órdenes ya se tiene creado un usuario llamado “monitoreo”.

En el directorio /home/monitoreo copiamos los Archivos:

- lserial.c
- gbase.c
- make
- instalar

y ejecutamos las ordenes:

```
$ ./instalar
```

al ejecutar este comando se crean los archivos lserial y gbase con los respectivos permisos de ejecución.

El siguiente paso es activar el programa cada vez que se inicia Linux, esto lo conseguimos activando mediante alguno de los 3 métodos:

- el comando crontab.
- archivo inittab
- archivo batch_profile

Dependiendo de la distribución de Linux en la que se este ejecutando y el metodo mas óptimo dependiendo de la aplicación.

Paso siguiente damos los permisos necesarios al puerto serial:

```
# chmod 666 /dev/ttyS0
```

Por ultimo reiniciamos el sistema operativo o ejecutamos el programa “lserial” que se ha creado anteriormente.

En este momento el sistema ya comienza a almacenar datos.

D. COOKIES

Debido a que el protocolo http es un protocolo sin estado y no se puede hacer un seguimiento de una sesión, utilizamos herramientas como son las cookies.

Página Web de un ejemplo para la creación de una cookie:

```
<?php
    setcookie("ejemplo", $nombre, time()+3600, "/", "");
?>
<html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de uso de cookie</H1>
Se ha establecido una cookie de nombre <b>ejemplo</b> con el valor: <b>
<?php
    echo $nombre;
?>
</b> que será válida durante 1 hora.
</body>
</html>
```

Página Web de un ejemplo para la obtención del valor de una cookie:

```
<html>
<head>
    <title>Ejemplo de PHP</title>
</head>
<body>
<H1>Ejemplo de uso de cookie</H1>
Se ha creado la cookie de nombre <b>ejemplo</b> vale: <b>
<?php
    echo $ejemplo; ?></b>
</body>
</html>
```

Al crearse una cookie se genera el archivo nombre_de_la_cookie.txt el cual contiene lo siguiente:

```
ejemplo
pruebas
localhost/
0
835269632
29667633
243699632
29667633
*
```

E. LISTA DE TABLAS Y FIGURAS

CAPITULO 1

FIGURA 1.1 DIAGRAMA A BLOQUES DEL MONITOREO DE RITMO CARDIACO VÍA INTERNET. (FUENTE DE AUTORES)	14
---	----

CAPITULO 2

FIGURA 2.1 EJEMPLO DE COMUNICACIÓN REMOTA VÍA ONDAS ELECTROMAGNÉTICAS. (FUENTE DE AUTORES).....	17
---	----

FIGURA 2.2 EJEMPLO DE SISTEMA DE TELEMETRÍA POR MEDIO DE INTERNET. (FUENTE DE AUTORES)	18
--	----

FIGURA 2.3 DISPOSITIVO PORTÁTIL DE MONITOREO DE PACIENTES.	19
---	----

FIGURA 2.4 MÓDULO PARA TELEMETRÍA VÍA INTERNET.....	20
---	----

CAPITULO 3

TABLA 3.1 DESCRIPCIÓN DEL PUERTO SERIE.	29
--	----

FIGURA 3.1 ACCESO AL DISPOSITIVO DE MONITOREO POR MEDIO DE INTERNET. (FUENTE DE AUTORES).....	32
---	----

FIGURA 3.2 DIAGRAMA A BLOQUES DE CONEXIÓN AL COMPUTADOR DEL DISPOSITIVO DE MONITOREO. (FUENTE DE AUTORES).	33
---	----

FIGURA 3.3 TRAMA DE DATOS ENTREGADA POR EL DISPOSITIVO DE MEDICIÓN. (FUENTE DE AUTORES).....	33
--	----

CAPITULO 4

FIGURA 4.1 DIAGRAMA DE BLOQUES DEL SISTEMA. (FUENTE DE AUTORES).....	38
--	----

FIGURA 4.2 TRAMA DE DATOS DE 12 CARACTERES ENTREGADA POR EL DISPOSITIVO DE MEDICIÓN. (FUENTE DE AUTORES).	39
--	----

FIGURA 4.3 MEDIDA CON VALORES REALES. (FUENTE DE AUTORES)	39
---	----

FIGURA 4.4 PANTALLA INDEX.HTML. (FUENTE DE AUTORES).	44
---	----

FIGURA 4.5 PANTALLA DE LOGIN DE MONITOREO HISTÓRICO. (FUENTE DE AUTORES).	45
--	----

FIGURA 4.6 PANTALLA DE MONITOREO DEL RITMO CARDIACO HISTÓRICO. (FUENTE DE AUTORES)..	48
--	----

FIGURA 4.7 PANTALLA DE MONITOREO DEL RITMO CARDIACO EN TIEMPO REAL. (FUENTE DE AUTORES).....	49
--	----

FIGURA 4.8 PANTALLA DE LOGIN PARA ACTUALIZACIÓN DE DATOS. (FUENTE DE AUTORES).	50
---	----

FIGURA 4.9 PANTALLA DE ACTUALIZACIÓN DE DATOS DE PACIENTES. (FUENTE DE AUTORES).	52
---	----

FIGURA 4.10. PROTOTIPO DEL SISTEMA DE MONITOREO DEL RITMO CARDIACO. (FUENTE DE AUTORES).....	80
--	----

CAPITULO 5

FIGURA 5.1 PANTALLA DE BIENVENIDA. (FUENTE DE AUTORES)	82
--	----

FIGURA 5.2 PANTALLA DE LOGIN PARA INGRESO A MONITOREO HISTÓRICO. (FUENTE DE AUTORES).83	
---	--

FIGURA 5.3 PANTALLA DE MONITOREO DEL RITMO CARDIACO HISTÓRICO. (FUENTE DE AUTORES)..	83
--	----

FIGURA 5.4 PANTALLA DE RESULTADO DE MONITOREO DEL RITMO CARDIACO HISTÓRICO DEL PACIENTE 1. (FUENTE DE AUTORES).....	84
---	----

FIGURA 5.5 PANTALLA DE MONITOREO DEL RITMO CARDIACO EN TIEMPO REAL. (FUENTE DE AUTORES).....	85
--	----

FIGURA 5.6 PANTALLA DE RESULTADO DE MONITOREO DE RITMO CARDIACO EN TIEMPO REAL DEL PACIENTE 1. (FUENTE DE AUTORES).	85
--	----

FIGURA 5.7 PANTALLA DE EJECUCIÓN DEL PROGRAMA “L SERIAL” CON ERROR. (FUENTE DE AUTORES).....	86
--	----

FIGURA 5.8 PANTALLA DE EJECUCIÓN DEL PROGRAMA “L SERIAL” CON ERROR DE LECTURA. (FUENTE DE AUTORES).	87
FIGURA 5.9 PANTALLA DE EJECUCIÓN DEL PROGRAMA “L SERIAL” SIN ERRORES. (FUENTE DE AUTORES).....	88
CAPITULO 6	
FIGURA A.1 PANTALLA DE INICIO DE SESIÓN QUE PRESENTA EL MENÚ PRINCIPAL. (FUENTE DE AUTORES).....	94
FIGURA A.2 PANTALLA PARA VALIDAR USUARIO Y PASSWORD DE MONITOREO HISTÓRICO. (FUENTE DE AUTORES).	95
FIGURA A.3 PANTALLA MENSAJE DE USUARIO NO AUTORIZADO AL MONITOREO. (FUENTE DE AUTORES).....	96
FIGURA A.4 PANTALLA DE MONITOREO DEL RITMO CARDIACO HISTÓRICO. (FUENTE DE AUTORES)..	97
FIGURA A.5 PANTALLA DE MONITOREO DEL RITMO CARDIACO EN TIEMPO REAL. (FUENTE DE AUTORES).....	98
FIGURA A.6 PANTALLA PARA VALIDAR USUARIO Y PASSWORD DE ACTUALIZACIÓN DE DATOS DE PACIENTES. (FUENTE DE AUTORES).....	99
FIGURA A.7 PANTALLA DE MENSAJE USUARIO NO AUTORIZADO A LA ACTUALIZACIÓN. (FUENTE DE AUTORES).....	100
FIGURA A.8 PANTALLA DE ACTUALIZACIÓN DE DATOS DE PACIENTES. (FUENTE DE AUTORES).	101