



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“RESERVACIÓN DE CONSULTAS DE UNA
CLÍNICA DE ESPECIALIDADES MÉDICAS”**

**TRABAJO DE TITULACIÓN
EN LA MODALIDAD DE SEMINARIOS Y
CURSOS DE ACTUALIZACIÓN Y
CAPACITACIÓN PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
ROBERTO OLIVARES GALVÁN**

ASESOR: M. EN C. MARCELO PEREZ MEDEL

MÉXICO

2005.

0352534



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A mis padres:

Por el apoyo incondicional que he recibido durante todo este tiempo y les agradezco infinitamente ese camino del éxito. Y ese camino que fue señalado por ustedes.

A mis hermanas:

Por apoyarme en todo momento, y sacar adelante mis proyectos.

A mi esposa:

Por ayudarme a desarrollar la más profunda superación, por enseñarme los valores del amor a la verdad y la bondad.

A mi hijo:

Por darme valor y fuerzas para seguir adelante, por transformar mi vida de disciplina de confianza y fe..... Te amo.

A mis profesores:

Por darme el apoyo y el tiempo necesario para sacar adelante mi proyecto, en especial a mi asesor.

A mis amigos:

Por el apoyo brindado dentro y fuera de la universidad.

A Dios:

Por permitirme llegar hasta donde estoy.



Modulo 1

Página

INTRODUCCIÓN.....1

- ❖ Introducción a servicios de Internet
- ❖ Instalación y configuración de servicios para WWW
 - Web Server
 - DBMS
 - JVM integrada al servidor

Modulo 2

ANÁLISIS Y DISEÑO DEL SISTEMA DE CONSULTAS.....31

- ❖ Ingeniería de requerimientos
- ❖ Herramientas de A y D con UML
 - Casos de uso: documental
 - Diagrama de casos de uso
 - Diagrama de secuencia
 - Diagrama de interacción
 - Diagrama de componentes
- ❖ Diseño de la base de datos:
 - Conceptos básicos
 - Diagrama de clases
- ❖ Modelo Vista Controlador, MVC.

Modulo 3

PROGRAMACIÓN ORIENTADA A OBJETOS.....52

- ❖ Introducción a OO
- ❖ Introducción a Java
- ❖ JDK y Eclipse



Modulo 4

Página

IMPLEMENTACION DE LA CAPA DE USUARIO DEL SISTEMA..... 72

- ❖ Introducción a HTML
- ❖ Diseño de la aplicación con respecto a perfiles de usuario
- ❖ Diseño de interfaz
 - Usabilidad
 - Navegación

Modulo 5

CAPA DE DATOS.....86

- ❖ Introducción a SQL
- ❖ SQL con DB2 UDB versión 8.1.
- ❖ Tareas comunes de administración de la base de datos

CONCLUSIONES.....100

GLOSARIO.....102

BIBLIOGRAFIA.....105



INDICE **MODULO 1** **Página**

Introducción a servicios de Internet..... 4

1. ¿Qué es Internet?4

 1.1 Direccionamiento IP y servicios DNS.....5

 1.2 Clases de Direcciones IP.....5

2. Servicios de Internet.....5

 2.1 El Correo Electrónico.....6

 2.2 World Wide Web.....7

 2.3 Características de la Web.....7

 2.4 Qué tipo de información maneja la web.....8

 2.5 Servicio de Transferencia de archivos: FTP.....9

 2.6 Servicios de FTP.....9

 2.7 Comandos de FTP.....9

 2.8 Sesiones Remotas de Trabajo: SSH.....10

Introducción al desarrollo en 3 Capas.....10

3 Capa de Presentación.....10

4 Capa de Negocio.....10

5 Capa de Datos.....11

 5.1 Estrategia UDA (Universal Data Access).....11

 5.2 Servidores de datos actuales.....12

Introducción a UNIX.....13

6 ¿Qué es UNIX?13

 6.1 Arquitectura.....13

 6.2 El Kernel.....14



6.3 El Shell..... 14

6.4 Intérpretes de Comandos..... 14

7 Comandos del UNIX..... 15

7.1 Tipos de Ficheros..... 15

7.2 Permisos de acceso..... 16

7.3 Cambiar los permisos..... 17

7.4 Procesos en UNIX..... 18

7.5 Terminar un proceso..... 18

8 Introducción a JDK..... 19

8.1 Directorio personal..... 20

8.2 Papelera..... 20

8.3 Iconos dinámicos para los soportes removibles..... 20

8.4 Panel..... 20

8.5 El Menú Principal..... 21

8.6 Mostrar Escritorio..... 21

8.7 Applet para Cambio de Escritorio..... 21

8.8 Applet de cambio de resolución de pantalla..... 21

8.9 Cambiar la apariencia de su escritorio..... 22

8.10 Cambiando la resolución de la pantalla..... 23

8.11 Administrando archivos..... 24

8.12 Ventana principal..... 24

9 Instalar JDK..... 25

10 Instalar Apache..... 27

Elementos de software para un proyecto de desarrollo en WWW..... 27

11 Manejador de Base de Datos: MYSQL..... 27



Java Virtual Machine Integral al servidor.....	28
12 Jakarta Tomcat.....	28
13 Instalar Tomcat.....	28
13.1 Configuración.....	29
13.2 Inicializar el servidor de Yakarta.....	29
13.3 Verificar la instalación.....	29
13.4 Configuraciones Adicionales.....	29

1. ¿Qué es Internet?

Es una red Mundial de equipos de cómputo que se comunican mediante un lenguaje común conocido como protocolo de red.

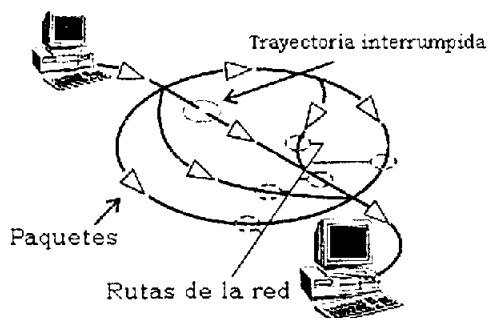


FIGURA 1.1 Red Mundial

Es similar al sistema internacional de teléfonos: nadie posee ni controla todo el conjunto, pero esta conectado de forma que funcione como una gran red.

Dentro de Internet hay presencia de organizaciones de tipo Comercial (.com), Gubernamentales (.gob) y hasta clubes.

1.1 Direccionamiento IP y servicios DNS

Direccionamiento IP

El direccionamiento IP es un protocolo de la capa de red encargado del proceso de ruteo de los paquetes a través de la red.

Las direcciones del nivel de red en Internet pueden representarse de manera simbólica o numérica. Una dirección simbólica es por ejemplo www.pntic.mec.es Una dirección numérica se representa por cuatro campos separados por puntos, como 193.144.238.1, los cuales no pueden superar el valor 255 (11111111 en binario). La correspondencia entre direcciones simbólicas y numéricas las realiza el DNS (Domain Name System).



1.2 Clases de Direcciones IP

Existen cinco clases de direcciones IP:

- Clase A
- Clase B
- Clase C
- Clase D
- Clase E

Las Clases A, B, y C son comerciales, pero las Clases D y E son reservadas para grupos de multidifusión o uso experimental.

2 Servicios de Internet

Internet ofrece una serie de servicios creados para darle sentido a esa gran conexión mundial y que, a lo largo de los años, ha crecido y se ha fortalecido para bien de los usuarios.

Estos servicios son:

- World Wide Web: comúnmente llamada “web”, se trata del servicio de más impacto y uso en internet, algunas personas dicen que internet es la web, pero sólo es parte de ella.
- Correo Electrónico: “el e-mail” es el servicio de envío y recepción de mensajes de texto, similares al correo convencional.
- Foros de discusión en tiempo real y no presenciales
 - Foros
 - Listas de discusión de correos
 - Chat (MSN y Yahoo messenger, entre otros)
- Servicio de transferencia de archivos: FTP
- Servicio de trabajo compartido en servidores
- Servicios de noticias y algunos más.....



2.1 El Correo Electrónico

El correo electrónico ("E-Mail" ó "Electronic Mail" en Inglés) es el segundo servicio más usado de la Red Internet (El primero es la navegación por la World Wide Web). Dos personas que tengan acceso a una cuenta de correo en Internet pueden enviarse mensajes escritos desde cualquier parte del mundo a una gran velocidad. Lo normal es que un mensaje tarde entre unos pocos segundos y unos pocos minutos, dependiendo de la cantidad de texto que se envíe.

Los Proveedores de Internet dan una o dos cuentas de correo a sus clientes cuando solicitan una conexión a la Red. De este modo, es casi seguro que si podemos conectarnos a Internet, podamos enviar y recibir correo electrónico.

Para manejar el correo electrónico, existen programas como el Eudora o el Pegasus, que están especializados en esta tarea. Sin embargo, es más cómodo usar un navegador como Netscape o Internet Explorer, puesto que incorporan la posibilidad de gestionar el correo en el propio navegador. Esto ofrece la ventaja de que con un sólo programa se tiene acceso a los dos servicios más usados de la Red.



2.2 World Wide Web

Se denomina *World Wide Web*, la 'telaraña que cubre el mundo', al colectivo formado por toda la información disponible en Internet que puede ser accedida o vista a través de un cliente Web. Por lo regular se utiliza indistintamente los términos Web, WWW o World Wide Web.

Desde un cliente Web, toda la información disponible se ve como un 'universo plano', en el cual la mayor parte de ella está accesible tras una o varias pulsaciones del ratón, ocultando todos los detalles necesarios para acceder a los datos.

Sin embargo, es importante recordar que por debajo existe un entramado de ordenadores, en los que se ejecutan aplicaciones clientes y servidores, que son los encargados de almacenar e intercambiar la información.

En este punto, cabe hacer una distinción clara entre Internet y el WWW. Este último es uno más de los servicios de información disponibles en Internet, una red de ordenadores que se comunican a través del protocolo TCP/IP. La evolución de los clientes Web como interfaz *casí* única de acceso a numerosos servicios de Internet contribuye al equívoco que supone utilizar indistintamente ambos términos.

2.3 Características de la Web

- Es muy fácil de utilizar, las interfaces de los clientes Web son muy intuitivas, y cualquier persona se hace con su funcionamiento rápidamente.
- "Todo lo que se puede saber sobre el Web está en el propio Web". A través de un cliente Web se puede acceder a ayudas sobre su manejo, aprender más sobre el Web, conocer cómo desarrollar nuevas páginas.
- Por su flexibilidad, puede dar soporte a multitud de servicios diferentes, por lo que resulta interesante para empresas, centros de enseñanza, etc. A través del Web se puede acceder a servicios comerciales y publicitarios, cursos y guías, bases de datos,...
- Es muy fácil publicar nueva información, así como incorporar información en formato electrónico de la que se disponía previamente, para hacerla accesible a todo el mundo.
- Está en continua evolución, y cada día sus capacidades de acceso y representación de información se vuelven más sofisticadas.
- Ha sido la principal causa del espectacular crecimiento que Internet ha tenido en los últimos cuatro años, tanto en el número de usuarios como en el volumen de información disponible.
- También sirve como soporte para las denominadas "Intranets", que son versiones reducidas de Internet para empresas, grupos de trabajo, etc., que utilizan la tecnología de publicación del Web, para intercambiar información entre grupos de trabajo.



2.4 Qué tipo de información maneja la web

Las páginas Web están diseñadas para contener tipos muy variados de información, con lo que se pueden construir sofisticadas presentaciones multimedia. De todos los posibles contenidos, cada cliente Web sabe representar determinados tipos de datos (ficheros) directamente; en caso de que no sepa qué hacer con el contenido de un fichero (imágenes TIFF o presentaciones ASAP, por ejemplo), puede buscar una aplicación adecuada para manejarlo. Los tipos de información más comunes son:

- Texto, con diferentes tipos de letra, formato, alineación, etc. Prácticamente lo mismo que se puede conseguir con un buen procesador de textos.
- Imágenes, insertadas junto al texto. Se tiene control sobre la posición, tamaño y ajuste de texto alrededor de la imagen.
- Elementos activos, que pueden ser texto o imágenes. El texto activo aparece subrayado y en distinto color. Las imágenes se rodean de un borde fino. En cualquier caso, en un entorno gráfico, el cursor cambia de forma cuando pasa sobre una imagen activa.
- *Imagemaps*, imágenes activas en las que se ha definido una relación entre zonas de la imagen y URLs. De esta forma, la información que el usuario recibe depende del punto en el que realiza la pulsación con el ratón. El ejemplo más claro es un mapa de España, en el que se proporciona información sobre una región pulsando sobre su lugar en el mapa.
- Tablas, que permiten representar gran variedad de formatos, contener imágenes u otras tablas, y alterar su presentación (color, bordes, alineación, etc.).
- Formularios, para recoger información proporcionada por un usuario. Inserción de texto, botones, diferentes tipos de selectores de opciones, etc.



2.5 Servicio de Transferencia de archivos: FTP

El Protocolo de Transferencia de Archivos (FTP) es una de las utilidades de la suite de Protocolos de TCP/IP. FTP permite a los usuarios copiar archivos entre sistemas remotos en una red IP. FTP se ha diseñado para su uso de forma interactiva por usuarios finales o por programas de aplicación. Las funciones esenciales de Transferencia de Archivos permiten a los usuarios realizar tareas básicas como copiar, mover, renombrar y trabajar con directorios de forma remota.

2.6 Tipos de servicios FTP

Normalmente los sistemas se configuran para autenticar los inicios de sesión, esto es pues, colocando un password y una contraseña, de esta manera se puede acceder al sistema. FTP tiene dos modalidades de uso:

FTP Anonymous: Esto supone un servidor FTP configurado para permitir el acceso público, es decir, el sistema se ajusta a una clave de acceso público para permitir el acceso anónimo a todos los archivos que se han compartido.

FTP Privado: En este caso el servidor se basa en autenticación a partir de la base de datos de usuarios locales, por lo tanto, sólo pueden iniciar sesión los usuarios que hayan sido dados de alta en dicho sistema, normalmente este tipo de FTP es utilizado por compañías que requieren de acceso remoto o en entornos donde la información es confidencial.

2.7 Comandos de FTP:

¿Qué tipos de comandos se pueden enviar a través de una conexión de control FTP?

- **Comando de Transferencia de Archivos:** Los comandos de Transferencia de Archivos permiten al usuario copiar un único archivo de un host a otro, copiar varios archivos de un host a otro, añadir un archivo local a un archivo remoto, copiar un archivo y añadir un número al nombre para que sea único
- **Comandos de Gestión de Archivos:** Los Comandos de Gestión permiten a los usuarios listar los archivos de un directorio, identificar el directorio en uso y cambiar de directorio, crear y eliminar directorios, cambiar el nombre a un archivo o borrarlo.
- **Comandos de Control:** Los Comandos de Control permiten a los usuarios identificar si se va a transferir texto ASCII o Binario, establecer si el archivo está estructurado como una serie de bytes o como una secuencia de registro, describir cómo se va a transferir el archivo, por ejemplo, como un flujo de efectos.



2.8 Sesiones remotas de trabajo: SSH

Actualmente las Redes de Computadoras son medios digitales más usados en todos los ámbitos de la sociedad para la transferencia de información.

Secure Shell (SSH) es un programa que permite realizar conexiones entre máquinas a través de una red abierta de forma segura, así como ejecutar programas en una máquina remota y copiar archivos de una máquina a otra.

Provee fuerte autenticación y comunicación segura sobre un canal inseguro y nace como un remplazo a los comandos telnet, ftp, rlogin, rsh, y rcp, los cuales proporcionan gran flexibilidad en la administración de una red, pero sin embargo, presenta grandes riesgos en la seguridad de un sistema.

Introducción al desarrollo de 3 capas

3 Capas de Presentación

Los servicios de presentación generalmente son identificados con la interfaz de usuario, y normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final. Aún así, existen oportunidades para identificar servicios que residen en componentes separados.

El cliente proporciona el contexto de presentación, generalmente un browser como Microsoft Internet Explorer o Netscape Navigator, que permite ver los datos remotos a través de una capa de presentación HTML.

La capa de servicios de presentación es responsable de:

- Obtener información del usuario.
- Enviar la información del usuario a los servicios de negocios para su procesamiento.
- Recibir los resultados del procesamiento de los servicios de negocios.
- Presentar estos resultados al usuario.

4 Capa de Negocio

Los servicios de negocios son el "puente" entre un usuario y los servicios de datos. Responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes.

Una tarea de negocios es una operación definida por los requerimientos de la aplicación, como introducir una orden de compra o imprimir una lista de clientes. Las reglas de negocio (business rules) son políticas que controlan el flujo de las tareas.



El nivel de servicios de negocios es responsable de

- Recibir la entrada del nivel de presentación.
- Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar (por ejemplo, la preparación de impuestos por ingresos, el procesamiento de ordenes y así sucesivamente).
- Enviar el resultado procesado al nivel de presentación.

Algunos de los servicios DNA para la capa de Negocios son los siguientes:

- Servicios Web a través de Microsoft Internet Information Server (IIS)
- Transacciones y Servicios de Componentes, Microsoft Transaction Server (MTS)
- Servicios Asíncronos, Microsoft Message Queue Server (MSMQ).
- Server-side Scripting, vía Active Server Pages (ASP).

5 Capa de Datos

- El nivel de servicios de datos es responsable de:
- Recuperar los datos.
- Almacenar los datos.
- Mantener los datos.
- La integridad de los datos. Los servicios de datos tienen una variedad de formas y tamaños, incluyendo los sistemas de administración de bases de datos relacionales (SABDs) como Microsoft SQL Server, servidores de correo electrónico como Microsoft Exchange Server y sistemas de archivos tales como el Sistema de Archivos NTFS.

5.1 La estrategia UDA (Universal Data Access)

Según la propuesta Microsoft, aceptada por muchos fabricantes, es la estrategia UDA (**Universal Data Access**), que se basa en una extensión de OLE, denominada **OLE-DB**. Mediante OLE-DB, se puede acceder a datos de formatos heterogéneos sin que eso suponga un problema de modificación de las formas de acceso. Se puede abrir un correo electrónico, o un documento de Word, una hoja de cálculo Excel o incluso crear una tabla con el contenido de los ficheros existentes en un directorio. Lógicamente, habrá un *traductor*, igual que sucedía con ODBC. Ese traductor que permite el acceso es el *Proveedor OLE-DB*, de los que existen ya un buen número, entre los que se cuenta un Proveedor OLE-DB para ODBC, en el caso de que el servidor de datos al que pretendemos acceder no tenga todavía disponible el proveedor OLE-DB nativo.



5.2 Servidores de datos

Un servidor de datos es llamado también DBMS (DataBase Management System) Sistema Manejador de Base de Datos, el cual es un software usado para gestionar datos de una o más aplicaciones.

Los DBMS más populares son:

- Oracle Es el mejor servidor de base de datos Objeto-Relacional.
- SQL-Server Es un servidor de base de datos fácil manejo de Microsoft
- MySQL Es el servidor de base de datos relacionales más popular y fácil de manejar
- PostgreSQL Es un servidor de base de datos de gran potencia

También hay otros como:

- DB2
- Informix
- Sybase
- Adabas

Introducción a UNIX

6 ¿ Qué es UNIX?

El UNIX es un sistema operativo de propósito general, interactivo, multiusuario, multitarea, portable y flexible¹.

Cada una de estas características hace que el UNIX sea un sistema que se adapta fácilmente a todo tipo de aplicaciones, pueda ejecutarse en diferentes entornos (desde una PC a un mainframe) y sea fácilmente integrable con otro tipo de plataformas.

6.1 Arquitectura

La arquitectura del UNIX consiste en tres diferentes capas:

El *kernel* es el corazón del sistema operativo. Es el responsable de controlar el hardware y de asignar los diferentes recursos a los procesos.

El *shell* es el intérprete de comandos y es generalmente con quién interactúan los usuarios.



FIGURA 1.2 Diferentes capas de Unix



6.2 El Kernel

El kernel es quién controla el ordenador y hace que los recursos del mismo estén disponibles a los procesos de los usuarios.

Entre las diferentes tareas del Kernel están: controlar el hardware, manejar los dispositivos de Entrada/Salida, soportar comunicaciones, asignar recursos del sistema, asignar mecanismos de protección, mantener la estructura de ficheros, etc.

6.3 El Shell

El shell o intérprete de comandos es el encargado de interactuar con el usuario y ejecutar los distintos comandos solicitados por éste.

Sus características principales son:

- Entorno de ejecución configurable, que permite crear y modificar comandos, prompts y otras utilidades
- Flujo flexible de datos, que permite usar la salida de un comando como entrada de otro
- Lenguaje de comandos de alto nivel, que incluye un conjunto completo de estructuras de control
- Permite la programación de la ejecución de los comandos; redirección de entrada/salida
- Permite guardar la salida de un comando a un fichero o utilizar un fichero como entrada de, un comando.

6.4 Intérpretes de Comandos

Existen diferentes intérpretes de comandos o shells dentro del UNIX, cada uno orientado a diferente tipo de utilización o usuario:

Bourne Shell (sh): Es el estándar y tiene gran facilidad de uso sobre todo para programación.

C-Shell (csh): Es similar al Bourne, pero tiene mayor flexibilidad para su uso interactivo. Provee mecanismos adicionales para guardar historia de comandos, reejecución y sustitución de comandos.

Korn Shell (ksh): Nuevo intérprete de comandos, se está convirtiendo en el nuevo estándar. Es totalmente compatible con el Bourne Shell, pero incorpora, entre otras, facilidades del C-Shell y control de trabajos en background.

Restricted Shell (rsh): version restringida del Bourne Shell. Sólo permite la ejecución de un limitado número de operaciones en el sistema y está diseñado para manejar más eficientemente la seguridad.

Restricted C-Shell (rcsh): version restringida del C-Shell.

Restricted Korn Shell (rksh): version restringida del Korn Shell.

7 Comandos del UNIX

El sistema operativo UNIX tiene más de 200 comandos. El administrador y cada uno de los usuarios pueden crear sus propios comandos escribiéndolos en algún lenguaje de programación (C, Cobol, etc.) o combinando los ya existentes dentro del UNIX con programas de shell. Normalmente después del prompt se introduce un comando. Cada comando realiza una tarea específica, como ejecutar una aplicación, imprimir un fichero, etc.

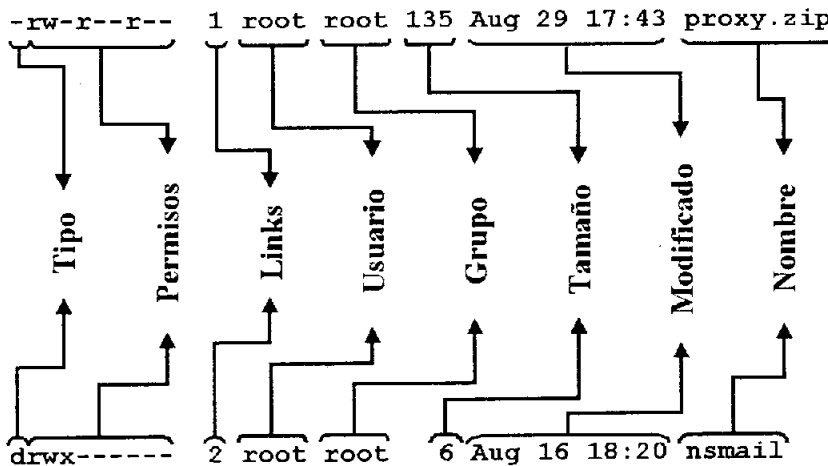


FIGURA 1.3 Ejemplo de algunos comandos de Unix

7.1 Tipos de Ficheros

UNIX tiene tres tipos diferentes de ficheros (más algunos otros que no son de tan clara identificación). Cada uno de ellos son nombrados de acuerdo a la misma convención, pero son manejados internamente de muy diferente manera.

Los distintos tipos son:

Ficheros regulares, normales o comunes: son aquellos donde se guarda la información.

Directorios: los directorios son ficheros que contienen la tabla de contenido o lista de los ficheros de un Filesystem de UNIX.



7.2 Permisos de acceso

Los ficheros manejados por los usuarios son normalmente los dos primeros, los otros tres son de uso especial y generalmente los usuarios no los manejan en forma directa.

El siguiente grupo de 9 caracteres indica los permisos de acceso que tiene el fichero. Los primeros tres son los permisos para el dueño del fichero, los tres siguientes los permisos para el grupo dueño del fichero y los últimos tres los permisos para el resto de los usuarios².

Los símbolos que aparecen en cada grupo tienen diferente significado dependiendo si se trata de un fichero ordinario o de un directorio.

Usuario	Grupo	Otros
r w x	r w x	r w x

FIGURA 1.4 Tipos de ficheros y directorios

Para los ficheros ordinarios el significado es:

- r lectura
- w escritura
- x ejecución
- cuando aparece en lugar de r, w o x el permiso está denegado

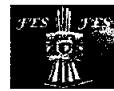
Para los directorios el significado es:

- r listar el contenido
- w agregar o borrar entradas al directorio
- x búsqueda
- cuando aparece en lugar de r, w o x el permiso está denegado

Cuando se crea un fichero o un directorio se establecen sus permisos:

para ficheros: `rw-r--r--`

para directorios: `rwxr-xr-x`



7.3 Cambiar Permisos

Los permisos de acceso a los ficheros y directorios se pueden cambiar con el comando `chmod`. Sólo el dueño del directorio o fichero puede modificar sus permisos³.

La sintaxis de este comando es:

```
chmod permisos fichero
```

Los permisos se dividen en tres partes:

1. A quien afectan: u para el dueño, g para el grupo y o para otros
2. La acción que se va a realizar: + para agregar el permiso y - para eliminarlo
3. El acceso que se modifica: r de lectura, w de escritura y x de ejecución

Los ficheros son aquellos a los cuales se les van a modificar los permisos.

La forma de especificar los permisos es la siguiente: se pone a quién o quienes va a afectar el cambio, cuál es la acción que se va a hacer y qué permisos serán afectados.

Todos los caracteres van uno a continuación del otro sin espacios en blanco.

Ejemplos:

```
chmod o-r file*
```

Se retira el permiso de lectura para otros usuarios (que no sean el dueño o el grupo dueño) de todos los ficheros llamados `file*`.

```
chmod go-rwx fichero
```

Se retiran todos los permisos al grupo dueño y resto de usuarios para acceder a `fichero`. En este caso el único que tiene acceso a `fichero` es el dueño del mismo.

```
chmod go+r *
```

Se agrega permiso de lectura para todos los usuarios a todos los ficheros del directorio de trabajo.

El comando `chmod` tiene una sintaxis alternativa. En ésta a cada simbolo de acceso le es asignado un número: r=4, w=2 y x=1. Para cada categoría (dueño, grupo y otros) los números de cada permiso se suman, de tal forma que se obtendrá un número de 3 dígitos que debe ser usado en el lugar de permisos.



Por ejemplo:

```
  r  w  -  r  -  -  -  -  -  
  4  2  1  4  2  1  4  2  1  
  
  6      4      0
```

Esto quiere decir que si se quiere modificar los permisos de acceso a un fichero para que el dueño pueda leer y escribir, el grupo dueño pueda leer y el resto de los usuarios no tengan acceso a él se deberá usar el número 640 con el comando `chmod`:

```
chmod 640 file
```

Se puede usar cualquier versión de `chmod` indistintamente de acuerdo a las necesidades o comodidad del usuario.

7.4 Procesos en UNIX

Todo programa que se está ejecutando es un proceso dentro de UNIX. El UNIX mantiene una lista de todos los procesos que se están corriendo en un momento dado en el sistema.

Los procesos compiten por el tiempo de la CPU hasta que su ejecución finaliza. El Kernel va asignando tiempos de CPU a los procesos en base a diferentes parámetros.

El login shell es un proceso, el primero que se ejecuta para cada usuario cada vez que hace login al sistema, El shell crea un nuevo proceso por cada comando que se ejecuta.

7.5 Terminar un proceso

Los procesos interactivos (no aquellos que se corren en background) se pueden terminar presionando la tecla `` (en algunos sistemas se la conoce como `<Supr>`, en otros se usa la tecla `<Ctrl>+<C>`). Al presionar esta tecla el programa recibe una señal que lo hace cancelar y el shell muestra el prompt para continuar con la ejecución de otros comandos.

Los procesos que se corren en background no son afectados por ``. También es posible que algunos comandos interactivos no sean afectados por ninguna de las teclas. Eso se debe a que el programador "inhibió" la posibilidad de cancelarlos.

8 Introducción a JDK

En Linux existen diferentes tipos de entorno gráfico. Todos ellos son muy parecidos entre sí y se hacen muy fácil de utilizar por el parecido que tiene respecto a Windows. Los más comunes o utilizados son el KDE y el Gnome. Todos ellos son muy parecidos entre sí. Durante la instalación de Linux se podrán elegir los entornos gráficos a instalar. Para acceder al entorno gráfico de Linux, se podrá teclear desde la línea de comandos el comando `startx` y una vez ejecutado dicho comando se entrara en el entorno gráfico que este instalado por omisión para su entrada, en el caso de que se quiera entrar en otro entorno diferente al que se ejecute por omisión se podrá teclear por ejemplo:

`startx KDE`

`startx Gnome`

Entre otros entornos gráficos, según el entorno en el que se quiera entrar, siempre anteponiendo el comando `startx` al nombre del entorno gráfico que se quiera utilizar. Si durante la instalación de Linux se ha elegido la opción de arrancar en modo gráfico, saldrá el prompt en modo gráfico,



FIGURA 1.5 Configuración predeterminada del escritorio de KDE

8.1 Directorio personal

Da acceso a todos sus archivos personales. Bajo los sistemas operativos tipo UNIX (Mandrake Linux es uno de ellos), cada usuario tiene un directorio personal que por lo general se denomina `/home/nombre_de_usuario`, donde `nombre_de_usuario` es el nombre de conexión del usuario.



FIGURA 1.6 Directorio Personal de KDE

8.2 Papelera

Da acceso a todos los archivos borrados (el equivalente de la Papelera de reciclaje de Windows)



FIGURA 1.7 Papelera de KDE

8.3 Iconos dinámicos para los soportes removibles

Hay iconos para los dispositivos removibles en su sistema (unidad de CD-ROM, unidad de disquetes, unidades ZIP/JAZ, etc.).



FIGURA 1.8 Iconos dinámicos de KDE

8.4 Panel

El panel es una barra que se ve en la parte inferior del escritorio donde tiene los siguientes componentes principales:



FIGURA 1.9 Panel de KDE



8.5 El Menú Principal

Da acceso a todos los programas instalados en su sistema. Es el equivalente del menú Inicio de Windows. Los programas están organizados en categorías convenientes, de manera que se pueda encontrar fácilmente la aplicación que necesita/desea ejecutar.



FIGURA 1.10 Menú de inicio

8.6 Mostrar Escritorio

Este botón sirve para minimizar todas las ventanas abiertas.



FIGURA 1.11 Botón para minimizar ventanas

8.7 Applet para Cambio de Escritorio

Hace que el cambio entre escritorios virtuales sea tan fácil como un, dos, tres.



FIGURA 1.12 Paginador

8.8 Applet de cambio de resolución de pantalla

Le permite cambiar la resolución de la pantalla. Con resoluciones mayores, se tiene más espacio para correr aplicaciones en el escritorio.



FIGURA 1.13 Cambio de resolución en la pantalla



8.9 Cambiar la apariencia de su escritorio

Para cambiar el esquema de colores del escritorio es Sistema+Configuración+KDE+LookNFeel->Colores en el menú principal. En la lista Esquema de colores hay esquemas de colores predefinidos. Elija el que le guste y haga clic sobre el botón Aplicar.

Al hacer clic sobre Guardar esquema se podrá guardar el esquema de colores para un uso posterior; preguntara el nombre del esquema, completarlo y hacer clic sobre Aceptar. Al hacer clic sobre Eliminar Esquema, se eliminara el esquema de colores seleccionado en ese momento.

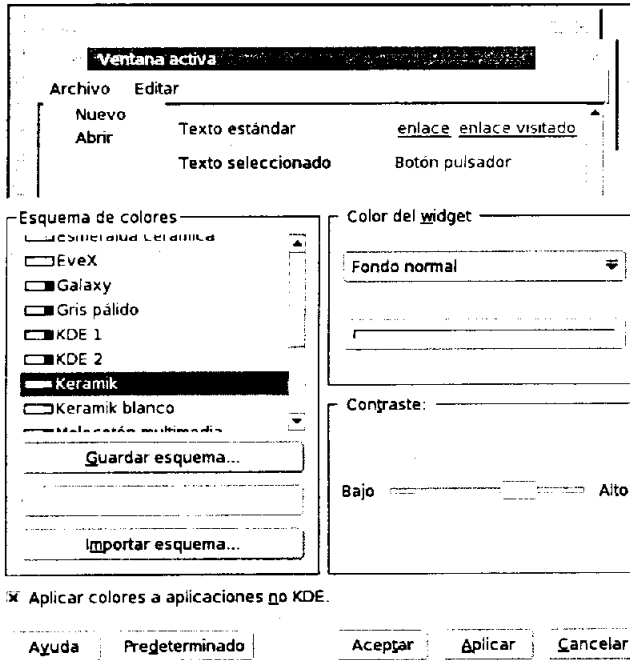


FIGURA 1.14 Personalizar los colores de ventanas y textos

8.10 Cambiando la resolución de la pantalla

Se hace clic sobre el applet de cambio de resolución de pantalla mostrara una lista de todos los tamaños y tasas de refresco disponibles para la combinación del monitor y tarjeta de vídeo. Los ajustes activos están marcados con una tilde.

Para cambiar cualquier parámetro, simplemente se debe seleccionar en la lista y aparecerá una ventana que pregunta si esta satisfecho con los ajustes. Si se hace clic sobre el botón Aceptar configuración, los ajustes se aplicarán de inmediato, de lo contrario se retienen los ajustes previos.

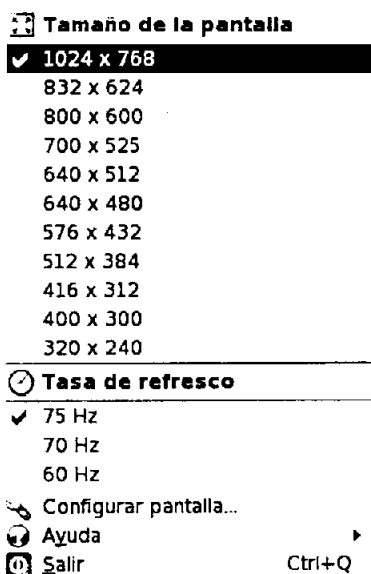


FIGURA 1.15 Personalizar tamaño de pantalla

8.11 Administrando archivos

Los administradores de archivos han crecido para convertirse en herramientas multitarea, que no sólo se ocupan de las tareas básicas tales como copiar y mover archivos de un lado a otro. De hecho con Konqueror, se puede administrar archivos, examinar una red LAN, reproducir archivos de audio tales como MP3s, navegar por la web, y más.

8.12 Ventana principal

Se puede acceder al administrador de archivos haciendo clic sobre el icono personalizado ubicado en la parte superior izquierda del escritorio.

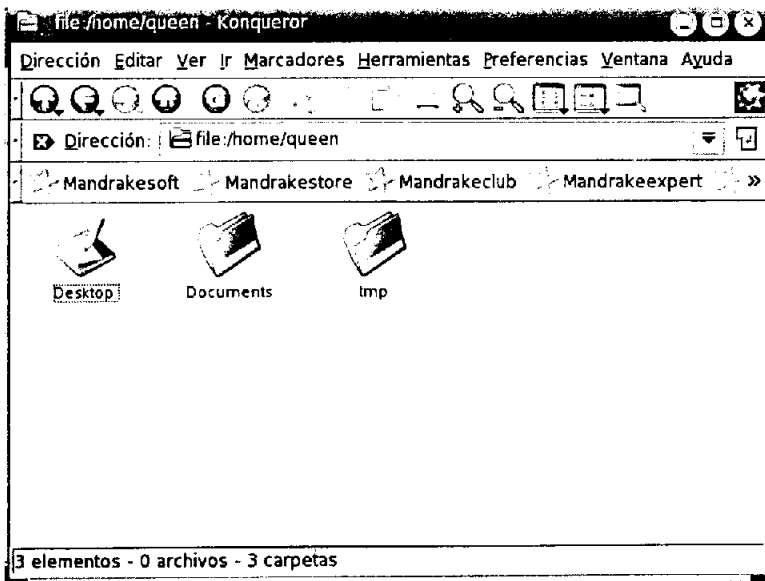


FIGURA 1.16 Administrador de archivos

9 Instalar JDK

Generalmente es un ejecutable que realiza la instalación en el directorio raíz del disco duro. Dejando el ambiente de trabajo en `c:\jdkVERSION`, donde versión corresponde al número de la versión del entorno Java.

Para este caso el directorio será `c:\jdk1.4.2`

Definir `JAVA_HOME` en las variables del sistema

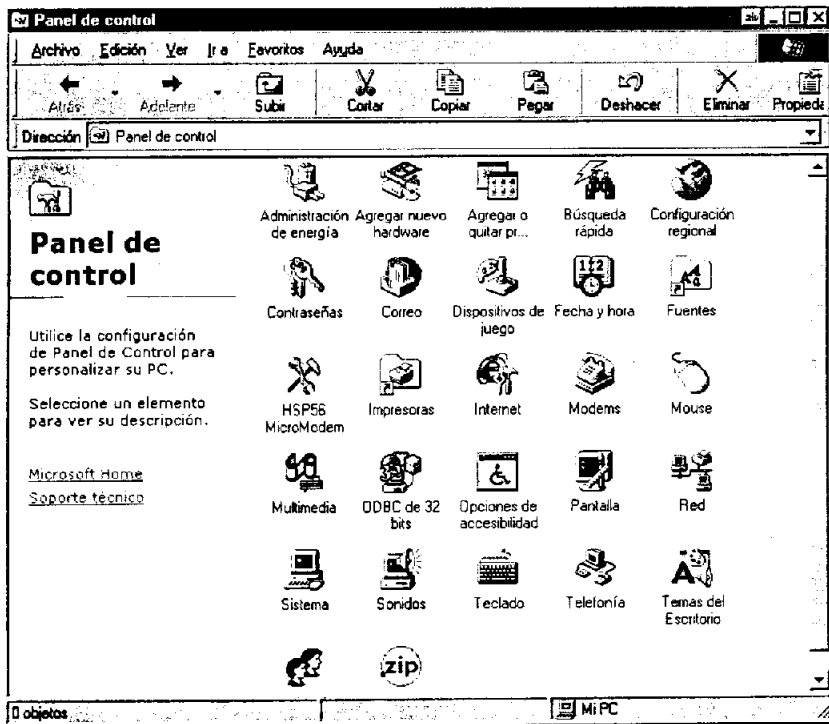


FIGURA 1.17 Panel de control

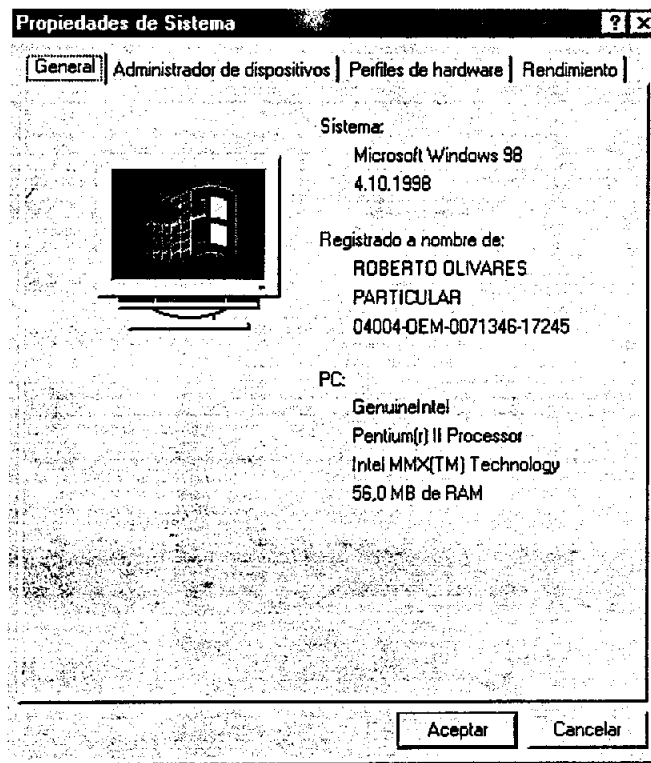


FIGURA 1.18 Propiedades del sistema

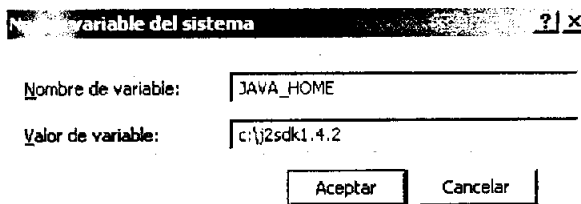


FIGURA 1.19 Variables del Sistema

Verificar que la variable está creada y definida correctamente: Abrir una ventana de comandos del sistema operativo y ejecutar el comando `echo %JAVA_HOME%`
Debe aparecer el directorio que se registro asociado a esta variable



10 Instalar Apache

Iniciar el archivo de instalación. Generalmente se puede hacer desde Inicio, Ejecutar o bien desde un Explorador de Windows

Al abrir el instalador pide aceptación de Licencia de Uso del producto, se debe aceptarla para poder continuar.

En el tipo de instalación hay 2 opciones "Typical" y "Custom", elegir **Custom**.

En la siguiente ventana, en la parte de abajo aparece el directorio donde Apache será instalado, junto un botón que dice Change

Al dar change se elige el directorio donde se instalara apache que es **c:\diploweb**

El objetivo de cambiar el directorio de instalación es asegurar para evitar errores por los nombres que Windows permite en sus archivos que tengan espacios, etc.

Al dar un nuevo directorio se debe continuar con la instalación hasta que indique que la instalación termino.

El servidor de apache queda instalado en el directorio **c:\diploweb\Apache2**

Elementos de software para un proyecto de desarrollo en WWW

11 Manejador de Base de Datos: MYSQL

MySQL es un sistema de administración de bases de datos relacionales, donde una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un solo lugar.

La ventaja es que esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se necesitan consultar datos.

La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es en general el lenguaje más usado y estandarizado de acceso a bases de datos relacionales.



Java Virtual Machine Integral al servidor:

12 Jakarta Tomcat

Primero se necesita saber que Jakarta es un proyecto que fue creado y mantenido bajo soluciones open source en la plataforma Java. Esto explica que el servidor donde se instale tenga previamente instalada la plataforma Java (JSDK).

Los productos Jakarta son desarrollados y distribuidos a través de varios subproyectos. La solución Tomcat que es una implementación de JavaServer y Server 2.3 es sólo una parte del proyecto Jakarta

13 Instalar Tomcat

Para desempaquetar el archivo **jakarta-tomcat-4.1.24.tar.gz**, es recomendable que el usuario se coloque en el directorio donde esté situado el archivo.

Ahora se debe desempaquetar y descomprimir el archivo con la siguiente instrucción:

```
tar -zxvf jakarta-tomcat-4.1.24.tar.gz
```

Esto debió crear un directorio llamado **jakarta-tomcat-4.1.24** el cual contiene todos los archivos de tomcat. En el ejemplo anterior se observa que el nombre del directorio es muy largo y difícil de recordar, por lo que se recomienda cambiarlo por uno más sencillo. Convencionalmente se le pone el nombre de jakarta-tomcat con la siguiente instrucción:

```
mv jakarta-tomcat-4.1.24 jakarta-tomcat
```

Ahora se pueden cambiar los archivos a un directorio donde puedan ser ejecutados por todos los usuarios, como por ejemplo /usr/local. Este paso se puede hacer de la siguiente manera:

```
mv jakarta-tomcat /usr/local
```

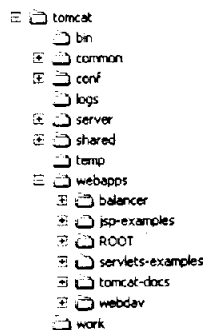


FIGURA 1.20 Directorios de Tomcat

Cada directorio tiene un propósito, pero por ahora sólo se necesita saber que en **bin** están los **scripts** de arranque del servidor, y en **webapps** las aplicaciones Web.



13.1 Configuración

Para que funcione correctamente es necesario declarar una variable de ambiente llamada **CATALINA_HOME** que contenga la ruta donde está instalado **jakarta-tomcat**, y esto se puede lograr editando el archivo `/etc/profile` de la siguiente manera:

```
vi /etc/profile
```

Y agregando la siguiente línea al final del archivo:

```
export CATALINA_HOME=/usr/local/jakarta-tomcat
```

Para que el cambio se realice se debe obligar al procesador a que lea el archivo, para que las variables de ambiente sean creadas. Lo anterior se logra con la siguiente instrucción:

```
source /etc/profile
```

Ahora **jakarta-tomcat** tiene todo lo necesario para poder funcionar correctamente.

13.2 Inicializar el servidor de Jakarta

Para que se inicie el servicio se debe teclear la siguiente instrucción:

```
SCATALINA_HOME/bin/startup.sh
```

13.3 Verificar la instalación

Sólo resta verificar que el servidor de **jakarta** esté funcionando correctamente, para lo cual, se debe teclear la siguiente dirección URL en el navegador de web:

```
http://localhost:8080/
```

El número 8080 en la dirección, le indica al navegador que se conecte al puerto 8080 del equipo. Como resultado, se debe observar en el navegador una página que indica que se ha instalado correctamente **Jakarta-Tomcat**. Dicha página contiene unos pequeños ejemplos que se pueden probar para asegurarse de que el servidor está correctamente instalado.

13.4 Configuraciones Adicionales

En caso de que sea necesario que el servidor de **jakarta-tomcat** atienda peticiones por el puerto 80, se debe modificar el archivo:

```
SCATALINA_HOME/conf/server.xml
```

**NOTA:**

Debe tomarse en cuenta que los puertos menores al 1024 sólo pueden ser utilizados por el usuario "root").

El archivo **SCATALINA_HOME/conf/server.xml** también puede ser utilizado para modificar algunas otras opciones del servidor de jakarta-tomcat.

Si se hiciera algún cambio en el archivo **SCATALINA_HOME/conf/server.xml** se debe detener el servicio y después se debe volver a iniciar, a fin de que los cambios puedan ser apreciados. Esto se logra con la instrucción:

SCATALINA_HOME/bin/shutdown.sh

Recordar que para iniciar el servidor, se debe utilizar la instrucción:

SCATALINA_HOME/bin/startup.sh



INDICE	MODULO 2	Página
Análisis y Diseño del Sistema de Consultas..... 33		
1	¿Qué es UML?	33
1.1	Diagramas de Implementación.....	33
1.2	Diagramas de implementación (2).....	33
1.3	Diagramas de implementación (3).....	33
1.4	Diagramas de Comportamiento.....	34
1.5	Diagramas de Comportamiento (2).....	34
1.6	Diagramas de Comportamiento (3).....	34
1.7	Diagramas de Comportamiento (4).....	34
2	Diagrama de casos de uso.....	35
3	Diagramas de Clases.....	36
4	Elementos de un Diagrama de colaboración.....	36
4.1	Elementos de un Diagrama de colaboración (2).....	37
4.2	Elementos de un Diagrama de colaboración (3).....	37
4.3	Elementos de un Diagrama de colaboración (4).....	37
4.4	Elementos de un Diagrama de colaboración (5).....	38
5	Elementos de los Diagramas de Estados.....	38
5.1	Elementos de los Diagramas de Estados (2).....	38
5.2	Elementos de los Diagramas de Estados (3).....	39
6	Diagramas de componentes.....	40
7	Diagramas de despliegue (deployment).....	41
8	Implementación de una Clínica de Especialidades Médicas.....	42



9	Objetivo General.....	42
10	Objetivo Específico.....	42
11	¿Qué hace?	42
12	¿Cómo lo hace?	42
13	Usuarios.....	43
14	Back Office.....	43
15	Servicios Básicos.....	44
16	Diagrama de clases de una clínica de especialidades médicas.....	48
17	Diccionario	49
18	Diagrama de casos de uso de una clínica de especialidades médicas.....	50
19	Diagrama de colaboración de una clínica de especialidades médicas.....	51



Análisis y Diseño del Sistema de Consultas

1 ¿Qué es UML?

- Es una metodología para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos
- Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software

1.1 Diagramas de Implementación

- Muestran los aspectos físicos del sistema.
- Incluyen la estructura del código fuente y la implementación, en tiempo de implementación
 - Diagrama de componentes
 - Diagrama de plataformas de despliegue

1.2 Diagramas de implementación (2)

- Diagrama de componentes
 - Muestra la dependencia entre los distintos componentes de software, incluyendo componentes de código fuente, binario y ejecutable. Un componente es un fragmento de código software (un fuente, binario o ejecutable) que se utiliza para mostrar dependencias en tiempo de compilación.

1.3 Diagramas de implementación (3)

- Diagrama de despliegue
 - Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución



1.4 Diagramas de Comportamiento

- Muestran interacciones entre objetos en un escenario del sistema.
 - Diagrama de secuencia.
 - Diagrama de colaboración.
 - Diagrama de estado.
 - Diagrama de actividad.

1.5 Diagramas de Comportamiento (2)

- Diagrama de secuencia.
 - Muestran las interacciones entre un conjunto de objetos, ordenadas según el tiempo en que tienen lugar.
 - El objeto puede existir sólo durante la ejecución de la interacción, se puede crear o puede ser destruido durante la ejecución de la interacción

1.6 Diagramas de Comportamiento (3)

- Diagrama de colaboración.
 - Muestra la interacción entre varios objetos y los enlaces que existen entre ellos
 - Muestra las relaciones entre los objetos, no la secuencia en el tiempo en que se producen los mensajes
 - Los diagramas de secuencias y los diagramas de colaboraciones expresan información similar, pero en una forma diferente.

1.7 Diagramas de Comportamiento (4)

- Diagramas de actividad
 - se utilizan para mostrar el flujo de operaciones que se desencadenan en un procedimiento interno del sistema.

2 Diagrama de casos de uso

- sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas (Actores)

Procesar una orden de compra del cliente

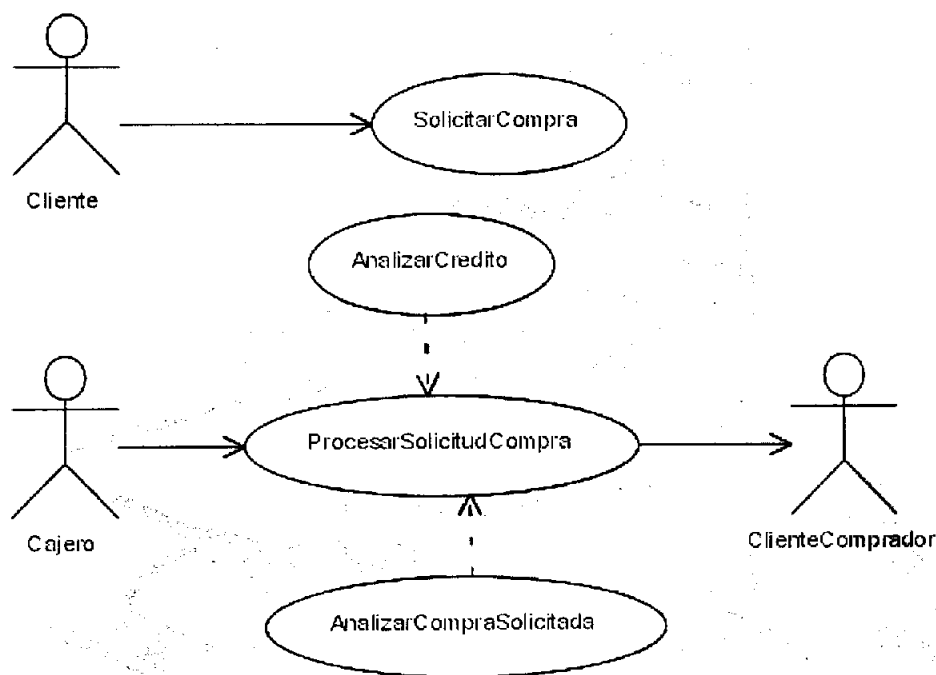


FIGURA 2.1 Ejemplo de diagrama de casos de uso

3 Diagramas de Clases

Representan un conjunto de elementos del modelo que son estáticos, como las clases y los tipos, sus contenidos y las relaciones que se establecen entre ellos.

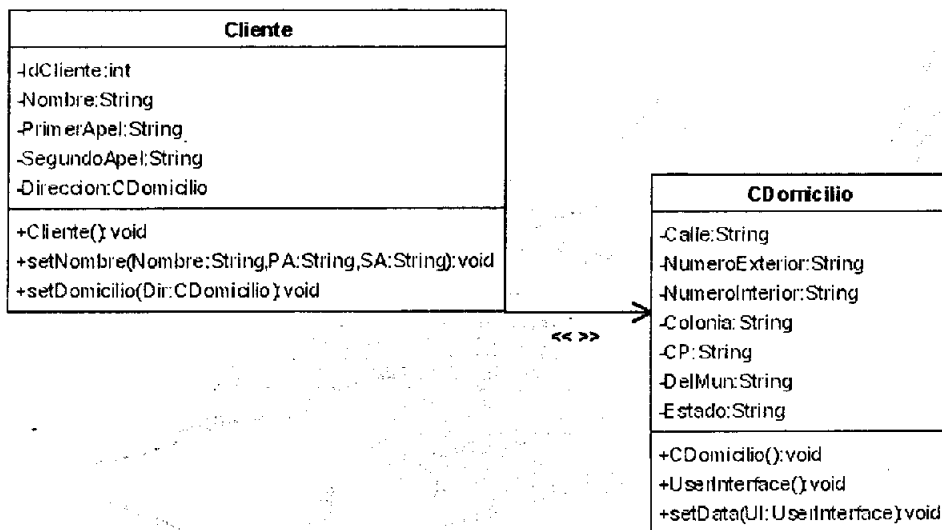


FIGURA 2.2 Ejemplo de diagrama de clases

4 Elementos de un Diagrama de colaboración

- **Objeto:** Un rectángulo que contiene el nombre y la clase del objeto en un formato

nombreObjeto : nombreClase.



4.1 Elementos de un Diagrama de colaboración (2)

- **Enlaces**
 - Es una instancia de una asociación en un diagrama de clases.
 - Se representa como una línea continua que une a dos objetos, acompañada por un número que indica el orden dentro de la interacción.

4.2 Elementos de un Diagrama de colaboración (3)

- **Enlaces.....**
 - Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones.
 - Pueden utilizar estereotipos para indicar si el objeto que recibe el mensaje es un atributo, un parámetro de un mensaje anterior, si es un objeto local o global.

4.3 Elementos de un Diagrama de colaboración (4)

- **Flujo de mensajes:** Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cerca de un enlace.
- **Marcadores de creación y destrucción de objetos:** Puede mostrarse en la gráfica qué objetos son creados y destruidos, agregando una restricción con la palabra *new* o *delete* respectivamente.

4.4 Elementos de un Diagrama de colaboración (5)

Cobro de estancia en un estacionamiento con tarifas por hora o fracción

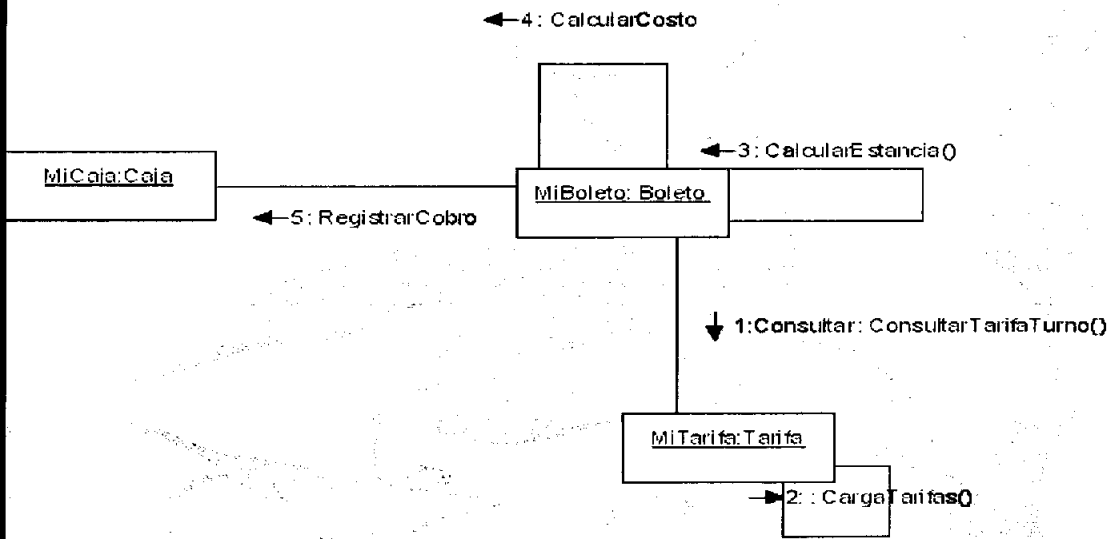


FIGURA 2.3 Ejemplo de diagrama de colaboración

5 Elementos de los Diagramas de Estados

Estado: Identifica un período de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos.

5.1 Elementos de los Diagramas de Estados (2)

Eventos: Es una ocurrencia que puede causar la transición de un estado a otro de un objeto. Esta ocurrencia puede ser una de varias cosas:

- Condición que toma el valor de V o F
- Recepción de una señal de otro objeto en el modelo.
- Recepción de un mensaje.
- Paso de cierto período de tiempo, después de entrar al estado

5.2 Elementos de los Diagramas de Estados (3)

Transición: Es una relación de tres o más estados en una transición de múltiples fuentes o múltiples destinos.

Diagramas de Estados

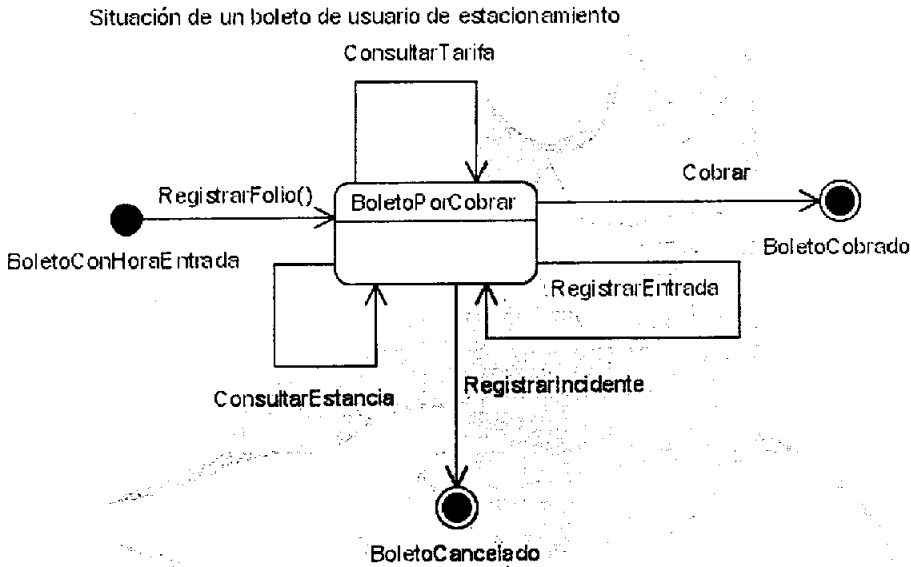


FIGURA 2.4 Ejemplo de diagrama de estado

6 Diagramas de componentes

Representa las componentes físicas de la aplicación.

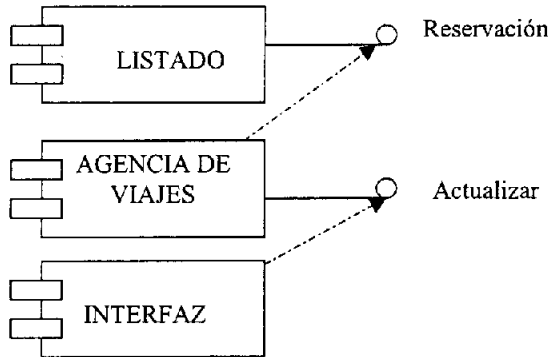


FIGURA 2.5 Ejemplo de diagrama de componentes

7 Diagramas de despliegue (deployment)

Representa la visualización de los componentes sobre los dispositivos físicos.

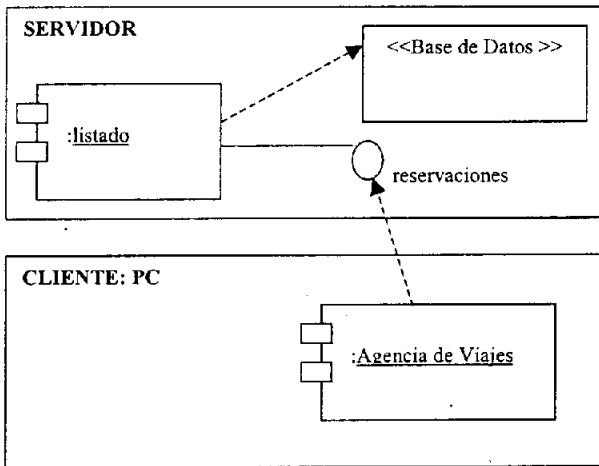


FIGURA 2.6 Ejemplo de diagrama de despliegue



8 Implementación de una Clínica de Especialidades Médicas

Es una Clínica de Especialidades Médicas privada e independiente que le ofrece sus servicios de salud, con una alta experiencia a favor de las comodidades, que nos permite atenderlo con la calidad que usted se merece.

9 Objetivo General:

Garantizar al **Paciente** un servicio que satisfagan sus necesidades en forma oportuna y eficiente por medio de Internet.

10 Objetivo Especifico:

Diseñar un Sistema que permita Administrar, Solicitar o Reservar una Consulta de una forma eficiente para el Paciente por medio de Internet

11 ¿Qué hace?

Registra a los Pacientes por diferentes Especialidades.

12 ¿Cómo lo hace?

Navegando por medio de Internet.



13 Usuarios:

- Público en General
- Cualquier Usuario en Internet
- Operador
- Administrador

14 Back Office:

- Llevar un control de información de los Pacientes por Especialidad
(Datos Personales, Domicilio, consultas, etc.)
- Llevar un control de información de los Especialistas
(Turno, Consultorio, Cirugía, Descansos, etc.)



15 Servicios Básicos:

- **Centro de Especialidades Médicas**

Cardiología:

Dermatología

Fisioterapia

Pediatría

Ginecología

Psicología

Urología

Traumatología

Oftalmología

Neurología

Alergología

Cirugía Plástica

Proctología

Falta definición general de cada especialidad

- **Otras**

Cardiología - Pediatría

Alergología - Hematología



- **También se cuenta con Terapias**

- Terapia Física
- Terapia Respiratoria
- Terapia de Varices
- Terapia de dolor Crónico
- Terapia de Lenguaje
- Estimulación temprana

- **Y entre otros Servicios están:**

- Laboratorio Clínico
- Radiografías
- Vacunas
- Farmacia
- Enfermeras en turno a domicilio
- Médicos en turno a domicilio
- Ambulancias



- **Registro del Paciente** { Políticas al Registrar un Paciente
 - Nombre
 - Edad
 - Sexo
 - Domicilio
 - Ciudad o Estado
 - País
 - Código Postal
 - Teléfono
 - Tipo de Sangre
 - Alérgico a

- **Reservación por Especialidad**
 - Tipo de Especialidad
 - Doctor
 - Fecha
 - Horario

- **Horarios disponibles**
 - Por especialidad
 - Por Fecha
 - Por horario



- **Información de Costos**

Consulta

Cirugía

Parto

Vacunas

- **Información de Pago**

Tarjeta de crédito

- **Descuentos especiales**

- **Confirmar Consulta**

- **Cancelación de Consulta**

{ Políticas al Cancelar Cita

- **Ubicación de la Clínica (Mapa)**

16 Diagrama de clases de una Clínica de Especialidades Médicas

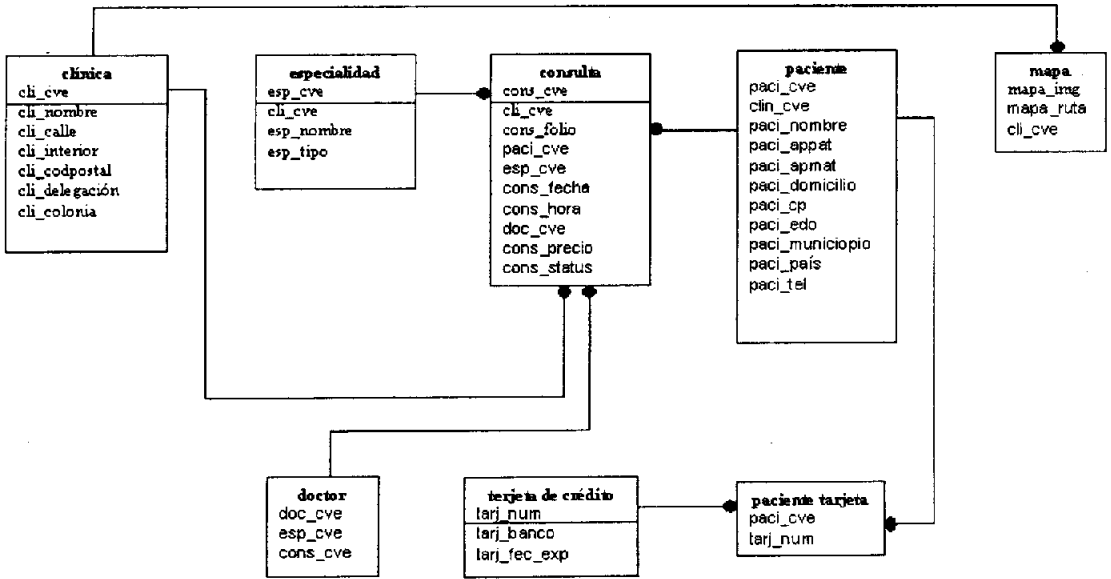


FIGURA 2.8 Ejemplo de diagrama de clases de una clínica



17 Diccionario de una Clínica de Especialidades Médicas

ENTIDAD	NOMBRE DE ENTIDAD	DESCRIPCION	OBSERVACIONES	TIPO
clin_cve	clinica	clave de la clínica		numérico
clin_nombre	clinica	nombre de la clínica		texto
clin_domicilio	clinica	domicilio de la clínica		texto
clin_especialidad	clinica	especialidad de la clínica	Alergología, Cardiología, Nutrición Dermatología, Fisioterapia, Pediatría, Ginecología, Psicología, Urología, Neurología,	numérico
esp_cve	especialidad	clave de especialidad		numérico
esp_nombre	especialidad	nombre de especialidad		numérico
esp_tipo	especialidad	tipo de especialidad		numérico
clin_cve	clínica	clave de la clínica		numérico
cons_folio	clínica	folio de la consulta		numérico
paci_cve	clínica	clave del paciente		numérico
esp_cve	clínica	clave de especialidad		numérico
cons_fecha	clínica	fecha de consulta		numérico
cons_hora	clínica	hora de consulta		numérico
doc_cve	clínica	clave del doctor		numérico
cons_precio	clínica	precio de la consulta		numérico
cons_status	clínica	status de la consulta		numérico
clin_cve	paciente	clave de la clínica		numérico
paci_cve	paciente	clave de paciente		numérico
paci_nombre	paciente	nombre de paciente		texto
paci_appat	paciente	apellido paterno del paciente		texto
paci_apmat	paciente	apellido materno del paciente		texto
paci_domicilio	paciente	domicilio del paciente		texto
paci_cp	paciente	código postal del paciente		numérico
paci_edo	paciente	estado del paciente		texto
paci_municipio	paciente	municipio del paciente		texto
paci_pais	paciente	país del paciente		texto
paci_tel	paciente	teléfono del paciente		numérico
doc_cve	doctor	clave del doctor		numérico
esp_cve	doctor	clave de especialidad		numérico
cons_cve	doctor	clave del consultorio		numérico
tarj_num	tarjeta crédito	numero de tarjeta		numérico
tarj_banco	tarjeta crédito	tipo de banco de tarjeta		numérico
tarj_fec_exp	tarjeta crédito	fecha de expedición de tarjeta		numérico
paci_cve	paciente tarjeta	clave de paciente		numérico
tarj_num	paciente tarjeta	numero de tarjeta		numérico
mapa_imag	mapa	imagen de mapa		numérico
mapa_ruta	mapa	ruta de mapa		numérico



18 Diagrama de casos de uso de una Clínica de Especialidades Médicas

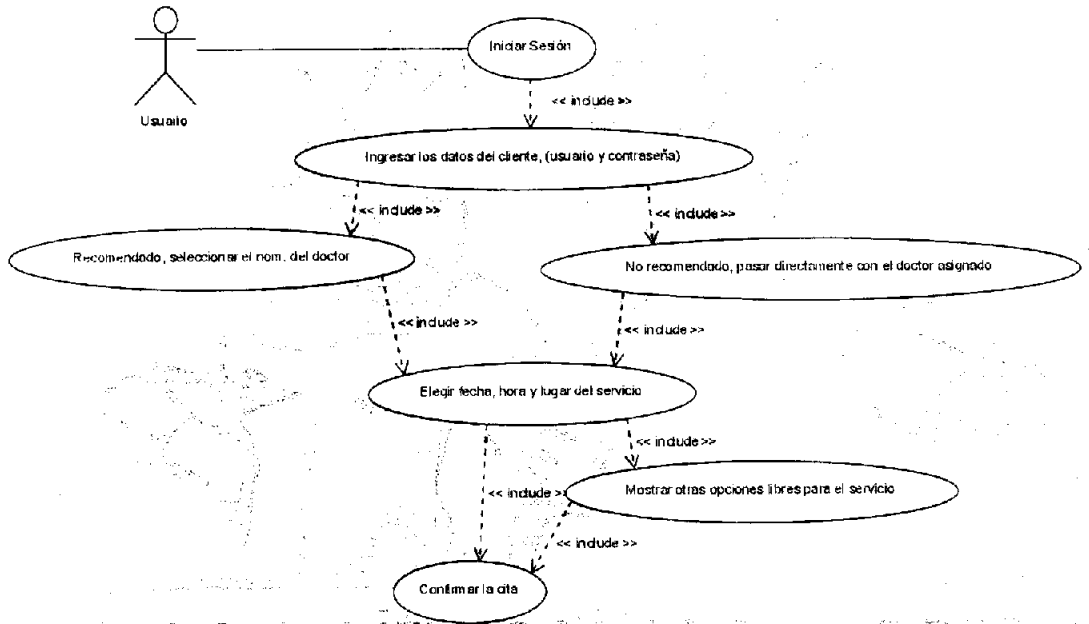


FIGURA 2.9 Ejemplo de diagrama de casos de uso de una clínica



19 Diagrama de colaboración de una Clínica de Especialidades Médicas

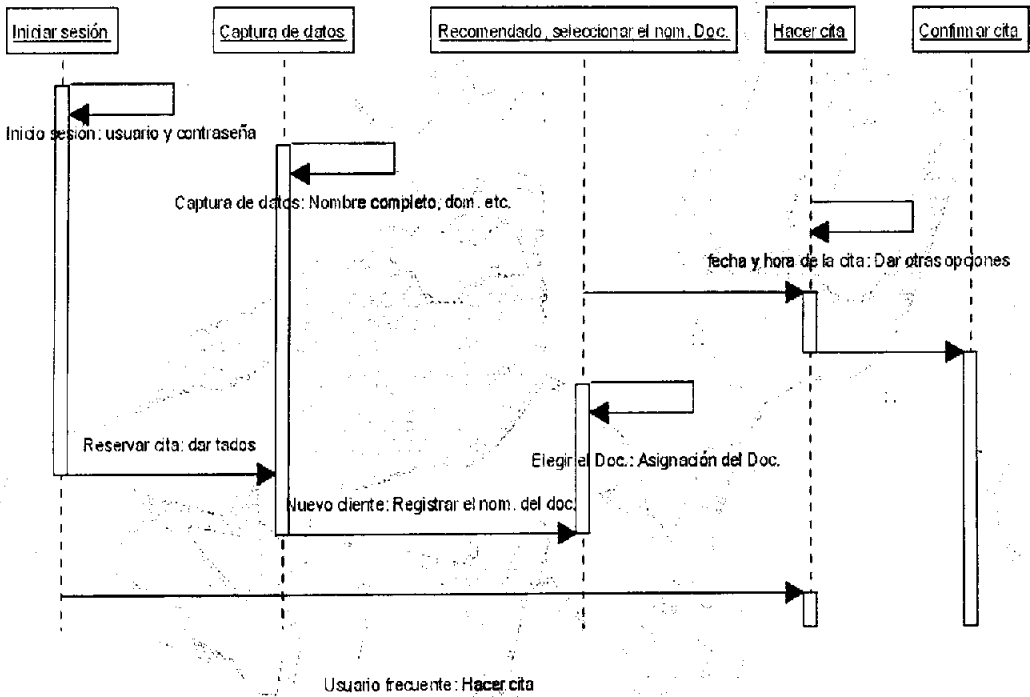


FIGURA 2.10 Ejemplo de diagrama de colaboración de una clínica



INDICE	MODULO 3	Página
Programación Orientada a Objetos.....		54
1 Enfoque de Orientación a Objetos.....		54
1.1 El objeto.....		54
1.2 El paradigma de orientación a objetos.....		54
1.3 Clases e Instancias		54
2 Características de un Lenguaje OO.....		55
2.1 Encapsulamiento.....		55
2.1.2 Polimorfismo.....		55
2.2 Características de un Lenguaje OO (2).....		55
2.3 Características de un Lenguaje OO (3).....		55
2.3.1 Herencia.....		55
2.4 Características de un Lenguaje OO (4).....		56
2.4.1 Herencia.....		56
3 Asociaciones estáticas.....		56
3.1 Asociaciones estáticas (2).....		56
3.2 Asociaciones de agregación o composición.....		56
4 Proceso de Análisis de Requerimientos.....		57
5 Modelo de Requerimientos.....		57
6 Modelo de Casos.....		57
6.1 Modelo de Casos (2).....		58
6.2 Modelo de Casos (3).....		58
6.3 Modelo de Casos (4).....		58



7 Modelo de Interfaz.....	58
7.1 Modelo de Interfaz (2).....	59
7.2 Modelo de Interfaz (3).....	59
8 Modelo del Dominio del Problema.....	59
9 Introducción a Java.....	60
9.1 Características de Java.....	60
9.2 Potente.....	61
9.3 Simple.....	62
9.4 Interactivo y orientado a red.....	62
9.5 Seguridad.....	63
9.6 Lenguaje basado en C++.....	64
9.7 Gestión de la Entrada/Salida.....	64
9.8 Diferentes tipos de aplicaciones.....	64
JDK y Eclipse.....	65
10 El ambiente de desarrollo Eclipse.....	65



1 Enfoque de orientación a objetos

- El enfoque de orientación a objetos es una forma de observar la realidad, que se basa en el concepto de Objeto:

Es todo aquello que tiene características que lo hacen único e indivisible dentro del entorno al que pertenece. Siempre es posible establecer propiedades o atributos de un objeto, lo mismo que su grado de respuesta a estímulos externos (comportamientos del objeto).

1.1 El objeto

- Tiene características que lo hacen único
- Todo objeto participa en un entorno
- Un objeto tiene
 - Características
 - Comportamientos o respuestas a estímulos de su entorno
 - Relación con otros objetos de su entorno

1.2 El paradigma de orientación a objetos

- Propone resolver problemas en función de la descripción de objetos en un entorno dado.
- Llama la atención por que propone reutilización de código, generación de sistemas estables y mantenibles
- Enfoca la metodología en el uso de objetos, relaciones de composición y herencia, así como en la definición de clases.

1.3 Clases e Instancias

- La clase trata de un concepto abstracto del objeto.
- Cuando definimos características de objetos y existen objetos que comparten algunas de ellas o que son parecidas decimos que forman parte de la misma clase
- Una instancia es la ejemplificación de un objeto.
- La clase es la definición y la instancia es un objeto.



2 Características de un Lenguaje OO

2.1 Encapsulamiento

Capacidad de “ocultar” en la misma unidad funcional de información a los atributos y su comportamiento

2.1.2 Polimorfismo

Es la característica de que un objeto que pide la ejecución de una función a otro objeto no necesita saber la naturaleza del objeto dueño de la función, simplemente la “usa” o “solicita” sin saber detalles del otro objeto.

2.2 Características de un Lenguaje OO (2)

- Un ejemplo del polimorfismo sería la operación suma.
 - Al enviar dos operandos de cadena se aplica una concatenación
 - Al enviar dos operandos numéricos se aplica una operación aritmética

2.3 Características de un Lenguaje OO (3)

2.3.1 Herencia

- Este concepto se asocia a la generalización - especialización en la definición de clases.
- Permite definir estructuras de objetos más cercanos a su representación en el mundo

Un ejemplo sería

- Trabajador
 - ❖ Trabajador de Confianza
 - ❖ Trabajador sindicalizado

Ambos tienen RFC, Nombre, Teléfono, pero solo el segundo tiene clave sindical y cuota sindical.



2.4 Características de un Lenguaje OO (4)

2.4.1 Herencia

- La clase de la cual se crea una relación de herencia se llama superclase
- La clase resultante de una relación de herencia se llama subclase
- Una clase base, siempre es abstracta

Una clase abstracta no puede ser instanciada

3 Asociaciones estáticas

- Es un vínculo entre dos objetos
- Las asociaciones en el caso de clases implica que hay una interacción pero en ningún momento que se trata de colecciones de objetos
- La relación se expresa mediante una línea continua que además puede llevar un nombre y una cardinalidad

3.1 Asociaciones estáticas (2)

- Dada la clase Cliente y Agente, según la regla del negocio: "todo cliente tiene asignado un Agente" que definamos una relación entre dichas clases.
- Una clase puede tener más de una relación con otras clases, siendo fundamental respetar la cardinalidad y propósito de la asociación.

3.2 Asociaciones de agregación o composición

- Un caso particular de asociación estática es el cuando se encuentra que un objeto es parte del otro.
- Una Factura estará agregada por su Detalle Factura
- Un Artículo estará agregado por su Estructura de Precios



4 Proceso de Análisis de Requerimientos

- La primera actividad en el desarrollo de Sistemas involucra al equipo de trabajo en el conocimiento del problema para establecer los requerimientos en términos confiables y especificaciones formales concretas.
- El proceso de Análisis de Requerimientos pretende acelerar las curvas de aprendizaje mediante subprocesos que permitan extraer los conocimientos clave del equipo.
- El PAR nos permite generar un Modelo de Requerimientos.
- El Modelo de Requerimientos se compone de:
 - **Modelo de casos**
 - **Modelo de Interfaz**
 - **Modelo de Dominio del Problema**

5 Modelo de Requerimientos

- El **Modelo de Casos** extrae el conocimiento funcional fundamental del problema de una forma estructurada y progresiva, siendo la base para establecer la estructura del sistema.
- El **Modelo de Interfaz** establece el vínculo visual entre el desarrollador y el usuario para concretar aspectos de la interacción que el sistema pudiese tener con su entorno
- El **Modelo del Dominio del Problema** se establecerán los principales objetos que constituirán al sistema y las relaciones que tienen entre sí

6 Modelo de Casos

- Plantea que todo principio sea el establecer las principales transacciones que contendrá el sistema.
- Una transacción es una interacción del sistema con sus "agentes externos"
- La labor del Analista es escuchar todas las posibles transacciones de parte del usuario y filtrar aquellas que no sean fundamentales.



6.1 Modelo de Casos (2)

- Cada transacción que se modela recibe el nombre de Caso
- Cada Caso requiere que se especifique no solo su nombre sino también la secuencia de pasos necesarios para realizarlo
- Cada interacción con el sistema será realizada por un Agente externo a él
- Un Agente Externo se denomina Actor

6.2 Modelo de Casos (3)

- Los actores no forman parte del sistema, sólo interactúan con él.
- Un Actor no necesariamente tiene que ser una persona
- El Caso puede tener Flujos Alternos o bien Flujos de Excepción para los casos que no contempla la secuencia de pasos principales.

6.3 Modelo de Casos (4)

- En la notación de UML el modelo de Casos se define a través de un Diagrama de Casos de Uso.
- Un actor se define con la figura de alambre de una persona
- Un Caso de uso se representa por medio de un Ovalo con su nombre al centro del mismo
- La relación Actor-Caso se define mediante flechas

7 Modelo de Interfaz

- Establece un vínculo entre el Usuario y el Analista
- Muestra gráficamente cada caso planteado
- Consta de la definición de las interfaces principales que participarán en la ejecución de un caso cuando el sistema exista.
- Interfaces significa: Pantallas, Reportes, Llamadas a Sistemas



7.1 Modelo de Interfaz (2)

- El modelo estará compuesto por las interfaces más significativas
- Las pantallas no necesariamente provienen de un programa funcional incluso pueden ser dibujadas mediante una herramienta de gráficos ya que su único objetivo será describir al usuario junto con el modelo de Casos el comportamiento del sistema

7.2 Modelo de Interfaz (3)

- En el caso de los reportes, éstos se pueden establecer por medio de procesadores de texto o bien a mano en papel.
- Para cuando se definen interfaces con otros sistemas deben establecerse la forma en que se estará dando dicha llamada

Nombre_Del_Proceso(Argumentos): Valor de retorno

8 Modelo del Dominio del Problema

- Tiene como objetivo identificar los objetos de información y las relaciones que guardan entre sí.
- Se plasma en un Diagrama de Clases.
- Para cada caso del Modelo de Casos existe un Diagrama de Clases



9 Introducción a Java

Java fue diseñado en 1990 por James Gosling. De Sun Microsystems, como software para dispositivos electrónicos de consumo. Curiosamente, todo este lenguaje fue diseñado antes de que diese comienzo la era World Wide Web, puesto que fue diseñado para dispositivos electrónicos como calculadoras, microondas y la televisión interactiva.

En los primeros años de la década de los noventa, Sun Microsystems decidió intentar introducirse en el mercado de la electrónica del consumo y desarrollar programas para pequeños dispositivos electrónicos. Tras unos comienzos dudosos, Sun decidió crear una filial, denominada FirstPerson Inc., para dar margen de maniobra al equipo responsable del proyecto.

Inicialmente, Java se llamó Oak (roble en inglés), aunque tuvo que cambiar de denominación, debido a que dicho nombre ya estaba registrado por otra empresa.

9.1 Características de Java

No es arriesgado afirmar que Java supone un significativo avance en el mundo de los entornos software, y esto viene avalado por tres elementos claves que diferencian a este lenguaje desde un punto de vista tecnológico:

- Es un lenguaje de programación que ofrece la potencia del diseño orientado a objetos con una sintaxis fácilmente accesible y un entorno robusto y agradable.
- Proporciona un conjunto de clases potente y flexible.
- Pone al alcance de cualquiera la utilización de aplicaciones que se pueden incluir directamente en páginas Web (aplicaciones denominadas *applets*).

Java aporta a la Web una interactividad que se había buscado durante mucho tiempo entre usuario y aplicación.

A lo largo de este apartado se estudian en detalle las principales características de Java.



9.2 Potente

a) Orientación a objetos

En este aspecto Java fue diseñado partiendo de cero, no siendo derivado de otro lenguaje anterior y no tiene compatibilidad con ninguno de ellos.

En Java el concepto de objeto resulta sencillo y fácil de ampliar. Además se conservan elementos "no objetos", como números, caracteres y otros tipos de datos simples.

b) Riqueza semántica

Pese a su simpleza se ha conseguido un considerable potencial, y aunque cada tarea se puede realizar de un número reducido de formas, se ha conseguido un gran potencial de expresión e innovación desde el punto de vista del programador.

c) Robusto

Java verifica su código al mismo tiempo que lo escribe, y una vez más antes de ejecutarse, de manera que se consigue un alto margen de codificación sin errores. Se realiza un descubrimiento de la mayor parte de los errores durante el tiempo de compilación, ya que Java es estricto en cuanto a tipos y declaraciones, y así lo que es rigidez y falta de flexibilidad se convierte en eficacia. Respecto a la gestión de memoria, Java libera al programador del compromiso de tener que controlar especialmente la asignación que de ésta hace a sus necesidades específicas. Este lenguaje posee una gestión avanzada de memoria llamada gestión de basura, y un manejo de excepciones orientado a objetos integrados. Estos elementos realizarán muchas tareas antes tediosas a la vez que obligadas para el programador.

d) Modelo de objeto rico

Existen varias clases que contienen las abstracciones básicas para facilitar a los programas una gran capacidad de representación. Para ello se contará con un conjunto de clases comunes que pueden crecer para admitir todas las necesidades del programador.

Además la biblioteca de clases de Java proporciona un conjunto único de protocolos de Internet.

El conjunto de clases más complicado de Java son sus paquetes gráficos AWT (*Abstract Window Toolkit*) y *Swing*. Estos paquetes implementan componentes de una interfaz de usuario gráfica básica común a todos los ordenadores personales modernos.



9.3 Simple

a) Fácil aprendizaje

El único requerimiento para aprender Java es tener una comprensión de los conceptos básicos de la programación orientada a objetos. Así se ha creado un lenguaje simple (aunque eficaz y expresivo) pudiendo mostrarse cualquier planteamiento por parte del programador sin que las interioridades del sistema subyacente sean desveladas.

Java es más complejo que un lenguaje simple, pero más sencillo que cualquier otro entorno de programación. El único obstáculo que se puede presentar es conseguir comprender la programación orientada a objetos, aspecto que, al ser independiente del lenguaje, se presenta como insalvable.

b) Completado con utilidades

El paquete de utilidades de Java viene con un conjunto completo de estructuras de datos complejas y sus métodos asociados, que serán de inestimable ayuda para implementar *applets* y otras aplicaciones más complejas. Se dispone también de estructuras de datos habituales, como *pilas* y *tablas hash*, como clases ya implementadas.

Existirá una interfaz *Observer/Observable* que permitirá la implementación simple de objetos dinámicos cuyo estado se visualiza en pantalla.

El JDK (*Java Development Kit*) suministrado por Sun Microsystems incluye un compilador, un intérprete de aplicaciones, un depurador en línea de comandos, y un visualizador de *applets* entre otros elementos.

9.4 Interactivo y orientado a red

a) Interactivo y animado

Uno de los requisitos de Java desde sus inicios fue la posibilidad de crear programas en red interactivos, por lo que es capaz de hacer varias cosas a la vez sin perder rastro de lo que debería suceder y cuándo. Para dar soporte a la utilización de múltiples hilos de programación (*multithread*).

Las aplicaciones de Java permiten situar figuras animadas en las páginas Web, y éstas pueden concebirse con logotipos animados o con texto que se desplace por la pantalla. También pueden tratarse gráficos generados por algún proceso. Estas animaciones pueden ser interactivas, permitiendo al usuario un control sobre su apariencia.



b) Arquitectura neutral

Java está diseñado para que un programa escrito en este lenguaje sea ejecutado correctamente independientemente de la plataforma en la que se esté actuando (Macintosh, PC, UNIX...). Para conseguir esto utiliza una compilación en una representación intermedia que recibe el nombre de *códigos de byte*, que pueden interpretarse en cualquier sistema operativo con un intérprete de Java

c) Trabajo en red

Java anima las páginas Web y hace posible la incorporación de aplicaciones interactivas y especializadas. Aporta la posibilidad de distribuir contenidos ejecutables, de manera que los suministradores de información de la Web pueden crear una página de hipertexto (*página Web*) con una interacción continuada y compleja en tiempo real; el contenido ejecutable es transferido literalmente al ordenador del usuario.

d) Applets

Una *applet* (miniaplicación) es un pequeño programa en Java transferido dinámicamente a través de Internet. Presentan un comportamiento inteligente, pudiendo reaccionar a la entrada de un usuario y cambiar de forma dinámica. Sin embargo, la verdadera novedad es el gran potencial que Java proporciona en este aspecto, haciendo posible que los programadores ejerzan un control sobre los programas ejecutables de Java que no es posible encontrar en otros lenguajes.

9.5 Seguridad

Existe una preocupación lógica en Internet por el tema de la seguridad: virus, caballos de Troya, y programas similares navegan de forma usual por la red, constituyendo una amenaza palpable. Java ha sido diseñado poniendo un énfasis especial en el tema de la seguridad, y se ha conseguido lograr cierta inmunidad en el aspecto de que un programa realizado en Java no puede realizar llamadas a funciones globales ni acceder a recursos arbitrarios del sistema, por lo que el control sobre los programas ejecutables no es equiparable a otros lenguajes.

Los niveles de seguridad que presenta son:

- Fuertes restricciones al acceso a memoria, como son la eliminación de punteros aritméticos y de operadores ilegales de transmisión.
- Rutina de verificación de los *códigos de byte* que asegura que no se viole ninguna construcción del lenguaje.
- Verificación del nombre de clase y de restricciones de acceso durante la carga.
- Sistema de seguridad de la interfaz que refuerza las medidas de seguridad en muchos niveles.

En futuras versiones se prevé contar también con encriptación y técnicas similares.



9.6 Lenguaje basado en C++

Java fue desarrollado basándose en C++, pero eliminando rasgos del mismo poco empleados, optándose por una codificación comprensible. Básicamente, encontramos las siguientes diferencias con C++:

- Java no soporta los tipos *struct*, *union* ni punteros.
- No soporta *typedef* ni *#define*.
- Se distingue por su forma de manejar ciertos operadores y no permite una sobrecarga de operadores.
- No soporta herencia múltiple.
- Java maneja argumentos en la línea de comandos de forma diversa a como lo hacen C o C++.
- Tiene una clase *String* que es parte del paquete *java.lang* y se diferencia de la matriz de caracteres terminada con un nulo que usan C y C++.
- Java cuenta con un sistema automático para asignar y liberar memoria, con lo que no es necesario utilizar las funciones previstas con este fin en C y C++.

9.7 Gestión de la Entrada/Salida

En lugar de utilizar primitivas como las de C para trabajar con ficheros, se utilizan primitivas similares a las de C++, mucho más elegantes, que permiten tratar los ficheros, sockets, teclado y monitor como flujos de datos.

De este modo se pueden utilizar dichas primitivas para cualquier operación de Entrada/Salida.

9.8 Diferentes tipos de aplicaciones

En Java podemos crear los siguientes tipos de aplicaciones:

- *Aplicaciones*: Se ejecutan sin necesidad de un navegador.
- *Applets*: Se pueden descargar de Internet y se observan en un navegador.
- *JavaBeans*: Componentes software Java, que se puedan incorporar gráficamente a otros componentes.
- *JavaScript*: Conjunto del lenguaje Java que puede codificarse directamente sobre cualquier documento HTML.
- *Servlets*: Módulos que permiten sustituir o utilizar el lenguaje Java en lugar de programas CGI (Common Gateway Interface) a la hora de dotar de interactividad a las páginas Web.



JDK y Eclipse

10 El ambiente de desarrollo Eclipse

La instalación y configuración de Eclipse, en un entorno de desarrollo integrado (IDE) que sirve para muchos lenguajes. Eclipse tiene interesantes utilidades para Java y además es gratuito.

Como se ha dicho, un IDE puede hacer el trabajo mucho más sencillo, sobre todo si el desarrollo ya va manejando un buen número de Clases. Además estos entornos nos permiten mucha más versatilidad para depurar programas puesto que tienen debuggers mucho más avanzados.

Eclipse es un IDE de código abierto. Hay más herramientas similares de código abierto disponibles pero he decidido usar esta, pues es la que creo que tiene mejor relación calidad y facilidad.

Su instalación es muy sencilla. Para ejecutarlo solo hay que arrancar el fichero Eclipse.exe. Una vez arrancado lo único que pedirá es la ruta por defecto donde eclipse guardara los proyectos que se crearán:

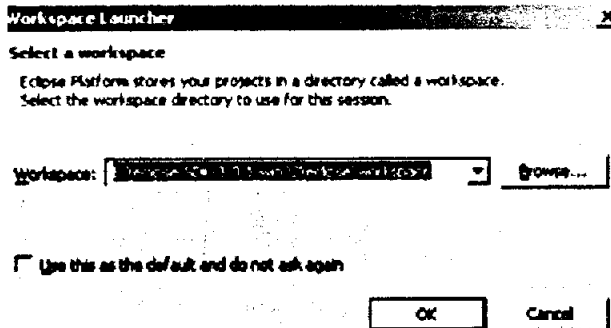


FIGURA 3.1 Ventana de ruta donde Eclipse guarda los proyectos

Aparecerá la ventana principal de Eclipse:

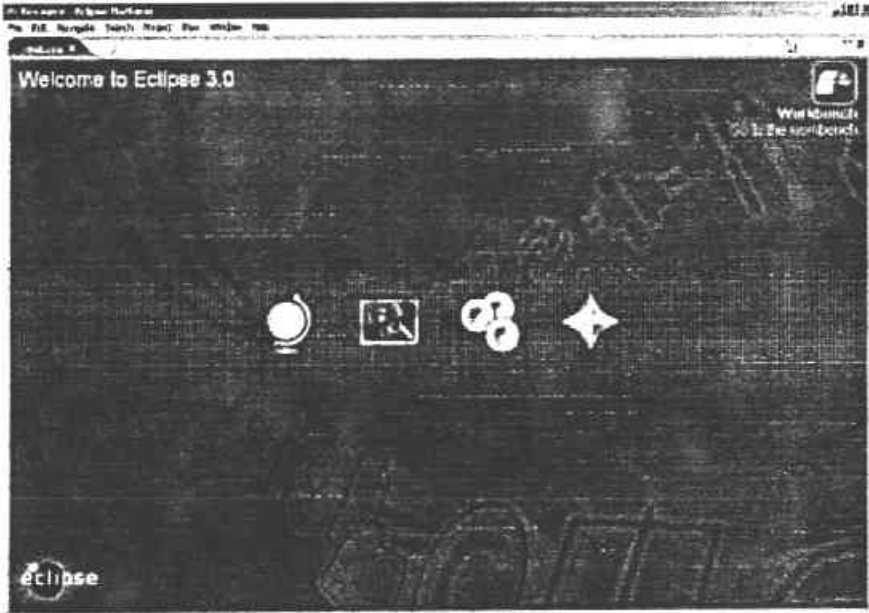


FIGURA 3.2 Ejemplo de una ventana principal de eclipse versión 3.0

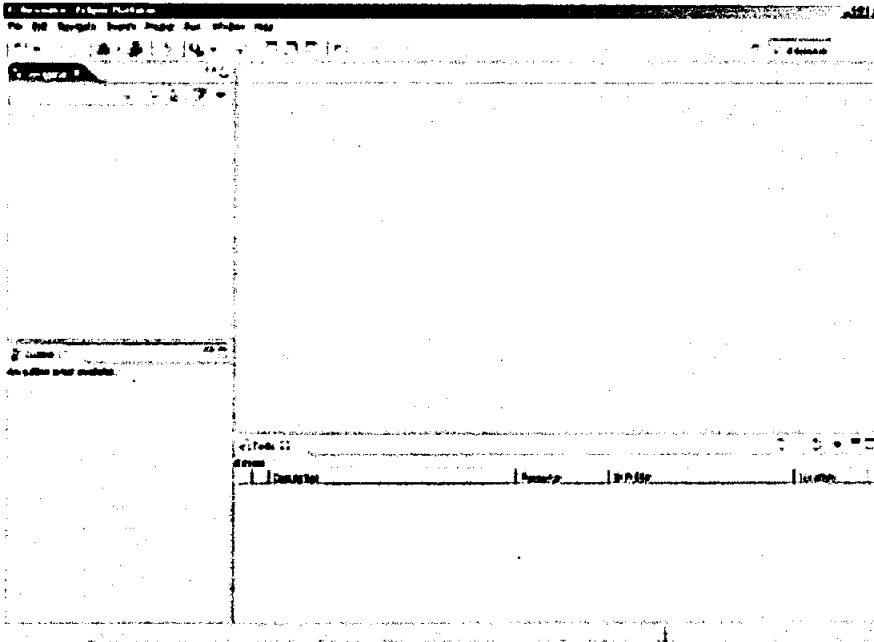


FIGURA 3.3 Ejemplo de ventana de eclipse para empesar a desarrollar un proyecto

Eclipse puede usar varias perspectivas en su ventana principal dependiendo del tipo de desarrollo que se vaya a realizar.

Ahora se abrirá la perspectiva "Java":

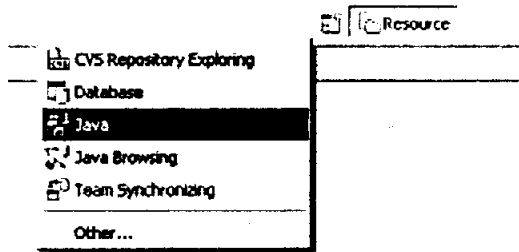


FIGURA 3.4 Ejemplo para abrir Java

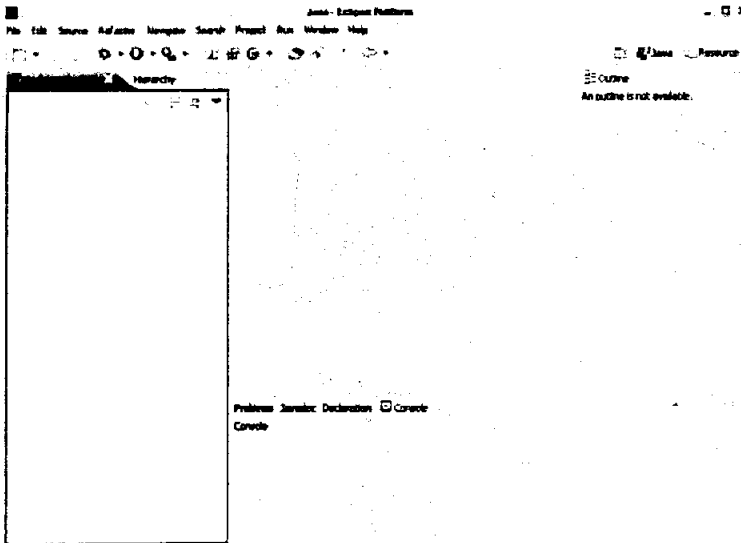


FIGURA 3.5 Ejemplo de una de las perspectivas de la ventana principal de eclipse

Es conveniente que todos los desarrollos se hagan dentro de un proyecto por mucho que el ejercicio sea tan simple. Por lo tanto se creara como ejemplo un proyecto para un desarrollo:

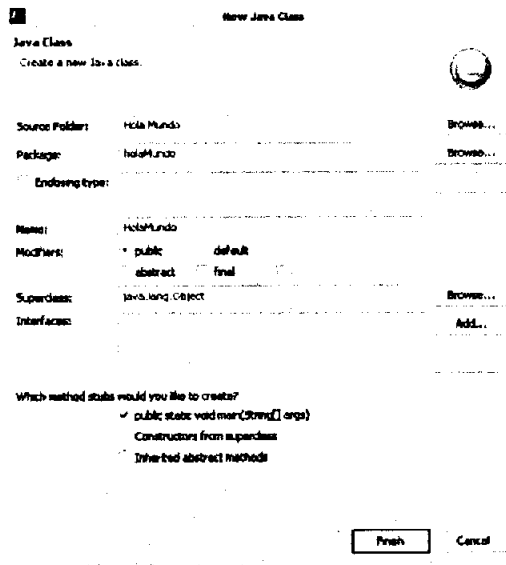


FIGURA 3.6 Ejemplo de cómo se crea un proyecto para un desarrollo

Como se ha podido comprobar al escribir los nombres de la clase y del paquete avisa de ciertas reglas para la nomenclatura de estos. Se puede avanzar que los nombres de las clases siempre empiezan en mayúscula y los paquetes en minúscula.

Al darle a finalizar crea una plantilla que se podrá ver en el centro de la pantalla. Esta área es la que corresponde al editor y aquí es donde se escribirán los programas en Java.

Cuando se ha hecho el habitual hola mundo se escribe lo que queda para que quede como en la figura. Es decir, se introduce `System.out.println("Hola Mundo\n");` dentro del método "main" de la clase.

Ya solo queda ejecutar el programa para ver que funcione. Para hacerlo funcionar se puede utilizar el menú "run" o directamente mediante los iconos de la barra de herramientas.

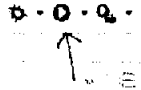


FIGURA 3.7 Ejemplo de cómo ejecutar un programa para que funcione

Al ejecutar el "run" un asistente dará a elegir el tipo de ejecución para el código en Java. Simplemente escoger "Java Application" en el menú con un doble "clic" y creará un "apéndice" de configuración de ejecución para el código en concreto, como se podrá ver:

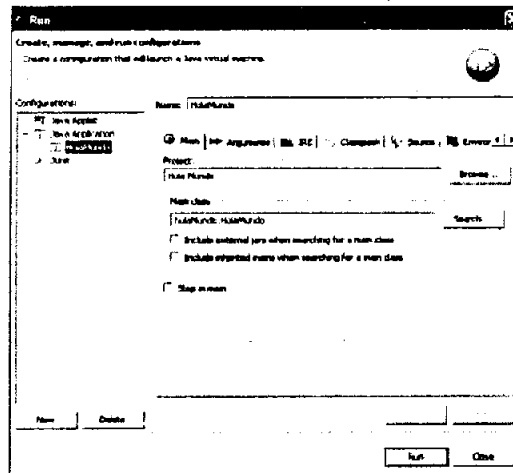


FIGURA 3.8 Ejemplo para elegir el tipo de ejecución para el código en java

En principio y sin más detalles le da click a "Run" y se verán los resultados:

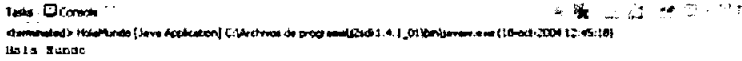


FIGURA 3.9 Ejemplo de un resultado en la pestaña "console" cuando se ejecuta un programa

Como se podrá comprobar, abajo hay una pestaña que pone "console". Esa pestaña muestra una consola que hace las veces de la línea de comandos desde la que se ejecutara el programa de Java en el primer ejemplo. Por tanto es en ese espacio en el que se podrá ver la salida del programa: "Hola Mundo".



INDICE

MODULO 4

Página

Implementación de la Capa de Usuario del Sistema..... 73

1 Introducción a HTML..... 73

 1.1 Estructura básica de un documento HTML..... 73

 1.2 Listas..... 74

 1.3 Tablas..... 75

 1.4 Imágenes..... 76

 1.5 Formulario..... 77

 1.6 Etiquetas <FORM>...</FORM>..... 78

 1.7 Proceso del diseño..... 79

 1.8 Criterios de Usabilidad..... 80

 1.9 Estructura y Organización de las paginas de la interfaz 80

2 Mapa de Navegación..... 82

3 Bosquejos..... 83



1 Introducción a HTML

El *HTML* (*Hyper Text Markup Language*) es un sistema para estructurar documentos. Estos documentos pueden ser mostrados por los visores de páginas Web en Internet, como Netscape, Mosaic o Microsoft Explorer. Por el momento no existe un estándar de *HTML*, aunque existen diferentes revisiones o niveles de estandarización, el 1.0, el 2.0, el 3.0, el 3.2 y el 4.0, lo que produce que algunos visores no "comprendan" en su totalidad el contenido de un documento.

1.1 Estructura básica de un documento HTML

Un documento escrito en HTML contendría básicamente las siguientes directivas:

<HTML>	Indica el inicio del documento.
<HEAD>	Inicio de la cabecera.
<TITLE>	Inicio del título del documento.
</TITLE>	Final del título del documento.
</HEAD>	Final de la cabecera del documento.
<BODY>	Inicio del cuerpo del documento.
</BODY>	Final del cuerpo del documento.
</HTML>	Final del documento.

El documento se hallará situado en algún ordenador al que se pueda acceder a través de Internet. Para indicar la situación del documento en Internet se utiliza la *URL* (*Uniform Resource Locator*). La *URL* es el camino que ha de seguir nuestro visor a través de Internet para acceder a un determinado recurso, bien sea una página Web, un fichero, un grupo de noticias.

La estructura de una URL para una página Web suele ser del tipo *http://dominio/directorio/fichero*. El dominio indica el nombre del ordenador al que accedemos, el directorio es el nombre del directorio de ese ordenador y fichero el nombre del fichero que contiene la página Web escrita en HTML. Por ejemplo:

http://ares.six.udc.es/cine/corunha2.html

Donde

<i>http://</i>	es el indicador de página Web
<i>ares.six.udc.es</i>	es el Dominio (nombre) del ordenador
<i>/cine/</i>	es el Directorio dentro del ordenador
<i>corunha2.html</i>	es el Fichero que contiene la página Web

1.2 Listas

Una lista, o listado, es una enumeración de dos o más elementos y suelen disponerse de tal forma que se facilite la distinción entre ellos.

El lenguaje HTML define los siguientes tipos de listas:

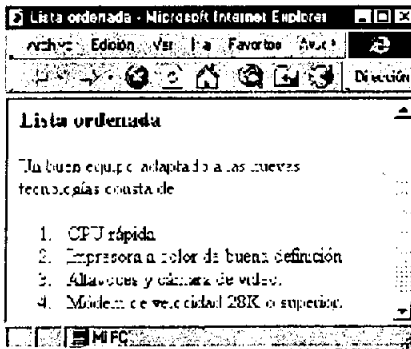


FIGURA 4.1 Ejemplo de lista numerada u ordenada

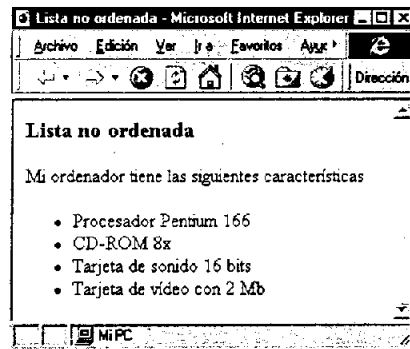


FIGURA 4.2 Ejemplo de lista no ordenada

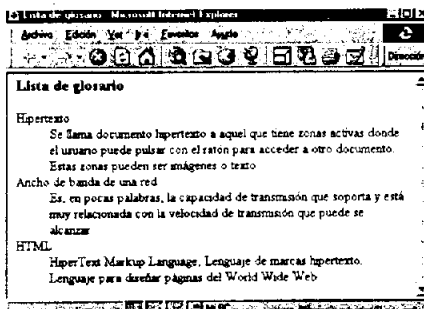


FIGURA 4.3 Ejemplo de lista de glosario

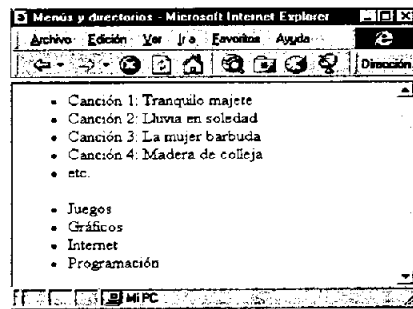


FIGURA 4.4 Ejemplo de lista de menús

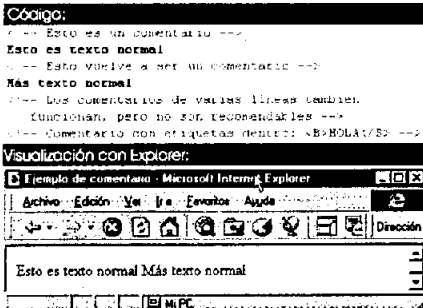


FIGURA 4.5 Ejemplo de lista usada por párrafos cortos

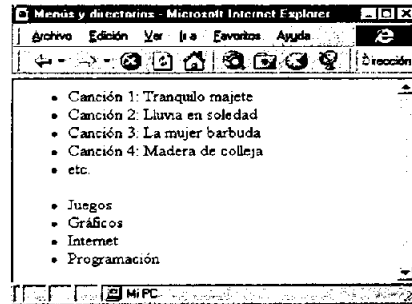


FIGURA 4.6 Ejemplo de lista de directorio

1.3 Tablas

Esta etiqueta consta de instrucción de inicio, <TABLE>, e instrucción de fin, </TABLE>. Entre ambas se tendrá que introducir otras tres etiquetas, que definirán la estructura de la tabla. Estas etiquetas son:

- TR: La etiqueta *Table Row* nos permitirá insertar filas en la tabla.
- TH: Es el primero de los dos tipos de celdas existentes en el lenguaje HTML. El nombre de la etiqueta proviene del inglés *Table Header*, encabezado de tabla.
- TD: Es el segundo tipo de celda de las tablas HTML. El nombre de la etiqueta, *Table Data* (Datos de tabla), se induce a pensar que su función será introducir todos los datos que se quiera en las celdillas definidas de esta forma.

R1C1	R1C2	R1C3
R2C1	R2C2	R2C3

FIGURA 4.7 Ejemplo de una tabla con varias etiquetas



1.4 Imágenes

Sin duda uno de los aspectos más vistosos y atractivos de las páginas web es el gráfico. La introducción en el texto de imágenes puede ayudar a explicar más fácilmente la información y darle un aire mucho más estético.



FIGURA 4.8 Ejemplo de un logotipo de imagen gif



FIGURA 4.9 Ejemplo de un imagen transparente



1.5 Formulario

Los formularios son una de las herramientas de que se dispone a la hora de hacer páginas web interactivas, en el sentido de que permite recopilar información de la persona que ve la página, procesarla y responder a ella, pudiendo de esta forma responder adecuadamente a sus acciones o peticiones.

El proceso comienza con la creación en la página de un formulario de entrada de datos, que va a contener diversos campos diferentes en su funcionalidad y que va a permitir recopilar toda aquella información que se desea de la persona que ve la página. Estos datos, una vez completado el formulario, serán enviados normalmente al servidor para su procesamiento o a uno directamente, mediante correo electrónico.

Los formularios están formados por una serie de controles distintos, cada uno de los cuales está asociado a un tipo concreto de datos o una acción predeterminada: botones de envío y borrado de datos, listas de selección, cajas de entrada de texto, etc.

registro.asp

Nombre:	<input type="text"/>	Núm.	<input type="text"/>
Apellidos:	<input type="text"/>		
F.Nac.:	Dia Mes Año <input type="text" value="1"/> <input type="text" value="1"/> <input type="text" value="1940"/>		
Calle y número:	<input type="text"/>		
Colonia:	<input type="text"/>		
Código Postal:	<input type="text"/>	Ciudad:	<input type="text"/>
Provincia:	<input type="text"/>		
Telefono(s):	<input type="text"/>	ext	<input type="text" value="cel044"/>
Estado Civil:	<input type="text" value="Soltero(s)"/> <input type="checkbox"/> Hombre <input type="checkbox"/> Mujer		
Correo E.:	<input type="text"/>		
Comentarios personales:	<input type="text"/>		
Pulse aquí:	<input type="button" value="Enviar datos"/>	<input type="button" value="Borrar los datos"/>	

FIGURA 4.10 Ejemplo de un formulario en una página web



1.6 Etiquetas <FORM>...</FORM>

Estas etiquetas son las que delimitan el comienzo y el final de un formulario. Si queremos mostrar en una página un elemento aislado de un formulario, aunque sea con otra finalidad diferente a la del envío de los datos que contenga al servidor (por ejemplo para tratar esa información mediante Javascript), es necesario delimitar ese elemento de formulario mediante las etiquetas <FORM> y </FORM>. Si no lo hacemos así, I.Explorer sí mostrará el elemento, pero Netscape no.

La pareja de etiquetas se introduce en el código HTML dentro del BODY de la página, en el lugar en que se quiera que aparezcan los elementos del formulario.

El diseño de aplicaciones Web con metodología de tres capas, se divide en tres componentes principales

- **Cliente**, es la computadora en el escritorio del usuario. Este accede a los servicios a través del navegador o Browser.
- **Servidor**, este es un sistema centralizado que realiza la lógica de negocios y los cálculos, así como la manipulación de los mismos.
- **Base de Datos**, esta capa mantiene los datos guardados en un sistema relacional de base de datos.

La ventaja de esta arquitectura es la independencia de una aplicación cliente, pues el acceso se realiza a través del navegador, la presentación de los datos es con HTML y la comunicación entre cliente y el servidor se realiza a través del protocolo HTTP.

A esta presentación de datos a través de una pantalla se le conoce como interfaz, y tiene las siguientes características

- Es en el cliente donde el usuario interactúa con la información
- La información se despliega en el navegador, por lo que esta es presentada como un documento HTML.
- El HTML es un lenguaje para presentar documentos multimedia y acceder a recursos disponibles en Internet.



La interfaz es el medio por el cual el usuario puede hacer todas sus operaciones a la base de datos, a través de operaciones básicas de internet, por lo que su diseño se pone a:

- Idear, imaginar la página, definir como va ser
- Identificar los elementos con que se constituirá o formará: textos, imágenes, colores, referencias, audio, ligas, las partes de la página.
- Definir el aspecto visual
- Definir el contenido informativo
- Definir los recursos de interacción o de comunicación del usuario
- Definir los nexos o ligas
- Definir su estructura

1.7 Proceso del diseño

El diseño de la interfaz se basa en los procesos y tareas de los usuarios, de lo contrario el sistema sería un fracaso, se debe sensibilizar para únicamente ofrecer lo necesario para el usuario trabajar. Para esto es necesario seguir los siguientes pasos:

1. Entender quien usara el sistema y para que
2. Utilizar un estándar o normatividad de la empresa u organización
3. Bosquejar un diseño
4. Crear un prototipo
5. Evaluarla con los usuarios
6. Construcción



1.8 Criterios de Usabilidad

Algunos aspectos fundamentales que se deben procurar en el diseño son:

- Diseño amigable
- Congruencia con los procesos
- Distribución lógica
- Navegación lógica
- Retroalimentación
- Consistencia
- Omplementación de ayuda

1.9 Estructura y Organización de las paginas de la interfaz

Una parte importante del diseño de la interfaz corresponde al equilibrio entre la estructura y las relaciones entre las páginas y sus procesos. El objetivo es construir una jerarquía de menús y páginas que aparezcan naturales y bien estructuras al usuario.

Los métodos tradicionales son:

- Lineales
- Planos
- Jerárquicos
- Vertical
- Vertical con opciones

Sin embargo en Sistemas Web, la estructura la dan los procesos y generalmente es del tipo vertical con opciones.



Los elementos más utilizados en Sistemas Web son:

- Encabezados
- Títulos
- Imágenes
- Frames
- Listas
- Formularios
- Tablas
- Botones
- Cuadros de diálogo



2 Mapa de Navegación

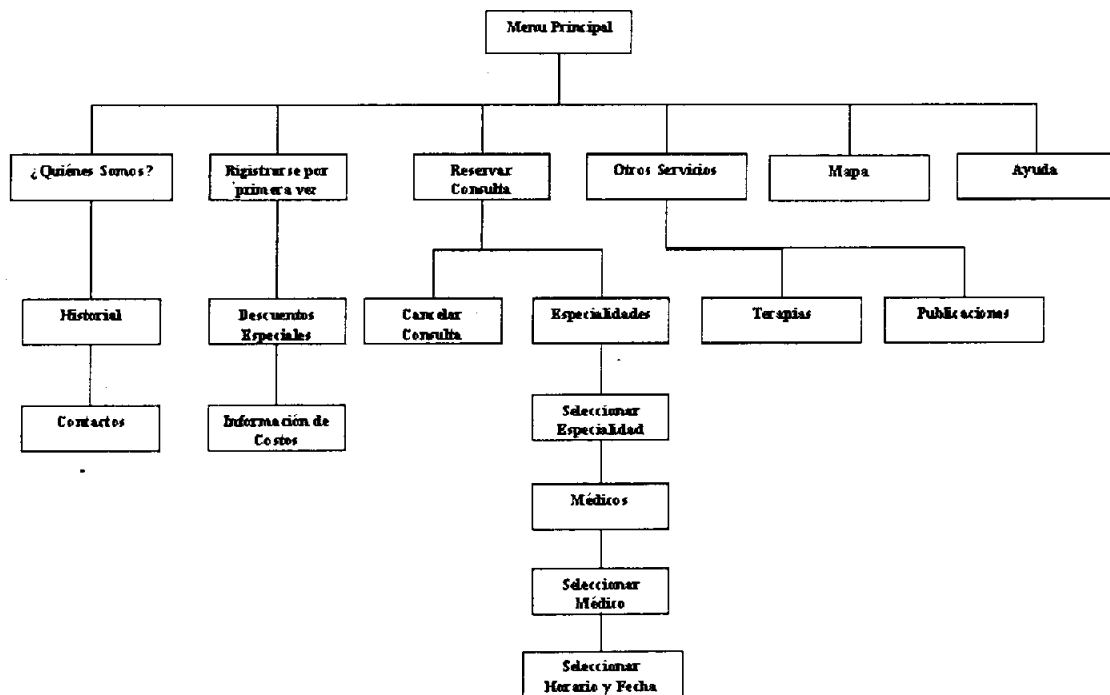


FIGURA 4.11 Ejemplo de mapa de navegación de una clínica



3 Bosquejos

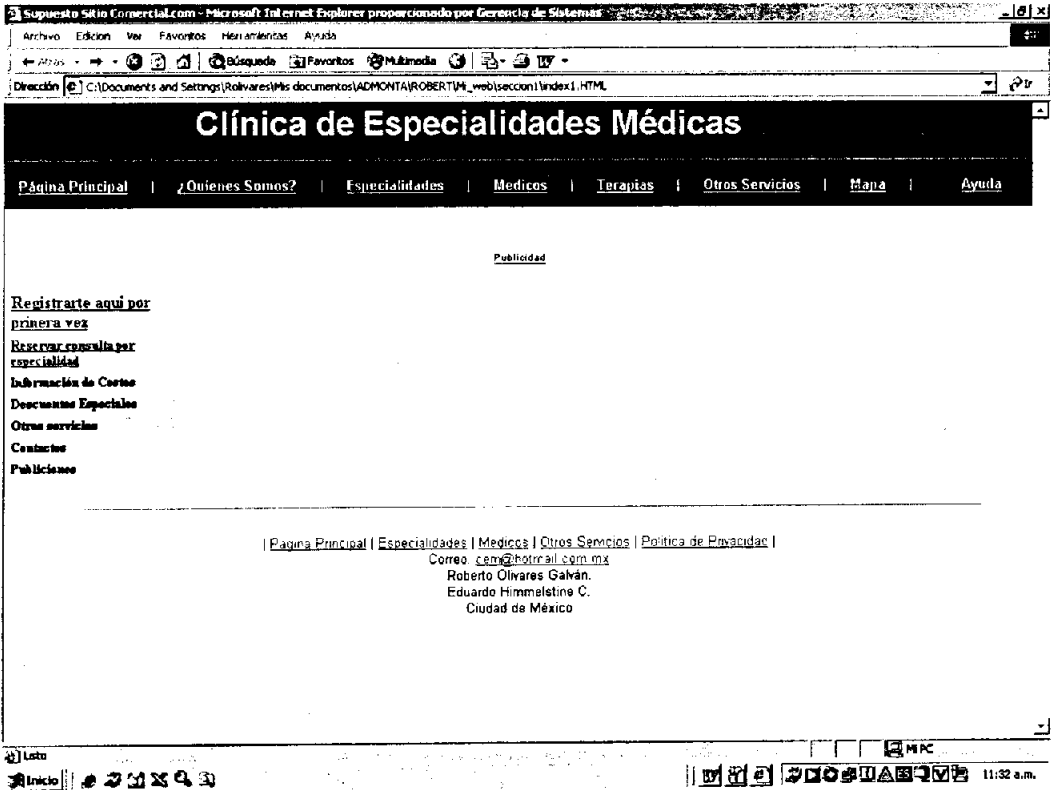


FIGURA 4.12 Página principal de una clínica



Supuesto Sitio Comercial.com - Microsoft Internet Explorer proporcionado por Gerencia de Sistemas

Archivo Edición Ver Favoritos Herramientas Ayuda

Búsqueda Favoritos Multimedia

Dirección C:\Documents and Settings\Robi\ivos\Mis documentos\ADMONTA\ROBERT\M_web\seccion1\index.html

Clínica de Especialidades Médicas

Reservata now

Nombre:			Núm.	
Apellidos				
F.Nac.:	Día	Mes	Año	
	1	1	1940	
Calle y número:				
Colonia:				
Código Postal:	Ciudad			
Provincia:				
Telefono(s):		ext.	cel.044	
Estado Civil:	Soltero(s) <input type="radio"/> Hombre <input type="radio"/> Mujer			
Correo E.				
Comentarios personales				

Listo Inicio

11:35 a.m.

FIGURA 4.13 Página para registrar a un paciente

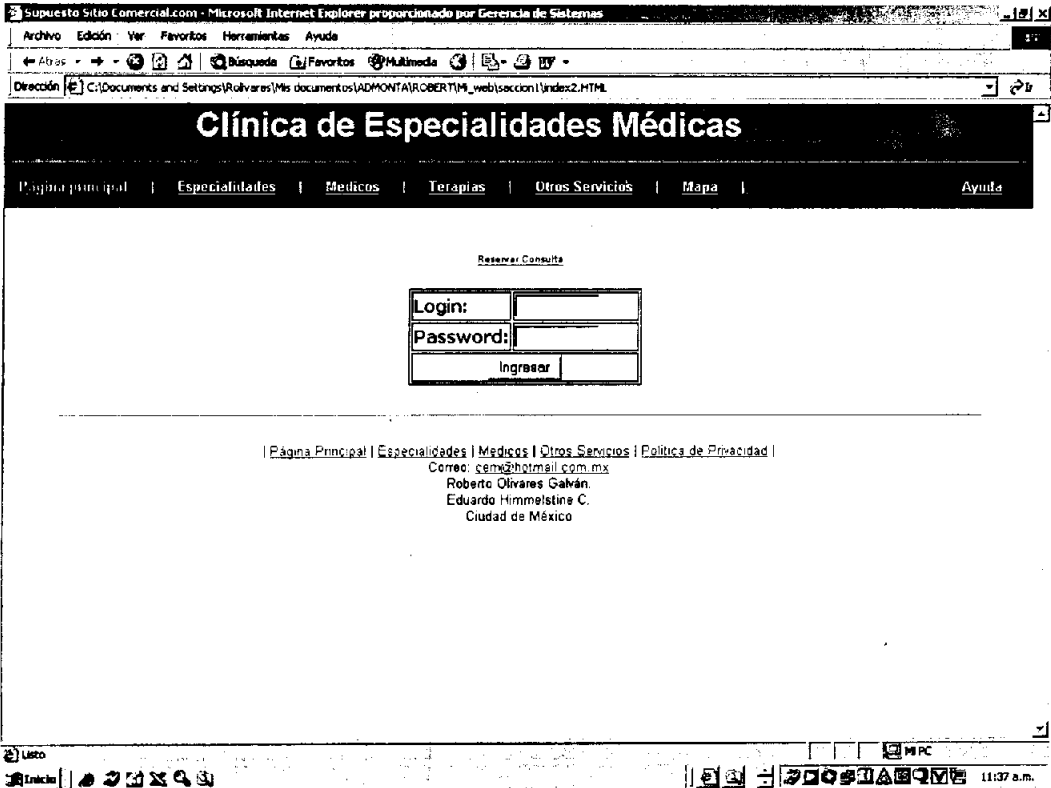


FIGURA 4.14 Página para reservar una consulta



INDICE	MÓDULO 5	Página
1 Capa de Datos.....		87
Conceptos Generales de Base de Datos.....		87
2 Base de Datos.....		87
2.1 Modelo de Datos.....		87
2.2 Modelo Basado en Objetos.....		87
2.3 Modelos lógicos basados en registros.....		88
2.4 Modelos físicos de datos		89
3 SGBD		90
3.1 Arquitectura de los Sistemas de Bases de Datos		90
3.2 Independencia Lógica y Física de Datos.....		91
4 Arquitectura de los SGBD.....		91
5 Componentes de un SGBD		92
6 Generalidades de un SGBD.....		92
7 Introducción a SQL.....		93
7.1 Consultar datos de una tabla.....		94
7.2 Consultar datos que cumplan ciertas condiciones.....		95
7.3 Mostrar datos de varias tablas		95
7.4 Operaciones elementales.....		96
7.5 Buscar un cierto texto.....		96
7.6 Modificaciones repetitivas.....		96
7.7 Ejemplos.....		97
8 DB2.....		98



1 Capa de datos

Es la base de una aplicación de Bases de Datos basada en Web, engloba todos los componentes del sistema que almacenan los datos. Está formada básicamente por el Sistema Gestor de Bases de Datos (Data Base Management System) y un paquete encargado de aislar el acceso a los datos.

Conceptos Generales de Bases de Datos

2 Base de Datos

La Base de Datos es el Conjunto de datos almacenados en una estructura física específica, entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización.

2.1 Modelo de Datos

Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

Los modelos de datos se dividen en tres grupos:

2.2 Modelo basado en objetos

Se usan para describir datos en los niveles conceptual y de visión,

Modelo Entidad-Relación

Es Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de *entidades*, que son objetos que existen y que se distinguen de otros por sus características.

Las entidades pueden ser de dos tipos:

Tangibles:

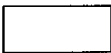



Son todos aquellos objetos físicos que podemos ver, tocar o sentir.

Intangibles:

Todos aquellos eventos u objetos conceptuales que no podemos ver, aun sabiendo que existen, por ejemplo: la entidad materia, sabemos que existe, sin embargo, no la podemos visualizar o tocar.



Las características de las entidades en base de datos se llaman *atributos*, por ejemplo el nombre, dirección teléfono, grado, grupo, etc. son atributos de la entidad alumno; clave, número de seguro social, departamento, etc., son atributos de la entidad empleado. A su vez una entidad se puede asociar o relacionar con más entidades a través de *relaciones*.

Símbolo	Representa
	Entidad
	Relación
	Atributos
	Ligas

2.3 Modelos lógicos basados en registros

Se utilizan para describir datos en los niveles conceptual y físico.

En estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores



Los tres modelos de datos más ampliamente aceptados son:

- **Modelo relacional**

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a los cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla;

- **Modelo de red**

Este modelo representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros

- **Modelo jerárquico**

Es similar al modelo de red en cuanto a las relaciones y datos, ya que estos se representan por medio de registros y sus ligas.

2.4 Modelos físicos de datos

Estos modelos se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos.

Existen dos Clasificaciones de este tipo que son:

- **Modelo unificador**
- **Memoria de elementos**

3 SGBD

El sistema de gestión de la base de datos (SGBD) es un conjunto de aplicaciones que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma.

3.1 Arquitectura de los Sistemas de Bases de Datos

Las bases de datos respetan la arquitectura de tres niveles definida, para cualquier tipo de base de datos, por el grupo ANSI/SPARC. En esta arquitectura la base de datos se divide en los niveles externo, conceptual e interno

- **Nivel interno:**

Es el nivel más bajo de abstracción, y define cómo se almacenan los datos en el soporte físico, así como los métodos de acceso.

- **Nivel conceptual:**

Es el nivel medio de abstracción. Se trata de la representación de los datos realizada por la organización, que recoge las vistas parciales de los requerimientos de los diferentes usuarios y las aplicaciones posibles.

- **Nivel externo:**

Es el nivel de mayor abstracción. A este nivel corresponden las diferentes vistas parciales que tienen de la base de datos los diferentes usuarios.

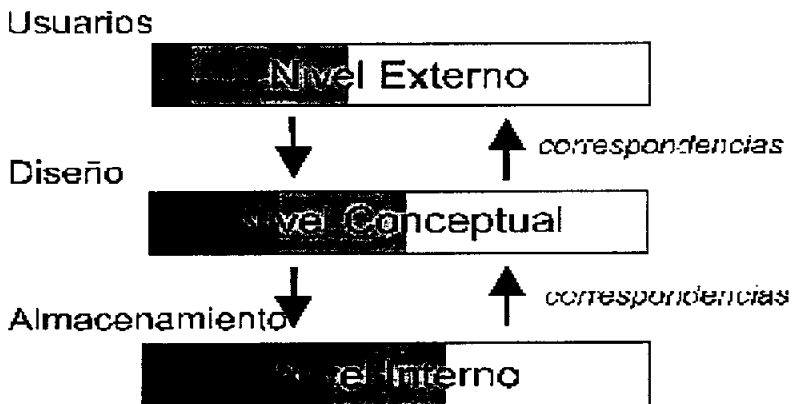


FIGURA 5.1 Arquitectura de un sistema de bases de datos



3.2 Independencia Lógica y Física de Datos

- LDD
- LMD

En ocasiones puede encontrarse el nivel conceptual dividido en dos niveles, conceptual y lógico.

El primero de ellos corresponde a la visión del sistema global desde un punto de vista organizativo independiente, no informático.

El segundo correspondería a la visión de la base de datos expresada en términos del sistema que se va a implantar con medios informáticos.

El modelo de arquitectura propuesto permite establecer el principio de independencia de los datos. Esta independencia puede ser lógica y física.

4 Arquitectura de los SGBD

Hay tres características importantes inherentes a los sistemas de bases de datos: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos.

Esta arquitectura divide la base de datos en tres niveles:

- **El nivel externo:**

Es la representación de los datos tal y como los ve el usuario.

- **El nivel lógico:**

Es una representación abstracta (no física como en el nivel interno) del contenido total de la base de datos.

- **El nivel interno:**

Es el más cercano a la máquina. Es una representación a bajo nivel de la BD en la que se define la forma en la que los datos se almacenan físicamente en la máquina.

5 Componentes de un SGBD

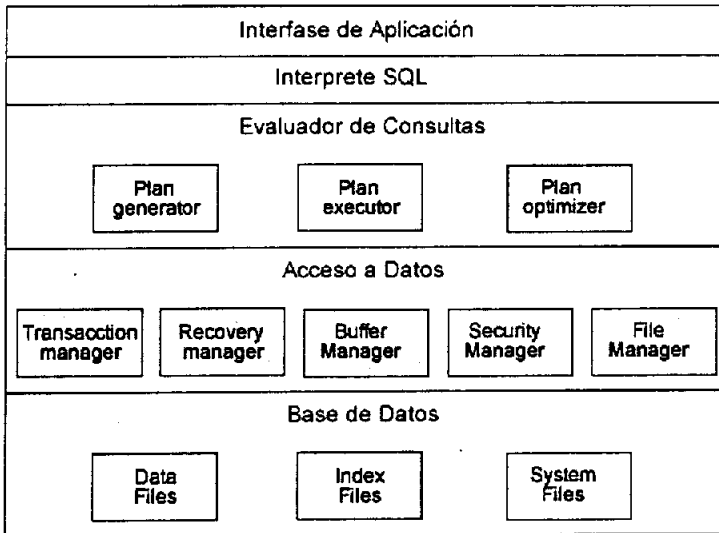


FIGURA 5.2 Componentes de un SGBD

Un SGBD tiene varios módulos, cada uno de los cuales realiza una función específica. El sistema operativo proporciona servicios básicos al SGBD, que es construido sobre él.

6 Generalidades de un SGBD

Modelo Lógico

- Relacional
- De Red
- Jerárquico
- Orientado a Objetos



Funciones

- Almacenar datos
- Descripciones de los datos
- Aseguramiento de la Integridad
- Manejo de Concurrencia
- Manejo de Seguridad de los datos
- Recuperación de Daños a los datos

7 Introducción a SQL

El SQL, son las siglas *Standard Query Language*, que es el language con que trabajan las bases de datos como SQL server, Mysql, Access, etc. de ahí su importancia que se estudie, conociendo el SQL se podrá interactuar mejor con las bases de datos.

Con las sentencias SQL se podrá trabajar en la base de datos dándole ordenes como las de insertar, modificar, borrar o solo consultarla.

Parece razonable suponer que quien quiere manejar SQL tiene que tener algunos conocimientos elementales de bases de datos, y que entiende conceptos como tabla, campo, registro, etc. Aun así, las ideas básicas, prescindiendo de todo tipo de formalismos (incluso demasiado) son:

- Una tabla es un conjunto de datos con una cierta homogeneidad, por ejemplo un fichero que contenga todos los datos de nuestros amigos (nombre, domicilio, teléfono, fecha de nacimiento, etc).
- Un registro sería cada uno de los bloques de datos que hemos guardado: la ficha de uno de nuestros amigos (con su nombre, su domicilio, etc).
- Un campo es cada uno de los datos que componen un registro: el nombre sería un campo, el domicilio sería otro campo, la fecha de nacimiento sería otro campo, y así sucesivamente.



La forma habitual de representar una tabla es esta:

Nombre	Domicilio	Teléfono	Fecha nacimiento
Juan Pérez	C/ La Pera, 1	111-11-11	1/1/1950
José López	C/ La Manzana, 2	222-22-22	20/10/1965
Andrés Martínez	C/ La Naranja, 3	333-33-33	15/8/1980

FIGURA 5.3 Ejemplo de una tabla en SQL

En esta representación:

- El conjunto de todos los datos es la tabla.
- Cada fila representa un registro de la tabla (una ficha).
- Cada columna son los distintos valores que ha tomado cada uno de los campos.

7.1 Consultar datos de una tabla.

La orden básica para crear una consulta en lenguaje SQL es la orden SELECT. Esta se emplea así:
 SELECT campo1, campo2, ... campon FROM tabla

Un primer ejemplo muestra ciertos campos de una tabla de clientes podría ser:
 SELECT nombre, apellidos, domicilio, localidad, provincia FROM clientes

Tanto las órdenes como los nombres de los campos y/o las tablas se pueden escribir en mayúsculas o minúsculas, pero en este se empleará mayúsculas para las órdenes de SQL y minúsculas para los nombres de campos y/o tablas, a fin de que se pueden distinguir con mayor facilidad.

Para mostrar todos los campos de una tabla, no hace falta enumerarlos uno a uno, sino que se puede emplear el asterisco (*):

SELECT * FROM clientes



7.2 Consultar datos que cumplan ciertas condiciones.

Si se quiere visualizar los datos que cumplan una cierta condición, se emplea la sentencia WHERE:

```
SELECT campo1, campo2, ... campon FROM tabla WHERE condición
```

Un ejemplo para mostrar datos de clientes de Madrid podría ser:

```
SELECT nombre, apellidos, domicilio, localidad, provincia FROM clientes WHERE localidad="Madrid"
```

Los operadores de comparación son los habituales en informática: =, <, >, <= (menor o igual), >= (mayor o igual), <> (distinto de).

Se pueden encadenar varias condiciones con AND (y), OR (ó), NOT (no):

```
SELECT nombre, apellidos, FROM clientes WHERE localidad="Madrid" AND totalCompras > 500000
```

Para saber si un campo se ha dejado en blanco, no se deberá mirar si es = 0, sino usar la condición IS NULL.

Por ejemplo, para saber que clientes no tienen el teléfono sería:

```
SELECT nombre FROM clientes WHERE telefono IS NULL
```

Si se quieren los datos ordenados según algún campo, se usa la cláusula ORDER BY:

```
SELECT nombre, apellidos, FROM clientes ORDER BY totalCompras
```

En caso de que se quieran ordenar de mayor a menor, se añade DESC después de ORDER BY:

```
SELECT nombre, apellidos, FROM clientes ORDER BY totalCompras DESC
```

7.3 Mostrar datos de varias tablas.

En el caso más sencillo, basta con indicar los nombres de las tablas después de FROM:

```
SELECT campo1, campo2, ... campon FROM tabla1, tabla2, ... tablan
```

Si coinciden nombres de campo en varias tablas, se escribe el nombre de la tabla y un punto antes del nombre de la tabla, para evitar la ambigüedad:

```
SELECT cliente.nombre, vendedores.nombre FROM clientes, agentes
```

Es muy frecuente que se tomen datos de varias tablas relacionadas, y esto en muchas bases puede obligarnos a indicar también una condición que refleje esa relación:

```
SELECT cliente.nombre, vendedores.nombre FROM clientes, agentes WHERE agentes.codigo = clientes.codigoVendedor
```

O en un caso que emplee 3 tablas

```
SELECT articulos.titulo, revistas.titulo, nombre FROM articulos, revistas, autores WHERE articulos.codigoRev = revistas.codigo AND articulos.codigoAutor = autores.codigo
```



7.4 Operaciones elementales.

También se pueden realizar operaciones elementales como la suma (SUM), promedio (AVERAGE), máximo (MAX) y mínimo (MIN) de datos numéricos:

```
SELECT SUM(totalCompras) FROM clientes
```

Se puede realizar más de una operación en la misma consulta:

```
SELECT MAX(importe), SUM(importe), AVERAGE(importe) FROM clientes
```

```
SELECT NOMBRE, SUM (totalCompras) FROM clientes
```

7.5 Buscar un cierto texto.

Es muy frecuente buscar los registros que contengan un cierto texto. En este caso, se debe usar LIKE en vez de "=" y emplear el carácter de tanto por ciento (%) como comodín:

```
SELECT nombre, apellidos FROM clientes WHERE apellidos LIKE "Fer%"
```

La consulta anterior mostraría los cliente cuyo apellido comience por "Fer". Si queremos buscar aquellos en cuya dirección aparece la palabra "Madrid", usaríamos:

```
SELECT nombre, apellidos FROM clientes WHERE localidades LIKE "%Madrid%"
```

En algunos sistemas de gestión de bases de datos, puede emplearse el asterisco (*) en vez del símbolo de tanto por ciento (%).

7.6 Modificaciones repetitivas.

Existen otras órdenes distintas de SELECT. Una muy útil es la que nos permite modificar valores de la base de datos: UPDATE, que se utiliza así:

```
UPDATE tabla SET campo=valor
```

Por ejemplo, para que el campo "Iva" de todos los clientes tenga el valor 16% podría ser:

```
UPDATE clientes SET iva=0.16
```

Si queremos que se modifiquen sólo los que cumplen cierta condición, emplearemos WHERE. Por ejemplo, para que aquellos clientes cuya localidad sea "Alicante" pasen a tener como localidad "Alicante (España)" haríamos:

```
UPDATE clientes SET localidad="Alicante (España)" WHERE localidad="Alicante"
```



7.7 Ejemplos.

Mostrar todos los datos de los clientes:

```
SELECT * FROM clientes
```

Nombre y domicilio de todos los clientes:

```
SELECT nombre, domicilio FROM clientes
```

Nombre y domicilio de los clientes de Alicante:

```
SELECT nombre, domicilio FROM clientes WHERE localidad = "Alicante"
```

Nombre y domicilio de los clientes que no sean de Alicante y que hayan comprado productos por un importe total superior o igual a 500.000 pesetas:

```
SELECT nombre, domicilio FROM clientes WHERE localidad <> "Alicante" AND totalCompras >= 500000
```

Nombre y domicilio de los clientes que se apellidan López (suponiendo que el apellido se haya guardado también dentro del campo nombre):

```
SELECT nombre, domicilio FROM clientes WHERE nombre LIKE "%López%"
```

Nombre de los clientes de Madrid, ordenados de mayor a menos volumen de compras:

```
SELECT nombre FROM clientes WHERE localidad = "Madrid" ORDER BY totalCompras DESC
```

Nombre del cliente de Madrid que mayor volumen de compras ha realizado:

```
SELECT nombre FROM clientes WHERE localidad = "Madrid" AND totalCompras = MAX(totalcompras)
```

Mayor importe de compras, menor y promedio de todos ellos:

```
SELECT MAX(totalcompras), MIN(totalcompras), AVERAGE(totalcompras) FROM clientes
```

Clientes para los que no se ha anotado el vendedor:

```
SELECT nombre FROM clientes WHERE codVendedor IS NULL
```

Escribir "(No indicado)" como número de Fax de aquellos clientes cuyo Fax no se haya teclado:

```
UPDATE clientes SET fax="(No indicado)" WHERE fax IS NULL
```

Multiplicar por 0.8 el valor actual de todos los productos

```
UPDATE productos SET valorActual = valorActual * 0.8
```



8 DB2

Las ventajas de DB2 son simples:

Tiene una Potencia real a menor costo de cualquier base de datos, con DB2 se puede acceder, administrar y analizar todo tipo de información de una empresa.

El DB2 puede ejecutarse en varias plataformas

- En Windows
- En LINUX
- En UNIX

Desde el punto de vista de la programación hay buenas y malas noticias. Las buenas noticias: el SQL de DB2 es muy potente. Las malas noticias: los procedimientos almacenados de DB2 deben programarse en lenguajes externos: C, Java, VisualBasic, etc. ¿Por qué esto es malo? No sólo porque obliga a utilizar un lenguaje adicional, sino principalmente porque nos hace depender de otro compilador, que casi siempre está ligado a la plataforma o sistema operativo.



CONCLUSIONES



A continuación se presentan las conclusiones a que se llegó con el trabajo desarrollado, y los logros obtenidos.

El presente trabajo ha contribuido al mejoramiento del manejo de la información en una Clínica de Especialidades Médicas.

Se Diseñó un sistema que permita operar y administrar eficientemente las labores cotidianas que se realizan en una clínica, tales como: reservaciones, expedientes, horarios de atención, próximas citas etc.

Las herramientas de manejo de sistemas y de bases de datos resultaron muy adecuadas para todos los propósitos planteados para esta tesis.

De manera que CEM es el proyecto para mejorar la operación de las clínicas públicas y privadas, es un proyecto sólido que puede aplicarse en cualquier sector médico con amplias posibilidades de éxito, ya que se hizo con las herramientas más generales posibles.

Se han logrado los siguientes beneficios con el desarrollo del sistema CEM:

Operación total a través del web permitiendo que el paciente pueda reservar consultas por tipos de Especialidad, Médicos, Fechas, y Horarios.

Reducir Gastos y Costos de Operación.

Control de Información de los Pacientes registrados por Especialidad. Llevar un control de información de los Médicos por especialidad. Administrar los gastos que se generen por concepto de consultas, cirugías, partos, vacunas, etc. Como también controlar pagos realizados en efectivo y/o Tarjetas de crédito. Llevar un control de descuentos a pacientes por especialidad según sea el caso, y la obtención de reportes en los momentos que se necesitan para la toma de decisiones.

En términos generales, las nuevas tecnologías que vayan apareciendo y dejando obsoletas a las actuales, serán los motivos para mantenerse en el camino de la actualización y el mejoramiento de los sistemas actuales.

Los sistemas de información seguirán siendo el pilar para la operación y la toma de decisiones en las pequeñas, medianas, y grandes empresas y ahora con las nuevas tecnologías, como las grandes redes de datos harán que las empresas cambien sus modelos de operación, y aquellas que no incursionen en estos ámbitos, simplemente verán pasar el progreso de sus competidores que si aprovechan la nueva tecnología. Gracias a toda esta nueva tecnología nosotros los ingenieros en computación tenemos porque seguir estudiando y aprendiendo cada día más.



GLOSARIO



ASCII: American Standard Code of information Interchange: Código normalizado estadounidense para el intercambio de la información. Código que permite definir caracteres alfanuméricos; se usa para lograr compatibilidad entre diversos procesadores de texto.

BASE DE DATOS: Conjunto de datos organizados de modo tal que resulte fácil acceder a ellos, gestionarlos y actualizarlos.

CEM: Clínica de Especialidades Médicas.

CGI: Common Gateway Interface: Es una norma para establecer comunicación entre un servidor web y un programa, de tal modo que este último pueda interactuar con Internet.

DBMS: Database Management System Sistema: Manejador de Base de Datos, el cual es un software usado para gestionar datos de una o más aplicaciones.

DNS: Domain Name Server: Es el encargado de convertir la dirección de IP en un nombre de dominio generalmente fácil de recordar y viceversa.

FRAMEWORK: Es la extensión de un lenguaje mediante una o más jerarquías de clases que implementan una funcionalidad y que (opcionalmente) pueden ser extendidas.

GNOME: Network Object Model Environment: Es un entorno de escritorio de código fuente abierto construido con componentes que se ejecutan totalmente a las directrices del desarrollo de aplicaciones de código fuente abierto.

HTML: Hyper Text Markup Language: Es un sistema para estructurar documentos. Estos documentos pueden ser mostrados por los visores de páginas Web en Internet, como Netscape, Mosaic o Microsoft Explorer.

HTTP: Hypertext Transfer Protocol: Es la base de toda comunicación desarrollada en la web.

JPEG: Joint Photographic Experts Group: Nombre del comité que diseño un estándar para la compresión del usuario en un sistema online.

KDE: K Desktop Environment: Entorno de escritorio K, donde K no significa nada en particular.

MVC: Es un patrón que define la organización independiente del Model (Objetos de Negocio), la View (interfaz con el usuario u otro sistema) y el Controller (controlador del workflow de la aplicación).

PASSWORD: Contraseña.

SGBD: Sistema de Gestión de Base de Datos: Es un conjunto de aplicaciones que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma.



SQL: Structured Query Language: Lenguaje de programación que se utiliza para recuperar y actualizar la información contenida en una base de datos.

SSH: Secure Shell: Es un programa que permite realizar conexiones entre máquinas a través de una red abierta de forma segura, así como ejecutar programas en una máquina remota y copiar archivos de una máquina a otra.

STRING: Cadena: Una palabra o número, porción de éste, o incluso un carácter se denomina String. Este puede ser de 1 a 255 caracteres. Cada carácter es de un byte.

STRUTS: Es un framework que implementa el patrón de arquitectura MVC en Java.

TCP: Protocolo de control de Transmisión: Es el que divide el mensaje original en datagramas de menor tamaño (múltiplo de 32 bits), y por lo tanto, mucho más manejables.

TFP: Protocolo de transferencia de archivos.

UDA: Universal Data Access: Es una estrategia que se basa en una extensión de OLE, denominada **OLE-DB**. Mediante OLE-DB, se puede acceder a datos de formatos heterogéneos sin que eso suponga un problema de modificación de las formas de acceso.

UML: Unified modeling lenguaje: Es una metodología para especificar, construir un sistema de software orientado a objetos.

URL: Uniform Resource Locator: Es el camino que ha de seguir nuestro visor a través de Internet para acceder a un determinado recurso, bien sea una página Web, un fichero, un grupo de noticias.



BIBLIOGRAFIA



- Desarrollo Web con PHP 5 y MYSQL (Guías Practicas) de Gutiérrez Gallardo, Juan Diego. Editorial: Anaya Multimedia - Anaya Interactiva, Primera edición. Ed. 2004.
- Fundamentos de Dreamweaver MX de West, Ray. Editorial: McGraw-Hill. Primera edición. Ed. 2004.
- Fundamentos de SQL de Houlette, Forrest. Editorial: McGraw-Hill. Segunda edición. Ed. 2003.
- Guía de Aprendizaje MYSQL de Ullman, Larry. Editorial: Pearson Educación. Primera edición. Ed. 2003.
- Herramientas de Programación para el Shell de UNIX de Medinets, David. Editorial: McGraw-Hill. Segunda edición. Ed. 2001.
- HTML la guía Completa de Kennedy, Bill y Musciano. Editorial: McGraw-Hill. Segunda edición. Ed. 1999.
- JAVA 2 de Martín Tordio, Miguel Angel. Editorial: Anaya Multimedia - Anaya Interactiva. Primera edición. Ed. 2000.
- Microsoft SQL Server 2000 de Delgado, Alberto. Editorial: Pearson Educación. Primera edición. Ed. 2001.
- Profesional Apache Tomcat 5 de VV. AA. Editorial: Anaya Multimedia - Anaya Interactiva. Primera edición, Ed. 2004.
- UML Gota a Gota de Fowler, Martín y Scott, Kendall Editorial: Alhambra Mexicana S.A. Primera edición. Ed. 1999.
- UNIX (Edición Especial) de David Bandel y Robert Napier Editorial: Prentice Hall. Sexta edición. Ed. 2001.
- UNIX a Fondo de AMSTRONG, James C. Editorial: Anaya Multimedia - Anaya Interactiva. Primera edición. Ed. 2000.
- UNIX de VV.AA y Moritsugu. Editorial: Pearson Educación. Primera edición, Ed. 2000.
- Tipos de Información que maneja la Web:
<http://edec.unican.es/libro/WWW.htm>
- Introducción a JDK:
<http://es.geocities.com/ticiajosep/Starter101/index.html>
<http://es.geocities.com/ticiajosep/Starter101/internet-browsers-plugins.html>
- Instalación de Tomcat:
<http://manuales.dgsca.unam.mx/jsp/jakarta.html>