



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**IMPLEMENTACIÓN DE CASO PRÁCTICO BAJO
SOFTWARE LIBRE**

Trabajo escrito en la modalidad de
“Seminarios y Cursos de Actualización y
Capacitación Profesional”, que para obtener
el título de ingeniero en computación

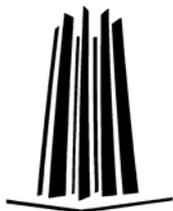
PRESENTA:

WENDY ARIDAY MAYEN CONTRERAS

ASESOR:

ING. RODOLFO VÁZQUEZ MORALES

MÉXICO, 2007





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.



FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN

JEFATURA DE CARRERA
INGENIERÍA EN COMPUTACIÓN

OFICIO: FESAR/JACO/0452/07.

ASUNTO: Designación de Revisores.

MAT. LUIS RAMÍREZ FLORES

ING. ROBERTO BLANCO BAUTISTA

ING. RODOLFO VÁZQUEZ MORALES

ING. BLANCA ESTELA CRUZ LUÉVANO

ING. MARÍA GABRIELA GONZÁLEZ HERNÁNDEZ

Informamos a ustedes de la autorización que se le concede a la alumna **WENDY ARIDAY MAYEN CONTRERAS**, para que pueda desarrollar el trabajo de titulación: **"IMPLEMENTACIÓN DE CASO PRÁCTICO BAJO SOFTWARE LIBRE"**, bajo la modalidad de **"Seminarios y Cursos de Actualización y Capacitación Profesional"** dirigido por el **Ing. Rodolfo Vázquez Morales**, solicitando a ustedes sean tan amables de revisar el avance del mismo y hacer las observaciones que consideren pertinentes, o en su caso, indicar a la alumna si dicha revisión se hará a la conclusión del trabajo de tesis.

Sin otro particular, le es grato enviarles un cordial saludo.

A TENTASMENTE
"POR MI RAZA HABLE EL ESPIRITU"
San Juan de Aragón, Estado de México, 12 de septiembre de 2007.
EL JEFE DE CARRERA

M. EN C. MARCELO PÉREZ MEDEL

MPM*mrh.

Agradecimientos

Haber concluido mi carrera Universitaria representa uno de los más grandes logros de mi vida, proceso en el cual ha participado mucha gente de singular importancia para mí, y a la que, sin lugar a duda tengo que agradecerle infinitamente su apoyo incondicional.

Sin embargo, tengo que destacar mi agradecimiento a una persona que admiro, amo y respeto, porque en base a su ejemplo, a su dedicación, y a su esfuerzo, ha hecho de mí una persona con grandes valores y dispuesta a ser cada día un mejor ser humano; por esto y por una infinidad de cosas más no puedo dejar de agradecer a la mejor mamá del mundo todo lo que ha hecho por y para mí. Gracias Lucía Contreras Iturbe. Este trabajo, es trabajo es dedicado a ti. Te amo.

No puedo dejar de mencionar al resto de mi familia a quienes sin escatimar esfuerzo alguno han sacrificado gran parte de su vida para formarme y educarme: Damaris, Berenice, Greys, Jesús, y sobre todo a Karen, quien ha sido y será mi mejor amiga y quien en todo momento me ha apoyado incondicionalmente.

También quisiera agradecer a la UNAM por abrirme sus puertas y dejarme ser parte de ella, porque para mí es un orgullo ser Universitario y la camiseta la llevo puesta día a día, y no sólo es agradecimiento a la institución, sino a todo lo que conlleva a ella: compañeros de clase, profesores, instalaciones y sobre todo a mi Asesor, Rodolfo Vázquez Morales por haberme guiado en este proceso por su apoyo y su confianza.

Hay una infinidad de personas a las que me gustaría agradecer, pero sería imposible debido al poco espacio que tengo, sin embargo, aunque tu nombre no aparezca aquí, en el momento en que lees estas líneas sabrás que te estoy eternamente agradecido por dedicarle unos minutos de tu tiempo a la lectura de mi trabajo.

A todos ustedes, Muchas gracias.

Introducción	4
Diplomado “Desarrollo de Sistemas bajo Software Libre”	6
1.1 Modulo 1: Sistema Operativo Linux	6
Características	6
Distribuciones	7
Componentes del Sistema Operativo	8
Sistema de Archivos	9
Atributos de los archivos	9
Comandos y Utilerías básicas	10
Ayuda del sistema	11
Procesos	11
1.2 Modulo 2: Instalación y Administración de Linux	12
Perfil y actividades del administrador	12
Instalación	13
Dar de alta y/o baja el sistema	14
Mantenimiento de claves de usuario	14
Instalación y mantenimiento de dispositivos	15
Interfaces gráficas en Linux	15
1.3 Modulo 3: Editores para la creación de Páginas Web	16
Estructura básica de un documento HTML	16
Elementos Avanzados	17
1.4 Modulo 4: Administración de servidores de WWW con Linux	18
Servidor WWW	18
Funcionamiento de HTTP	18
Instalación del servidor WWW Apache	19
1.5 Modulo 5: Programación con PHP	20
Lenguaje de programación PHP	20
Características de PHP	20
Usos de PHP	20
Ventajas de PHP	21
Desventajas de PHP	21
Funcionamiento del intérprete	21
1.6 Modulo 6: Interacción de WWW con bases de datos	21
Gestores de Bases de Datos Libres	22
Introducción a SQL	22
1.7 Modulo 7: Introducción a la Seguridad en Cómputo	23
Seguridad Computacional	23
Políticas de Seguridad	23
Mecanismos de Seguridad	23
Planes de Contingencia	24
Servicios de Seguridad propuestos por OSI	24
1.8 Funcionamiento de una aplicación Web	25
Arquitectura Web	26
Aplicaciones Multinivel	26
Ventajas de una aplicación Web	27
Caso Práctico	29
2.1 Ejercicio:	29
2.2 Desventajas del proceso actual	31

2.3 Ventajas al contar con un sistema informático	32
2.4 Diagrama de Flujo del Proceso Actual	32
2.5 Metodología Usada	33
Rational Unified Process (RUP)	34
Entregables del proyecto	34
Gráfica de Gantt	37
<i>Análisis de Caso Práctico</i>	38
3.1 Requerimientos Funcionales	38
3.2 Propósito del Sistema	41
3.3 Alcance	42
3.4 Diagramas de Casos de Uso	56
3.5 Diagramas de Secuencia	66
Preinscribir	66
Agregar actividades	67
Modificar actividades	67
Baja de Actividades	68
Inscribir sin preinscripción	69
Inscribir con preinscripción	70
Reservar en las instalaciones del gimnasio	71
Reservar actividades vía Web	72
Cancelar actividades vía Web	73
Cancelar actividades en las instalaciones del gimnasio	74
Actualizar Información del socio vía Web	75
Actualizar Información del socio por el personal Administrativo	76
Asistencias	77
Inasistencias	78
<i>Diseño de Caso Práctico</i>	79
Modelo Vista Controlador	79
Arquitectura de Sistema	80
Diseño De Capa de Negocio	80
Diseño de la Capa de Persistencia	82
Diseño de Vista	83
<i>Programación</i>	86
Elementos de Programación	86
Herramienta de desarrollo	86
Lenguaje de programación	87
Contenedor de servlets	87
Codificación Vista	87
Codificación Controlador	87
Codificación Modelo	88
Componente Service Locator.	88
Componente Business Delegate.	88
Componente: Facade.	89
Componente: Cache de Segundo Nivel.	90
Componente: Session Factory Acceso a Base de Datos.	90
<i>Implementación</i>	92
Diagrama de Arquitectura.	92
Requerimientos de Hardware y Software	92

Implementación de Caso Práctico bajo Software Libre

Usuarios creados en el sistema Operativo _____	92
Acciones de Seguridad _____	93
Recomendaciones para la operación del sistema _____	93
Pruebas de operación _____	93
Entregables de los procesos de codificación, prueba e instalacion _____	94
Formato de satisfacción del cliente _____	95
Conclusiones _____	97
Bibliografía _____	99

Introducción

A lo largo de la historia han existido periodos en los que han aparecido tecnologías que han revolucionado la vida cotidiana convirtiéndose en límites entre lo viejo y lo nuevo; definiendo también las generaciones marcadas por esos cambios tecnológicos; Internet se ha convertido en la tecnología que mas recientemente ha cambiado la forma en que las personas interactúan entre si y la forma en que hacen negocios. Las transacciones y las comunicaciones ocurren ahora en tiempos casi instantáneos.

En la actualidad es necesario contar con sistemas que permitan la automatización del seguimiento y control de procesos que se desarrollan en diversas empresas y oficinas. Con esta automatización se tiene un uso más eficiente de los recursos y un control más preciso de los mismos, lográndose con ello, compartir información entre diversas áreas y oficinas, a través de la red local e incluso empleando Internet o la intranet corporativa.

Con la aparición de las aplicaciones Web que dependen cada vez más de tecnologías de servidor, los departamentos de informática de las empresas necesitaban alguna forma de desarrollar aplicaciones que sean sustentables y escalables. Es necesario diseñar estas aplicaciones de forma que pudieran atender a miles de usuarios en forma simultanea, 24 horas al día, 7 horas a la semana sin ningún tipo de actividad.

Hoy en día, las herramientas de trabajo, lenguajes de programación, metodologías, mejores prácticas de desarrollo, han evolucionado tanto que la construcción de un sistema de información se ha vuelto casi transparente para los desarrolladores, sin dejar de lado que ahora lo que lleva o a lo que se le tiene que invertir tiempo es a la selección de las herramientas y lenguajes más adecuadas al tipo de sistema que se quiera construir.

Es por ello que es importante tener un conocimiento previo de estas herramientas, sus características, ventajas y desventajas frente a otras herramientas; saber en qué circunstancias utilizarlas y sacarles el mejor provecho posible.

En la actualidad se ha vuelto necesario hacer conciencia de que la seguridad es una responsabilidad compartida; por lo que todos los usuarios de computadoras requieren un conocimiento esencial de los elementos de la seguridad con que actualmente disponemos en el mundo de la computación; la seguridad por lo tanto debe ser una de las consideraciones principales al conectar cualquier aplicación a una red; recordando que una red es la columna vertebral de la computación distribuida y es por lo tanto el camino que utilizan los hackers para infiltrarse en un sistema de computación.

Una parte fundamental en la construcción de un sistema es la inversión que se hará en herramientas y tomando en cuenta que éstas son muy caras cuando se habla de licencias; es también válido echar un vistazo a las herramientas libres actuales y por qué no decirlo, muchas veces resultan ser mejores que por aquellas herramientas por las que necesitas pagar una licencia.

A lo largo de este trabajo, se verán todas las etapas por las que pasa la construcción de un sistema de Información así como las herramientas que se utilizaran en cada una de ellas.

El haber cursado este diplomado, y el realizar el siguiente trabajo, planteó algunos objetivos:

Implementación de Caso Práctico bajo Software Libre

- Aprender a utilizar en un ámbito profesional las herramientas vistas en el diplomado, siguiendo una metodología de desarrollo de software.
- Conocer diferentes plataformas de desarrollo, sus ventajas y desventajas, para tomar en cada caso la plataforma que mas se adecuó a las necesidades del software a desarrollar.
- Tener la experiencia y el conocimiento sobre el proceso de transición entre la etapa de construcción y de producción.
- Comprobar el potencial que el software libre tiene y la fuerza que día a día va tomando como alternativa para la industria del software.
- Por último, se espera demostrar las ventajas y la versatilidad que tienen las aplicaciones Web, y cómo estas han ido desplazando a las aplicaciones stand alone.



Capítulo 1

Desarrollo de Sistemas Bajo Software Libre

Modulo 1: Sistema Operativo Linux

Modulo 2: Instalación y Administración de Linux

Modulo 3: Editores para la creación de Páginas Web

Modulo 4: Administración de servidores de WWW con Linux

Modulo 5: Programación con PHP

Modulo 6: Interacción de WWW con bases de datos

Modulo 7: Introducción a la Seguridad en Cómputo

Funcionamiento de una aplicación Web

Diplomado “Desarrollo de Sistemas bajo Software Libre”

Gracias al crecimiento, diversidad y potencial que se ha alcanzado con los recursos informáticos, el escenario de las aplicaciones ha tenido que evolucionar de la misma manera, siendo necesario implementar aplicaciones que exploten de forma eficiente los recursos con que contamos hoy en día. Uno de esos recursos son las herramientas de software libre, las cuales resultan una excelente alternativa en el desarrollo de sistemas. Ante esta situación, decidí cursar el diplomado “Desarrollo de Sistemas bajo Software Libre” de la Universidad Nacional Autónoma de México, que tiene por objetivo “Conocer nuevas herramientas administrativas que permitan desarrollar e implementar sistemas para el control de procesos e información, que funcionen de forma natural en red o por Internet, empleando herramientas de software libre que han demostrado tener una alta confiabilidad, alto desempeño y funcionalidad”, del cual presento el siguiente reporte.

1.1 Modulo 1: Sistema Operativo Linux

Linux es un sistema operativo de libre distribución, desarrollado originalmente por Linus Torvalds, un estudiante de la universidad finlandesa de Helsinki, quien, en 1991, se abocó a la tarea de reemplazar a Minix, un clon de Unix de pequeñas proporciones y finalidad académica desarrollado años antes por Andrew Tannenbaum.

Linus empezó escribiendo el núcleo del proyecto en ensamblador, y luego comenzó a añadir código en C, lo cual incrementó la velocidad de desarrollo. Linus nunca anunció la versión 0.01 de Linux (agosto 1991), esta versión no era ni siquiera ejecutable, solamente incluía los principios del núcleo del sistema, estaba escrita en lenguaje ensamblador y asumía que uno tenía acceso a un sistema Minix para su compilación. En octubre de 1991, anunció la primera versión "oficial" de LINUX, la 0.02, que ya era capaz de ejecutar el SHELL bash y el compilador gcc de GNU

LINUX es un sistema operativo, compatible Unix. El sistema lo forma el núcleo del sistema (kernel) mas un gran número de programas y librerías que hacen posible su utilización; se distribuye bajo licencia GNU¹, ha sido diseñado y programado por multitud de programadores alrededor del mundo. El núcleo del sistema sigue en continuo desarrollo bajo la coordinación de Linus Torvalds.

Características

Dentro de las características más sobresalientes con las que cuenta Linux se tienen las siguientes:

- Multiusuario. Distintos usuarios pueden acceder a los recursos del sistema simultáneamente aunque se trate de una instalación en una sola máquina.
- Multiplataforma. Funciona con la mayoría de plataformas del mercado: Intel 386/486/Pentium, Motorola 680, Sun Sparc, Alpha etc
- Shells² programables que lo convierten en el sistema más flexible que existe.
- Soporte para cualquier cantidad y tipo de dispositivos directamente en el núcleo.
- Soporte para la mayoría de sistemas de archivos.

¹ Interfaz del sistema operativo que interactúa con el usuario y pasa los comandos para ser ejecutados

² Interfaz del sistema operativo que interactúa con el usuario y pasa los comandos para ser ejecutados

Implementación de Caso Práctico bajo Software Libre

- Multitarea: Habilidad de ejecutar varios programas al mismo tiempo.
- Multiprocesador: Soporte para sistemas con mas de un procesador.
- Protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.
- Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.
- Compatible con POSIX, System V y BSD a nivel fuente.
- Soporte para muchos teclados nacionales o adaptados y es bastante fácil añadir nuevos dinámicamente.
- Consolas virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas .
- Acceso transparente a particiones MS-DOS (o a particiones OS/2 FAT) mediante un sistema de archivos especial.

Distribuciones

Una distribución no es otra cosa, que una recopilación de programas y archivos, organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet, o comprando los CD's de las mismas, los cuales contendrán todo lo necesario para instalar un sistema Linux bastante completo y en la mayoría de los casos un programa de instalación que ayudara en la tarea de una primera instalación.

Algunas de las distribuciones más comunes son:

 S.u.S.E. http://www.novell.com/linux/suse/	 CALDERA http://www.caldera.com/
 http://www.esware.com/	 http://www.gentoo.org/
 http://www.es.debian.org/	 http://www.slackware.com/
 http://www.mklinux.org/	 redhat http://www.redhat.com/

 http://www.linux-mandrake.com/es/	 http://www.hispafuentes.com/
--	---

Componentes del Sistema Operativo

Linux está compuesto básicamente de 4 capas:

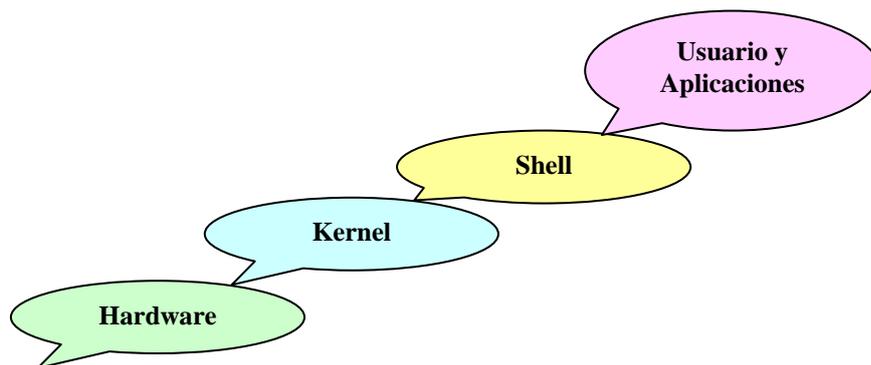


Figura 1.1 Componentes de un sistema operativo

- **Hardware:** Conjunto de piezas físicas del equipo de cómputo. Discos, terminales, CPU, memoria y demás dispositivos.
- **Kernel:** Es el corazón del sistema operativo y es iniciado cada vez que el sistema es arrancado. Interpreta las instrucciones proporcionadas por el usuario y las convierte en lenguaje máquina e indica al hardware lo que tiene que realizar con dicha información. Maneja todos los recursos del sistema y los presenta al usuario de manera coherente. Controla el acceso a la computadora y a sus archivos, asigna recursos a las distintas actividades que se llevan a cabo en la computadora, registra el trabajo realizado por el CPU de manera que los procesos que cada usuario ejecute puedan ser efectuados tan eficientemente como sea posible.
- **Shell:** Interprete de comandos; por este medio es como el usuario se entiende con el sistema. Un shell es un programa que interpreta y ejecuta los comandos conforme se proporciona desde la Terminal. El shell se utiliza con tres propósitos diferentes:
 1. Interprete de comandos
 2. Para personalizar el ambiente de trabajo dentro de una sesión.
 3. Como lenguaje de programación.
- **Aplicaciones:** Programas y aplicaciones que se le han agregado al sistema como hojas de cálculo, lenguajes de programación, manejadores de bases de datos, procesadores de texto, etc.

Sistema de Archivos

Un sistema de archivos es un método lógico de organización y almacenamiento de gran cantidad de información, de modo que facilita al usuario el manejo de la misma. El archivo es la unidad más pequeña en que la información es almacenada; todo archivo tiene un nombre.

El sistema de archivos nativo de Linux es el EXT2. Ahora existen otros sistemas de archivos con journalising (si se arranca sin haber cerrado el sistema, no necesitan hacer un chequeo sino que recuperan automáticamente su último estado), los más conocidos son EXT3, ReiserFS y XFS.

Linux esta compuesto de un sistema de archivos jerárquico, el cual muchas veces se denomina árbol, en el cual no existen unidades de disco, en su lugar cada unidad de almacenamiento así como cada dispositivo de hardware son reconocidos como un archivo o directorio dentro del sistema. En la parte superior de la jerarquía está el directorio especial llamado root, representado por el símbolo de / a partir de este punto se desprenden diferentes ramas de directorios como los siguientes:

Directorio	Descripción
/	Es la raíz del sistema, base de todos los archivos y directorios.
/etc	Contiene los archivos de configuración de la mayoría de los programas.
/home	Contiene los archivos personales de los usuarios.
/bin	Contiene comandos básicos y muchos programas.
/dev	Contiene archivos simbólicos que representan dispositivos reales físicos como lo son discos duros, memoria, etc.
/mnt	Contiene subdirectorios donde se montan (se enlaza con) otras particiones de disco duro, CDRoms, etc.
/tmp	Archivos temporales o de recursos de programas.
/usr	Programas y librerías instalados con la distribución
/usr/local	Programas y librerías instalados por el administrador
/sbin	Comandos administrativos
/lib	Librerías usadas por varios programas y lenguajes
/var	Datos varios como archivos de log (registro de actividad) de programas, bases de datos, contenidos del servidor web, copias de seguridad...
/proc	Información temporal sobre los procesos del sistema.

Un directorio puede contener archivos normales y subdirectorios, en Linux, se considera un subdirectorio como un archivo especial.

Atributos de los archivos

Dentro del sistema operativo se encuentran diferentes tipos de archivos, que se distinguen por el primer bit de sus propiedades, que indica si el elemento es un directorio o un archivo, así como el tipo de archivo de que se trata. Los archivos más comunes en Linux son los directorios (d), archivos ordinarios (-), ligas (l) archivos de tipo carácter (c) y los archivos de bloque (b).

Permisos

Existen tres tipos de permisos que se aplican tanto a los archivos como a los directorios: lectura (r), escritura (w) y ejecución (x). Estos permisos se encuentran asignados en tres bloques: usuario o dueño del elemento listado, grupo al que pertenece el usuario y a otros usuarios del

Implementación de Caso Práctico bajo Software Libre

sistema, de ello que los atributos se componen de 10 bits, el primero indica el tipo de elemento y los siguientes 9 representan los tipos de permisos asignados a cada elemento listado. Se pueden modificar los permisos a un archivo con el comando **chmod**.

Comandos y Utilerías básicas

El Sistema Operativo Linux incluye una gran variedad de programas de utilidad que se pueden utilizar fácilmente para realizar tareas específicas. Estos programas son flexibles, adaptables, portables y modulares, y pueden ser usadas junto con filtros y redireccionamientos para hacerlos más poderosos, por lo cual reciben el nombre de utilerías.

Una utilería se puede definir como una herramienta integrada del sistema operativo, que permite el tratado eficiente de datos para lograr obtener información de relevancia para la organización o usuarios que la necesiten; se pueden encontrar en forma de comandos los cuales son ejecutados como órdenes a través del intérprete de comandos del sistema operativo (shell).

Estas rutinas se encuentran en los directorios :

/bin
/usr/bin

Estos directorios incluyen las utilerías estándar de Linux. Por convención, /bin contiene las utilerías (archivos ejecutables) usadas con más frecuencia, mientras que en /usr/bin están las utilidades más complicadas, así como programas específicos de ciertos equipos.

La estructura básica de los comandos es:

\$comando [-opciones o modificadores] [parámetros] archivo(s)

En donde el signo de pesos (\$) es el prompt o símbolo de sistema; después se indica el comando a ejecutar y después pueden o no especificarse modificadores u opciones al comando; algunos comandos trabajan con **parámetros** que deben especificarse después de los modificadores del comando; y, finalmente se indica el archivo o archivos sobre los que actuará el comando.

A continuación se muestra una lista de los principales comandos y utilerías de Linux.

- **pwd**: muestra el directorio actual en el que se está trabajando.
- **ls**: nombre de los archivos contenidos en un directorio.
- **cd** : Permite cambiar el directorio actual de trabajo.
- **cat**: Despliega el contenido de uno o mas archivos.
- **more**: Despliega los archivos en la Terminal de forma paginada.
- **mkdir**: Crea uno o mas directorios.
- **rm**: Borra uno o mas archivos.
- **cp**: Copiar un archivo a otro, o un conjunto de archivos a un directorio.
- **head**: Imprime las primeras líneas de un archivo.
- **tail**: Muestra las últimas líneas de un archivo.
- **mv**: Mover archivos o estructuras de directorios de un lugar a otro y renombrar archivos o directorios.
- **hostname**: Muestra el nombre del servidor
- **uname**: Indica el tipo de Linux que se está utilizando

Implementación de Caso Práctico bajo Software Libre

- **pico:** Programa que permite editar archivos de texto
- **chmod:** Permite modificar permisos a los archivos
- **who:** Muestra un listado sencillo de quien se encuentra conectado en el servidor.
- **w:** Muestra un listado con detalles de los usuarios conectados en el servidor
- **grep:** Imprime las líneas de un archivo que cumplan un patrón especificado
- **wc:** Cuenta líneas, palabras y caracteres.
- **sort:** Ordena una lista de palabras del archivo que se le pasa como argumento.
- **ps:** Muestra los procesos del sistema
- **tar:** Empaqueta archivos.
- **Gzip:** Comprime archivos.

Caracteres Especiales

Linux maneja varios caracteres especiales, cada uno de ellos realiza una función en específico, en ocasiones pueden ser combinados para aumentar el potencial del comando en ejecución.

Estos caracteres son:

; , < > | >> ^ \$ & / ~ \ [] ' " () . * ?

Comodines

Existen dos caracteres llamados comodines los cuales pueden sustituir un carácter o conjunto de caracteres durante la ejecución de comandos; estos son * ?

Redireccionamiento

En Linux existe una salida estándar que es el monitor, de igual manera hay una entrada estándar que es el teclado. El carácter mayor que > permite redireccionar la salida de un programa a un archivo. El carácter menor que < permite redireccionar un archivo hacia un programa. El carácter pipe | funciona como un filtro mandando la salida de un programa hacia la entrada de otro programa, enlazando la salida de uno con la entrada de otro. El signo doble mayor >> permite redireccionar la salida de un programa hacia un archivo, si el archivo no existe lo crea, pero si el archivo existe el resultado del comando lo añade al final del archivo existente.

Existe una salida estándar de verdadero y una salida de falso, a la salida de verdadero le corresponde el número 1 y a la salida de falso le corresponde el número 2, de tal forma que si en algún comando no se cumple una condición en lugar de que el error se imprima en pantalla pueda ser redireccionado a un archivo

Ayuda del sistema

Linux cuenta con un manual en línea de los comandos, el cual puede ser desplegado en pantalla mediante el comando man teniendo como argumento el comando que se quiere consultar; es posible acceder también a él para obtener información acerca del funcionamiento del comando man.

Procesos

Un proceso es un programa que se encuentra corriendo dentro del servidor, cada uno tiene un tiempo de vida que va desde el momento en que se presiona la tecla enter hasta que se termina

Implementación de Caso Práctico bajo Software Libre

la ejecución del proceso. Todos los procesos son invocados por un usuario desde el intérprete de comandos de tal forma que ningún otro usuario puede afectar las tareas que otra persona esta realizando. Los estados en los que se puede encontrar un proceso son:

- Zombie: Cuando un proceso corre en forma independiente, incluso del shell o del proceso que lo invocó.
- En ejecución, se encuentra en el CPU.
- Stopped o detenido. Cuando el proceso ya no esta corriendo en el CPU.
- Sleeping Durmiendo en espera de recibir una solicitud para iniciar su trabajo.
- Dormido sin interrupción posible, normalmente relacionado con entrada/salida.

1.2 Modulo 2: Instalación y Administración de Linux

Perfil y actividades del administrador

El administrador de sistemas es la persona responsable de configurar, mantener y actualizar el sistema o conjunto de sistemas que forman una red; cuidando el funcionamiento del software, hardware y periféricos de forma que estén disponibles para ser utilizados por los usuarios; el perfil del administrador de sistemas debe cumplir o realizar lo siguiente:

Tareas administrativas comunes:

- Administrar usuarios.
- Configuración de dispositivos.
- Hacer respaldos.
- Capacitar usuarios.
- Asegurar el sistema.
- Registrar los cambios del sistema.
- Asesorar a los usuarios.

Actividades del administrador

- Mantenimiento de claves de usuarios.
- Instalación y mantenimiento de dispositivos.
- Instalación y actualización de software (comercial y dominio público).
- Configuración de las interfaces de red.
- Administración de los recursos (cpu, memoria y disco).
- Atención a usuarios.
- Monitoreo del sistema.
- Detección de fallas.
- Auditoria e implantación de la seguridad del sistema.

Conocimientos del Administrador

- Técnicas de programación.
- Dominio de al menos un lenguaje de programación.
- Funcionamiento del sistema operativo.
- Técnicas de administración del sistema operativo.
- Conocimientos básicos de hardware y mantenimiento de dispositivos.
- Comprensión profunda sobre redirección, tuberías, procesamiento en segundo plano.
- Manejo de vi, pues es el común denominador entre los sistemas UNIX.

Implementación de Caso Práctico bajo Software Libre

- Programación shell.

Instalación

Requerimientos

La distribución de Linux que se utilizó en el diplomado fue Slackware, por lo que se será esta distribución la que se tratará en el resto del trabajo. Linux Slackware no requiere de un sistema extremadamente potente para ejecutarse; los requerimientos mínimos para instalar y ejecutar Slackware son:

- Procesador 386.
- 16MB en RAM.
- 50 megabytes de espacio libre en disco duro.
- Unidad de 3,5 ".
- Hardware adicional tarjeta de vídeo para ejecutar X Windows y una tarjeta de red por si se desea tener acceso a algún tipo de red.

Cabe mencionar que para instalar Slackware se debe seleccionar las series de paquetes que se instalarán. Linux Slackware es una de las distribuciones de Linux más antiguas, contiene un conjunto de paquetes que se reparten en series, cada una de estas series esta etiquetada con una letra del abecedario comenzando con la letra A hasta llegar a la Z.

Hoy en día las distribuciones de Linux son distribuidas en CD-ROM e incluyen todos los paquetes completos.

Discos de Arranque

Cuando no se cuenta con una unidad de CD-ROM, es necesario contar con los discos de arranque para la instalación: de BOOT y de ROOT.

Si la instalación se realiza por CD-ROM, automáticamente se cargan las imágenes de los discos boot y root; siempre y cuando esté habilitada la opción en el BIOS del sistema.

La instalación de Linux Slackware no tiene mayor dificultad que seguir los pasos que se muestran en la pantalla de instalación del sistema. Linux detectará la mayor parte del hardware que está instalado en el equipo.

Particiones

En Linux es habitual tener más de una partición, el área de swap y el sistema de archivos Linux en la mayoría de los casos.

Tener en una misma partición todo el Linux dificulta la tarea de los administradores de sistemas; puede dificultar la capacidad para actualizar o hacer copias de seguridad, incluso algunos archivos dañados pueden provocar problemas, lo que significa que la jerarquía de un directorio dañado puede afectar a los restantes, o puede obligar a reinstalar Linux; éstos problemas se pueden evitar creando una partición independiente para cada uno de los sistemas principales de archivos. Dicha configuración mejora la seguridad y permite gestionar las copias de seguridad y su posterior recuperación. La existencia de varias particiones ofrece varias ventajas entre las que están:

- Sencilla gestión de las copias de seguridad y actualizaciones.
- Capacidad para controlar la forma en que se monta cada sistema de archivos.
- Protección contra programas SUID renegados.

Implementación de Caso Práctico bajo Software Libre

- Evita la denegación de servicio accidental y protege de desbordamientos al sistema de archivos raíz.

El termino montar hace referencia a la forma en que Linux permite utilizar los distintos sistemas de archivos. Cuando Linux monta un sistema de archivos local o externo, conecta el sistema a un dispositivo o directorio local, lo que permite un punto de acceso o punto de montaje.

Dar de alta y/o baja el sistema

En muchos sistemas operativos, no es necesario realizar ningún procedimiento especial para desconectarse, el sistema se puede apagar en cualquier momento.

Linux no funciona de este modo, sino que necesita tiempo para cerrar los procesos abiertos y guardar los datos en el disco.

Para apagar el sistema Linux, se necesita utilizar el comando shutdown, diseñado para desconectar Linux de forma segura. Durante el proceso de apagado, shutdown realiza las siguientes acciones:

- Notifica a los restantes procesos y usuarios que el apagado es inminente.
- Apaga otros procesos que aún se están ejecutando.
- Notifica a root a medida que se desconecta cada servicio.
- Si así se especifica, reinicia el sistema.

Mantenimiento de claves de usuario

En Linux, Root es el encargado de administrar a los usuarios individuales a los grupos y los archivos.

Una cuenta de usuario consta de dos elementos:

1. Un nombre de usuario y una contraseña valida.
2. Un directorio inicial.

Cuando un usuario intenta iniciar una sesión, Linux comprueba si se cumplen estos requisitos, examinando el archivo passwd, en el cual se encuentran los datos correspondientes a cada usuario: username, password, userID, groupID, real name, user home y shell.

Para agregar o eliminar un usuario, existen varias formas de hacerlo:

- Utilizando herramientas gráficas.
- Utilizando herramientas de línea de comandos.
- Modificando /etc/passwd manualmente

Los grupos se almacenan en /etc/group. En su estructura, cada línea almacena un registro y cada registro se almacena en cuatro campos delimitados por dos puntos (:).group name, group password ,group ID (GID), group users.

Muchas de las primeras distribuciones Linux almacenaban las contraseñas de los usuarios en /etc/passwd, lo que no resultaba seguro, ya que /etc/passwd es y debe ser legible. De ahí que cualquier usuario pueda ver el contenido de /etc/passwd.

Las contraseñas Linux están encriptadas, de esta manera aunque /etc/passwd es público la contraseña es ilegible para cualquier usuario curioso. La base de datos de contraseñas es el

Implementación de Caso Práctico bajo Software Libre

archivo `/etc/shadow`, es un archivo especial que almacena no solo las contraseñas de los usuarios sino también de indicadores de reglas especiales.

El administrador debe proteger los archivos del sistema, asignándoles permisos adecuados. Existen 3 tipos de permisos para los archivos.

r	lectura	Permite ver el contenido.
w	escritura	Permite modificar el contenido.
x	ejecución	Permite ejecutar el contenido.

Para el caso de los directorios, los permisos tienen un significado un poco diferente.

r	lectura	Permite ver los nombres de los archivos.
w	escritura	Permite hacer cambios al directorio.
x	ejecución	Permite buscar o cambiarse en el directorio.

Instalación y mantenimiento de dispositivos

Las distribuciones de Slackware Linux incluyen un kernel reciente con todos los módulos compilados y puede detectar hardware en el momento del arranque, de forma que un administrador o un usuario en general no debe encargarse de esto. Sin embargo puede haber algunos casos excepcionales:

Interfaces gráficas en Linux

Las interfaces gráficas para Linux y para todos los sistemas tipo Unix en general son conocidas como ambientes XWindow. Para el caso de Linux se utiliza una implementación de XWindow con código abierto (open source) denominada XFree86 que también está disponible para otros sistemas tipo Unix como los BSD (FreeBSD, OpenBSD y NetBSD), Solaris, etc.

La arquitectura de un sistema X es del tipo cliente-servidor. El servidor se encarga de tomar la entrada de los usuarios y hacerla llegar a las aplicaciones clientes, además de recibir y redireccionar las salidas de estas aplicaciones. Por lo general el servidor X se ejecuta en la misma estación que las aplicaciones clientes.

La forma más acabada de una interfaz gráfica para Linux es lo que se conoce como Entorno Gráfico de Escritorio o GDE (Graphical Desktop Environments). Estos se caracterizan fundamentalmente por la cantidad y variedad de las aplicaciones que ofrecen, la interrelación entre las mismas, las facilidades para la interacción con el usuario, y las posibilidades de configuración visual que poseen de acuerdo al gestor de ventanas empleado.

Los dos ejemplos clásicos de entornos de escritorio son:

- KDE (K Desktop Environment)
- GNOME (GNU Network Object Environment)

1.3 Modulo 3: Editores para la creación de Páginas Web

El HTML, acrónimo inglés de Hyper Text Markup Language (lenguaje de marcación de hipertexto), es un lenguaje informático diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

HTML sirve para crear los documentos que se publican en la Web, éstos pueden contener imágenes, sonido, texto y/o archivos multimedia.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en el navegador el texto, así como también las imágenes y los demás elementos, en la pantalla de la computadora.

Estructura básica de un documento HTML

Las etiquetas básicas para crear una página HTML son:

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <p>ejemplo</p>
  </body>
</html>
```

- **<HTML>**: Es la etiqueta que define el inicio del documento html, le indica al navegador que todo lo que viene a continuación debe tratarlo como una serie de códigos html.
- **<HEAD>**: Define la cabecera del documento html, esta cabecera suele contener información sobre el documento que no se muestra directamente en el navegador. Como por ejemplo el título de la ventana de su navegador. Dentro de la cabecera **<HEAD>** podemos encontrar:
 - **<TITLE>**: Define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana
 - **<LINK>**: Utilizada para definir algunas características avanzadas, como por ejemplo las hojas de estilo usadas para el diseño de la página.
- **<BODY>**: Define el contenido principal o cuerpo del documento, esta es la parte del documento html que se muestra en el navegador, dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes. Dentro del cuerpo **<BODY>** se pueden encontrar diferentes etiquetas. A continuación se indican algunas a modo de ejemplo:
 - **<H1>, <H2>, ... <H6>**: Encabezados o títulos del documento en diferentes tamaños de fuente
 - **<P>**: Indica un párrafo nuevo
 - **
**: Salto de línea forzado
 - **<TABLE>**: Comienzo de una tabla (las filas se identifican con **<TR>** y las celdas dentro de las filas con **<TD>**)
 - **<A>**: Indica la existencia de un hipervínculo o enlace, dentro o fuera la página web. Debe definirse el parámetro de pasada por medio del atributo href (ejemplo: **Google** se representa como Google)
 - **<DIV>**: Comienzo de un área especial en la página
 - ****: Indica la existencia de una imagen para mostrarse en el navegador

Implementación de Caso Práctico bajo Software Libre

A continuación se muestran algunos ejemplos de interfaces gráficas realizadas con HTML.



Figura 1.2 Interfaz del sitio de la UNAM

Elementos Avanzados

Formularios

Los formularios son muy utilizados para realizar búsquedas o bien para introducir datos personales en una página Web por ejemplo en sitios de comercio electrónico. Los datos que el usuario introduce en estos campos son enviados al correo electrónico del administrador del formulario o bien a un programa que se encarga de procesarlo automáticamente.

Los formularios son definidos por medio de las etiquetas `<form>` y `</form>`. Entre estas dos etiquetas se colocan todos los campos y botones que componen el formulario. Dentro de esta etiqueta `<form>` debemos especificar algunos atributos:

- **METHOD:** Este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar este atributo son `post` y `get`.
- **ACTION:** Define el tipo de acción a llevar a cabo con el formulario.
- **ENCTYPE:** Indica la forma en la que viajará la información que se mande por el formulario. En el caso más corriente, enviar el formulario por correo electrónico, el valor de este atributo debe de ser `"text/plain"`.

Dentro de un formulario se puede recoger información utilizando los distintos elementos disponibles. La forma más usual es permitiendo que el usuario introduzca texto, aunque también se pueden elegir una opción entre una lista, pulsando uno o más botones, etc. Todo ello se puede realizar combinando tres etiquetas HTML: `INPUT`, `TEXTAREA` y `SELECT`.

Implementación de Caso Práctico bajo Software Libre

Frames

Los frames permiten partir la ventana del navegador en diferentes áreas. Cada una de estas áreas es independiente de las otras y deben ser codificadas con archivos HTML también independientes. Como resultado, cada frame o marco contiene las propiedades específicas que se le indiquen en el código HTML a presentar en ese espacio. Así mismo, y dado que cada marco es independiente, tendrán sus propias barras de desplazamiento, horizontales y verticales, por separado.

Se suelen utilizar para colocar en una parte de la ventana una barra de navegación, que generalmente se encuentra fija y permite el acceso a cualquier zona de la página Web. Una de las principales ventajas de la programación con frames viene derivada de la independencia de los distintos frames, pues podemos navegar por los contenidos de nuestro Web con la barra de navegación siempre visible, y sin que se tenga que recargar en cada una de las páginas que vamos visitando.

En la siguiente figura se muestran 2 formas de implementar distribuciones de los frames:

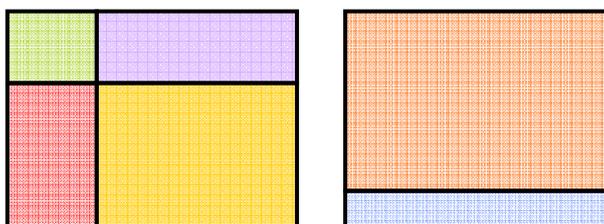


Figura 1.3. Áreas de una interfaz

1.4 Modulo 4: Administración de servidores de WWW con Linux

Servidor WWW

Un servidor Web es un programa que ofrece servicios dentro del WWW en Internet, es el encargado de ofrecer comunicación mediante el protocolo HTTP (hypertext transfer protocol).

Funcionamiento de HTTP

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que se suele conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear `www.unam.mx` en el navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. En este ejemplo el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Algunos servidores web para Linux más comunes actualmente son los siguientes:

- Roxen Challenger

Implementación de Caso Práctico bajo Software Libre

- Zope
- Xitami
- Zeus
- Iplanet
- cherokee
- Sioux
- SunOne
- Tomcat
- AOL Server
- Apache

El servidor sobre el que se hablará en este capítulo es Apache, que fue el servidor que se trató en el diplomado.

Instalación del servidor WWW Apache

La instalación de apache es muy sencilla, si no se cuenta con el software en un disco lo único que se tiene que hacer es conectarse a la red y descargarlo de la página <http://apache.org>

1. Para desempaquetar y descomprimir el archivo se utiliza el comando `$tar -xvzf apache1.3.31.tar.gz`
2. Dentro del directorio de apache, se encuentra el archivo `INSTALL`, en el que se pueden encontrar instrucciones que se deben de seguir para instalar apache.
 - a. **`./configure prefix=/usr/local/apache`**. `Configure` realiza una búsqueda del hardware y software del sistema y configura Apache de la manera más optima. Verifica todos los archivos, dependencias, módulos, etc que se van a necesitar en la instalación. Si todo sale bien con `./configure`, se procede a hacer `make`.
 - b. **`$make`**. Compila todos los archivos y dependencias para poder instalar apache.
 - c. **`$make install`**. Se encarga de copiar los archivos necesarios a los directorios especificados. Se crean los directorios y se instala apache.
3. Hasta aquí, esta terminada la instalación de apache, pero falta un punto muy importante, levantar el servicio, para esto dentro de la carpeta de apache hay un directorio llamado `bin`, dentro de ese directorio está el archivo para levantar (`start`) parar (`stop`) o reiniciar (`restart`) el servidor: `apachectl`.
4. Por último se comprueba que se levantó el demonio correctamente. Hay varias formas de realizarlo: una de ellas es abrir una ventana del navegador y teclear la siguiente dirección: `http://127.0.0.1` Si todo es correcto se debe mostrar la página de inicio de Apache. La otra forma es teclear en una terminal el comando `$ ps -fea | grep "httpd"`

Lo que resta de aquí en adelante es configurar el servidor de la manera que mas convenga; esto se hace con las directivas que se encuentran en el archivo `httpd.conf` que está en la carpeta de `conf`, dentro de Apache. El script `apachectl` está diseñado para actuar como un script estándar de tipo SysV `init`; puede tomar los argumentos `start`, `restart`, y `stop` y traducirlos en las señales apropiadas para `httpd`.

Un directorio virtual es un directorio que se encuentra en un directorio distinto del que se mapea en la URL. Esta directiva sólo se encuentra dentro del fichero de configuración `httpd.conf`.

1.5 Modulo 5: Programación con PHP

Lenguaje de programación PHP

PHP es un lenguaje de scripting que permite la generación dinámica de contenido en un servidor Web. Su nombre oficial es PHP: **Hypertext Preprocessor**³.

PHP fue creado originalmente por Rasmus Lerdorf como un conjunto de utilidades para añadir dinamismo a las páginas Web. Este conjunto de herramientas ganó rápidamente popularidad y fue posteriormente rediseñado por Zeev Suraski y Andi Gutmans y rebautizado como PHP 3.0. Más tarde se rediseñó completamente el intérprete, añadiéndole más potencia y nuevas funcionalidades, para dar lugar a PHP 4. Actualmente está disponible la versión 5 de PHP.

PHP es un lenguaje de programación que contiene muchos conceptos de C, Perl y Java. Su sintaxis es muy similar a la de estos lenguajes, haciendo muy sencillo su aprendizaje .

El código PHP está embebido en documentos HTML de manera que es muy fácil incorporar información actualizada en un sitio Web.

Características de PHP

PHP es un lenguaje para la creación de sitios Web del que se pueden destacar las siguientes características:

- Potente y robusto lenguaje de programación embebido en documentos HTML.
- Dispone de librerías de conexión con la gran mayoría de los sistemas de gestión de bases de datos para el almacenamiento de información permanente en el servidor.
- Proporciona soporte a múltiples protocolos de comunicación en Internet (http, IMAP, FTP, LDAP, SNMP).
- Código fuente abierto.
- Gratuito
- Portable y multiplataforma
- Eficiente
- Alta velocidad de desarrollo

Además dispone de facilidades para el procesamiento de archivos, funciones de tratamientos de textos, generación dinámica de imágenes, tratamiento de documentos XML.

Usos de PHP

Los principales usos del PHP son los siguientes:

- Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl, en Linux, Windows y Macintosh.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), que permite desarrollar aplicaciones de escritorio tanto para los sistemas operativos basados en Unix, como para Windows y Mac OS X.

³ El nombre PHP tiene su origen a principio de 1995 en un conjunto de macros conocido como Personal Home Page Tools

Ventajas de PHP

- La principal ventaja se basa en ser un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- Existe amplia documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.

Desventajas de PHP

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y php.
- La orientación a objetos es aún muy deficiente para aplicaciones grandes.

Funcionamiento del intérprete

PHP se ejecuta en el servidor antes de que la página sea enviada al usuario que realizó la petición. Cuando llega una petición a un servidor Web, éste localiza el documento solicitado por el cliente y, en función de una serie de parámetros de la propia configuración del servidor, decide la acción a realizar con el documento. Así, cuando un cliente pulsa sobre un enlace correspondiente a un documento que incluye código PHP, el navegador envía la solicitud correspondiente, y cuando detecta que se trata de un documento con código PHP pone en funcionamiento el intérprete de dicho lenguaje.

PHP puede ser instalado al menos en tres modos diferentes:

- Como un intérprete externo (modo CGI)
- Como una extensión del servidor
- Como un módulo interno del servidor (disponible sólo para Apache)

1.6 Modulo 6: Interacción de WWW con bases de datos

En la actualidad, muchas instituciones se han dado cuenta de la importancia que el Web tiene en el desarrollo de sus aplicaciones, ya que con ello pueden lograr una mejor comunicación con personas o instituciones situadas en cualquier lugar del mundo.

La mayor parte de información es presentada de forma estática a través de documentos HTML, lo cual limita el acceso a los distintos tipos de almacenamiento en que ésta pueda encontrarse. Pero, en la actualidad surge la posibilidad de utilizar aplicaciones que permitan acceder a información de forma dinámica, tal como a bases de datos, con contenidos y formatos muy diversos.

Una de las ventajas de utilizar el Web para este fin, es que no hay restricciones en el sistema operativo que se debe usar, permitiendo la conexión entre sí, de las páginas desplegadas en un navegador Web que funciona en una plataforma, con servidores de bases de datos alojados en

Implementación de Caso Práctico bajo Software Libre

otra plataforma. Además, no hay necesidad de cambiar el formato o estructura de la información dentro de las bases de datos.

Una de las principales ventajas que presenta el trabajar con páginas dinámicas es el poder almacenar los contenidos en bases de datos. De esta forma, podemos organizarlos, actualizarlos y buscarlos de una manera mucho más simple.

Gestores de Bases de Datos Libres

La siguiente tabla es una comparativa entre dos Gestores de Bases de Datos:

Base de Datos	Ventajas	Desventajas
MySQL	<ul style="list-style-type: none">✓ Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al disparar selects y demás.✓ Mejores utilidades de administración (backup, recuperación de errores, etc).✓ Aunque se cuelgue, no suele perder información ni corromper los datos.✓ Mejor integración con PHP.✓ No hay límites en el tamaño de los registros.✓ Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.✓ MySQL se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla "en caliente".	<ul style="list-style-type: none">✓ No soporta transacciones, "roll-backs" ni subselects.✓ No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.
Postgres	<ul style="list-style-type: none">✓ Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.✓ Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).✓ Tiene mejor soporte para triggers y procedimientos en el servidor.✓ Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL.✓ Además, tiene ciertas características orientadas a objetos.	<ul style="list-style-type: none">✓ Consume BASTANTES más recursos y carga más el sistema.✓ Límite del tamaño de cada fila de las tablas a 8k!!! (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).✓ Es de 2 a 3 veces más lenta que MySQL.✓ Menos funciones en PHP.

Introducción a SQL

El lenguaje **SQL** está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos. Existen dos tipos de comandos **SQL**:

- Los **DLL** que permiten crear y definir nuevas bases de datos, campos e índices.
- Los **DML** que permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Comandos **DLL**

Comando	Descripción
CREATE	Utilizado para crear nuevas tablas, campos e índices
DROP	Utilizado para crear nuevas tablas, campos e índices
TABLE	Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Comandos DML

Comando	Descripción
SELECT	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado
INSERT	Utilizado para cargar lotes de datos en la base de datos en una única operación.
UPDATE	Utilizado para modificar los valores de los campos y registros especificados
DELETE	Utilizado para eliminar registros de una tabla de una base de datos

Cláusulas

Las cláusulas son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

Comando	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar la condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico

1.7 Modulo 7: Introducción a la Seguridad en Cómputo

Seguridad Computacional

La seguridad es el conjunto de políticas y mecanismos que nos permiten garantizar la confidencialidad, integridad y disponibilidad de los recursos de un sistema. Un sistema de Cómputo es seguro si se puede confiar en que se comportará como se espera que lo haga, y que la información se mantendrá inalterada y accesible para los usuarios que él mismo decida.

La seguridad computacional es un término general que cubre una gran área de computación y procesamiento de la información. En la actualidad el activo más importante en una organización es la información. En algunas industrias, como el comercio electrónico, la disponibilidad y confianza de los datos pueden hacer la diferencia entre el éxito y el fracaso.

Un proceso fundamental para la seguridad informática es el análisis de riesgos, que realiza la identificación y evaluación del riesgo a sufrir un ataque y perder datos, tiempo y horas de trabajo, comparándolo con el costo que significaría la prevención de este suceso.

Políticas de Seguridad

Una política de seguridad de información es un conjunto de reglas aplicadas a todas las actividades relacionadas al manejo de la información de una entidad, teniendo el propósito de proteger la información, los recursos y la reputación de la misma. Las políticas son guías para asegurar la protección y la integridad de los datos dentro de los sistemas.

Mecanismos de Seguridad

Son la parte más visible de un sistema de seguridad. Se convierten en la herramienta básica para garantizar la protección de los sistemas. Se clasifican de acuerdo al siguiente diagrama:

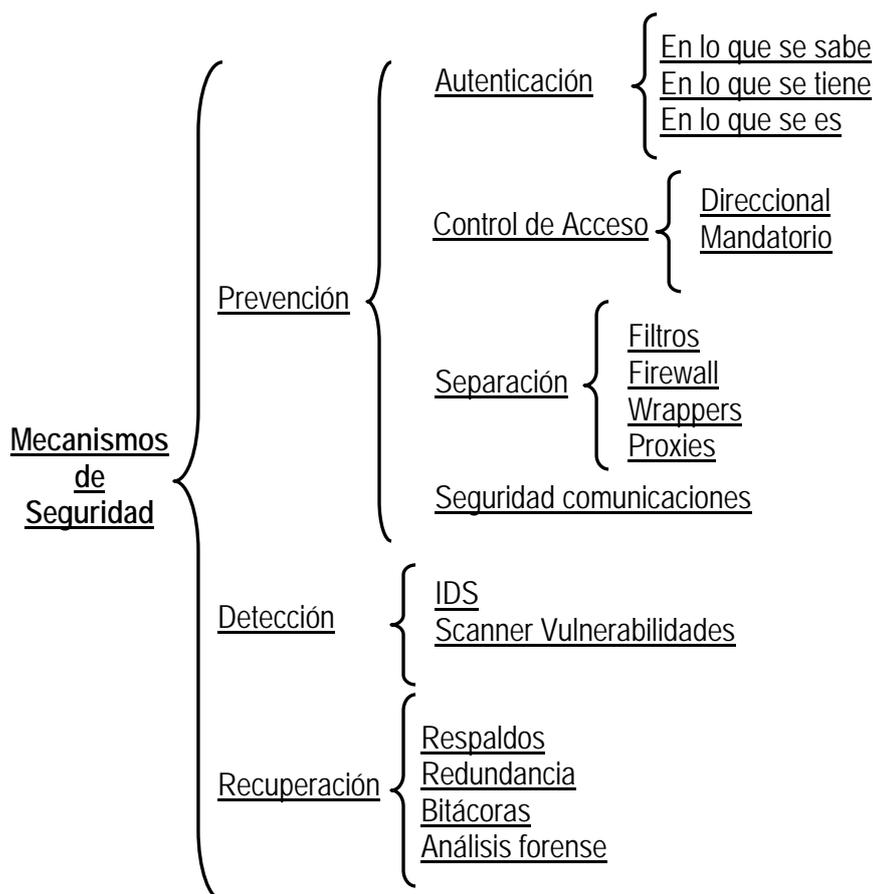


Figura 1.4. Diagrama de mecanismos de seguridad.

Planes de Contingencia

Consiste en un análisis pormenorizado de las áreas que componen una organización para establecer una política de recuperación ante un desastre; además de aumentar la seguridad de una empresa, se gana en el conocimiento de fortalezas y debilidades. Cuando no se cuenta con un plan de contingencia se pueden tener pérdidas de clientes, de imagen, de ingresos por beneficios, de ingresos por producción, de competitividad, de credibilidad en el sector.

Servicios de Seguridad propuestos por OSI

Un servicio de seguridad es una característica que debe tener un sistema para satisfacer una política de seguridad.

El estándar ISO 7498-2, que cubre las comunicaciones seguras entre sistemas abiertos define cinco clases de servicios de seguridad: Confidencialidad, Autenticación, Integridad, Autorización y No Repudio.

Un elemento fundamental de la seguridad informática es la criptografía, la cual tiene estos conceptos básicos:

- **Criptografía:** "Escritura oculta" se encarga de convertir un texto normal y comprensible en un formato incomprensible a menos que se posea un conocimiento secreto. Recientemente, se conoce como la ciencia de usar las matemáticas para cifrar y descifrar información.

Implementación de Caso Práctico bajo Software Libre

- **Criptanálisis:** Es el estudio de los métodos para obtener el sentido de una información cifrada, sin acceso a la información secreta requerida para obtener este sentido normalmente.
- **Cifrado:** El cifrado es el tratamiento de un conjunto de datos, contenidos o no en un paquete, a fin de impedir que nadie excepto el destinatario de los mismos pueda leerlos. Hay muchos tipos de cifrado de datos, que constituyen la base de la seguridad de la red.
- **Criptosistema:** Conjunto de procedimientos que garantizan la seguridad de la información y utilizan técnicas criptográficas. El elemento principal es la "llave"

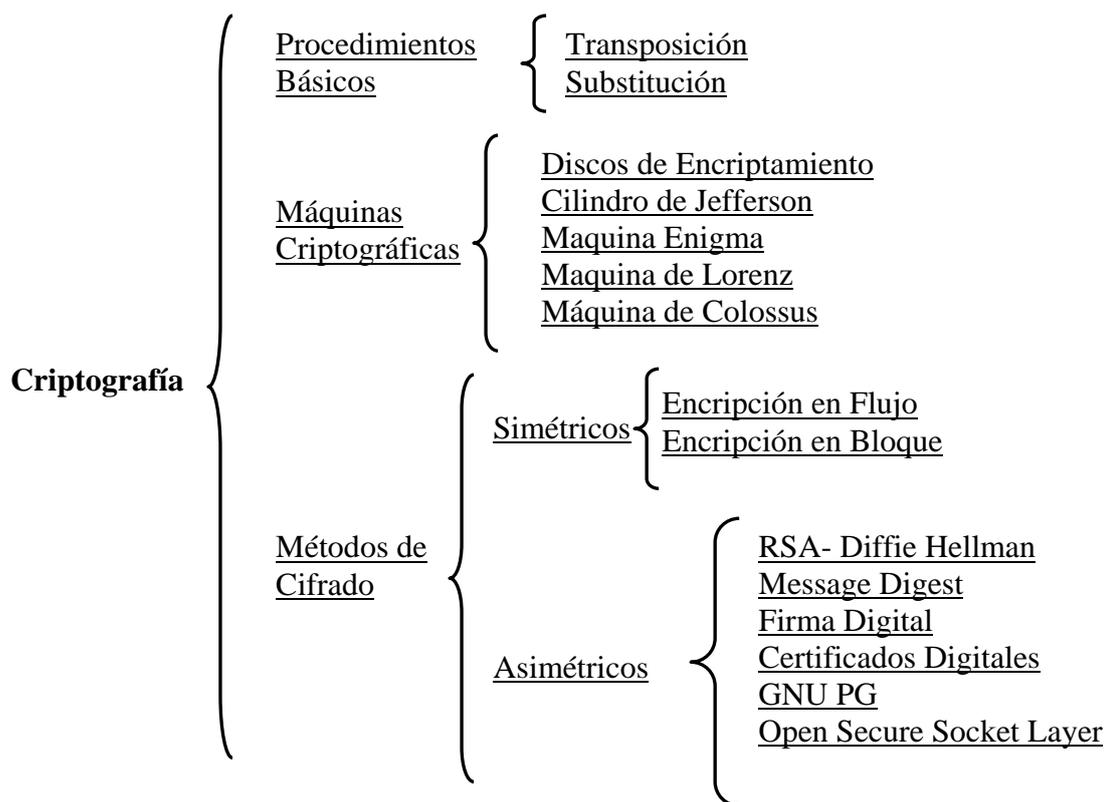


Figura 1.5. Elementos de la criptografía

1.8 Funcionamiento de una aplicación Web

La Web en sus orígenes fue pensada como un medio para desplegar información, esta reposa de manera estática en los servidores la cual es accedida a través de una consulta hecha por un navegador valiéndose del protocolo HTTP (Hyper Text Transfer Protocol).

Actualmente se maneja el mismo concepto en la comunicación cliente-servidor solo que no necesariamente el resultado de la comunicación debe provenir de la carga de una página estática, esta puede ser el resultado de la ejecución en el servidor de alguna lógica de programación. Esto último no necesariamente lo llamamos una aplicación Web, pero nos acerca al concepto. Una aplicación Web se considera como "un sitio Web donde la navegación a través de él y la entrada de datos por parte de un usuario, afectan el estado de la lógica del negocio. En esencia, una aplicación Web usa un sitio Web como entrada (front-end) a una aplicación típica. Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación Web".

Implementación de Caso Práctico bajo Software Libre

Bajo este concepto las aplicaciones Web no solo se encargan de desplegar información, sino que también, deben contener una lógica asociada que permita apoyar algún proceso propio del negocio para el cual fue diseñada.

Afortunadamente, existen herramientas potentes para realizar esto, ya que han surgido nuevas tecnologías que permiten que el acceso a una base de datos desde el Web. El único problema es decidir entre el conjunto de posibilidades la correcta para cada situación.

Cualquier sistema de complejidad no trivial, necesita ser analizado y modelado. Las aplicaciones Web, al igual que otras aplicaciones, necesitan métodos formales de análisis y diseño.

Arquitectura Web

Para abrir una página Web en un navegador, normalmente se teclea el correspondiente URL. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó, según se muestra en la siguiente figura

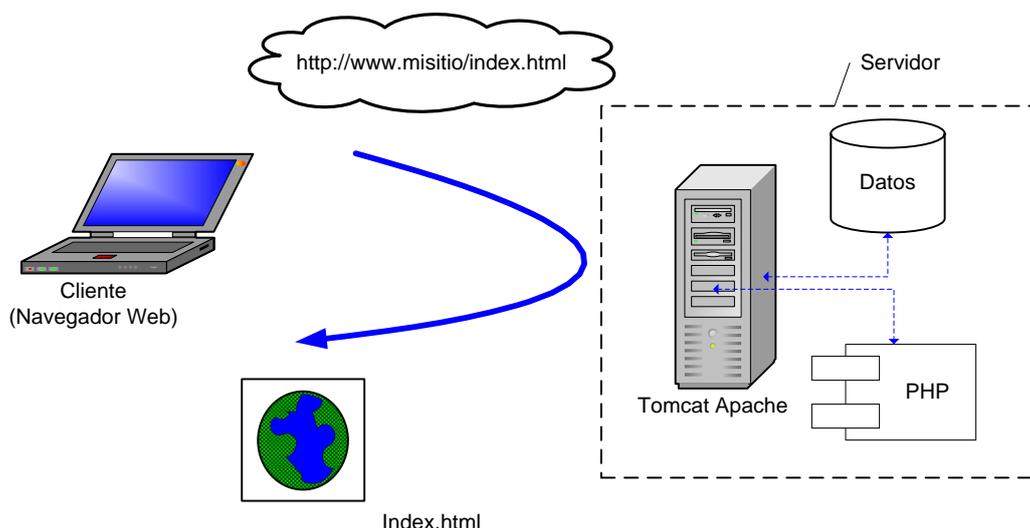


Figura 1.6: Arquitectura Web básica.

Aplicaciones Multinivel

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles. El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor Web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o script. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

Implementación de Caso Práctico bajo Software Libre

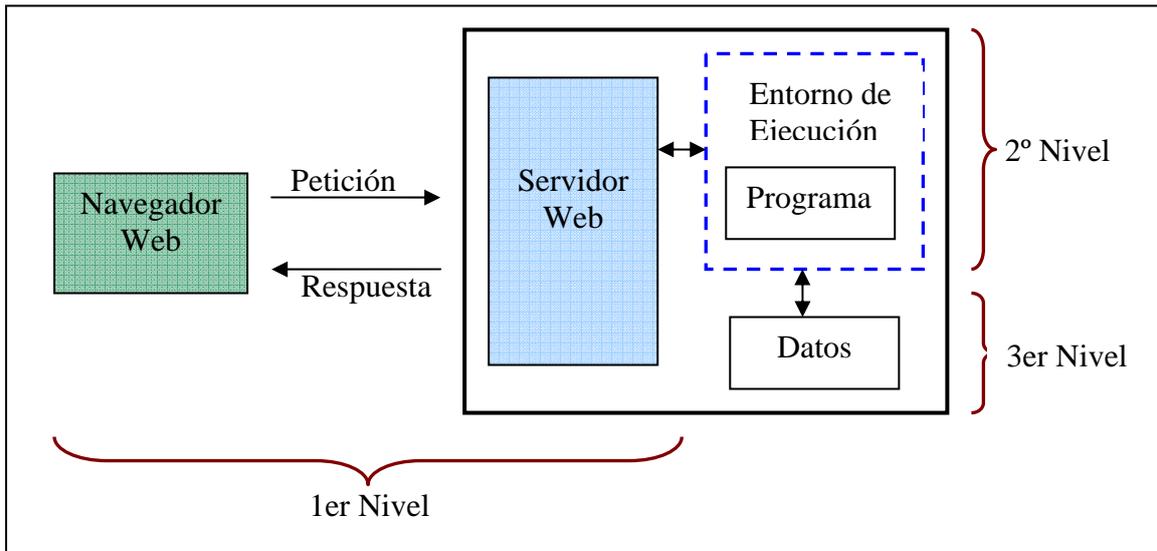


Figura 1.7 Arquitectura Web de tres niveles.

Pero convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola

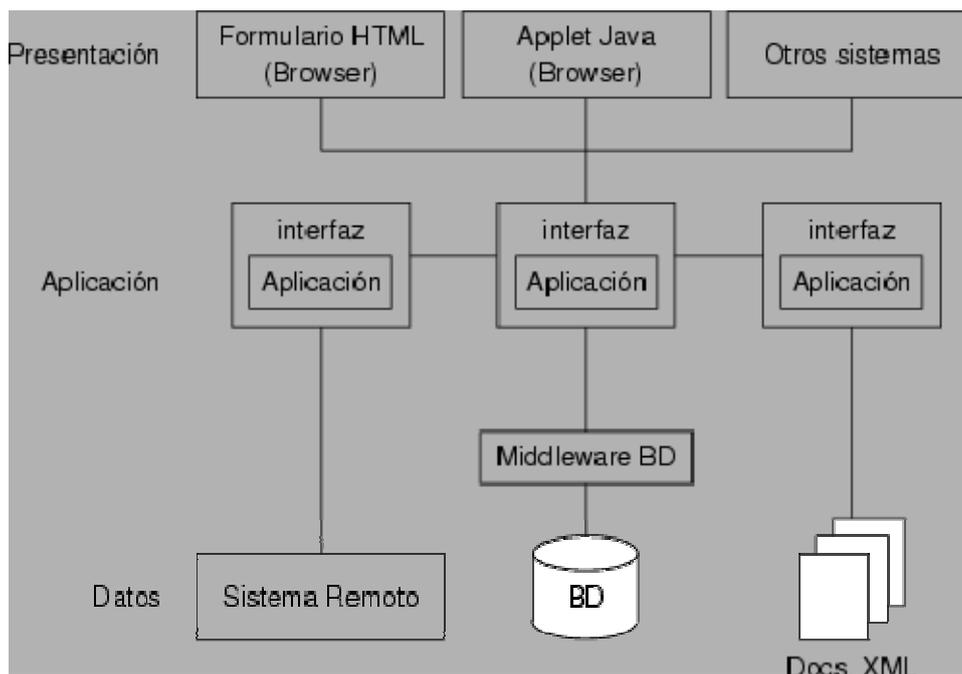


Figura 1.8: Arquitectura Multinivel.

Ventajas de una aplicación Web

Las aplicaciones Web ofrecen grandes ventajas que pueden ser aprovechadas por muchas organizaciones, sobre todo ahora que la globalización es una realidad. Entre las ventajas que se pueden mencionar están:

Implementación de Caso Práctico bajo Software Libre

- No requieren instalación, pues usan tecnología Web, lo cual nos permite el aprovechamiento de todas las características de Internet.
- Son fáciles de usar, no requieren conocimientos avanzados de computación.
- Alta disponibilidad, ya que puede realizar consultas en cualquier parte del mundo donde se tenga acceso a Internet y a cualquier hora.
- Acceso sin necesidad de distribución e, idealmente, con pocos requerimientos técnicos.
- Datos centralizados y fácil integración de datos de múltiples fuentes.
- Permiten el desarrollo de comunidades que dan valor a las aplicaciones (software social).

Con la introducción de Internet y de la Web en concreto, se han abierto infinidad de posibilidades en cuanto al acceso y uso de información desde cualquier parte del mundo. Con los avances en tecnología cada vez se demandan aplicaciones más rápidas, ligeras y robustas que permitan ser usadas sin importar el lugar u horario.

Debido al auge que ha tenido Internet en las últimas décadas, cada vez es mayor el número de empresas que necesitan contar con sistemas Web que se puedan acceder desde cualquier parte del mundo y a cualquier hora.

Tener una aplicación de este tipo ya no es un lujo de ningún negocio, sino mas bien una necesidad que día a día demandan los usuarios, con el fin de poder realizar cualquier tipo de transacción vía Internet; desde pagos, consultas, compras, etc. y todo de forma rápida y práctica, evitando en su mayoría los engorrosos trámites burocráticos.

Pues bien, pensando en esto, en un gimnasio que se encuentra en pleno crecimiento, se han hecho notoria la necesidad de un sistema en el cual se tenga centralizada la información que maneja dentro del gimnasio y que tanto dueños, instructores o los mismos clientes puedan tener acceso a ella y manejarla de acuerdo a sus perfiles y a lo que más convenga a la empresa.

Capítulo 2

Caso Práctico

Ejercicio:

Desventajas del proceso actual

Ventajas al contar con un sistema informático

Diagrama de Flujo del Proceso Actual

Metodología Usada

Caso Práctico

2.1 Ejercicio:

Una cadena de gimnasios requiere de un sistema que le ayude a su personal administrativo a realizar las actividades que actualmente se llevan a cabo de forma manual y monótona en el proceso de administración de socios y los servicios que éste puede realizar. Se pretende automatizar la captura de socios; así como las actividades que puede realizar un socio como son: reservar o cancelar clases, saber el estado de sus pagos, conocer las actividades extras que se planeen en el Gimnasio, inscribirse a torneos, etc.

El número de clientes ha crecido considerablemente de forma que llevar el control sobre ellos de forma manual se ha vuelto un caos, provocando inconsistencias en la información, esto se refleja en pérdidas financieras para el gimnasio, y en pérdidas de clientes, como consecuencia a que la atención para cada uno de ellos se hace cada vez mas deficiente ya que el personal que tiene contratado el gimnasio para atenderlos se ha vuelto insuficiente. Las principales características que se pueden observar en el control de la información son:

Característica	Consecuencia
Actualizada	La información que se maneje en el gimnasio debe ser real en todo momento sin que haya procesos manuales para obtenerla, esto es imposible si no se cuenta con un sistema informático para su control.
Conectividad con entidades externas	La información debe circular por fuera del gimnasio, es decir, cada socio puede consultar sus datos y los del gimnasio desde cualquier parte en la que tenga una conexión a Internet.
Relevante	Para poder proveer la cantidad exacta de información, se debe contar con sistemas que permitan tener análisis a diferentes niveles de detalle.
Alta velocidad y disponibilidad de acceso	El crecimiento de los negocios exige una alta velocidad en la información de la cual se puede proveer con conexiones permanentes en "línea" a las bases de datos.
Confiable	Qué tanto se puede creer en la información que se está recibiendo. Este factor se diseña mediante la implementación de procesamiento automático de información, establecimiento de seguridades a diferentes niveles, y la auditabilidad de las actividades, específicamente identificando quién hizo qué, cuando y desde donde.
Explicable	Se debe poder ver a todos los niveles de detalle el origen de toda información

No es difícil planificar estas características dentro de un sistema de información si se contemplan desde el inicio.

A continuación se detalla el proceso de inscripción que actualmente se lleva a cabo en el gimnasio.

La inscripción la realiza un cliente directamente en el gimnasio de acuerdo a lo siguiente:

1. El cliente llena una solicitud anotando en ella sus datos personales:
 - a. Nombre Completo
 - b. Dirección
 - c. Teléfono Particular
 - d. Celular
 - e. Fecha de Nacimiento
 - f. Sexo

Implementación de Caso Práctico bajo Software Libre

2. El cliente realiza su primer pago que consta de la inscripción y su primera mensualidad.
3. Una vez hecho esto, el personal administrativo llena un tarjetón con los datos del nuevo socio y con los pagos realizados. Este tarjetón es el control que se lleva en el gimnasio por cada socio; en él se anotan las visitas que cada cliente realizó durante el mes y llevan el control de sus pagos.

Este proceso se quiere automatizar de acuerdo a lo siguiente:

1. Si el cliente cuenta con una conexión a Internet, él podrá realizar su preinscripción a través del llenado de un formulario.
 - a. El sistema asignará de acuerdo a los datos introducidos, el correspondiente código de barras al cliente; además se le asignará un número de prerregistro.
2. Si el cliente desea realizar su inscripción directamente con el personal administrativo, deberá presentarse con él, para que le tomen sus datos.
 - a. Una vez llenado el formulario de inscripción, se asignará el código de barras y una clave única de 4 dígitos al cliente; con esta clave el socio podrá tener acceso al sistema vía Web
3. Se toma la foto al cliente.
4. El cliente tiene que realizar el pago correspondiente a la inscripción y a la primera mensualidad directamente con el personal administrativo.
5. Se imprime la credencial con los siguientes datos:
 - a. Nombre Completo
 - b. Foto
 - c. Código de barras

Una vez que el cliente terminó el proceso de inscripción, se le considera como un socio activo, con el derecho de acceder a los servicios vía Web que el Gimnasio proporcione.

Además de ser un centro de acondicionamiento físico, en el Gimnasio se imparten clases de algunas actividades como son: natación, tae kwon do, gimnasia preolímpica, jazz, ballet, tahitiano, etc. Para poder tomar alguna de estas actividades, es requisito que el socio las reserve ya sea vía Internet o en las instalaciones hasta con 15 minutos de anticipación con el personal administrativo.

En caso de que el socio no asista a más de 5 clases reservadas en un mes, no tendrá derecho a hacer reservaciones durante el siguiente mes.

En el gimnasio se pueden llegar a organizar actividades extras para los socios, entre las que pueden estar: conferencias, exhibiciones, torneos, etc. Para que un socio pueda participar en alguna de ellas, deberá confirmar su asistencia con la anticipación necesaria para cada uno de los eventos, Estos eventos son publicados en las instalaciones del gimnasio y se pretende que se publiquen en el sitio que será creado para que desde ahí un socio pueda confirmar su asistencia y participación en el evento.

Las actividades mencionadas, serán registradas por el personal administrativo, y una vez que concluyan dichas, serán dadas de baja automáticamente.

La siguiente figura muestra un resumen de los procesos que se pueden llevar a cabo en el gimnasio.

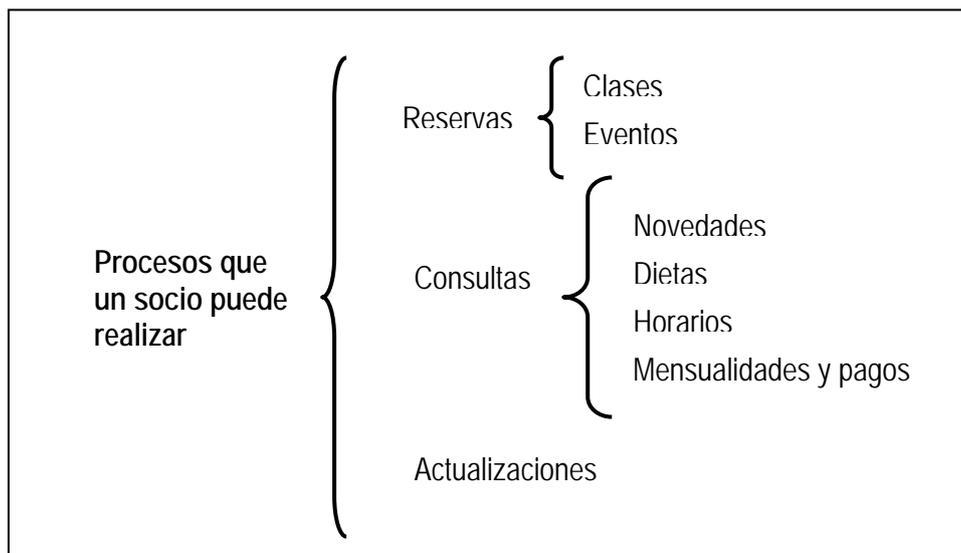


Figura 2.1 Procesos de socio.

Cabe mencionar que todas estas actividades, se pueden realizar a través del personal administrativo en las instalaciones del gimnasio.

2.2 Desventajas del proceso actual

Entre las desventajas que se tienen, al no implementar un sistema para automatizar los procesos se encuentran:

- Existe mucha fuga de información y por lo tanto, no se puede llevar a cabo una buena administración de los recursos.
- No se tiene un control exacto sobre el número de socios con que cuenta el gimnasio, y mucho menos las actividades y clases que son mas solicitadas, los horarios y fechas en los que hay mas demanda, etc.
- Existe mucho descontento por parte de los socios, en cuando al orden que se lleva a cabo la administración, lo que provoca falta de clientes, y por lo tanto, falta de ingresos.
- No se lleva un seguimiento de los socios en cuanto a asistencias, puntualidad de pagos, etc.
- No se lleva a cabo un mecanismo que controle la disponibilidad en las actividades que se realizan.
- No se tiene el control sobre los ingresos.
- Es difícil tomar decisiones en cuanto al número de personal requerido (temporadas u horarios), ya que no se tienen datos confiables sobre las necesidades del gimnasio.
- Es necesario acudir al gimnasio para inscribirse.
- Es necesario acudir al gimnasio para conocer instalaciones, horarios, actividades e información en cuanto al gimnasio se refiera.
- El socio tiene que estar preguntando sus fechas de vencimiento, ya que no existe ningún mecanismo que se lo haga saber.
- El personal tiene que hacer una búsqueda manual entre los tarjetones de todos los socios para saber algún tipo de información del socio; trabajo que se vuelve tedioso, propenso a errores y fastidioso para los socios.
- Resulta complicado para el personal administrativo llevar el control sobre todos los tarjetones de los socios.

Implementación de Caso Práctico bajo Software Libre

- Pérdida de tiempo para los dueños, al realizar un proceso monótono para obtener un análisis de ingresos y socios.

2.3 Ventajas al contar con un sistema informático

Todos los problemas mencionados anteriormente se pueden evitar implementando el sistema; además de que se obtendrán otras ventajas que se mencionan a continuación:

- Se llevará un control exacto sobre el número de socios con que cuenta el gimnasio, así como las actividades y clases que son mas solicitadas, los horarios y fechas en los que hay mas demanda, y sobre el personal requerido.
- Se llevará a cabo un control sobre la disponibilidad de las clases y eventos impartidos en el gimnasio, lo que provocará que no haya sobresaturación de gente en las diferentes áreas; esto se realizará a través de las reservaciones para cada una de las actividades.
- Tanto socios, personal administrativo y dueños podrán revisar de forma automática el estado de los socios y el seguimiento que estos han tenido durante su estancia en el gimnasio.
- El sistema hará que se tenga control sobre los ingresos que se tienen en el gimnasio, y servirá para hacer estadísticas de los posibles rendimientos que se tendrán en un periodo de tiempo.
- Los dueños podrán tomar decisiones en cuanto al personal requerido, ya que sabrán, en que horarios y en que épocas se necesita de mas gente para satisfacer las necesidades del cliente.
- El socio no tendrá que hacer el engorroso proceso de inscripción en las instalaciones del gimnasio, tomando en cuenta la pérdida de tiempo que esto significa.
- El socio podrá conocer las actividades impartidas en el gimnasio, así como costos e información que quiera obtener acerca del sistema, desde cualquier lugar en el que pueda obtener una conexión a Internet.
- El socio podrá saber sus fechas de pago, así como su estatus en el gimnasio; es decir, podrá ver las clases que tenga reservadas, algunas observaciones que se le tengan que hacer por parte del gimnasio, etc.
- Todas las búsquedas que el personal administrativo hace con los tarjetones, se eliminarán por completo, ya que éstos no existirán, las búsquedas de información ahorrarán tiempo, ya que éstas se harán de forma automática, de acuerdo a las necesidades del personal.
- No habrá fuga de información.

2.4 Diagrama de Flujo del Proceso Actual

El siguiente diagrama muestra el proceso actual del sistema, como se puede observar, no hay un control en la información que se maneja, ya que todo es a través del tarjetón de cada socio, lo que implica que al perder un tarjetón se pierda toda la información del cliente.

Implementación de Caso Práctico bajo Software Libre

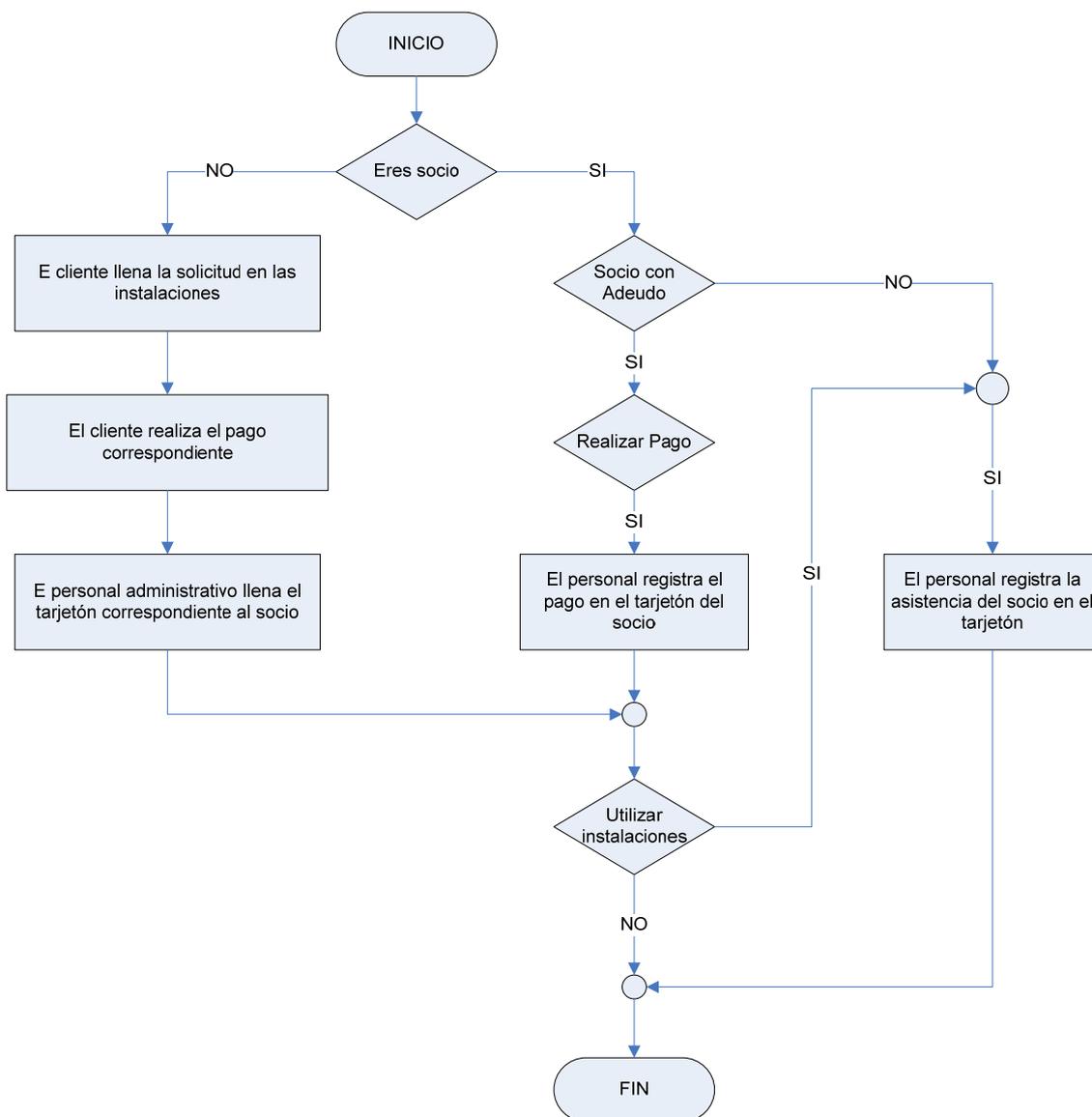


Figura 2.2: Diagrama de Flujo del Proceso Actual del Sistema

2.5 Metodología Usada

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo.

Muchas veces los usuarios finales, se dan cuenta de las cosas que dejaron de mencionar, recién en la etapa final del proyecto, pese a que se les mostró un prototipo del software en la etapa inicial del proyecto.

Los proyectos en problemas son los que salen del presupuesto, tienen importantes retrasos, o simplemente no cumplen con las expectativas del cliente.

La metodología utilizada para este proyecto es Racional Unified Process (RUP); de acuerdo a las necesidades y condiciones para el desarrollo del proyecto, ésta es la que más se adecua para tener un buen desarrollo. A continuación se detalla un poco esta metodología.

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

- ✓ **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto. En esta etapa se describe el objetivo, propósito y oportunidad de negocio que generará el proyecto; señala la situación actual y los beneficios que generará la implementación del proyecto.
- ✓ **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima. Una vez que se hizo el análisis de requerimientos del sistema, se debe estimar el esfuerzo y costo de los productos de trabajo que se van a desarrollar. Identificar
- ✓ **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- ✓ **Transición:** El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

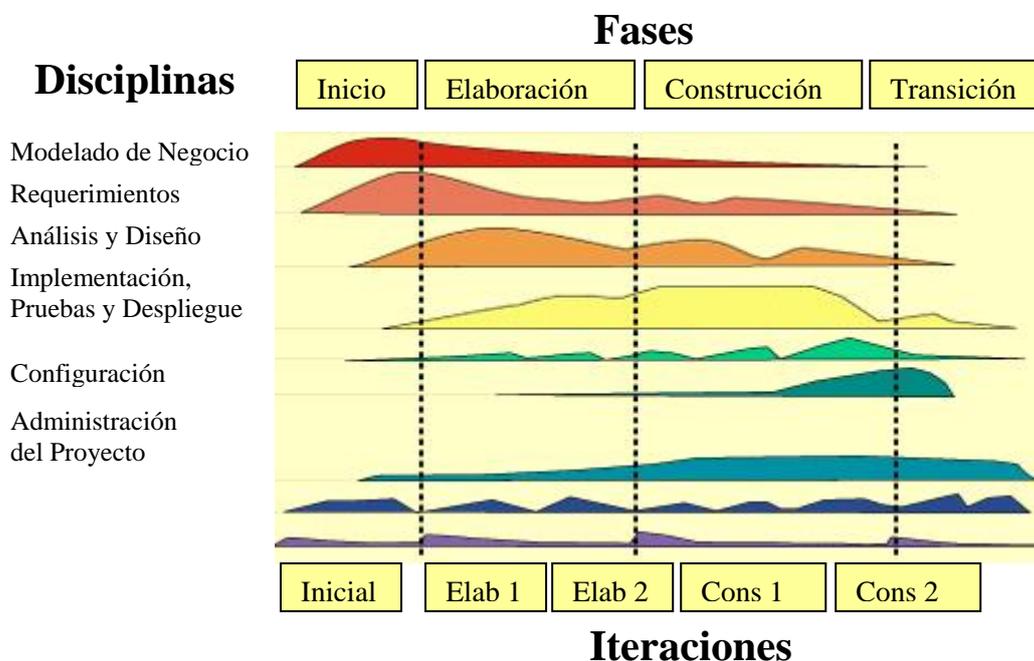


Figura 2.3 Fases e Iteraciones de la Metodología RUP

Entregables del proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

Implementación de Caso Práctico bajo Software Libre

1. Plan de Desarrollo del Software. Es el presente documento.
2. Modelo de Casos de Uso del Negocio. Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.). permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito.
3. Modelo de Objetos del Negocio. Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio.
4. Glosario. Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.
5. Modelo de Casos de Uso. El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas.
6. Visión. Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto.
7. Prototipos de Interfaces de Usuario. Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema.
8. Modelo de Análisis y Diseño. Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis hacia una de diseño, de acuerdo al avance del proyecto.
9. Modelo de Datos. Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos.
10. Modelo de Implementación. Este modelo es una colección de componentes y los subsistemas que los contienen.
11. Modelo de Despliegue. Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.
12. Casos de Prueba. Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Solicitud de Cambio. Los cambios propuestos para los artefactos se formalizan mediante este documento. Mediante este documento se hace un seguimiento de los defectos detectados, solicitud de mejoras o cambios en los requisitos del producto.
13. Plan de Iteración. Es un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.
14. Evaluación de Iteración. Este documento incluye la evaluación de los resultados de cada iteración, el grado en el cual se han conseguido los objetivos de la iteración, las lecciones aprendidas y los cambios a ser realizados.
15. Lista de Riesgos. Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.
16. Manual de Instalación. Este documento incluye las instrucciones para realizar la instalación del producto.
17. Material de Apoyo al Usuario Final. Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación, Guías de Mantenimiento y Sistema de Ayuda en Línea
18. Producto. Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación.

Implementación de Caso Práctico bajo Software Libre

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos del RUP son:

- ✓ **Actividades:** Son los procesos que se llegan a determinar en cada iteración.
- ✓ **Trabajadores:** Vienen hacer las personas o entes involucrados en cada proceso.
- ✓ **Artefactos:** Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

El proceso de planeación de proyectos describo cómo se realiza éste en el área de sistemas; incluye actividades tales como estimación de atributos de los productos de trabajo y sus tareas, determinar los recursos necesarios, negociar compromisos, producir plan de actividades e identificar y analizar riesgos.

Una vez que se han hecho las estimaciones de tamaño, esfuerzo y costo del proyecto hay que definir el plan de proyecto con esta información como un documento formal y aprobado, el cual es usado para administrar y controlar el proyecto.

Todas estas tareas se irán describiendo a lo largo de este documento; pero a continuación se muestra una tabla en la que aparecen las herramientas que se utilizaran en cada etapa.

Etapa	Herramientas
Modelado de Negocio	DB Designer,
Requerimientos	Poseidon
Análisis y Diseño	Poseidon
Implementación, Pruebas y Despliegue	Eclipse, Tomcat, MySQL
Administración de Proyecto	MindManager

Implementación de Caso Práctico bajo Software Libre

Gráfica de Gantt

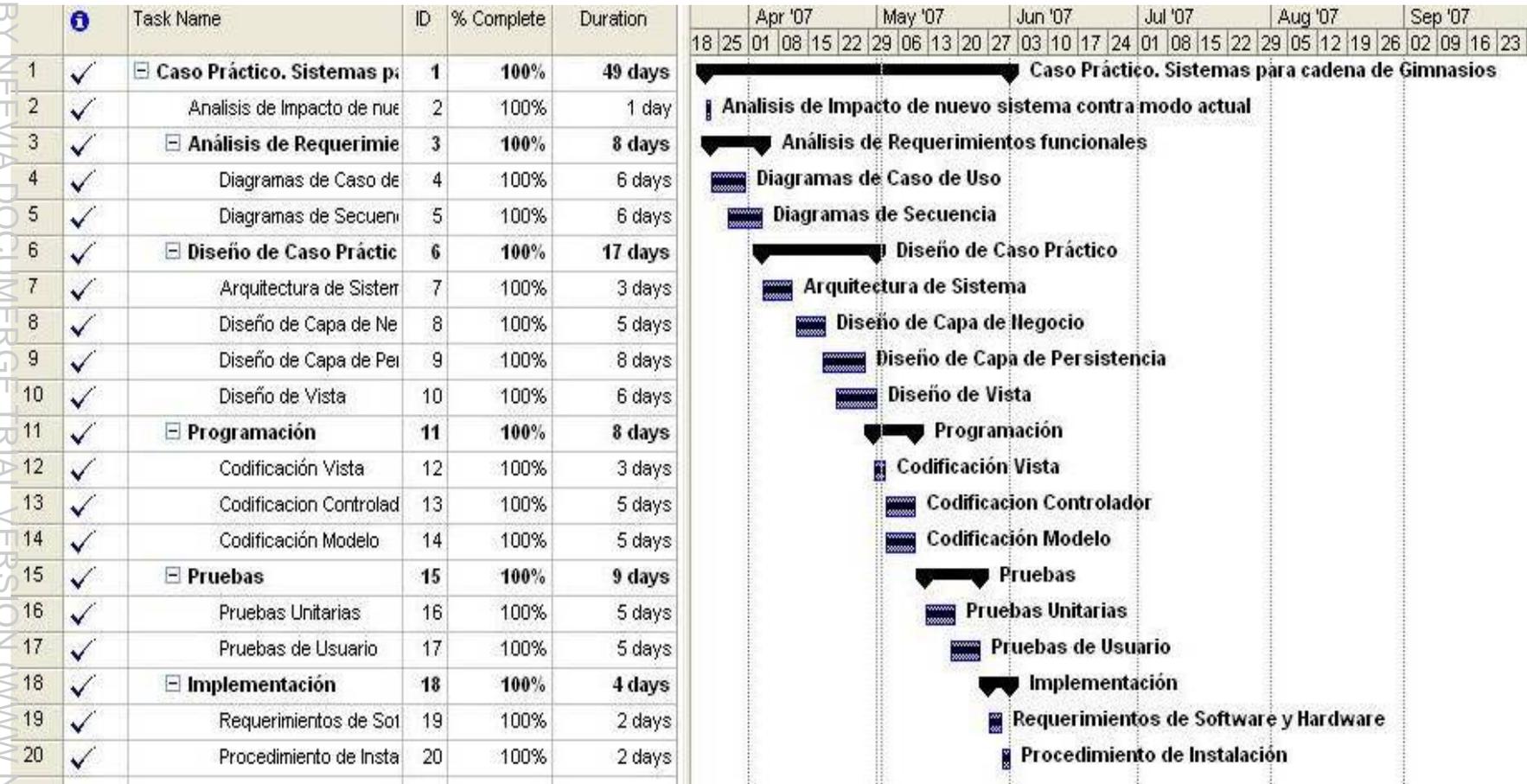


Figura 2.5. Gráfica de Gant



Capítulo 3

Análisis de Caso Práctico

Requerimientos Funcionales

Propósito del Sistema

Alcance

Diagramas de Casos de Uso

Diagramas de Secuencia

Análisis de Caso Práctico

3.1 Requerimientos Funcionales

En el Gimnasio “Beverly Hills” se tiene la necesidad de contar con un sistema que contemple todas las operaciones que se realizan en él de forma manual y monótona.

El sistema debe cubrir los procesos generales que se realizan en el Gimnasio en cuanto a un cliente/socio se refiere:

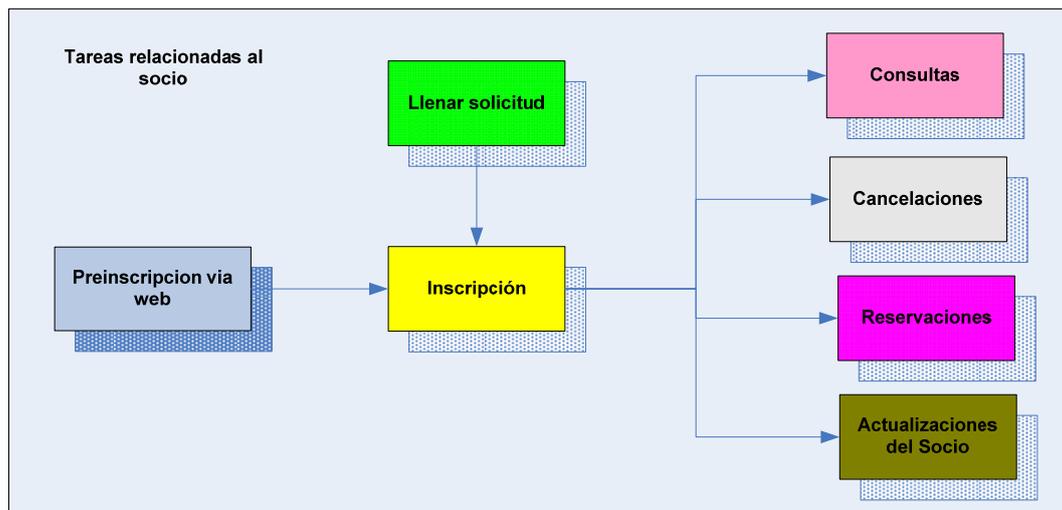


Figura 3.1 Diagrama de Procesos del Sistema relacionados con el socio

Inscripción: Es el inicio del proceso para reconocer a un cliente como socio activo del Gimnasio. Su propósito es establecer si un cliente cumple con los requisitos para poder formar parte del Gimnasio, y en cuyo caso, reconocerlo como socio activo; para que pueda realizar todas las actividades a las que un socio tiene derecho.

Reservaciones: En el proceso de reservaciones, los socios pueden apartar tanto clases como las actividades extemporáneas que se organicen; tales actividades pueden ser: torneos, conferencias, eventos de celebraciones del Gimnasio, etc.

Consultas: Es importante que un socio pueda saber en cualquier momento su situación con respecto al Gimnasio, sus fechas de pago, sus actividades reservadas, y algunas indicaciones que pueda tener los instructores para cada uno de los socios. Todo esto es parte del módulo de consultas.

Actualización de información del cliente: En el gimnasio siempre se debe contar con los datos actualizados de los socios; por lo que se debe contar con un proceso en el cual los socios puedan cambiar su información personal, datos como son: teléfono, domicilio, etc. También se deben poder actualizar todas las actividades que se realicen en el gimnasio: Horarios, torneos, eventos.

Hay procesos que el administrador tiene que realizar antes de que el socio pueda tener accesos a ellos. Por ejemplo, para que un socio pueda reservar o consultar clases, éstas deben estar dadas de altas, y este proceso de alta así como otros, los debe de realizar el administrador.

Implementación de Caso Práctico bajo Software Libre

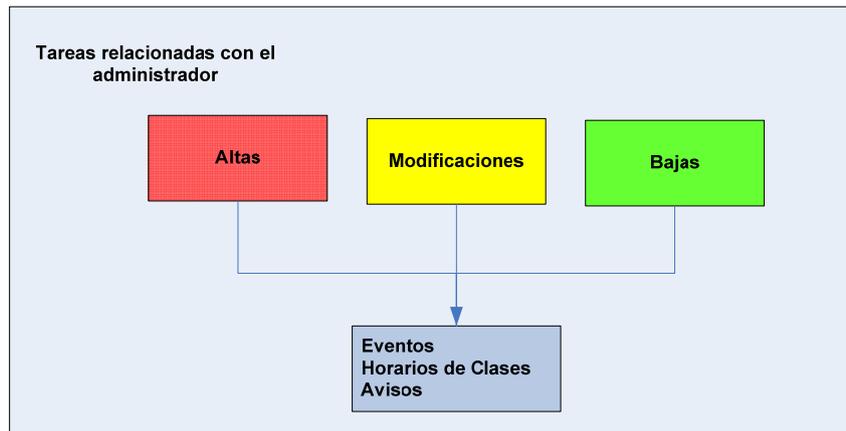


Figura 3.2 Diagrama de Procesos del Sistema relacionados con el administrador

Altas, Modificaciones y Bajas: Todas las actividades que se realizan en el Gimnasio deben estar disponibles en el sistema para que se puedan hacer las acciones correspondientes con ellas (consultas, reservaciones y actualizaciones).

Cancelaciones. Todos los socios tienen derecho a hacer cancelaciones de las actividades que se tengan reservadas.

Procesos Generales

Diagrama de flujo del proceso general de las actividades del socio.

Implementación de Caso Práctico bajo Software Libre

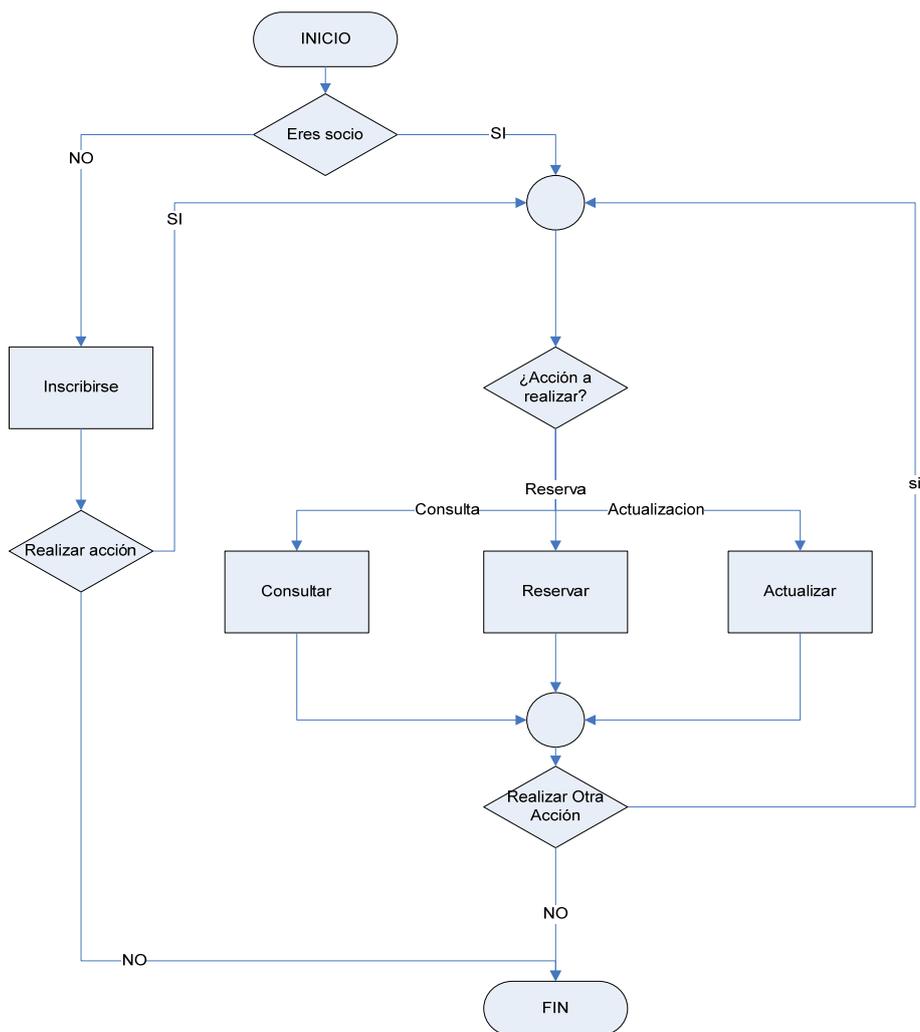


Figura 3.3 Diagrama de flujo del proceso general de las actividades del socio.

Implementación de Caso Práctico bajo Software Libre

Diagrama de flujo del proceso general de las actividades del socio.

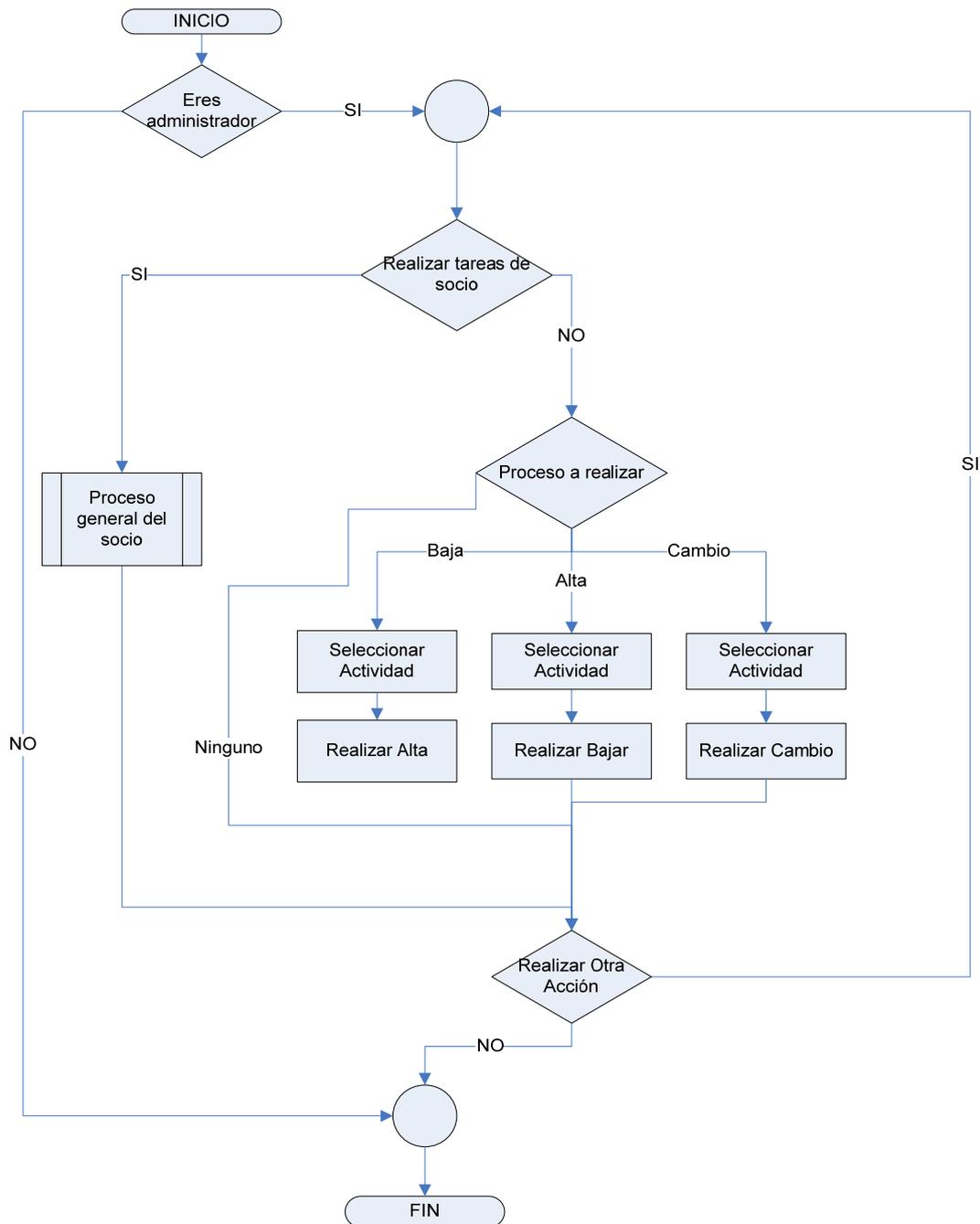


Figura 3.4 Diagrama de flujo del proceso general de las actividades del administrador.

3.2 Propósito del Sistema

Definir el requerimiento de automatización de los procesos realizados en el Gimnasio, especificado en términos de las características y necesidades claves de los usuarios, conteniendo el entorno y mejoras de los requisitos solicitados y dando las bases para los requerimientos técnicos detallados.

3.3 Alcance

Se contempla que con el sistema se pueda realizar lo siguiente:

3.3.1 Inscripción.

El proceso de inscripción se puede realizar de dos formas

a) Inscripción vía Web.

Un cliente puede realizar su proceso de inscripción vía Web si cuenta con una conexión a Internet. Para ello, debe entrar a la página del sistema y buscar el vínculo de Inscripción. Aquí aparecerá un formulario que el cliente debe de llenar para comenzar su proceso de inscripción. Cuando finalice con el llenado del formulario, el sistema hará las validaciones correspondientes a cada uno de los campos obtenidos, dichas validaciones se explican con más detalle posteriormente.

Si todos los datos son correctos el sistema dará al cliente un número de preinscripción para que termine con el proceso en las instalaciones del gimnasio y el monto a pagar por la inscripción y la primera mensualidad. El pago se hace directamente en la administración del gimnasio; este número de preinscripción solo será válido por una semana; de modo que si el cliente no termina el proceso de inscripción en una semana, sus prerregistro se perderá.

El personal administrativo busca el número de preinscripción del cliente para continuar con el proceso, y el sistema indica si la clave generada por el sistema ya esta repetida, en este caso, el personal del gimnasio verifica si se trata o no de la misma persona que ya estaba inscrita anteriormente y perdió su vigencia; en caso de que sea así lo único que se hace es volver a activar la cuenta del cliente una vez que este hace el pago. En caso de que no se trate de la misma persona, el personal del gimnasio podrá escoger otra clave para el cliente, recorriendo las letras del apellido para que quede guardada otra clave diferente; se recuerda que esta clave es la clave de usuario para identificarse en el sistema por lo que debe de ser única.

b) Inscripción dentro de las instalaciones.

Cuando el cliente realiza su inscripción en las instalaciones, tiene que pedir al encargado del gimnasio una solicitud y llenar los datos que se piden en esta. El cliente entrega la solicitud al personal del gimnasio y éste lo da de alta en el sistema.

Al dar de alta al usuario, la clave de identificación del socio se creará.

Los datos que se piden tanto en el formulario vía Web como en la solicitud son:

- ✓ Apellido Paterno
- ✓ Apellido Materno
- ✓ Nombre (s)
- ✓ Teléfono Particular
- ✓ Teléfono de emergencia
- ✓ Celular
- ✓ Fecha de Nacimiento
- ✓ Correo electrónico
- ✓ Sexo
- ✓ Dirección: Calle, número, colonia, código postal, estado.
- ✓ Enfermedades, alergias, tipo de sangre.

Implementación de Caso Práctico bajo Software Libre

A partir de este momento, el proceso es el mismo para ambos casos.

El personal del gimnasio, hace el cobro al cliente correspondiente a la inscripción y a la primera mensualidad.

Se le da al socio una clave de usuario para tener acceso al sistema vía Web, la cual consta de su primer nombre, la primera letra de su apellido paterno y su fecha de nacimiento. En caso de que esta clave este repetida, se verifica que no sea el mismo socio; si no es así, el administrador puede asignar la clave siguiendo la base para obtener la clave. Junto con la clave se entrega una contraseña provisional que podrá cambiar en el sitio Web cuando lo desee. Esta clave cuenta con 5 letras

Se le toma la foto y se imprime la credencial con la foto, código de barras, y el nombre del socio.

Una vez terminado el proceso de inscripción, el cliente pasa a ser socio activo del gimnasio.

Implementación de Caso Práctico bajo Software Libre

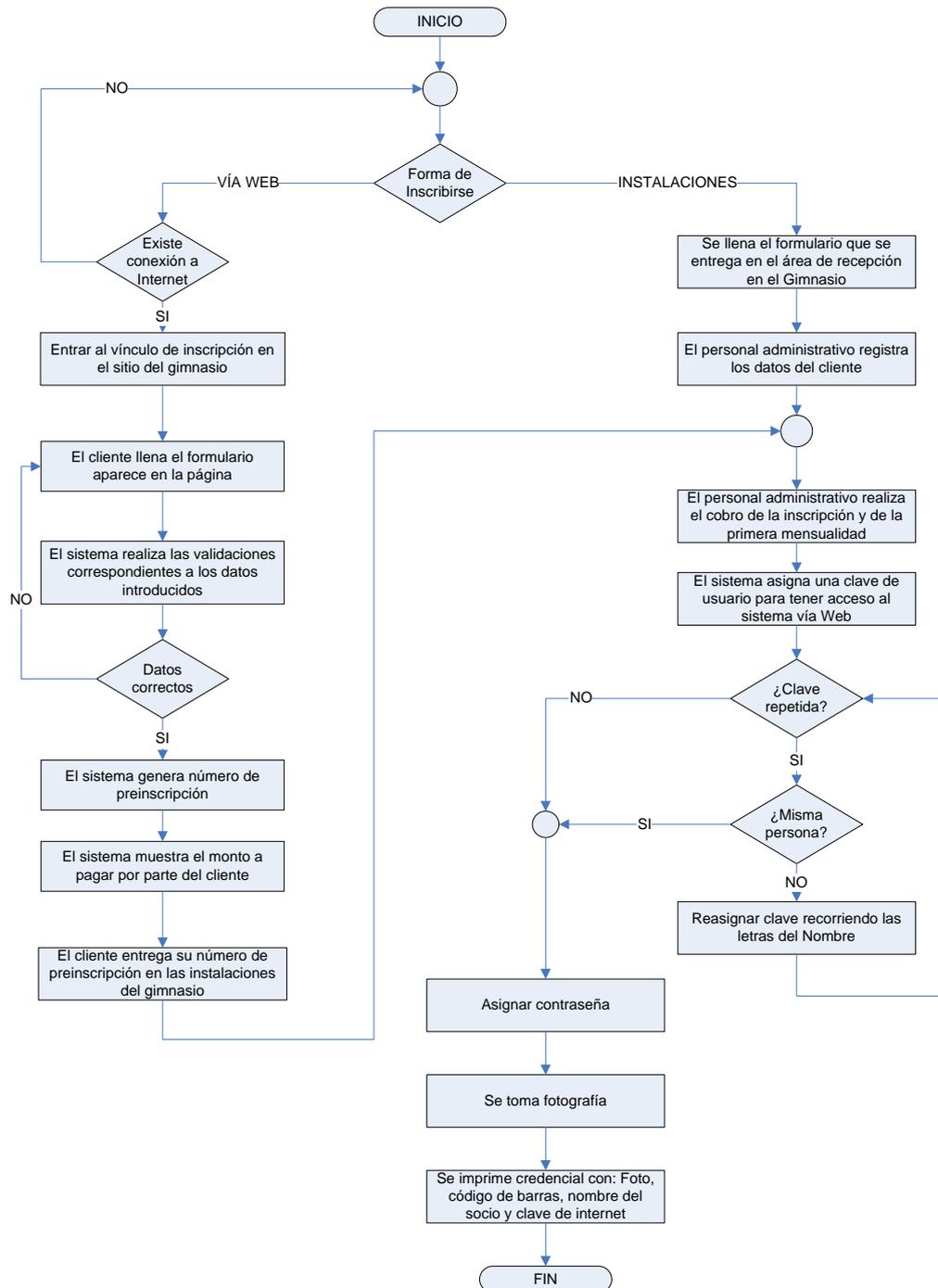


Figura 3.5 Diagrama de flujo del proceso de inscripción

3.3.2 Actualización de información del socio

Un cliente puede modificar cierta información que en el gimnasio se maneja sobre él, tal información puede ser dirección, teléfono, dirección de correo electrónico y contraseña para el acceso al sitio.

Este proceso si se puede realizar de dos formas: vía Internet y en las instalaciones del gimnasio, por lo regular.

Implementación de Caso Práctico bajo Software Libre

a) Vía Internet.

Un socio puede hacer estas modificaciones a través de Internet, para lo que necesita la clave de usuario y la contraseña. Cuando el socio entra al sitio Web del gimnasio se le pide su clave de usuario y la contraseña; el sistema hace las validaciones correspondientes a estos datos y si son correctos y además el socio está al corriente en sus pagos, se da el acceso al usuario al menú de socios, y entre sus opciones esta la de Información de Usuario; cuando se selecciona este menú aparece toda la información del usuario y los campos que pueden ser modificados aparecen habilitados para su captura.

El socio hace las modificaciones necesarias y guarda los cambios, el sistema hace las validaciones correspondientes a estos campos, y si hay errores en los campos regresa a la pantalla enviando el mensaje de error correspondiente. Si no existen errores, se guarda la información y se muestra al socio la pantalla de menú de socio.

Para el caso del cambio de contraseña, se le pedirá al socio su contraseña actual para realizar el cambio, si la contraseña es correcta, se le pedirá que digite la nueva contraseña y que la verifique, si las dos contraseñas coinciden, el proceso se completa y al socio se le pedirá la nueva contraseña en su próxima visita al sitio.

b) En las Instalaciones con el personal.

Para este caso, el proceso es muy parecido al anterior, El socio tiene que dar la clave de usuario al personal administrativo para que éste pueda hacer los cambios necesarios a la información perteneciente al socio.

El personal administrativo, entra al sistema con su clave y con su contraseña, aparece el menú con las opciones para el usuario administrativo y entra al menú de actualización de información. Se pide la clave de usuario del socio; si el socio esta al corriente en sus pagos, aparece toda la información del usuario y los campos que pueden ser modificados aparecen habilitados para su captura.

El socio hace las modificaciones necesarias y guarda los cambios, el sistema hace las validaciones correspondientes a estos campos, y si hay errores en los campos regresa a la pantalla de datos enviando el mensaje de error correspondiente. Si no existen errores, se guarda la información y se muestra al socio la pantalla de menú del usuario administrativo.

Implementación de Caso Práctico bajo Software Libre

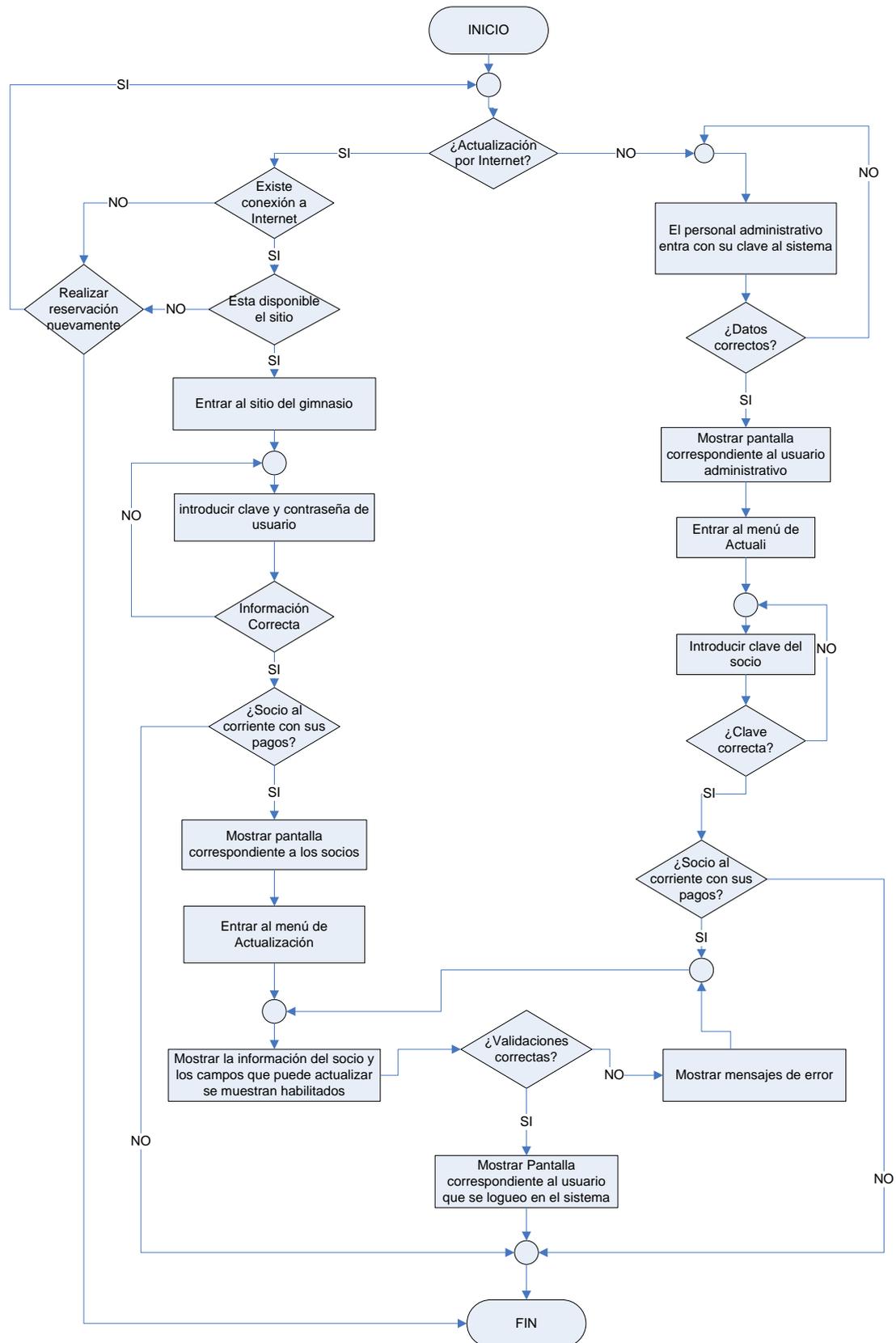


Figura 3.6 Diagrama de flujo del proceso de actualización de información de los socios

3.3.3 Reservaciones.

Todas las actividades que se realizan en el gimnasio requieren hacer una reservación, excepto cuando se va a asistir al área de pesas. Las reservaciones se pueden hacer por un día o por un periodo, es decir si un usuario va a ir todo el mes a la misma hora a una actividad puede hacer la reservación para todo el mes; esto se hace con el fin de que no se saturen los espacios asignados para cada actividad y el socio tenga un servicio de buena calidad.

Una reservación se puede hacer, al igual que la inscripción de dos formas:

a) Vía Internet.

Un socio puede hacer reservaciones para las clases que se imparten cotidianamente en el gimnasio a través de Internet; para esto sólo se necesita la clave de usuario y la contraseña. Cuando el socio entra al sitio Web del gimnasio se le pide su clave de usuario y la contraseña; el sistema hace las validaciones correspondientes a estos datos y si son correctos y además el socio está al corriente en sus pagos, se da el acceso al usuario al menú de socios, y entre sus opciones esta Reservas; cuando se selecciona este menú aparece un submenú con las opciones de horarios y de los eventos extemporáneos. Dependiendo de la opción a escoger será la pantalla que se muestre.

Cuando se selecciona la opción de horarios, se muestran todas las actividades que existen en el gimnasio, sus horarios, instructores, áreas en donde se imparte y todos los días del periodo en el que el socio puede hacer reservaciones; el socio puede activar varias opciones siempre y cuando no tenga ya el horario ocupado para otra actividad.

Cuando el socio selecciona la opción de actividades extemporáneas, se mostraran todas las opciones que existan para ellas; dependiendo de las necesidades de cada una. El socio podrá seleccionar las actividades para las cuales quiera hacer reservaciones.

Cuando el socio entre otra vez a cualquiera de estas opciones aparecerán las actividades que tiene reservadas y siempre podrá activar o desactivarlas como a él le convenga; de este modo puede cancelar su reservación.

b) En las Instalaciones con el personal.

Para este caso, el proceso es muy parecido al anterior, El socio tiene que dar la clave de usuario al personal administrativo para que éste pueda hacer la reservación a nombre del socio.

El personal administrativo, entra al sistema con su clave y con su contraseña, aparece el menú con las opciones para este usuario y entra a reservaciones. Se pide la clave de usuario del socio; si el socio esta al corriente en sus pagos, aparecen los submenús para reservaciones; el personal selecciona las opciones que el socio pida y al igual que en el caso anterior, cuando se vuelva a entrar al sistema con la misma clave de usuario, se podrán activar o desactivar las opciones según sea el caso.

El sistema muestra solo las actividades en las que hay disponibilidad y las que están en tiempo de reservación.

Implementación de Caso Práctico bajo Software Libre

Cuando el socio terminó de hacer sus reservaciones, el sistema guarda la información capturada en una base de datos, para que posteriormente se le pueda consultar y se permitan los accesos al socio.

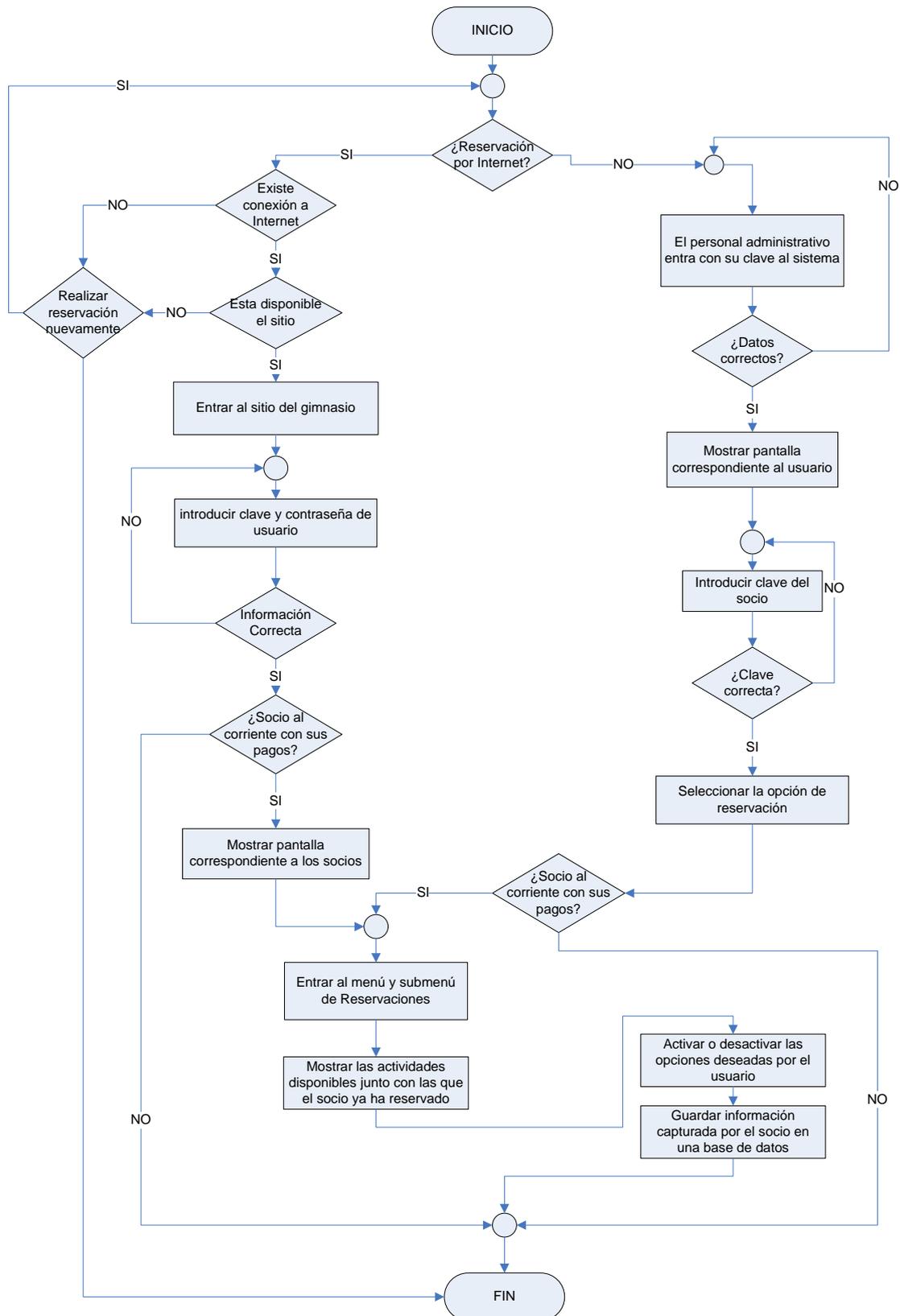


Figura 3.7 Diagrama de flujo del proceso de reservas

3.3.4 Cancelaciones.

Todas las actividades que se hayan sido reservadas por algún socio, pueden ser canceladas, hasta 15 minutos antes de la actividad.

Al igual que las reservas, éstas se pueden realizar de 2 formas:

Una cancelación se puede hacer, al igual que la inscripción de dos formas:

a) Vía Internet.

Un socio puede hacer cancelar sus clases reservadas a través de Internet; para esto sólo se necesita la clave de usuario y la contraseña. Cuando el socio entra al sitio Web del gimnasio se le pide su clave de usuario y la contraseña; el sistema hace las validaciones correspondientes a estos datos y si son correctos y además el socio está al corriente en sus pagos, se da el acceso al usuario al menú de socios, y entre sus opciones esta Reservas; cuando se selecciona este menú aparece una lista de actividades que el socio tiene reservadas, en la cual puede escoger cuales cancelar.

Las actividades que estén a 30 minutos de empezar, ya no aparecerán en la lista de actividades reservadas.

b) En las Instalaciones con el personal.

El socio tiene que dar la clave de usuario al personal administrativo para que éste pueda hacer la reservación a nombre del socio.

El personal administrativo, entra al sistema con su clave y con su contraseña, aparece el menú con las opciones para este usuario y entra a cancelaciones. Se pide la clave de usuario del socio; si el socio esta al corriente en sus pagos, aparecen los la lista de actividades que el socio tiene reservadas. El personal selecciona las actividades que el personal quiera cancelar

Implementación de Caso Práctico bajo Software Libre

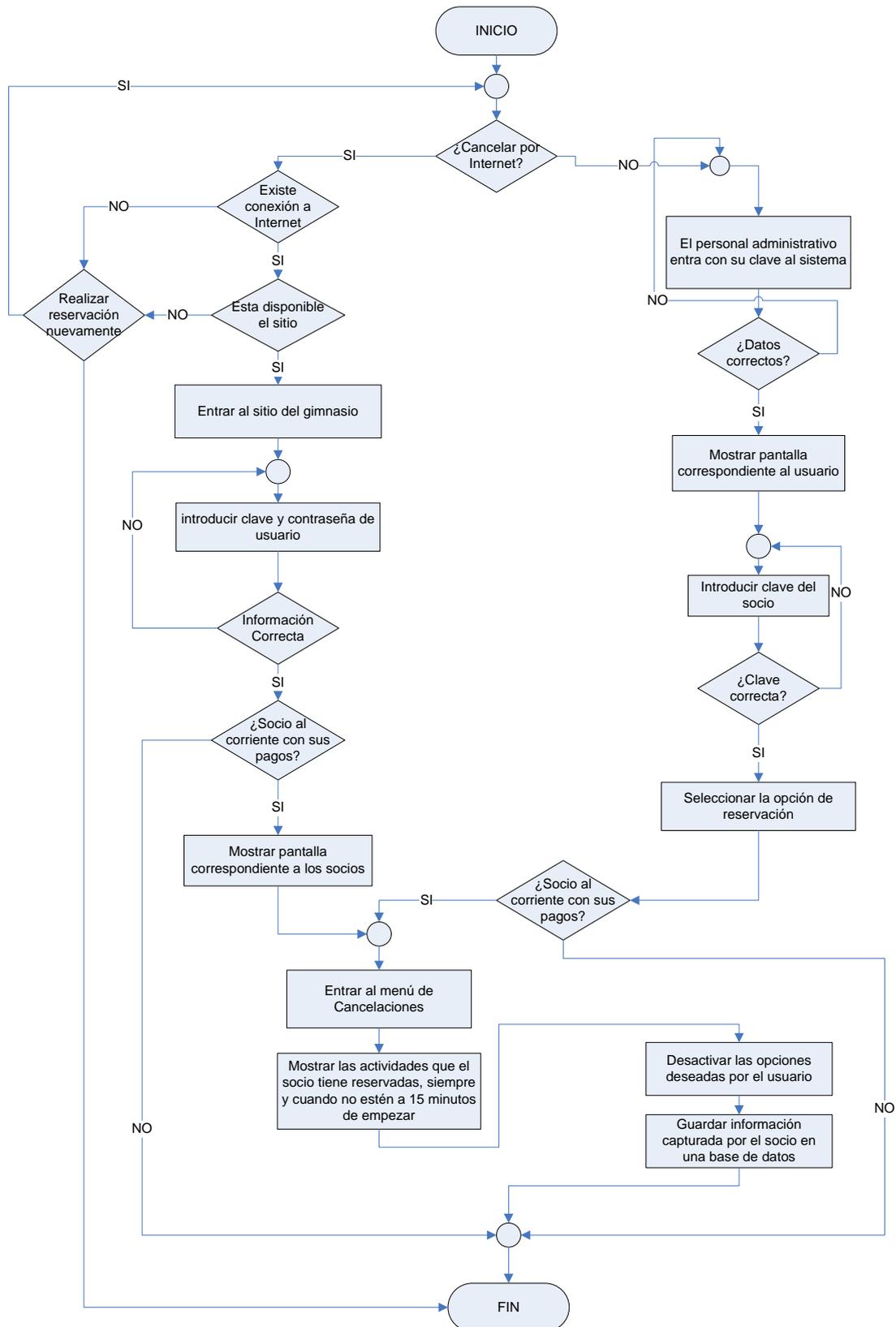


Figura 3.8 Diagrama de flujo del proceso de reservas

3.3.5 Asistencias.

El proceso del registro de asistencias sólo se lleva a cabo en las instalaciones del gimnasio, y esto se realiza de la siguiente forma:

Cada una de los salones o áreas existentes para realizar actividades tiene su lector de código de barras de modo que el socio pueda registrar sus asistencias a través de él y le pueda ser permitido el acceso a las clases reservadas.

Para que un socio pueda tomar alguna clase, tuvo que haber realizado su reservación previamente.

El socio llega al área donde se imparte la actividad que reservó y en la entrada tiene que deslizar su credencial por el lector de código de barras. Si el código de barras es correcto y tiene una reservación realizada, se verifica la hora de entrada, si todavía no pasa de la hora marcada para empezar la clase, se toma la asistencia del socio y se le permite el acceso a la clase. Si sobrepasa la hora de entrada, por ningún motivo se permite la entrada al socio, aunque para estadísticas, se marcara como retardo en el historial del socio.

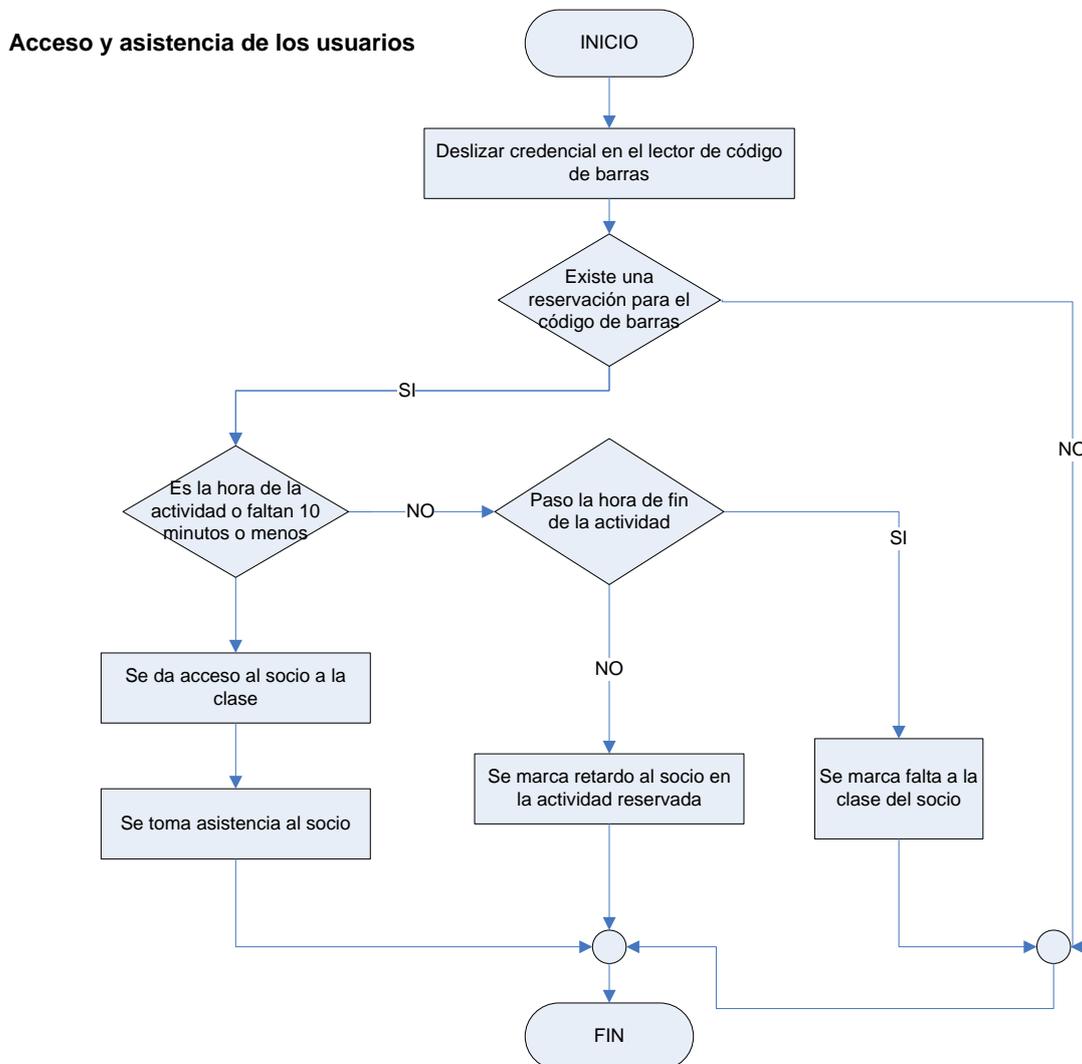


Figura 3.9 Diagrama de flujo del proceso de asistencia de los socios

Implementación de Caso Práctico bajo Software Libre

Cuando termina la actividad, el personal administrativo, entra al sitio del gimnasio, entra al menú de actividades, le aparecerá una lista de las actividades que se están impartiendo, selecciona la que terminó y la marca como terminada, El sistema pondrá a todos los socios que no se presentaron a la actividad falta

Registro de inasistencias

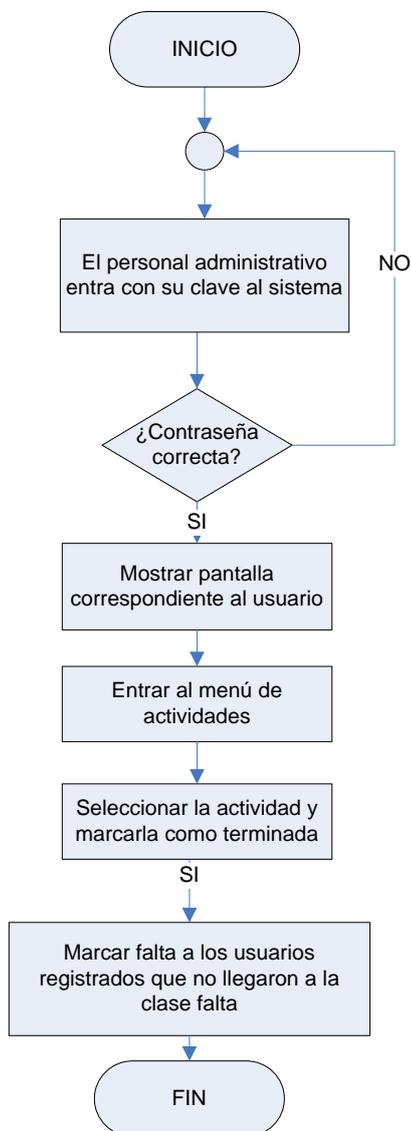


Figura 3.10 Diagrama de flujo del proceso de inasistencia de los socios

3.3.6 Olvido de contraseña

Todos los procesos mencionados anteriormente necesitan que un usuario se autentifique, para lo cual es indispensable una contraseña que, por lo regular suele olvidársele al usuario; es por ello que es de suma importancia contar con un proceso con el cual el socio pueda obtener una nueva contraseña.

El proceso para recuperarla es el siguiente.

El socio entra al sitio del gimnasio y en la parte de autenticación, está el vínculo de Olvido de contraseña, cuando el socio va a este vínculo se le muestra un formulario con unas preguntas

Implementación de Caso Práctico bajo Software Libre

que tiene que contestar adecuadamente, si todos los datos son correctos y coinciden con los que están dados de alta en el sistema, se le enviará una nueva contraseña a la dirección de correo electrónico que dio de alta al registrarse; en caso de que no cuente con una dirección de correo electrónico, el socio podrá ir a las instalaciones del gimnasio, y con su credencial, el personal administrativo le dará su nueva contraseña; en este caso, el encargado, entra al sitio y en el menú de socios, esta la opción de obtener contraseña, el sistema generara una nueva contraseña para el socio y el personal administrativo se la dará al momento.

Las preguntas que se le hacen al socio para darle su contraseña son:

- Nombre completo
- Dirección
- Teléfono
- Número de su código de barras

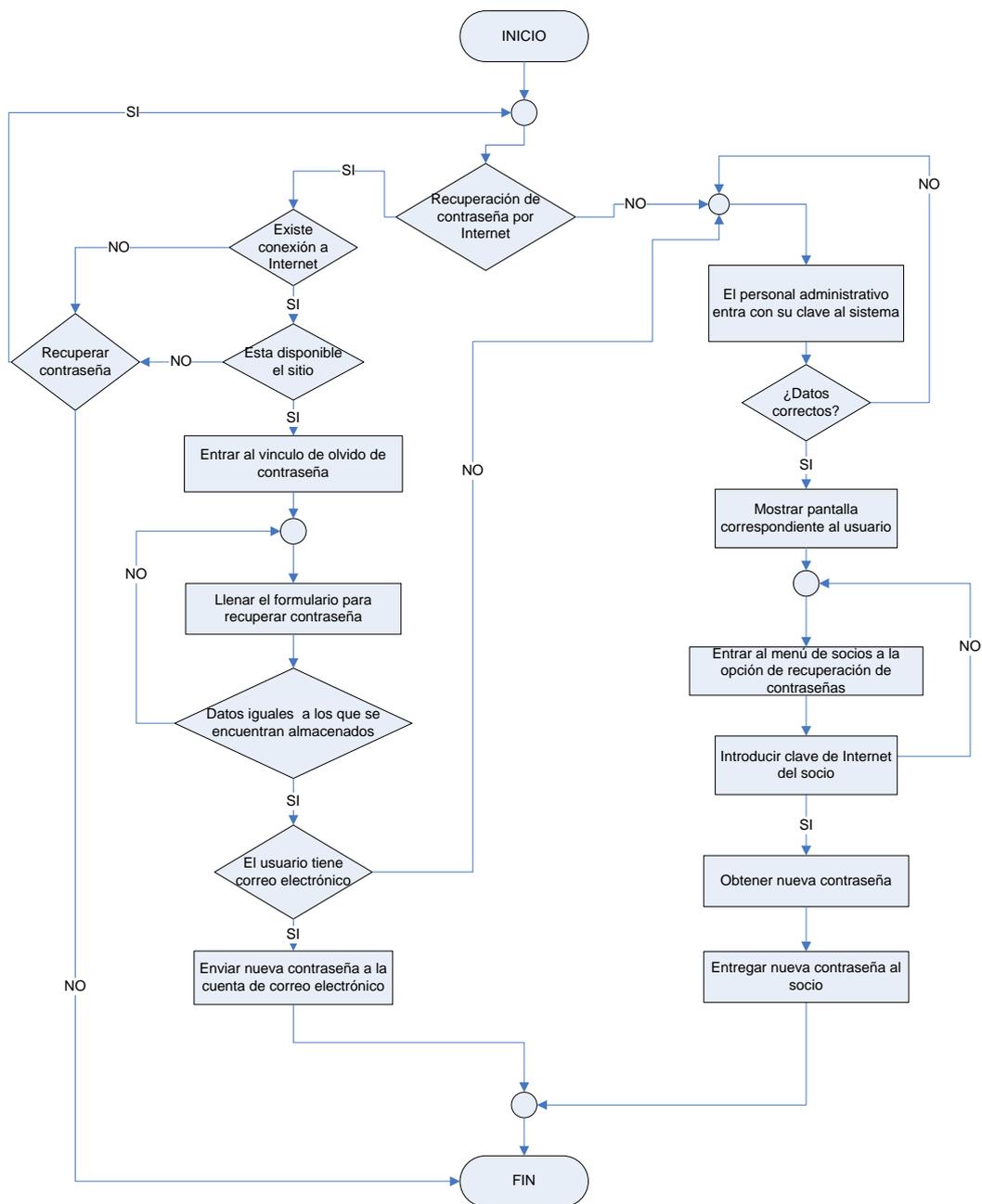


Figura 3.11 Diagrama de flujo del proceso de asistencia de los socios

3.3.7 Altas, cambios y bajas de actividades

Todas las actividades que un socio puede reservar, deben de estar dadas de alta en el sistema, este proceso sólo lo puede realizar el personal administrativo y se realiza de la siguiente forma:

Una vez que el personal administrativo se autentificó, aparecerá el menú correspondiente a este usuario, en donde estará la opción de actividades, para poder dar de alta una actividad deberá de llenar un formulario con los siguientes datos:

- Nombre de la actividad
- Hora de inicio de la actividad
- Hora de fin de la actividad
- Periodo de tiempo en que estará disponible la actividad
- Instructor o encargado de la actividad
- Cupo
- Sala o área donde se impartirá

Las actividades se dan de baja automáticamente cuando termina el periodo por el que estarán disponibles.

Y para los cambios, cuando el personal entre a la opción de cambios, aparecerá una lista de todas las actividades dadas de alta, al seleccionar alguna de ellas, aparecerá la pantalla de alta pero con los datos que se habían introducido originalmente.

Implementación de Caso Práctico bajo Software Libre

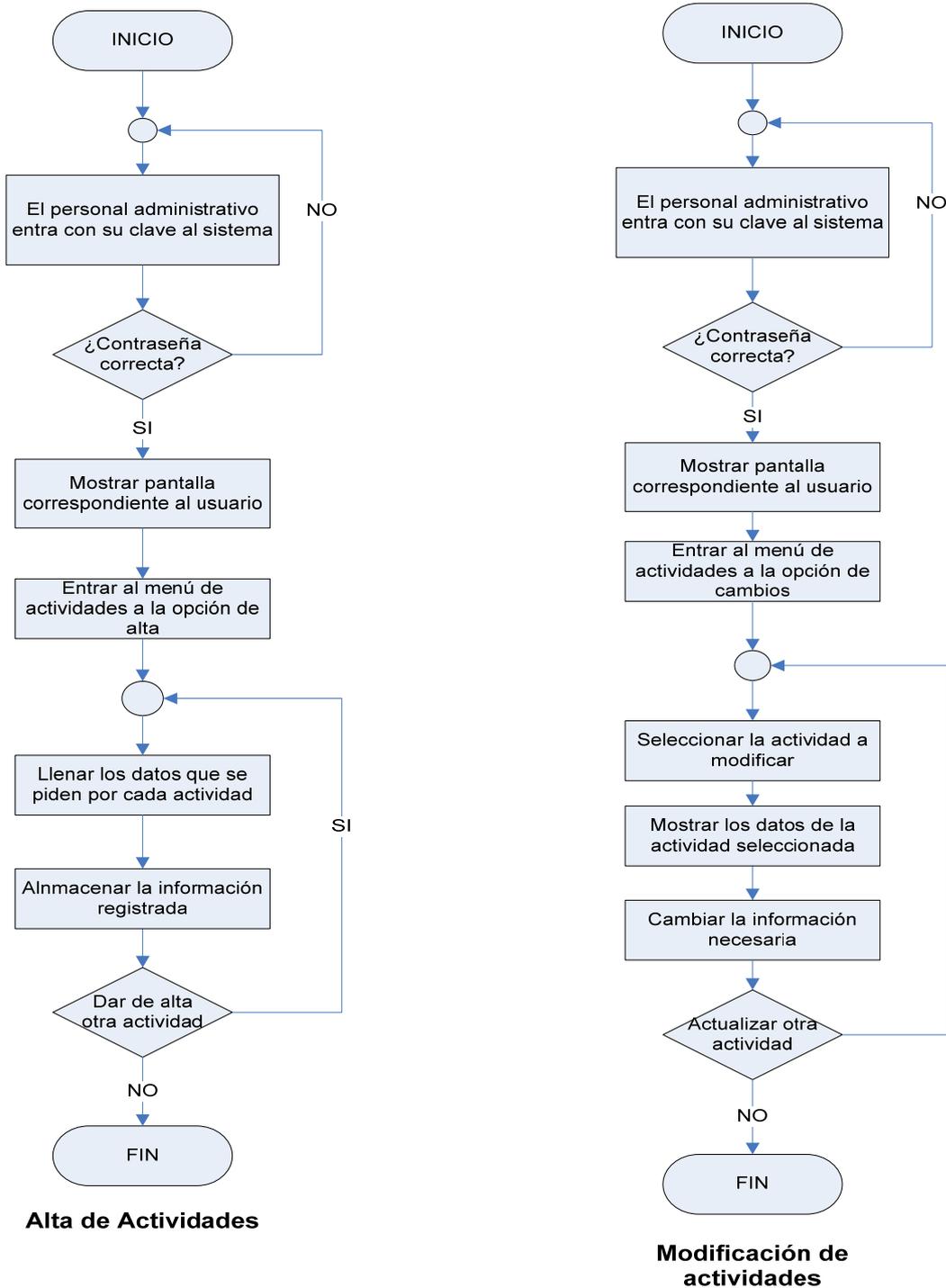


Figura 3.12 Diagrama de flujo del altas, bajas y cambios de actividades

3.4 Diagramas de Casos de Uso

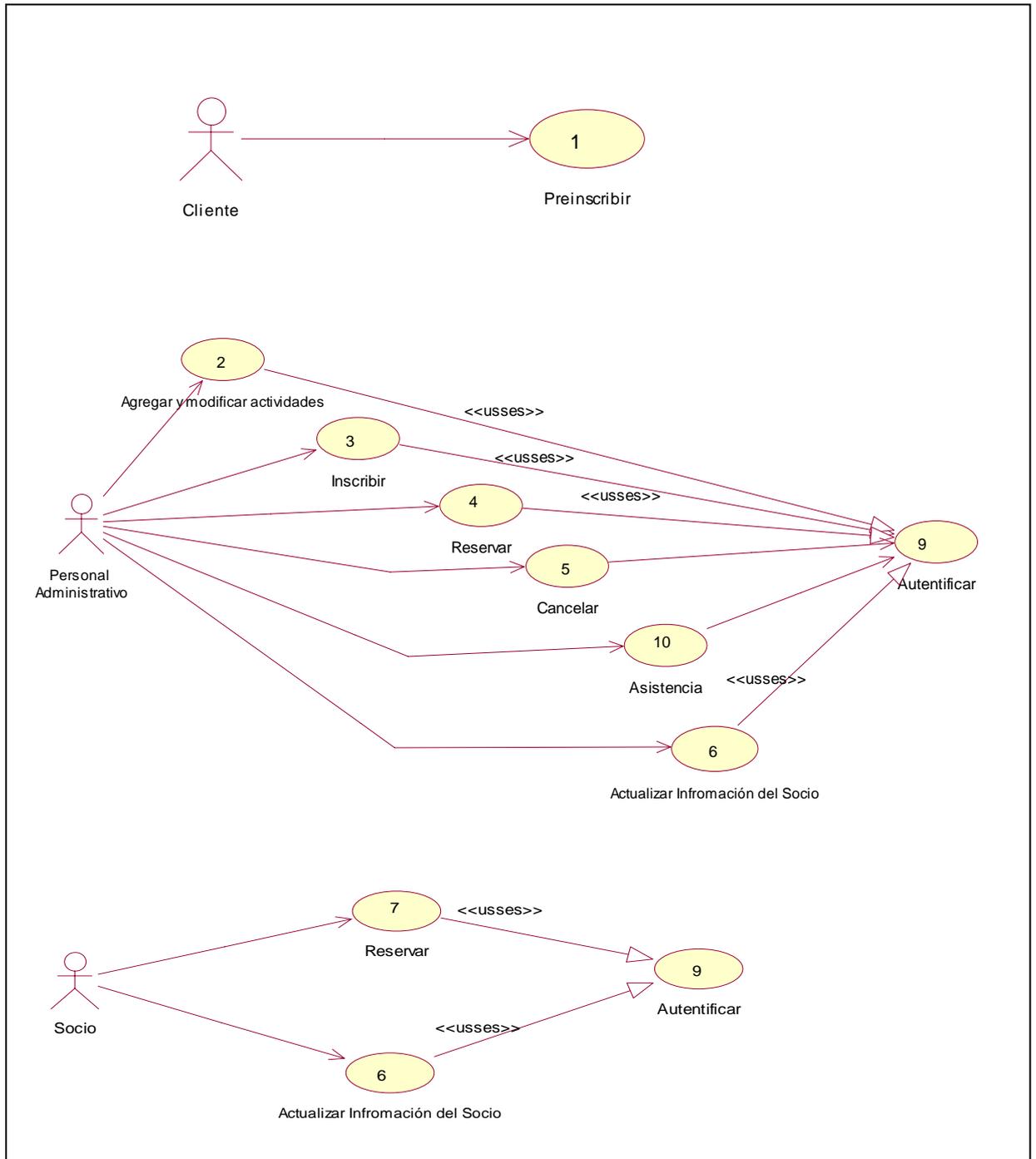


Figura 3.13 Diagrama Casos de uso del sistema

Implementación de Caso Práctico bajo Software Libre

Preinscribir					
Id. del CU	1	Prioridad	Alta / Media / Baja	Estado	En elaboración/ <u>Propuesto</u> /Validado
Actores participantes	Cliente – Es cualquier persona interesada en ser parte del gimnasio como socio.				
Breve Descripción	Permite registrar la información del cliente a través de Internet para agilizar su proceso de inscripción en las instalaciones del gimnasio.				
Pre-condiciones					
Flujo Principal	<ul style="list-style-type: none"> ▪ Entrar al sitio del gimnasio, en el vínculo de “Se parte del club”. ▪ Llenar el formulario que se mostrará con los datos correspondientes del cliente. ▪ Se realizan las validaciones correspondientes en cuanto a formato de datos introducidos (A-1). ▪ El sistema realiza las validaciones correspondientes en cuanto a promociones se refiere (A-2). ▪ El sistema genera el número de preinscripción (A-3). ▪ Se registra en la base de datos la información del cliente (E-1). ▪ Se muestra una pantalla de bienvenida al usuario, la cual contiene su nombre y el monto a pagar por cliente de acuerdo a la información introducida por él. ▪ El cliente sale del sitio del gimnasio. 				
Flujos Alternos	<p>A-1 Validación de campos introducidos por el cliente: Para aceptar el registro del cliente el sistema realiza las siguientes validaciones::</p> <ul style="list-style-type: none"> ▪ Nombre, Apellido Paterno. El sistema valida que no sobrepase los 20 caracteres cada uno, o bien que no tenga caracteres especiales. Dato obligatorio. (E-2) ▪ Apellido Materno. El sistema valida que no sobrepase los 20 caracteres cada uno, o bien que no tenga caracteres especiales (E-2) ▪ Teléfono, teléfono de emergencia. El sistema valida que no sobrepase los 8 caracteres numéricos. ▪ Celular. El sistema valida que no sobrepase los 13 caracteres numéricos. ▪ Fecha de Nacimiento. El sistema realiza la validación correspondiente a una fecha correcta. ▪ Sexo. Femenino, masculino ▪ Dirección: Calle, número, colonia, código postal, estado. Campos alfanuméricos y obligatorios. (E-2) ▪ Correo electrónico: Campo alfanumérico. (E-2) ▪ Enfermedades, alergias, tipo de sangre. Campos alfanuméricos <p>A-2 Verificar promociones disponibles.</p> <ul style="list-style-type: none"> ▪ Se verifica en la base de datos, en base a la fecha si existe algún descuento o promoción para el socio, en caso de haberlo, se realiza el descuento adecuado al precio base de la inscripción o mensualidad. <p>A-3 Generación de número de preinscripción.</p> <ul style="list-style-type: none"> ▪ El sistema genera por cada cliente un número consecutivo de 4 dígitos para que con éste termine su proceso de inscripción. Este digito solo tiene un mes de vigencia y cada mes se comienza el conteo de cero. (E-3) 				
Flujos de excepción	<p>E-1 Imposible guardar registro El sistema notifica: “Error en el sistema, favor de intentar más tarde”, se muestra la página de Inicio.</p> <p>E-2 Formato no válido El sistema notifica: “El dato ingresado es inválido con respecto a la estructura documental definida. Por favor, ingréselo de nuevo”, el sistema permite volver a ingresarlo.</p> <p>E-3 No se genera número de preinscripción. El sistema notifica: “Error en el sistema, favor de intentar más tarde”, se muestra la página de Inicio</p>				
Notas					

Implementación de Caso Práctico bajo Software Libre

Agregar y modificar actividades					
Id. del CU	2	Prioridad	<u>Alta</u> / Media / Baja	Estado	<u>En elaboración</u> /Propuesto/Validado
Actores participantes	Personal Administrativo – Es una persona contratada por el gimnasio, con el privilegio de realizar las tareas administrativas del gimnasio.				
Breve Descripción	Permite dar de alta en el sistema las actividades que se van a llevar a cabo.				
Pre-condiciones	<p>Del proceso</p> <ul style="list-style-type: none"> ▪ El personal administrativo debe de contar con una clave y una contraseña de administrador para tener acceso al sistema y poder realizar todas las actividades que le corresponden. <p>Del sistema</p> <ul style="list-style-type: none"> ▪ Debe estar en el sistema el formato que se debe llenar para registrar las actividades y estar la opción en el menú de registro y cambios de actividades. 				
Flujo Principal	<ul style="list-style-type: none"> ▪ Entrar al sistema con la contraseña de personal administrativo (E-1). ▪ Si quiere dar de alta una actividad entra el menú de alta de actividades <ul style="list-style-type: none"> ○ Llena el formulario de altas que aparece. ○ Envía el formulario para que el sistema lo valide ○ El sistema realiza las validaciones correspondientes (A-1). ○ Se inserta la actividad en base de datos (A-2). ▪ Si el usuario quiere modificar la información que se introdujo anteriormente, entra a al menú de cambio de actividades <ul style="list-style-type: none"> ○ Se muestra una lista de las actividades que hay disponibles, para que el usuario seleccione la actividad deseada y haga los cambios correspondientes. (A-3) ○ Aparecerá el mismo formulario que llenó a la hora de dar de alta la actividad pero con la información ya registrada. ○ Hace los cambios en los campos necesarios. ○ Envía el formulario para que el sistema lo valide ○ El sistema realiza las validaciones correspondientes (A-1). ○ Se realizan los cambios en la base de datos. ▪ Si lo que se quiere es dar de baja la actividad, el usuario debe entrar el menú de Baja de Actividades, y seleccionar de la lista la actividad o actividades que se desee dar de baja. (A-4) 				
Flujos Alternos	<p>A-1 Validación de campos introducidos por el personal administrativo: Para aceptar el registro que el personal introdujo se realizan las siguientes validaciones:</p> <ul style="list-style-type: none"> ▪ Nombre de la actividad. El sistema valida que no sobrepase los 20 caracteres cada uno, o bien que no tenga caracteres especiales. Dato obligatorio. (E-2) ▪ Hora de Inicio. El sistema realiza la validación correspondiente al formato de la hora. Dato obligatorio. ▪ Hora de fin de la actividad. El sistema realiza la validación correspondiente al formato de la hora y que esta no sea igual o menor que la hora de inicio. Dato obligatorio. (E-3) ▪ Instructor o encargado de la actividad. El sistema valida que no sobrepase los 20 caracteres cada uno, o bien que no tenga caracteres especiales. Dato obligatorio. (E-2) ▪ Cupo. Dato numérico obligatorio (E-2) ▪ Sala o área donde se impartirá. Dato alfanumérico obligatorio (E-2) <p>A-2 Registro de Actividad en base de datos Cuando se guarda una actividad en base de datos, se obtiene una clave única para el control de las actividades; éste número es un consecutivo de todas las actividades que se han registrado.</p> <p>A-3 Selección de Actividad a modificar Una vez que aparece la lista de actividades a modificar y se selecciona una, se envía la clave de la actividad para que se regresen los datos correspondientes a esta actividad de la base de datos. Se genera el formulario de actividades con la información que se sacó de base de datos.</p>				

Implementación de Caso Práctico bajo Software Libre

	<p>A-4 Baja de Actividades De la lista que se muestra de baja de actividades, se puede seleccionar una o mas actividades, una vez que se seleccionaron las actividades deseadas, aparece una pantalla de confirmación de borrado, si se acepta, se manda la clave de la actividad para que sea borrado de la base de datos; si se cancela, se regresa a la lista de actividades disponibles.</p>
Flujos de excepción	<p>E-1 Contraseña no válida El sistema notifica: "El usuario y contraseña ingresado es inválido. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-2 Formato no válido El sistema notifica: "El dato ingresado es inválido con respecto a la estructura documental definida. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-3 Hora Inválida El sistema notifica: "La hora de término no puede ser menor o igual a la hora de inicio de la actividad. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p>
Notas	

Implementación de Caso Práctico bajo Software Libre

Inscribir					
Id. del CU	3	Prioridad	Alta / Media / Baja	Estado	En elaboración/Propuesto/Validado
Actores participantes	Personal Administrativo – Es una persona contratada por el gimnasio, con el privilegio de realizar las tareas administrativas del gimnasio.				
Breve Descripción	Permite dar de alta en el sistema a los clientes que quieren ser parte del gimnasio como socios.				
Pre-condiciones	Del proceso <ul style="list-style-type: none"> ▪ El personal administrativo debe de contar con una clave y una contraseña de administrador para tener acceso al sistema y poder realizar todas las actividades que le corresponden. ▪ Si el cliente no cuenta con número de preinscripción, debe llenar un tarjetón que le entregará el personal administrativo. ▪ Si el cliente realizó una preinscripción debe contar con su número de preinscripción. 				
Flujo Principal	<ul style="list-style-type: none"> ▪ Entrar al sistema con la contraseña de personal administrativo (E-1). ▪ Se visualiza la página de Inicio de Sesión correspondiente al personal administrativo. ▪ Se selecciona la opción de Inscripción en el menú. ▪ Si el cliente realizó una preinscripción, se introduce el número de preinscripción para obtener todos los datos que el cliente ya había introducido. ▪ Si el cliente no realizó preinscripción, se realiza el siguiente procedimiento: <ul style="list-style-type: none"> ○ Llenar el formulario de inscripción. ○ Se realizan las validaciones correspondientes en cuanto a formato de datos introducidos (A-1). ○ El sistema realiza las validaciones correspondientes en cuanto a promociones se refiere (A-2). ○ Se registra en base de datos la información del cliente. ▪ Se muestra el monto a cobrar al cliente y se realiza dicho cobro. ▪ Se registra en base de datos el pago realizado. ▪ Se asigna una clave al socio para tener acceso al sistema vía Web (A-3) ▪ El sistema asigna contraseña al nuevo socio (A-4) ▪ El personal administrativo toma la foto al socio. ▪ Imprimir credencial con código de barras, foto, nombre del socio y clave de Internet ▪ El sistema regresa a la página principal 				
Flujos Alternos	<p>A-1 Validación de campos introducidos por el cliente:</p> <p>Para aceptar el registro del cliente el sistema realiza las siguientes validaciones::</p> <ul style="list-style-type: none"> ▪ Nombre, Apellido Paterno. El sistema valida que no sobrepase los 20 caracteres cada uno, o bien que no tenga caracteres especiales. Dato obligatorio. (E-2) ▪ Apellido Materno. El sistema valida que no sobrepase los 20 caracteres cada uno, o bien que no tenga caracteres especiales (E-2) ▪ Teléfono, teléfono de emergencia. El sistema valida que no sobrepase los 8 caracteres numéricos. ▪ Celular. El sistema valida que no sobrepase los 13 caracteres numéricos. ▪ Fecha de Nacimiento. El sistema realiza la validación correspondiente a una fecha correcta. ▪ Sexo. Femenino, masculino ▪ Dirección: Calle, número, colonia, código postal, estado. Campos alfanuméricos y obligatorios. (E-2) ▪ Correo electrónico: Campo alfanumérico. (E-2) ▪ Enfermedades, alergias, tipo de sangre. Campos alfanuméricos <p>A-2 Verificar promociones disponibles.</p> <ul style="list-style-type: none"> ▪ Se verifica en la base de datos, en base a la fecha si existe algún descuento o promoción para el 				

Implementación de Caso Práctico bajo Software Libre

	<p>socio, en caso de haberlo, se realiza el descuento adecuado al precio base de la inscripción o mensualidad.</p> <p>(A-3) Generación de clave de Internet La clave de Internet se genera de acuerdo a lo siguiente:</p> <ul style="list-style-type: none">▪ Primer nombre▪ Primera letra de su apellido paterno▪ Fecha de nacimiento. (E-3) <p>(A-4) Generación de contraseña de Internet La contraseña para la clave de Internet es una cadena de 5 letras que se genera aleatoriamente.</p>
Flujos de excepción	<p>E-1 Contraseña no válida El sistema notifica: "El usuario y contraseña ingresado es inválido. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-2 Formato no válido El sistema notifica: "El dato ingresado es inválido con respecto a la estructura documental definida. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-3 Clave inválida En caso de que esta clave este repetida, se notifica "Clave repetida, favor de introducir una nueva" se verifica que no sea el mismo socio; si no es así, el administrador puede asignar la clave siguiendo la base para obtener la clave, recorriendo las letras del nombre, apellido paterno y apellido materno</p>
Notas	

Implementación de Caso Práctico bajo Software Libre

Reservar, Cancelar Actividades					
Id. del CU	4, 5	Prioridad	Alta / Media / Baja	Estado	En elaboración/Propuesto/Validado
Actores participantes	<p>Personal Administrativo – Es una persona contratada por el gimnasio, con el privilegio de realizar las tareas administrativas del gimnasio.</p> <p>Socio – Es una persona que realizó su inscripción al gimnasio, y por lo tanto tiene acceso a todos los servicios que se proporcionan en él.</p>				
Breve Descripción	Permite a los socios realizar reservaciones para las actividades que deseen tomar en el tiempo en que realizaron su pago.				
Pre-condiciones	<p>Del proceso</p> <ul style="list-style-type: none"> ▪ El personal administrativo o en su caso, el socio debe de contar con una clave y una contraseña para tener acceso al sistema y poder realizar la reservación. ▪ Si el socio va a hacer su reservación vía Web, debe tener disponible una conexión a Internet para que pueda entrar al sitio del gimnasio. <p>Del sistema</p> <ul style="list-style-type: none"> ▪ Deben estar dadas de alta en el sistema las actividades que se impartirán en el gimnasio. 				
Flujo Principal	<p>Si la reservación se realiza en las instalaciones:</p> <ul style="list-style-type: none"> ▪ Entrar al sistema con la contraseña de personal administrativo (E-1). ▪ Entrar al menú de reservas ▪ Introducir la clave del socio (E-2). ▪ Seleccionar la opción de reservación (E-3) <ul style="list-style-type: none"> ○ Si el socio lo que quiere es cancelar una reservación, entonces el administrador tendrá que quitar la selección a las actividades que aparezcan reservadas y que el socio ya no desee. ▪ Guardar la información registrada por el personal administrativo. <p>Si la reservación se realiza vía Web:</p> <ul style="list-style-type: none"> ▪ Entrar al sitio del gimnasio ▪ Entrar al sistema con usuario y contraseña del socio (E-1). ▪ Entrar al menú de reservas (E-3) ▪ Seleccionar la opción de reservación <ul style="list-style-type: none"> ○ Si el socio lo que quiere es cancelar una reservación, entonces el administrador tendrá que quitar la selección a las actividades que aparezcan reservadas y que el socio ya no desee. ▪ Guardar la información registrada por el personal administrativo. 				
Flujos Alternos					
Flujos de excepción	<p>E-1 Contraseña no válida El sistema notifica: "El usuario y contraseña ingresado es inválido. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-2 Clave inexistente El sistema notifica: "El dato ingresado es no existe en base de datos. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-2 Usuario atrasado en pagos El sistema notifica: "El usuario no puede realizar reservación, no tiene su pago al corriente", el sistema regresa al menú general.</p>				
Notas					

Implementación de Caso Práctico bajo Software Libre

Actualizar Información del socio					
Id. del CU	6	Prioridad	Alta / Media / Baja	Estado	En elaboración/ <u>Propuesto</u> /Validado
Actores participantes	<p>Personal Administrativo – Es una persona contratada por el gimnasio, con el privilegio de realizar las tareas administrativas del gimnasio.</p> <p>Socio – Es una persona que realizó su inscripción al gimnasio, y por lo tanto tiene acceso a todos los servicios que se proporcionan en él.</p>				
Breve Descripción	Permite actualizar la información que se tiene registrada en el sistema sobre un socio.				
Pre-condiciones	<p>Del proceso</p> <ul style="list-style-type: none"> ▪ El personal administrativo o en su caso, el socio deben de contar con una clave y una contraseña para tener acceso al sistema y poder realizar la reservación. ▪ Si el socio va a hacer modificaciones vía Web, debe tener disponible una conexión a Internet para que pueda entrar al sitio del gimnasio. <p>Del sistema</p> <ul style="list-style-type: none"> ▪ El socio debe estar dado de alta en el sistema. 				
Flujo Principal	<p>Si la actualización la realiza el personal administrativo:</p> <ul style="list-style-type: none"> ▪ Entrar al sistema con la contraseña de personal administrativo (E-1). ▪ Entrar al menú de Socios ▪ Introducir la clave del socio (E-2). ▪ Aparece la información del socio (A-1). ▪ Modificar la información deseada (E-3) ▪ Guardar la información registrada por el personal administrativo. <p>Si la actualización la realiza el socio:</p> <ul style="list-style-type: none"> ▪ Entrar al sitio del gimnasio ▪ Entrar al sistema con usuario y contraseña del socio (E-1). ▪ Entrar al menú de Socio ▪ Aparece la información del socio (A-1). ▪ Modificar la información deseada (E-3) ▪ Guardar la información registrada. Los cambios se guardan en la base de datos. 				
Flujos Alternos	<p>A-1 Recuperar datos del socio.</p> <ul style="list-style-type: none"> ▪ Se verifica en la base de datos, en base a la clave del socio los datos que tiene registrados. ▪ Se muestra el último pago que realizó el socio y la fecha hasta la cual tiene vigencia su pago. ▪ Se muestran todos los datos que se tienen registrados del socio, exceptuando la contraseña. Solamente los datos que se pueden modificar aparecen habilitados, para que el socio los cambie, éstos son: <ul style="list-style-type: none"> ○ Teléfono, teléfono de emergencia. El sistema valida que no sobrepase los 8 caracteres numéricos. ○ Celular. El sistema valida que no sobrepase los 13 caracteres numéricos. ○ Dirección: Calle, número, colonia, código postal, estado. Campos alfanuméricos y obligatorios. (E-3) ○ Correo electrónico: Campo alfanumérico. (E-3) ○ Enfermedades, alergias. Campos alfanuméricos 				
Flujos de excepción	<p>E-1 Contraseña no válida</p> <p>El sistema notifica: "El usuario y contraseña ingresado es inválido. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p>				

Implementación de Caso Práctico bajo Software Libre

	<p>E-2 Clave incorrecta El sistema notifica: "El usuario o contraseña no existen. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-3 Formato no válido El sistema notifica: "El dato ingresado es inválido con respecto a la estructura documental definida. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo</p>
Notas	

Autenticar					
Id. del CU	9	Prioridad	Alta / Media / Baja	Estado	En elaboración/ <u>Propuesto</u> /Validado
Actores participantes	<p>Personal Administrativo – Es una persona contratada por el gimnasio, con el privilegio de realizar las tareas administrativas del gimnasio.</p> <p>Socio – Es una persona que realizó su inscripción al gimnasio, y por lo tanto tiene acceso a todos los servicios que se proporcionan en él.</p>				
Breve Descripción	Permite que un usuario del sistema pueda autenticarse para poder tener acceso a sus actividades de acuerdo al perfil del usuario.				
Pre-condiciones	<p>Del proceso</p> <ul style="list-style-type: none"> ▪ El usuario deben de contar con una clave y una contraseña para tener acceso al sistema y poder realizar la reservación. <p>Del sistema</p> <ul style="list-style-type: none"> ▪ El usuario debe estar dado de alta en el sistema. 				
Flujo Principal	<ul style="list-style-type: none"> ▪ Entrar al sistema ▪ Introducir clave de usuario ▪ Introducir contraseña de usuario ▪ El sistema verifica la información introducida contra la información que se tiene almacenada en base de datos. (A – 1) (E-1) 				
Flujos Alternos					
Flujos de excepción	<p>E-1 Contraseña no válida El sistema notifica: "El usuario y contraseña ingresado es inválido. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-2 Formato no válido El sistema notifica: "El dato ingresado es inválido con respecto a la estructura documental definida. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.</p> <p>E-3 Formato no válido El sistema notifica: "El dato ingresado es inválido con respecto a la estructura documental definida. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo</p>				
Notas					

Implementación de Caso Práctico bajo Software Libre

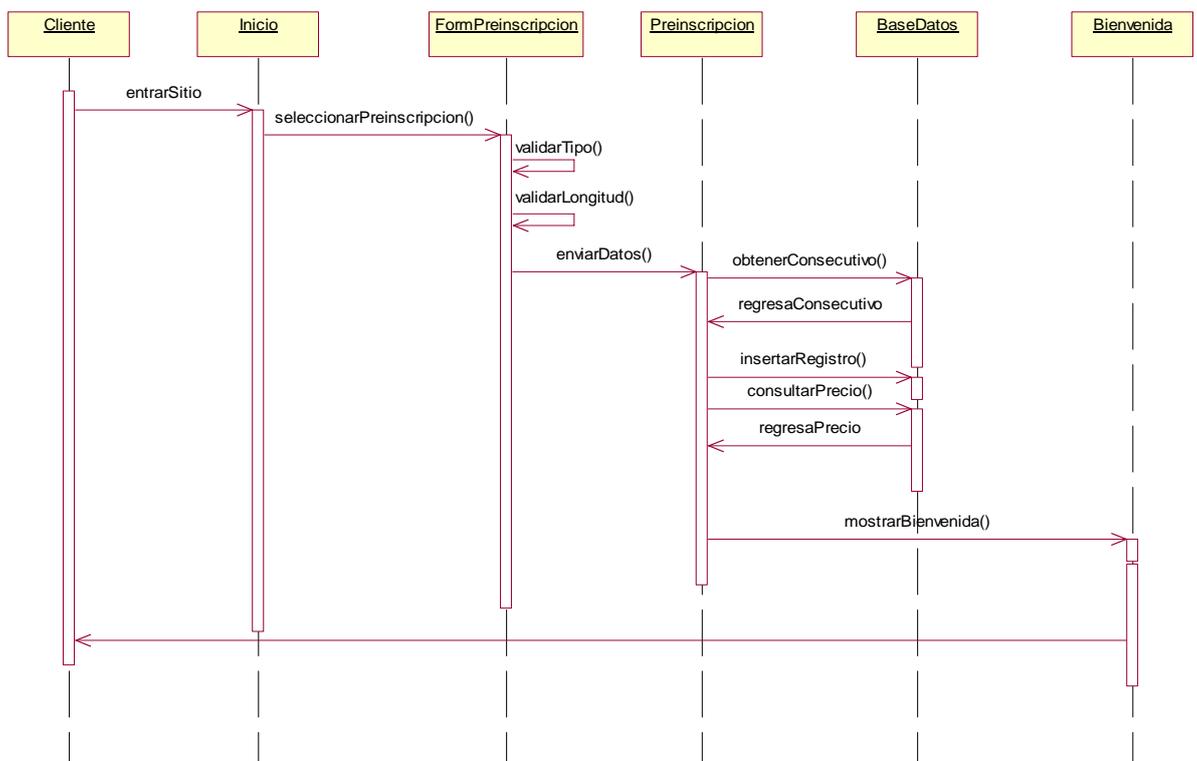
Asistencias					
Id. del CU	10	Prioridad	<u>Alta</u> / Media / Baja	Estado	En elaboración/ <u>Propuesto</u> /Validado
Actores participantes	<p>Socio – Es una persona que realizó su inscripción al gimnasio, y por lo tanto tiene acceso a todos los servicios que se proporcionan en él.</p> <p>Personal Administrativo – Es una persona contratada por el gimnasio, con el privilegio de realizar las tareas administrativas del gimnasio</p>				
Breve Descripción	Permite llevar un control sobre las actividades a las que cada socio acudió y que tenía reservadas.				
Pre-condiciones	Del sistema <ul style="list-style-type: none"> ▪ El socio debió haber reservado una actividad. 				
Flujo Principal	<ul style="list-style-type: none"> ▪ El socio desliza su credencial en el lector de código de barras, en el área donde tomará la actividad. ▪ Se verifica en base de datos si el socio tiene registrada la actividad, si es así: <ul style="list-style-type: none"> ○ Si faltan 10 minutos o menos o es la hora para que empieza la actividad, se registra asistencia al socio y se le da acceso a la sala. ○ Si paso la hora de fin de la actividad se marca falta al socio ○ Si todavía esta dentro de la hora de la actividad se marca retardo al socio. ▪ Cuado terminó la actividad, el personal administrativo: <ul style="list-style-type: none"> ▪ Entra al sitio del gimnasio ▪ Entra al sistema con usuario y contraseña del administrador (E-1). ▪ Entra al menú actividades ▪ Aparecerá una lista de las actividades que se están impartiendo y se marca como terminada la actividad. ▪ Se registra falta a todos los socios que tenían reservación y no llegaron. ▪ 				
Flujos Alternos					
Flujos de excepción	E-1 Contraseña no válida El sistema notifica: "El usuario y contraseña ingresado es inválido. Por favor, ingréselo de nuevo", el sistema permite volver a ingresarlo.				
Notas					

3.5 Diagramas de Secuencia

Típicamente uno examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se tiene modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces se pueden caminar esos pasos para descubrir qué objetos son necesarios para en la construcción del sistema.

Preinscribir

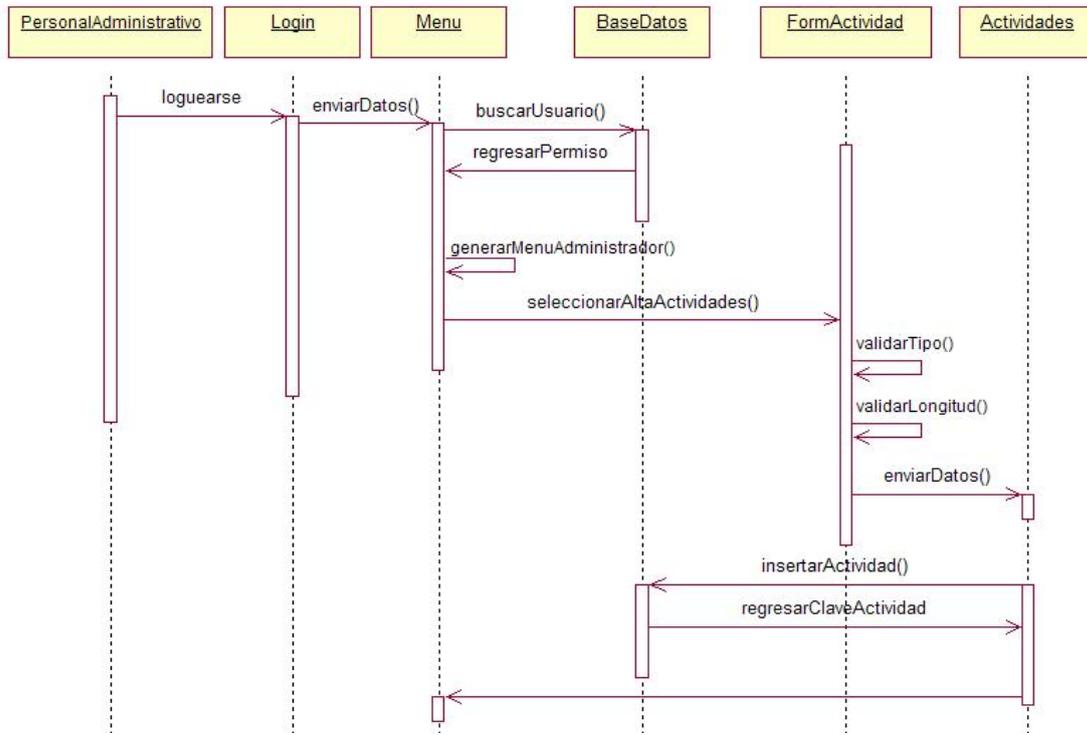
En este diagrama se muestra la secuencia que se debe seguir para registrar la información del cliente a través de Internet para agilizar su proceso de inscripción en las instalaciones del gimnasio.



Implementación de Caso Práctico bajo Software Libre

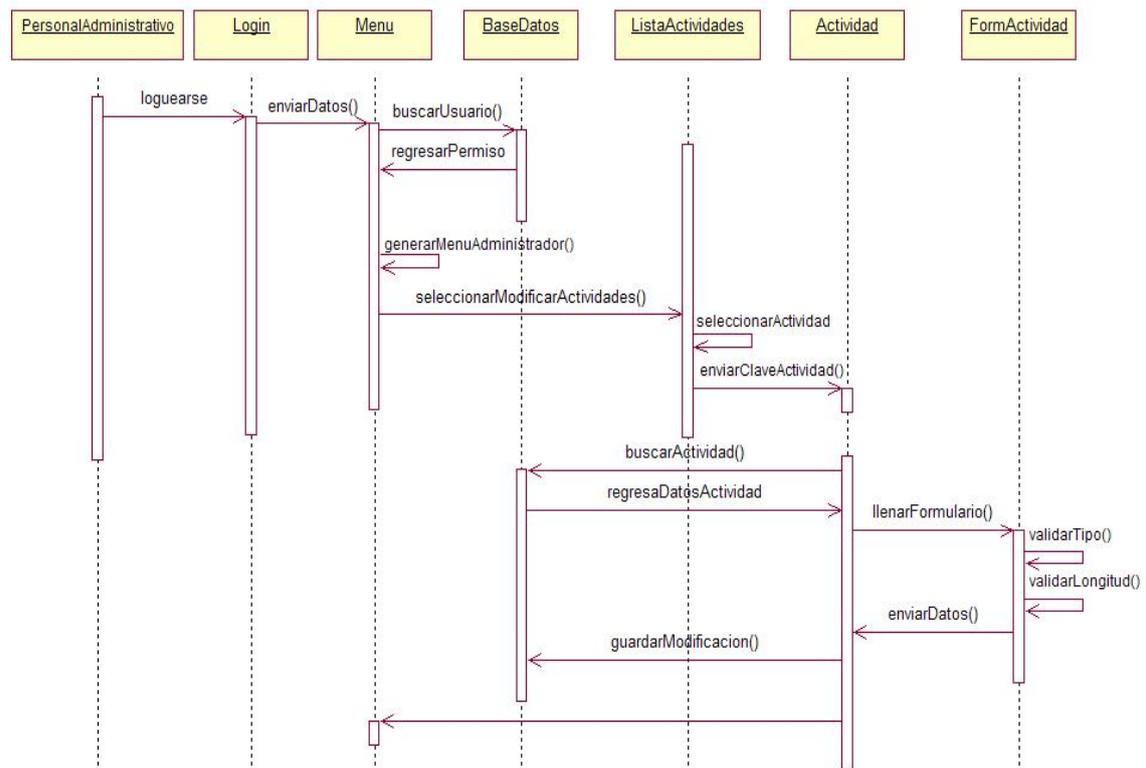
Agregar actividades

En este diagrama se muestra la secuencia que se sigue para dar de alta en el sistema las actividades que se pueden llevar a cabo en el gimnasio, cabe destacar que sólo el personal administrativo puede realizar esta actividad.



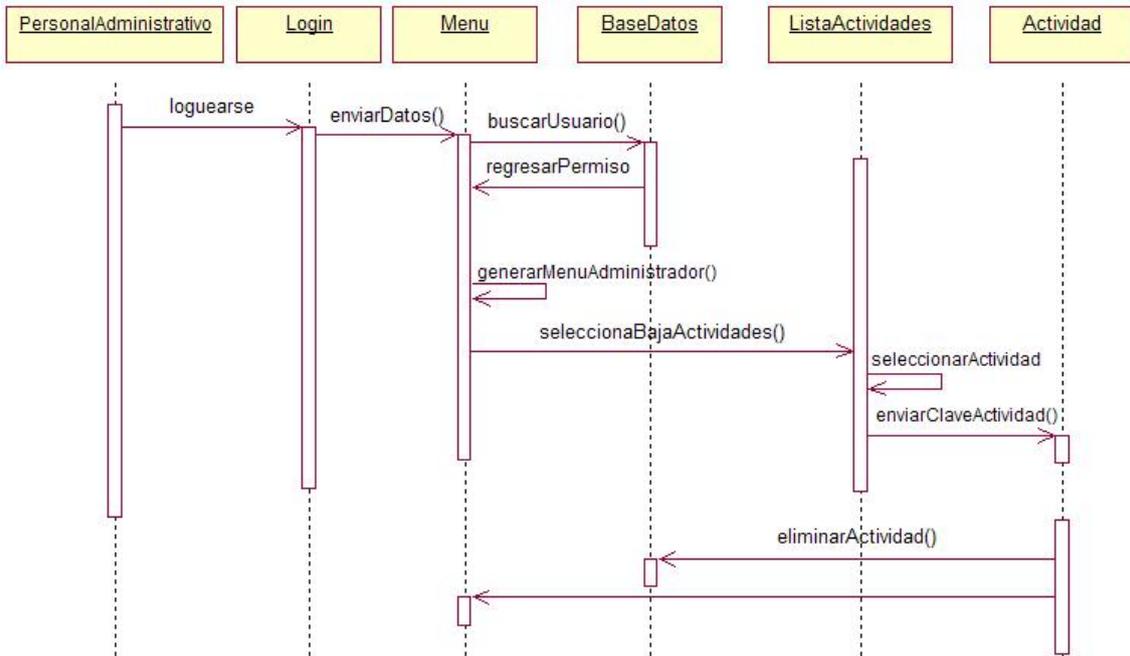
Modificar actividades

El proceso para realizar la modificación de actividades es de la siguiente forma:



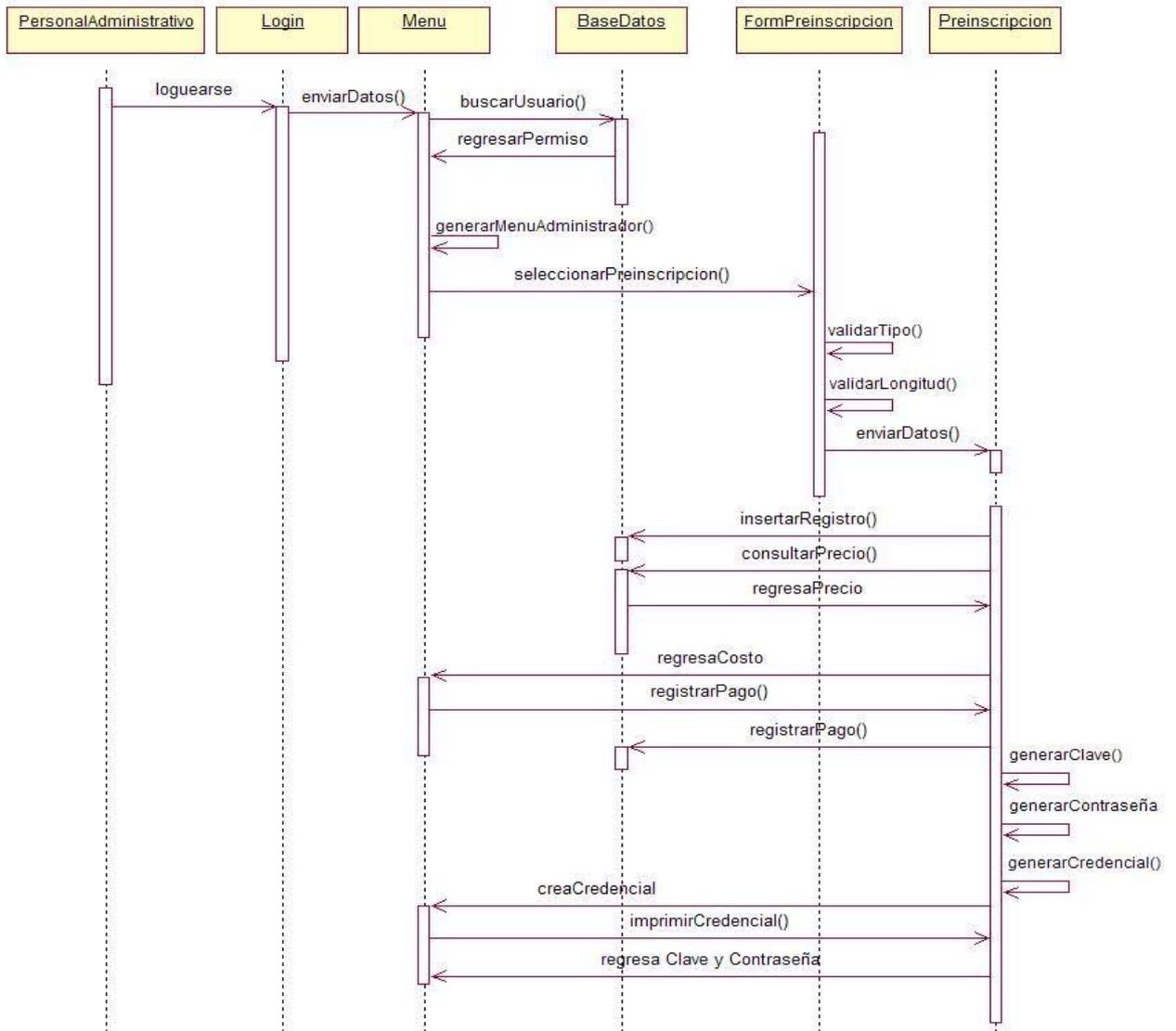
Baja de Actividades

El proceso para realizar la baja de actividades es de la siguiente forma:



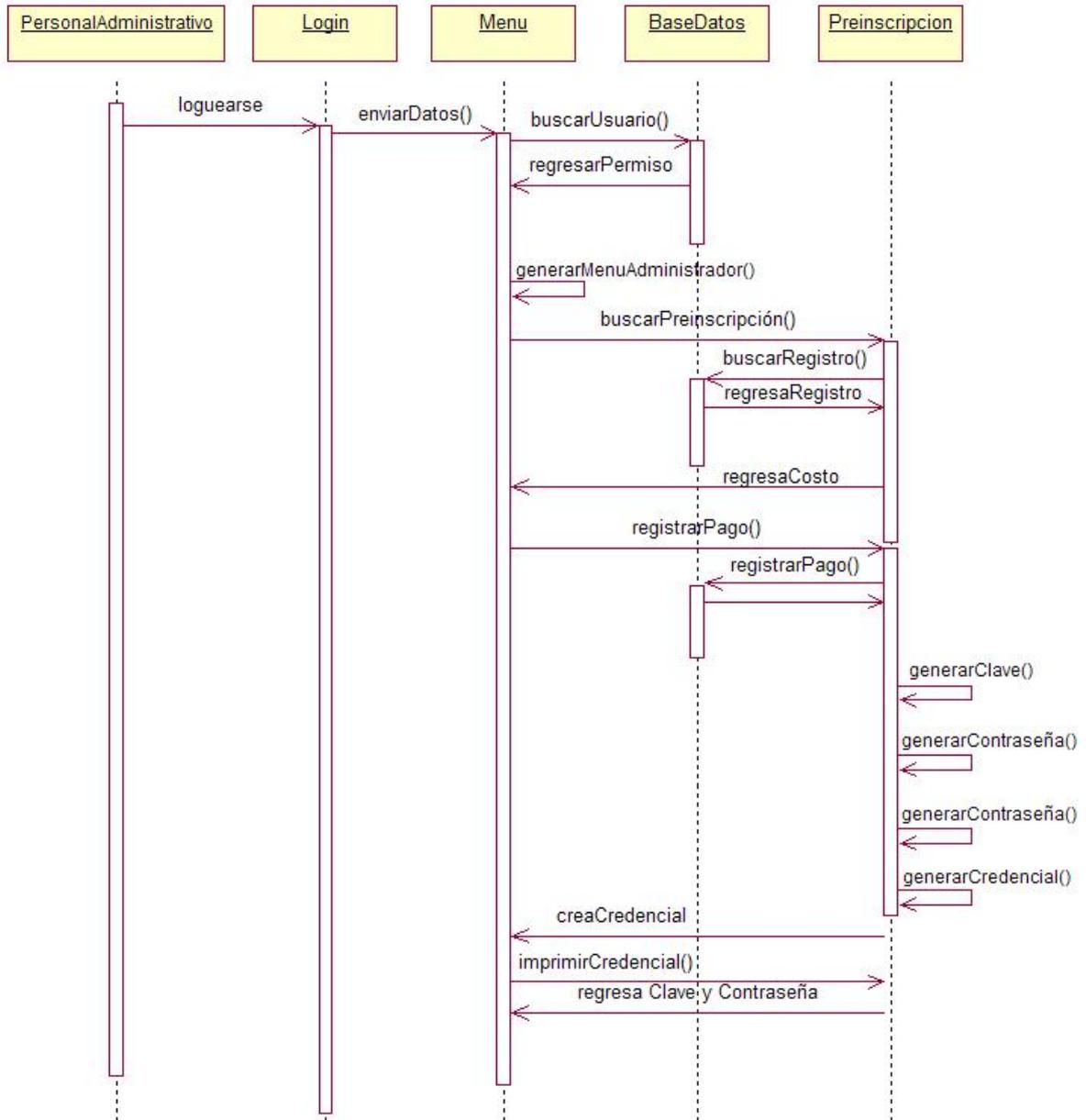
Inscribir sin preinscripción

El siguiente diagrama de secuencia, muestra la secuencia que se sigue en el caso de una inscripción de un socio que no realizó preinscripción



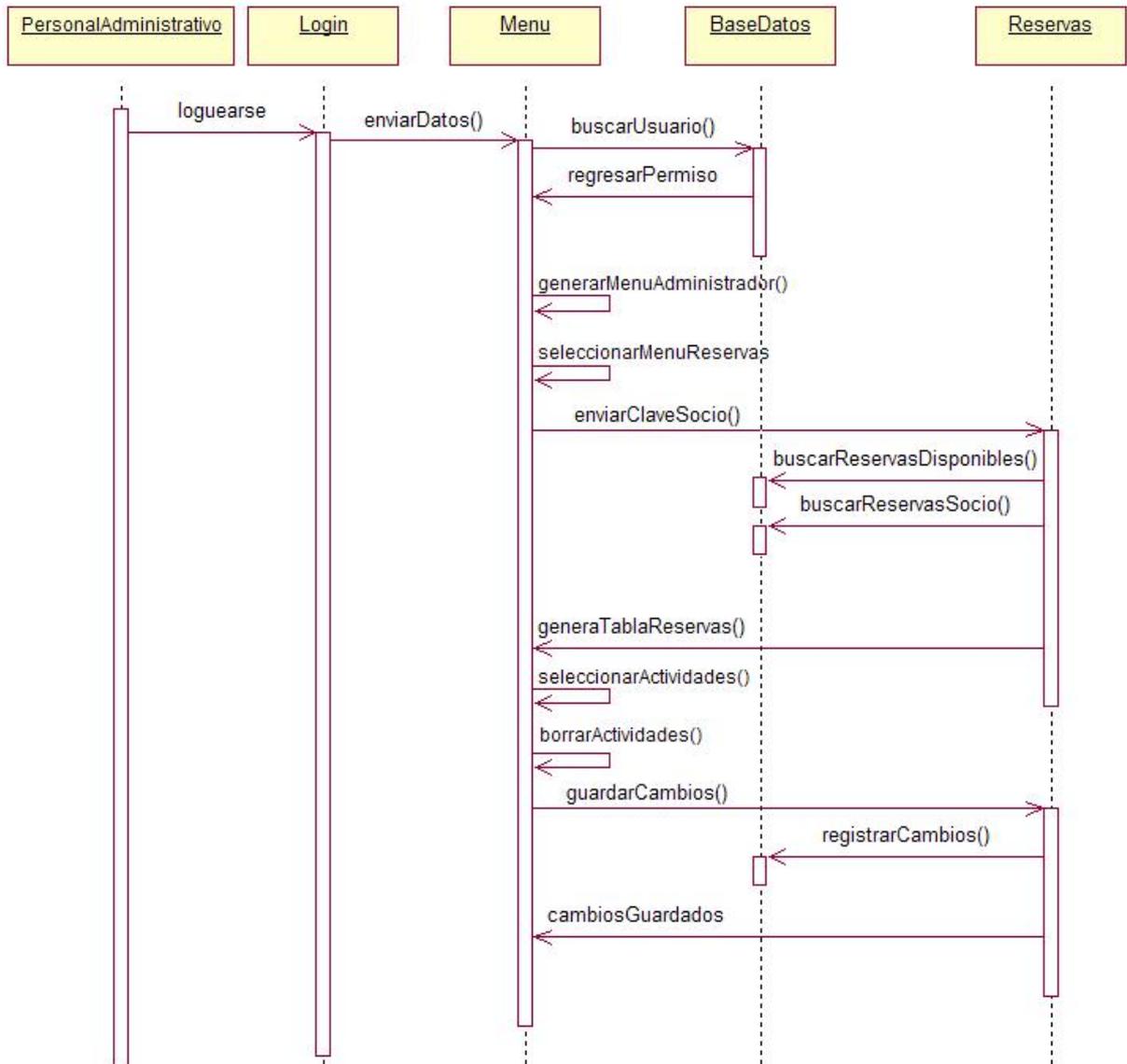
Inscribir con preinscripción

En el siguiente diagrama se muestra la secuencia que se debe seguir en caso de que un cliente haya realizado su preinscripción.



Reservar en las instalaciones del gimnasio

El diagrama siguiente muestra la secuencia que permite a los socios hacer reservaciones para las actividades que deseen tomar en el periodo válido de su pago; éste proceso se hace en las instalaciones del gimnasio por medio del personal administrativo.



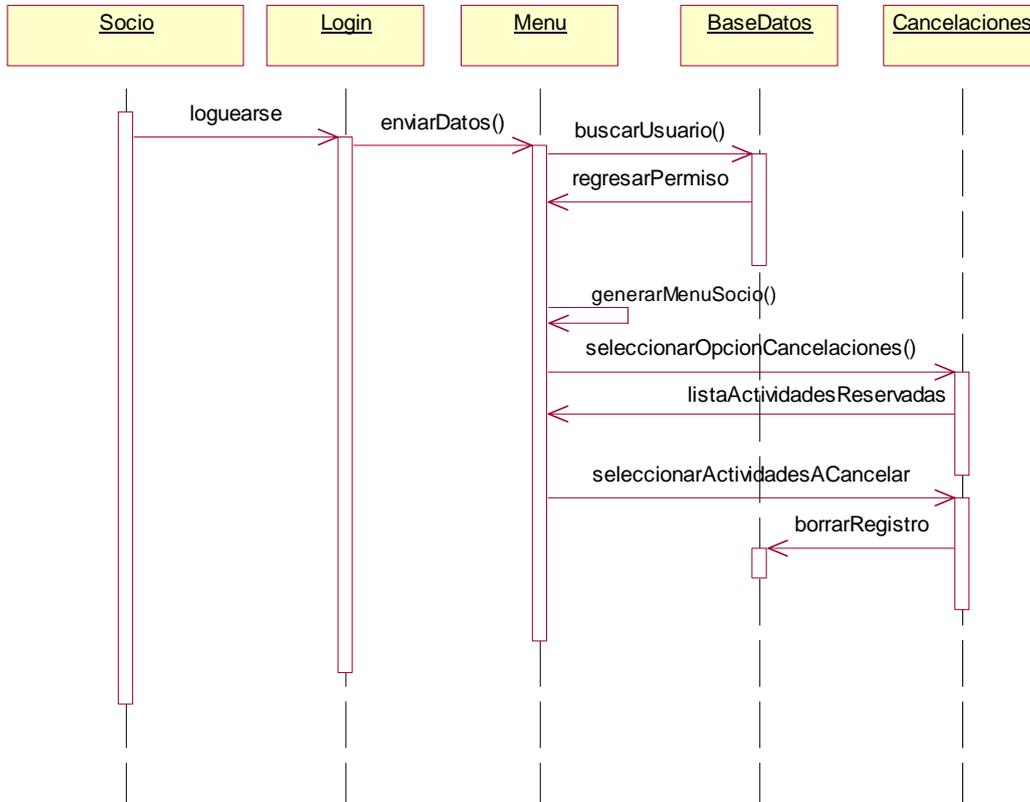
Reservar actividades vía Web

El diagrama siguiente muestra la secuencia que permite a los socios realizar reservaciones para las actividades que deseen tomar en el tiempo en que realizaron su pago. Estas reservaciones se realizan por el socio vía Web.



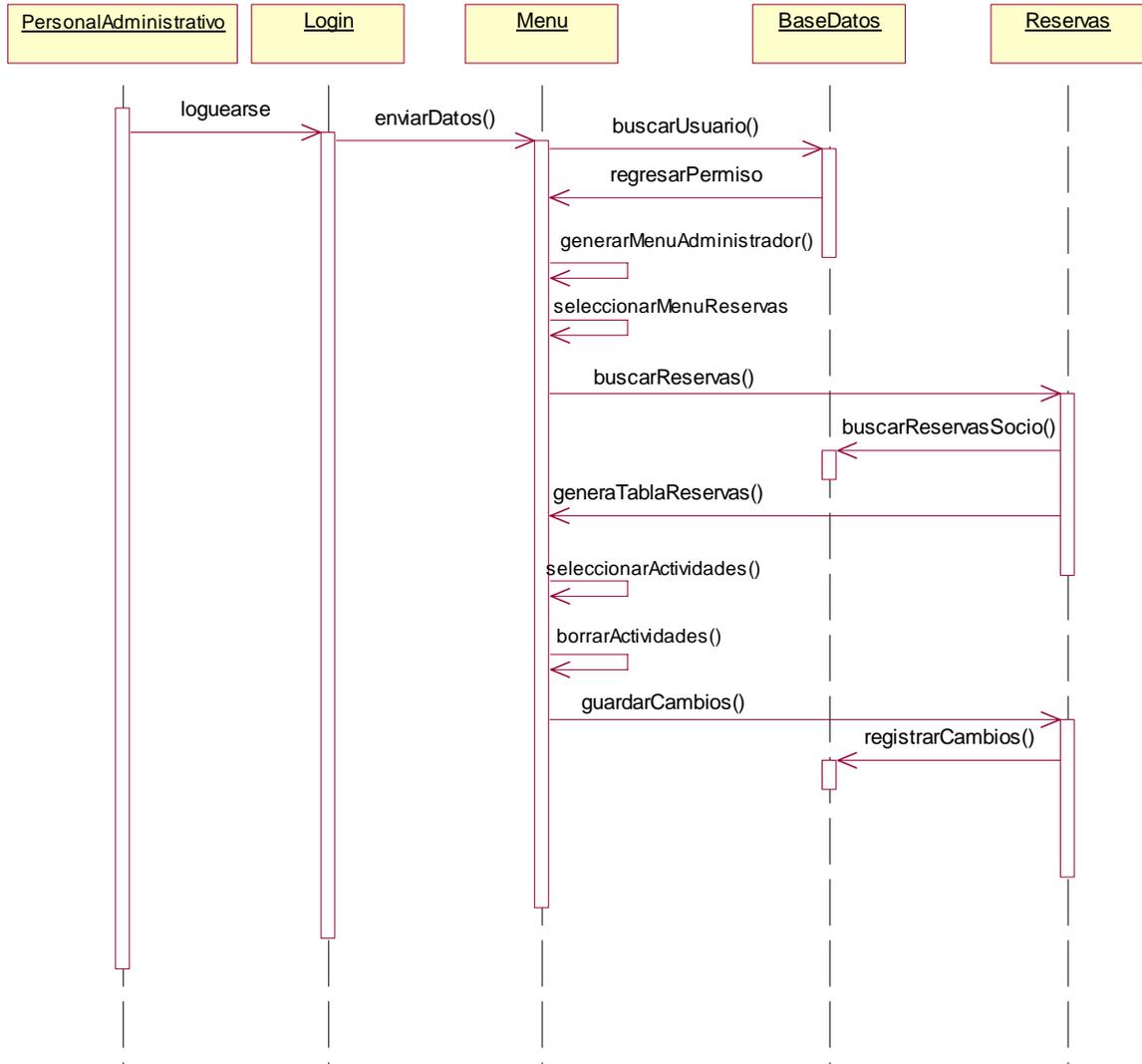
Cancelar actividades vía Web

El diagrama siguiente muestra la secuencia que permite a los socios cancelar las actividades que se habían reservado. Estas cancelaciones se realizan por el socio vía Web.



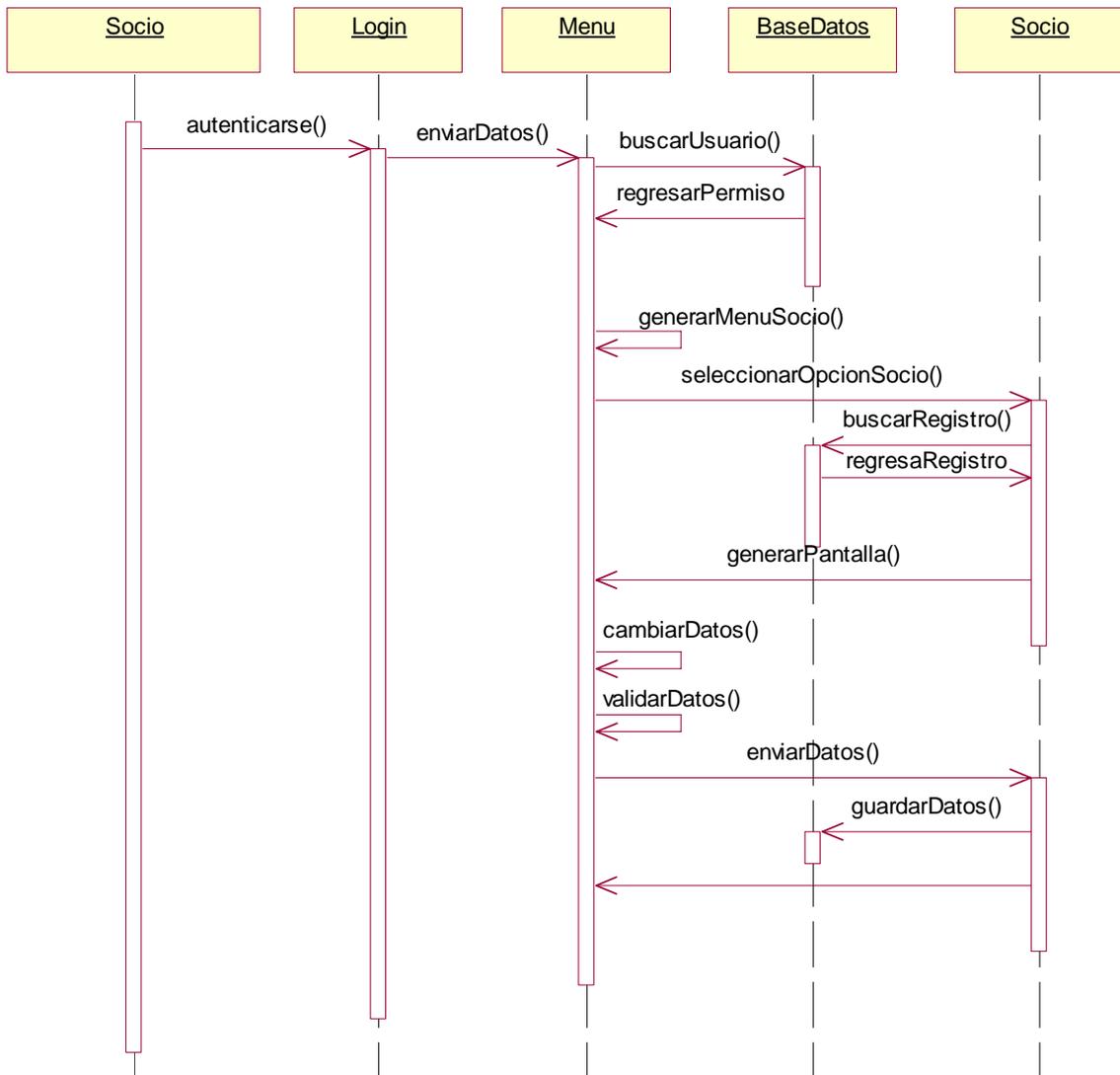
Cancelar actividades en las instalaciones del gimnasio

El diagrama siguiente muestra la secuencia que permite a los socios cancelar reservaciones para las actividades que tenía reservadas; éstas se realizan en las instalaciones del gimnasio por medio del personal administrativo.



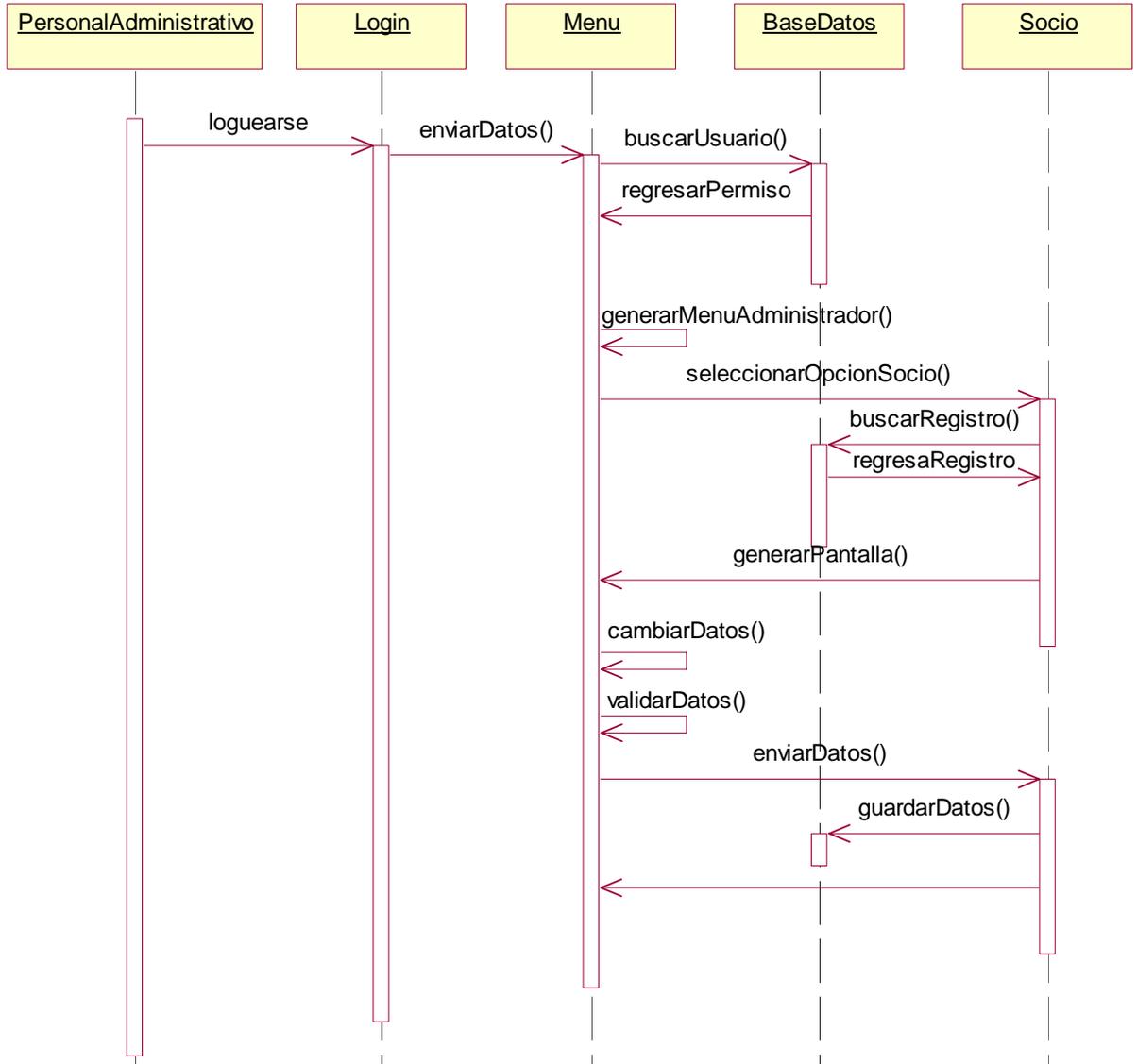
Actualizar Información del socio vía Web

Un cliente puede modificar cierta información que en el gimnasio se maneja sobre él, tal información puede ser dirección, teléfono, dirección de correo electrónico y contraseña para el acceso al sitio. Para que un socio pueda realizar estas modificaciones vía Web se sigue la siguiente secuencia



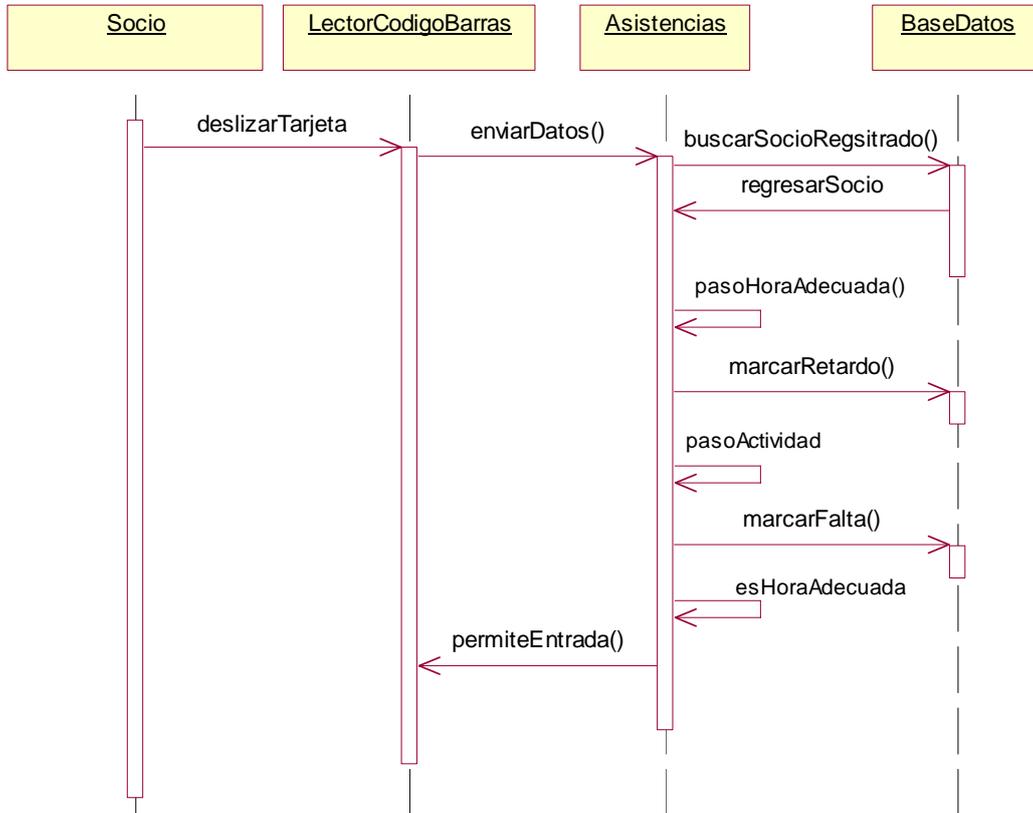
Actualizar Información del socio por el personal Administrativo

Cuando un socio no puede actualizar su información porque no cuenta con un acceso a Internet. Tiene que realizarlo a través del personal administrativo del gimnasio, y es éste quien hace los cambios. Se debe seguir la siguiente secuencia.

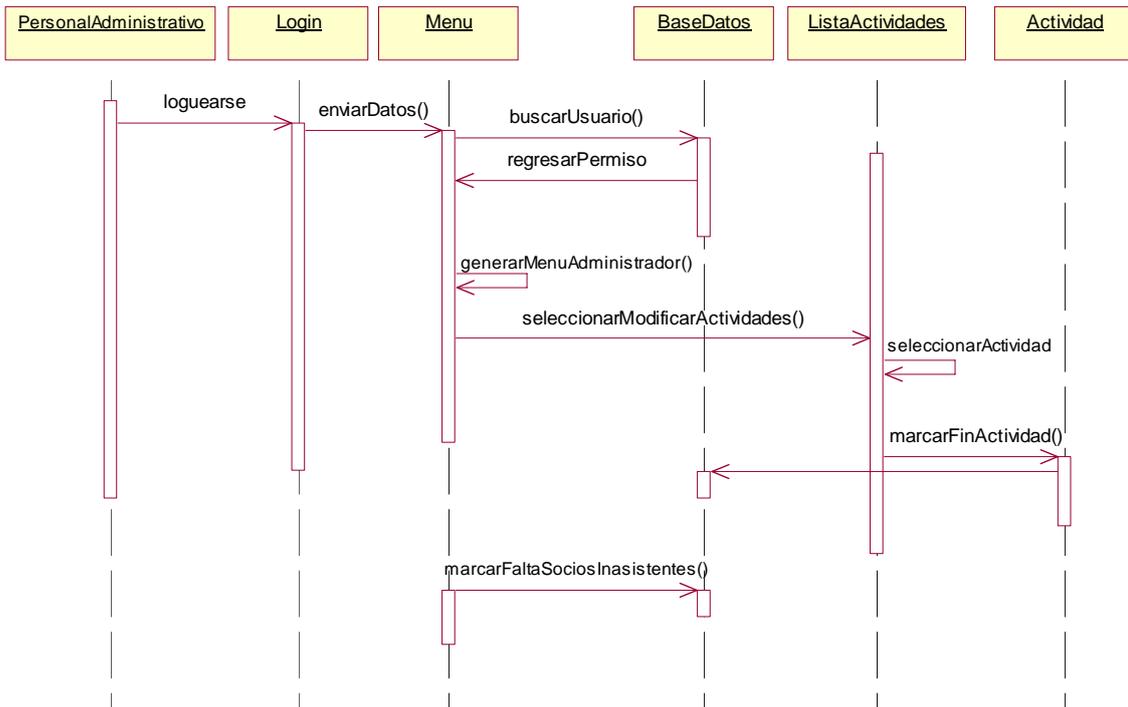


Asistencias

Permite llevar un control sobre las actividades a las que cada socio acudió y que tenía reservadas



Inasistencias



Una vez que se tiene el análisis de los requerimientos de procesos del sistema, se puede ir trabajando con la parte del diseño, con respecto a la vista del usuario.

Capítulo 4

Diseño de Caso Práctico

Modelo Vista Controlador

Diseño de Caso Práctico

Modelo Vista Controlador

MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

1. **El Modelo** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
2. **La Vista** es el objeto que maneja la presentación visual de los datos representados por el Modelo. Muestra los datos al usuario e interactúa con el Modelo a través de una referencia al propio Modelo.
3. **El Controlador** es el objeto que proporciona significado a las ordenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista.

El siguiente diagrama muestra un ejemplo de interacción con el patrón de diseño MVC:

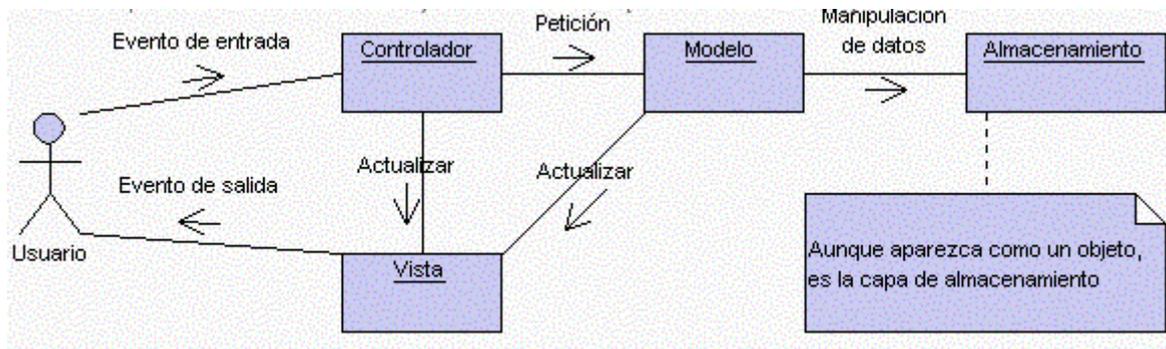


Figura 4.1 Patrón de diseño MVC

¿Que ventajas obtenemos de este modelo? Obviamente una separación total entre lógica de negocio y presentación. A esto se le pueden aplicar opciones como el multilinguaje, distintos diseños de presentación,.. etc sin alterar la lógica de negocio. La separación de capas como presentación, lógica de negocio, acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y más fácilmente mantenibles, lo que al final resulta en un ahorro de tiempo en desarrollo en posteriores proyectos.

En este caso, mediante Struts se aplica el MVC en toda una aplicación Web convencional

Arquitectura de Sistema

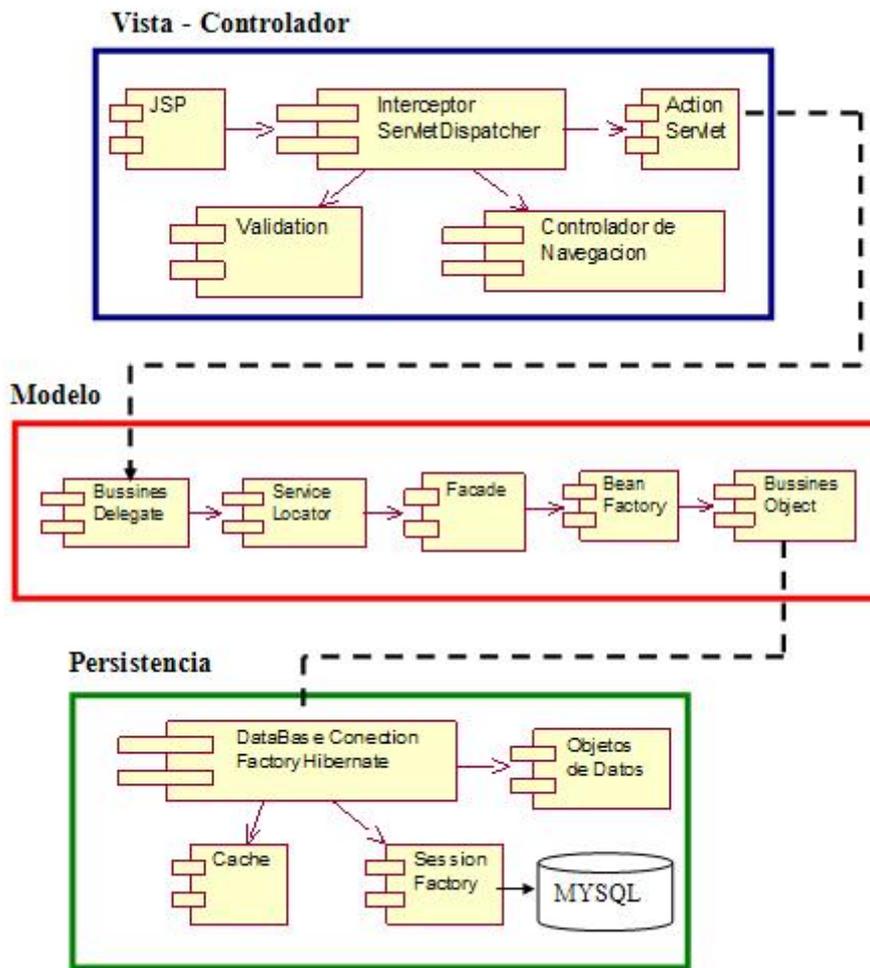


Figura 4.2 Arquitectura de Sistema de Gimnasio

Diseño De Capa de Negocio

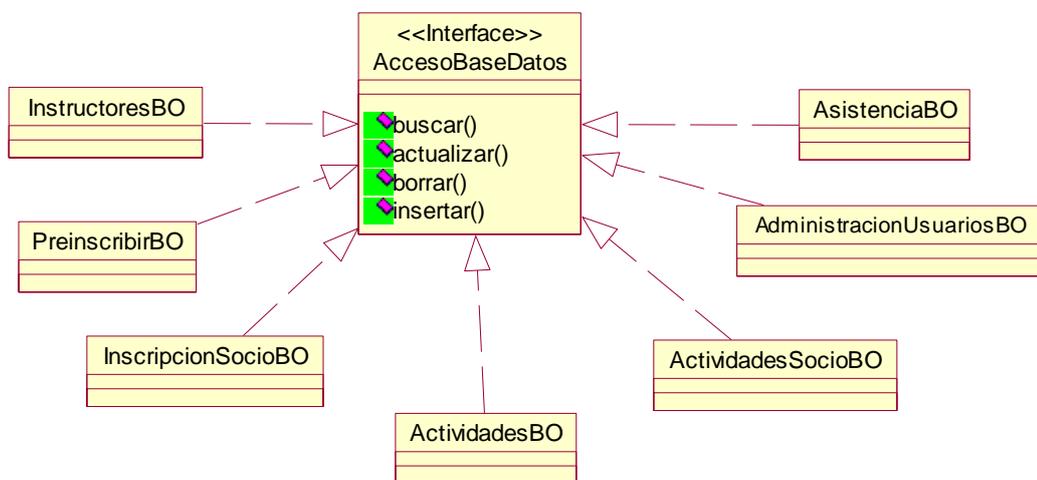


Figura 4.3 Capa de Diseño

Implementación de Caso Práctico bajo Software Libre

Seguridad de Sistema



Figura 4.4 Seguridad Sistema

CREATED BY NEEVIA DOCUMENTERGE TRIAL VERSION (WWW.NEEVIA.COM)

Diseño de la Capa de Persistencia

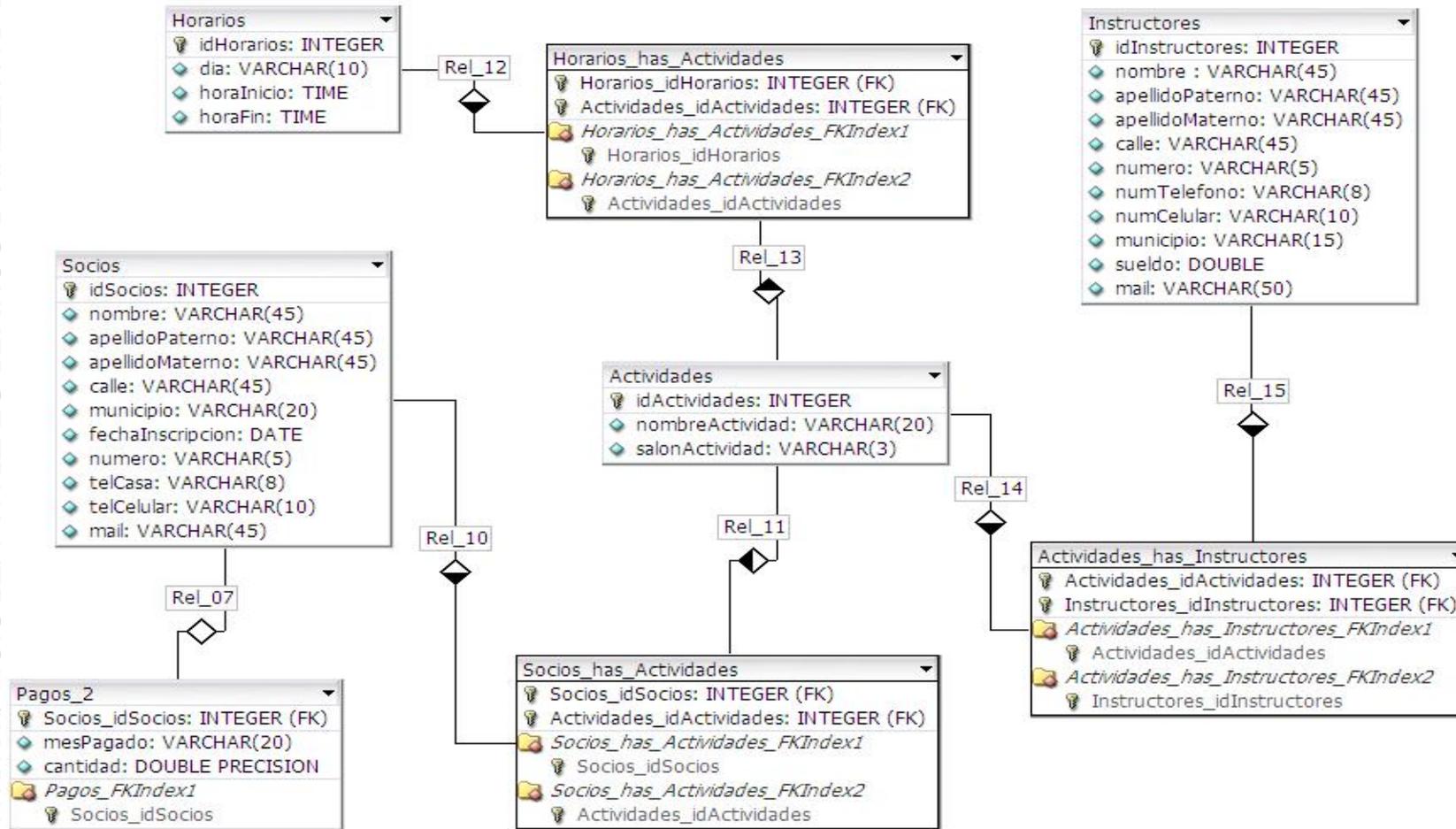


Figura 4.5 Capa de Persistencia

Implementación de Caso Práctico bajo Software Libre

Del diagrama entidad – relación, se obtienen las clases de persistencia que se necesitan para trabajar con Hibernate, a continuación se muestra el diagrama de clases de este modelo.

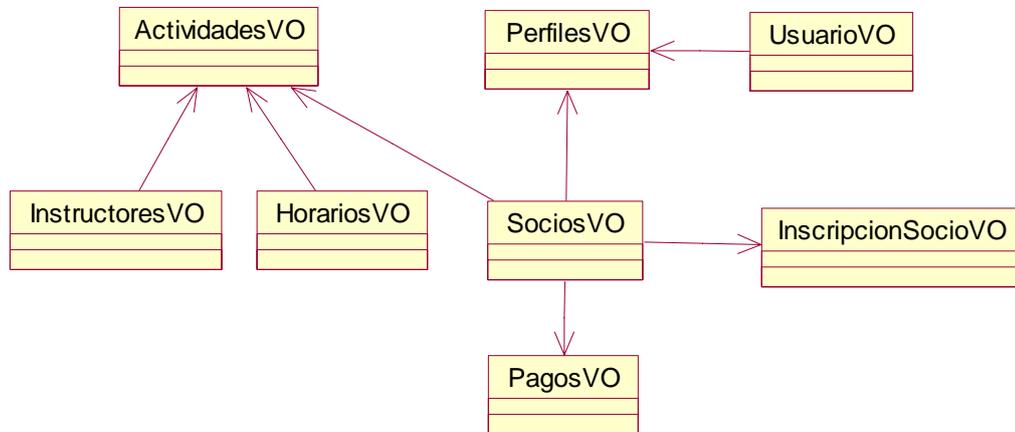


Figura 4.6 Modelado de Objetos de Hibernate

Diseño de Vista

Página Principal y de Autenticación

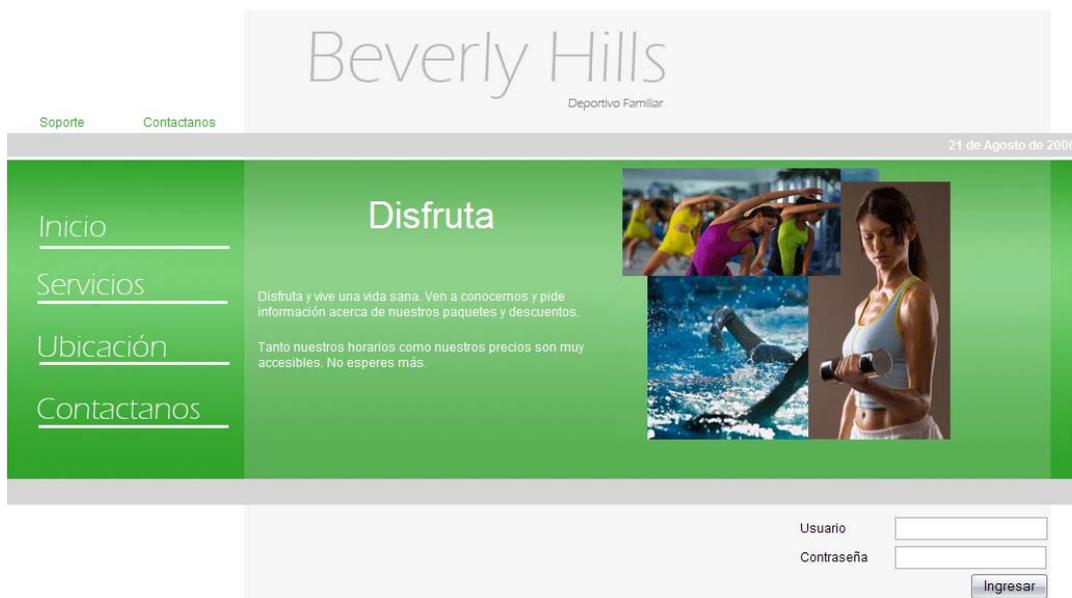


Figura 4.6 Página Principal

Página de Presentación de Información:



Figura 4.7 Presentación de Información

Página de Formulario de Preinscripción de un nuevo cliente:



Figura 4.8 Formulario de Preinscripción

Implementación de Caso Práctico bajo Software Libre

Cabe señalar que en el desarrollo de un sistema, el diseño es fundamental; el diseño gráfico no significa hacer un dibujo o, una imagen o crear una fotografía. Significa mucho más que todos esos elementos, aunque para poder conseguir, comunicar y transmitir visualmente un mensaje de forma efectiva, el diseñador debe conocer los diferentes recursos gráficos de los que dispone, junto con la imaginación, experiencia, buen gusto y el sentido común necesarios para combinarlos de forma correcta y adecuada; fue así que para esta aplicación se buscó un diseño que fuera agradable al usuario.

The Java logo is centered in the background, featuring a blue coffee cup with two red steam lines rising from it. The word "Java" is written in a large, light red font below the cup.

Capítulo 5

Programación

Elementos de Programación

Herramienta de desarrollo

Lenguaje de programación

Contenedor de servlets

Codificación Vista

Codificación Controlador

Codificación Modelo

Programación

Elementos de Programación

JSP. JavaServer Pages combinan HTML con fragmentos de Java para producir páginas Web dinámicas. Una página JSP es archivo de texto simple que consiste en contenido HTML o XML con elementos JSP. Cuando un cliente pide una página JSP del sitio web y no se ha ejecutado antes, la página es inicialmente pasada al motor de JSP, el cual compila la página convirtiéndola en Servlet, la ejecuta y devuelve el contenido de los resultados al cliente.

Utilizar JSP presenta varias ventajas sobre el uso de PHP, a continuación se presentan algunas:

1. Permite de forma más intuitiva una separación entre presentación y lógica. Con PHP se puede conseguir también, pero es algo menos intuitivo.
2. La programación de JSP/Servlet utiliza Java como herramienta lo cual tiene ventajas al ser un entorno más orientado al desarrollo de aplicaciones (las comprobaciones estáticas de tipos, herramientas de desarrollo como netbeans o eclipse, API más genéricas,).
3. Con Java (concretamente J2EE) se tiene una plataforma de desarrollo empresarial, con una inmensa cantidad de patrones de diseño, integración de estándares (Web Services, XML, seguridad, etc.), herramientas de desarrollo, gestores de objetos, frameworks de persistencia y presentación, etc.
4. Se publica 10 veces más libros sobre Java y todas sus tecnologías que sobre otros lenguajes de programación.

Struts. Framework para aplicaciones web java que implementa el modelo MVC, provee un conjunto de clases y TAG-LIBS que conforman el Controlador, la integración con el Modelo (o lógica de negocio) y facilitan la construcción de vistas. Struts es una plataforma sobre la que se monta la lógica de negocio, y esta plataforma nos permite dividir la lógica de la presentación entre otras cosas.

Spring Framework. Spring es un framework de aplicaciones Java/J2EE desarrollado usando licencia de OpenSource. Es potente en cuanto a la gestión del ciclo de vida de los componentes y fácilmente ampliable. Es interesante el uso de programación orientada a aspectos (IoC). Ofrece un ligero contenedor de bean para los objetos de la capa de negocio, DAOs y repositorio de Datasources JDBC y sesiones Hibernate. El objetivo de Spring es no ser intrusivo, aquellas aplicaciones configuradas para usar beans mediante Spring no necesitan depender de interfaces o clases de Spring, pero obtienen su configuración a través de las propiedades de sus beans.

Hibernate. Es una herramienta ORM (Object to Relational Mapping). Provee la capacidad de almacenar objetos en una base de datos utilizando un lenguaje de programación orientado a objetos, ocultando los detalles a la aplicación. Funciona como un servicio de almacenamiento para aplicaciones para guardar y recuperar objetos Java desde y hacia bases de datos relacionales.

Herramienta de desarrollo

Eclipse. Entorno integrado de desarrollo

Lenguaje de programación

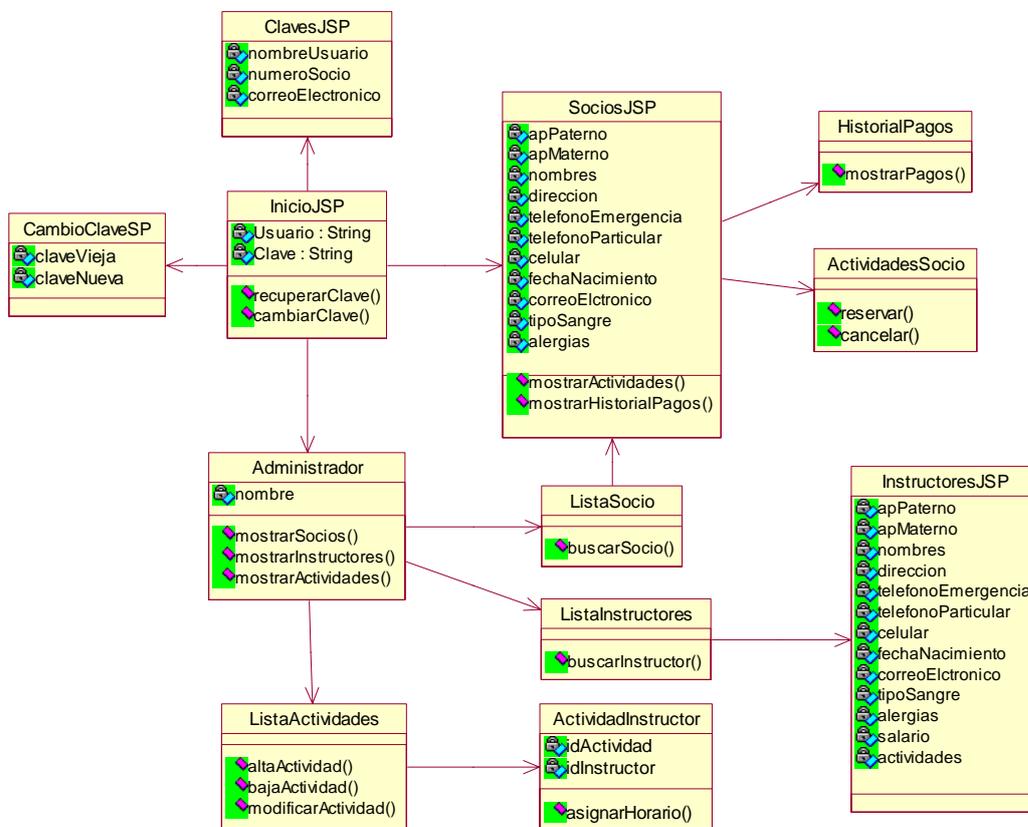
Java j2ee

Contenedor de servlets

Tomcat

Codificación Vista

Diagrama de Clases de Vista



Codificación Controlador

Struts-config.xml

```
<action-mappings>
  <action path="/inicio"
    type="com.gym.mx.action.inicioAction "
    name="incioGym"
    scope="session"
    validate="false">
    <forward name="continue" path="/incio.jsp"/>
    <forward name="error" path="/incio.jsp"/>
  </action>
</action-mappings>
```

```
</action>  
</action-mappings>
```

Codificación Modelo

Componente Service Locator.

Este componente propone la creación de un punto de entrada común para la instanciación de los servicios que se desean invocar. Esto permite cambiar fácilmente la implementación de los servicios hacia otros objetos sin necesidad de mantener el código que los invoca.

Codificación:

```
public class ServiceLocator {  
    static Logger logger = Logger.getLogger(ServiceLocator.class);  
    private ApplicationContext appContext;  
  
    public Object getService(ServletContext context,String beanName)  
        throws RotoException{  
        // The Application Context  
        Object object=null;  
        try{  
  
            this.appContext = WebApplicationContextUtils.getRequiredWebApplicationContext(context);  
  
            object=(Object) this.appContext.getBean(beanName);  
  
  
        }catch(Exception e){  
            logger.error("Error no se pudo instanciar el servicio requerido");  
            new RotoException("Ocurrio un error en la aplicacion, para detalles verificar log Error");  
        }  
        return object;  
    }  
}
```

Componente Business Delegate.

A este patrón se le da un uso particular para desarrollar los servicios de lógica de negocio; se encarga de crear, gestionar los objetos de lógica de negocio y de proporcionar un interfaz de los métodos de negocio que se pueden utilizar.

Codificación:

Implementación de Caso Práctico bajo Software Libre

```
public final class CommonBusinessDelegate {
    static private ServletContext servletContext;
    static Logger logger = Logger.getLogger(CommonBusinessDelegate.class);
    public CommonBusinessDelegate(){}

    public CommonBusinessDelegate(ServletContext context){
        if(servletContext==null){
            servletContext=context;
        }
    }

    public Object executeSelect(String beanName,java.lang.Object object)throws Exception{
        try{
            InterfaceCommonDao interfaceDao=(InterfaceCommonDao)(new ServiceLocator()).getService(servletContext,beanName);
            return(interfaceDao.select(object));
        }catch(Exception e){
            throw new Exception("Ocurrio un error en la aplicacion, para detalles verificar log Error");
        }
    }

    public Object executeInsert(String beanName,java.lang.Object object)throws Exception{
        try{
            InterfaceCommonDao interfaceDao=(InterfaceCommonDao)(new ServiceLocator()).getService(servletContext,beanName);
            return(interfaceDao.insert(object));
        }catch(Exception e){
            throw new Exception("Ocurrio un error en la aplicacion, para detalles verificar log Error");
        }
    }

    public Object executeUpdate(String beanName,java.lang.Object object)throws Exception{
        try{
            InterfaceCommonDao interfaceDao=(InterfaceCommonDao)(new ServiceLocator()).getService(servletContext,beanName);
            return(interfaceDao.update(object));
        }catch(Exception e){
            throw new Exception("Ocurrio un error en la aplicacion, para detalles verificar log Error");
        }
    }

    public Object executeDelete(String beanName,java.lang.Object object)throws Exception{
        try{
            InterfaceCommonDao interfaceDao=(InterfaceCommonDao)(new ServiceLocator()).getService(servletContext,beanName);
            return(interfaceDao.delete(object));
        }catch(Exception e){
            throw new Exception("Ocurrio un error en la aplicacion, para detalles verificar log Error");
        }
    }
}
```

Componente: Facade.

Presenta una interface única para un subsistema o un conjunto de componentes de negocio.

Codificación:

```
package com.gym.mx.common.services.dao;
/*
 * InterfaceDao
 * Autor:@Wendy Mayen
 * Date: Mayo 2006
 * Others:
 */
import com.gym.mx.exception.GymException;

public interface InterfaceCommonDao {

    public java.lang.Object select(java.lang.Object object)throws GymException;
    public java.lang.Object insert(java.lang.Object object)throws GymException;
    public java.lang.Object update(java.lang.Object object)throws GymException;
    public java.lang.Object delete(java.lang.Object object)throws GymException;
}
```

Implementación de Caso Práctico bajo Software Libre

```
<bean id="expedienteTx"
  class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean">
  <property name="transactionManager" ref="transactionManager" />
  <property name="target" ref="expedienteImpl" />
  <property name="transactionAttributes">
    <props>
      <prop key="get*">PROPAGATION_REQUIRED,readOnly</prop>
      <prop key="find*">PROPAGATION_REQUIRED,readOnly</prop>
      <prop key="load*">PROPAGATION_REQUIRED,readOnly</prop>
      <prop key="store*">PROPAGATION_REQUIRED</prop>
    </props>
  </property>
</bean>
```

Componente: Cache de Segundo Nivel.

Con una cache de segundo nivel los problemas de actualizaciones concurrentes entre sesiones. La caché de segundo nivel se sitúa al mismo nivel que el objeto *SessionFactory* de Hibernate, recogiendo y coordinando los objetos con los que trabajan las diferentes sesiones.

Una caché de segundo nivel, alivia de la responsabilidad de almacenar datos temporales, a la caché de primer nivel; responsabilidad que, por otra parte, era la causa de que los modelos de sesión por transacción de aplicación y de sesión por unidad de trabajo pudiera alentar las aplicaciones.

```
<bean>
  <property name="hibernateProperties">
    <props>
      <prop key="hibernate.dialect">
        net.sf.hibernate.dialect.SQLServerDialect
      </prop>
      <prop key="hibernate.show_sql">true</prop>
      <prop key="hibernate.cache">
        org.hibernate.cache.OSCache
      </prop>
    </props>
  </property>
</bean>
```

Componente: Session Factory Acceso a Base de Datos.

Un session Factor es un almacén de datos. Pueden usarse múltiples bases de datos creando múltiples archivos de configuración XML y creando múltiples objetos.

Presenta una interface única para un subsistema o un conjunto de componentes de negocio.

Codificación:

Implementación de Caso Práctico bajo Software Libre

```
<bean id="dataSource"
  class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName">
    <value>com.microsoft.jdbc.sqlserver.SQLServerDriver</value>
  </property>
  <property name="url">
    <value>
      jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=Expediente
    </value>
  </property>
  <property name="username">
    <value>sa</value>
  </property>
  <property name="password">
    <value>sa</value>
  </property>
</bean>
```

Cuando se tiene una metodología a seguir, la programación de un sistema debe de ser un proceso relativamente rápido y sencillo, tomando en cuenta que con el diseño se han establecido los elementos necesarios para que la tarea del programador implique un bajo costo de esfuerzo y tiempo. De hecho, las herramientas, patrones de diseño, y demás componentes han sido establecidos anteriormente por el diseñador.

En esta etapa, se deja el sistema preparado para que el personal encargado de realizar las pruebas comience con su trabajo, contemplando las pruebas de aplicación, las pruebas de volumen y las pruebas de usuario. Es importante dejar claro que un programador jamás tiene por qué realizar las pruebas de lo que él mismo construyó.

Capítulo 6

Implementación



Diagrama de Arquitectura .

Requerimientos de Hardware y Software

Usuarios creados en el sistema Operativo

Acciones de Seguridad

Recomendaciones para la operación del sistema

Pruebas de operación

Entregables de los procesos de codificación, prueba e instalación

Formato de satisfacción del cliente

Implementación

Diagrama de Arquitectura.

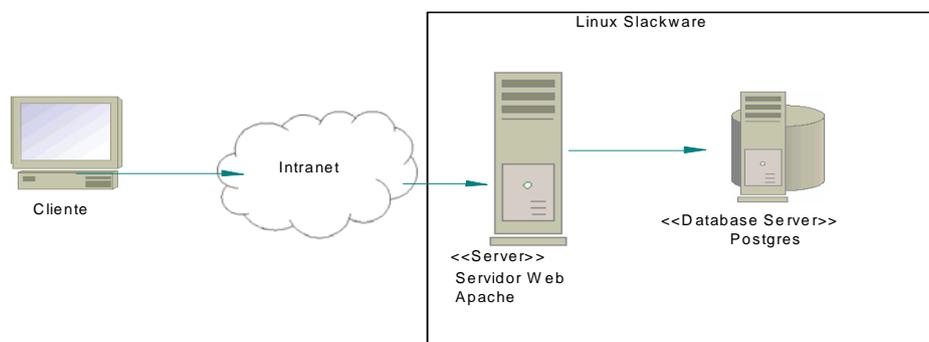


Figura 6.1 Diagrama de plataformas y elementos técnicos

Requerimientos de Hardware y Software

Componente	Hardware	Software
Cliente	<ul style="list-style-type: none"> • PC Multiplataforma • Procesador Intel Pentium III o equivalente de 996MHz • 256 MB RAM 	<ul style="list-style-type: none"> • Sistema operativo (Windows, Mac, Unix, Linux, etc) • Navegador de Internet (FireFox, Explorer, Netscape) con las últimas actualizaciones o SP
Web Server	<ul style="list-style-type: none"> • PC Multiplataforma • Procesador Intel Pentium IV o equivalente de 996MHz • 256 MB RAM 	<ul style="list-style-type: none"> • Servidor Web Apache • Sistema operativo Linux Slackware version 10 • JAVA J2SDK 1.4
Manejador de Base de Datos		<ul style="list-style-type: none"> • PostgreSQL

Usuarios creados en el sistema Operativo

Usuarios	Operaciones	Área Operativa.
Administradores	Validar parches de sistema operativo y aplicaciones instaladas, vigilar la seguridad del sistema, monitoreo	Administradores
Web Server	Redireccionamiento de todas las peticiones al Portal hacia el Application Server	Dirección de sistemas
Usuario Base Datos	Administración de la base de	Dirección de Sistemas

	datos del sistema Encuestas.	
--	------------------------------	--

Acciones de Seguridad

1. Cerrar puertos de red
2. Bloquear servicios de red
3. Utilizar protocolos seguros como SSH
4. Utilizar firewalls
5. Control de usuarios
6. Permisos de usuarios
7. Permisos de grupos
8. Criptografía
9. Huellas digitales

Recomendaciones para la operación del sistema

1. Manejo adecuado de contraseñas.
2. Plan de respaldos.
3. Sustentabilidad sobre los datos
4. Mantenibilidad a la infraestructura
5. Monitoreo de funcionamiento de aplicación

Pruebas de operación

Si un sistema falla es porque tiene al menos un defecto. Es por ello que hay que realizar pruebas, con la finalidad de eliminar los defectos. Las pruebas las deben hacer un grupo no involucrado con el desarrollo del sistema y éstas se deben prever desde el inicio del proyecto.

En el proceso de pruebas, previsto desde el comienzo del sistema, se fueron desarrollando los diferentes tipos de pruebas.

- ✓ Las Inspecciones son vistas como un elemento de aseguramiento de la calidad.
- ✓ Las Pruebas de Unidad se conocen como las pruebas de caja blanca y las pruebas de caja negra, el sistema se va probando por componente de forma tal que cuando un componente queda bien validado, se prueba en conjunto con otro componente como caja negra.
- ✓ Las Pruebas de Integración son la pruebas que se realizan de la aplicación con todos los componentes ya formando una sola aplicación.
- ✓ Finalmente, el sistema se prueba con muestras y casos de prueba con datos reales y con un grupo especializado de colaboradores. Es importante hacer notar que parte de lo que se prueba en esta última fase es:
 - Recuperación ante fallas del sistema.
 - Pruebas de seguridad.
 - Pruebas de estrés.
 - Pruebas de performance.

Entregables de los procesos de codificación, prueba e instalación

Proceso	Entregable
Codificación	<ul style="list-style-type: none">✓ Código de la aplicación.✓ Documentación del programa.
Prueba	<ul style="list-style-type: none">✓ Escenarios de prueba (plan de pruebas) y prueba de datos.Resultados de las pruebas de los programas y del sistema.
Instalación	<ul style="list-style-type: none">✓ Manuales: Técnico, Usuario y Operación✓ Plan de capacitación de los usuarios.✓ Plan de instalación de HW y SW.✓ Plan de remodelación/acondicionamiento de localidades.

Proceso de Soporte

Al concluir con la puesta en producción del sistema, debe de haber un Centro de Soporte a Usuarios (conocido también como Centro de Información), es un grupo de personas que están en la capacidad de responder preguntas y asistir a los usuarios, dentro de una organización, en un amplio rango de necesidades en computación.

Implementación de Caso Práctico bajo Software Libre

Formato de satisfacción del cliente

El realizar una encuesta al cliente o en su caso a usuarios del sistema, tiene como objetivo conocer el proceso del desarrollo del sistema, las condiciones y los resultados del que se obtuvieron desde otra perspectiva diferente; de tal forma que se puedan identificar los factores que incidan en un mejor servicio.

SISTEMA PARA CADENA DE GIMNASIOS							
ENCUESTA DE SERVICIOS							
Nombre: _____				Fecha: / /			
Responda: E = Excelente, B = Bueno, R = Regular, M = Malo				E	B	R	M
1	¿Cómo califica al tiempo que se le asignó al desarrollo del proyecto?						
2	El nivel de conocimientos de la persona que atendió su solicitud fue						
3	¿Cómo evalúa la actitud de la persona que le atendió?						
4	¿Cómo califica la disponibilidad para atender sus dudas o comentarios?						
5	¿Cómo fue la comunicación con Usted para informarlo del avance del proyecto?						
6	¿Cómo califica de forma general la atención y solución de su solicitud?						
7	¿Cuál es el nivel de cumplimiento del objetivo de su solicitud, de acuerdo a lo solicitado?						
8	Califique el grado de cumplimiento de la fecha de entrega de la solución de su solicitud						
9	Califique el grado de cobertura del documento de Plan de Trabajo de su requerimiento						
10	¿Cuál fue el nivel de claridad del Documento de Plan de Trabajo que se entregó para la solución de su requerimiento?						
11	¿Cómo califica el manual y/o capacitación requerida?						
Responda: S = SI o N = NO				S	N		
12	¿Se le entregó un documento de plan de trabajo sobre su requerimiento?						
13	Si hubo modificaciones al documento original y se rechazó, ¿Recibió un nuevo documento actualizado con los Cambios?						
14	¿Recibió la capacitación requerida?						
15	¿Se entregó la documentación necesaria para entender y operar la nueva funcionalidad solicitada?						
16	¿El requerimiento representó fallas al momento de empezar a utilizarlo en producción?						
17	¿Se atendieron con prontitud las fallas reportadas?						
18	¿Se recibió alguna documentación sobre la solución realizada para la falla reportada?						
Comentarios Adicionales							

Implementación de Caso Práctico bajo Software Libre

Los proyectos de sistemas de información en el mundo empresarial requieren tanto una herramienta de software adecuada, que responda a sus necesidades de gestión específicas, como una buena metodología de implantación de soluciones tecnológicas, que colabore al éxito de la misma.

En este momento, en que el sistema se ha puesto en producción y la respuesta del cliente es satisfactoria se puede considerar que el proyecto ha sido exitoso; sin dejar de un lado que los sistemas de información, así como tienen un ciclo de vida de desarrollo, también tienen un ciclo de vida productivo, que dependerá de las necesidades que el cliente vaya teniendo en su desarrollo empresarial, hasta que llegue el momento en el que su sistema sea obsoleto, tendremos en ese caso que empezar con un nuevo ciclo de desarrollo, contemplando las nuevas herramientas, metodologías, lenguajes que en su momento haya, considerando siempre lo más adecuado para la satisfacción del cliente.

Conclusiones

Como primer conclusión, me gustaría mencionar los resultados que se obtuvieron a partir de los objetivos que se plantearon al comienzo del presente trabajo.

En este proyecto se siguió la metodología RUP, que a mi punto de vista es de las más flexibles y utilizadas en el mercado, sin embargo, no es la panacea, debemos siempre tener claro que las necesidades de las empresas son de diferente índole, y factores como el tiempo son clave en la selección de alguna de las existentes, podrían llegarse a dar el caso en que ninguna de las metodologías conocidas nos ayuden o se adecuen a nuestro sistema, entonces tendríamos que establecer alguna nosotros mismos; es decir, debemos tener una base sobre la cual trabajar para obtener la respuesta deseada considerando tiempos, costo, esfuerzo, y sobre todo satisfacción al cliente para tener sistemas de calidad.

Yo creo que todo lo que se aprendió en el Diplomado es de suma importancia y de hecho son las bases para construir software de calidad, sin embargo, la experiencia profesional es algo indispensable y por lo tanto para concluir el ciclo, es obligatorio vivir la experiencia en la vida cotidiana, enfrentarse a los problemas y a la solución de los mismos cuando se pasa de la etapa de desarrollo a la etapa de producción, aprender a tratar al cliente, y satisfacer sus necesidades es algo que sólo da la práctica.

Uno de los aspectos expuestos en este trabajo y de singular importancia es el software libre. Software que a mi punto de vista, sigue sorprendiendo a mucha gente cuando descubre que un autor no sólo permite que su trabajo sea copiado y redistribuido libremente, sino que además anima al usuario a hacer lo mismo. Creo que hay una tendencia muy marcada al uso de software libre y el por qué es una situación muy predecible: la calidad que ha alcanzado el software libre y sus bajos o nulos precios. Uno de los principales objetivos para cualquier organización, es obtener mayores beneficios a menores costos, y el software libre, bajo esta premisa, seguirá tomando fuerza día a día, sobre todo teniendo la seguridad de que muchas veces tienen más calidad que el software propietario o de licencia, por el que se pagan cantidades enormes de dinero.

El haber realizado este trabajo y el que mi campo laboral pertenezca a los sistemas, me ha permitido darme cuenta de la fuerza tan grande que día a día los sistemas informáticos van tomando en este mundo en donde la computación se ha vuelto parte fundamental en nuestra vida.

En los últimos años se hizo mucho por el desarrollo de sistemas cada vez más potentes de acuerdo a las cantidades de información que se manejan, mas adecuados a las necesidades de las empresas y de los usuarios, con interfaces agradables y con tecnologías de vanguardia; pero esto no quita el hecho de que aún haya mucho y variado por hacer.

Los tecnología avanza día a día a pasos agigantados y la computación en conjunto con los sistemas informáticos deben de mantener el paso de este crecimiento. En pocos años, hemos sido testigos del auge que han tenido las aplicaciones y cómo han ido evolucionando desde las aplicaciones de escritorio hasta aplicaciones Web, de dispositivos móviles, etc., de la importancia y del peso que éstas están tomando alrededor del mundo, sin embargo, a pesar de todo ese crecimiento, de la tecnología y de las facilidades que hay para desarrollar sistemas de calidad, resulta desalentador darse cuenta que en un país como el de nosotros, los sistemas no han tomado aún la importancia y la seriedad que tienen.

Implementación de Caso Práctico bajo Software Libre

Con el software libre se han desarrollado infinidad de aplicaciones de mejor calidad incluso que las aplicaciones realizadas con software de licencia, eso lo podemos ver con las miles de aplicaciones publicadas día a día en Internet, cuando vemos el tipo de pagina que se despliega, antes solo se hablaba de html; ahora podemos ver extensiones como .jsp, .php, .asp, jsf, etc. etc; de las cuales, la mayoría siempre es de software libre.

Las empresas centradas en la Web están cambiando la forma en que se proporciona el acceso a sistemas distribuidos altamente eficientes que funcionan a lo largo y ancho de la empresa. La forma antigua de construir sistemas empresariales ya no puede resolver los requerimientos actuales de la informática. Por lo que para poder cumplir con la demanda de información instantánea que impone día y noche miles de usuarios concurrentes, los desarrollos de sistemas tienen que evolucionar día a día, la computación es una carrera para toda la vida, y cada nuevo día salen al mercado miles de posibilidades y tecnologías, hay que ser concientes de esto y tratar de seguir ese crecimiento.

Bibliografía

- 📖 AGUILAR Joyanes Luis, Zahoero Ignacio. "Estructuras de datos con Java"
McGraw-Hill ISBN: 8448156315
- 📖 BARRETT Daniel J., BYRNES Robert G., SILVERMAN Richard: "Linux Security Cookbook",
O'Reilly & Associates, Inc., United States of America, 2003, p. 332, ISBN 0-596-00391-9
- 📖 BLACK, Ulyses. "Redes de computadores protocolos, normas e interfaces"
McGraw-Hill, 1991. México
- 📖 CLARKE, Justin: "Network Security Tools"
O'Reilly & Associates, Inc., United States of America, 2005, p. 352, ISBN 0-596-00794-9
- 📖 COLMENAR A. "Gestión de Proyectos con Microsoft Project 2007"
Ra-ma, ISBN: 8478978062
- 📖 GALBRATH Ben, KALKNER Jayson, IRANI Romín. "Desarrollo Web con JSP"
Anaya Multimedia, ISBN-13: 9788 441513525
- 📖 GIL Rubio, Fco. Javier: "Creación de sitios Web con PHP 4."
McGraw-Hill, Madrid, 2001, p. 547, ISBN 84-481-3209-2
- 📖 GUTIÉRREZ Juan Diego: "MySQL"
Anaya Multimedia-Anaya Interactiva, Madrid, 2004, p. 304, ISBN 84-415-1683-9
- 📖 LERDOF rasmus, TATROE Kevin: "PROGRAMING PHP",
O'Reilly & Associates, Inc., United States of America, 2002, P.524, ISBN 1-56592-610-2
- 📖 LUMENS Chris, CANTRELL David, JOHNSON Logan: "Slackware Linux Essentials.", Ed.
2da., Slackware Linux, Inc., Canada, 2005, p. 284, ISBN 1-57176-338-4
- 📖 KAY, Trevor: "Linux+ Certification Bible."
Hungry Minds, Inc., Nueva York, 2002, p. 721, ISBN 0-7645-4881-6
- 📖 KENNEDY Bill, MUSCIANO Chuck: "HTML & XHTML: The Definitive Guide, 5th Edition", Ed.
5ta. O'Reilly & Associates, Inc., United States of America, 2002, p. 670, ISBN 0-596-00382-X
- 📖 MASLAKOWSKI Mark, "Aprendiendo MySQL en 21 días"
Pearson Educación, México, 2001, p. 534, ISBN 970-26-0036-7
- 📖 MOHAMMED J. kabir: "La Biblia de Servidor Apache".
Anaya Multimedia-Anaya Interactiva, Madrid, 1999, p. 688, ISBN 84-415-0807-0
- 📖 MORITSUGU Steve: "UNIX Serie práctica.". Trad. Maribel Martínez Moyano.
Pearson Educación, Madrid, 2000, p. 1032, ISBN 84-205-29508

Implementación de Caso Práctico bajo Software Libre

- 📖 NIEDERST Jennifer: "HTML Pocket Reference, Second Edition", Ed. 2da. O'Reilly & Associates, Inc., United States of America, 2002, p. 104, ISBN 0-596-00296-3
- 📖 SHEMA, Mike: "Claves Hackers de Sitios Web"
McGraw-Hill, México, 2005, p. ISBN 0-07-222784-2
- 📖 WELLING Luke, THOMSON Laura: "Desarrollo Web con PHP y MySQL"
Anaya Multimedia-Anaya Interactiva, Madrid, 2003, p. 909, ISBN 84-415-1569-7
- 📖 ZAWODNY, Jeremy D.: "MySQL avanzado"
Anaya Multimedia-Anaya Interactiva, Madrid, 2004, p. 336 , ISBN 84-415-1759-2

- 📖 [HTTP Server] Apache Web Server
<http://httpd.apache.org/>
- 📖 [PHP] PHP Net
<http://www.php.net/>
- 📖 [MySQL] Mysql Developer Zone
<http://www.mysql.org/>
- 📖 [LINUX] Linux Online
<http://www.linux.org/>
- 📖 [LC] Características de Linux
<http://www.fismat.umich.mx/~elizalde/curso/node156.html>
- 📖 [DTIC-SSL]Departamento de Tratamiento de la Información y Codificación
<http://www.iec.csic.es/criptonomicon/ssl.html>
- 📖 [WEB DESIGN] W3 Schools. Web Developer
<http://www.w3schools.com/>
- 📖 [OS] The Open Source
<http://www.opensource.org/osd.html>
- 📖 [3C] Community Created Content
http://turre.com/images/stories/books/webkirja_koko_optimoitu2.pdf
- 📖 [The Rise] THE RISE OF OPEN SOURCE LICENSING
http://pub.turre.com/openbook_valimaki.pdf
- 📖 [PFOSS] Perspectives on Free and Open Source Software
<http://mitpress.mit.edu/books/chapters/0262562278.pdf>
- 📖 [Proyecto GNU] El proyecto GNU y el software libre
<http://biblioweb.sindominio.net/pensamiento/softlibre/>

Implementación de Caso Práctico bajo Software Libre

-  [Free Software] Software libre para una sociedad libre
http://download.savannah.gnu.org/releases/rms-essays/free_software.es.pdf
-  [Libro Blanco] Libro Blanco del Software Libre en España (III)
http://libroblanco.com/joomla/document/III_libro_blanco_del_software_libre.pdf
-  [OSO] Open Source Origins
http://www.charlesriver.com/resrcs/chapters/1584503475_1stChap.pdf
-  [CRIPTO] Introducción a la Criptografía
<http://pateame.fciencias.unam.mx/cripto/notas/cripto.pdf>
-  [GPG] THE GNU PRIVACY GUARD
<http://www.gnupg.org/>