



---

---

**UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES**

**ANÁLISIS Y DISEÑO DE UN SISTEMA  
PARA EL MONITOREO DEL  
MEDICAMENTO EN UNA FARMACIA.**

**TRABAJO ESCRITO**

POR LA MODALIDAD DE  
SEMINARIOS Y CURSOS DE ACTUALIZACIÓN  
Y CAPACITACIÓN PROFESIONAL  
QUE PARA OBTENER EL TÍTULO DE :

**INGENIERO EN COMPUTACIÓN**

**P R E S E N T A :**

**XAVIER AUGUSTO OLIVARES CHÁVEZ**

**DIRECTOR DE LA TESIS: M.TI. OMAR MENDOZA GONZÁLEZ**

MEXICO, D. F.

2008



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

# DEDICATORIAS



*La suerte favorece solo a las mentes preparadas.*

**A mi esposa Lorna y mi hijo Luís Javier:**

Por que su presencia es un gran motivo de querer superarme día con día, y que me han enseñado otra faceta del amor.

**A mi padre el Dr. Javier Olivares Alvarado:**

Por que me enseñaste la importancia del conocimiento y me alentaste en cada uno de los proyectos que inicie y se que siempre te sentiste orgulloso de mi.

**A mi madre la Sra. Petra Chávez Castillo:**

Por que me has enseñado el valor de la rectitud y la constancia. Y todos los días siempre me das muestra de tu amor.

**A mi hermana la Dra. Ana Laura Olivares Chávez:**

Por que siempre me has apoyado en las buenas y en las malas. Con tu ejemplo me has mostrado la importancia del estudio.

**A mis padrinos, el Dr. Félix de León y la Profa. Rosalba M. Olivares:**

Por el apoyo que siempre me han brindado, y que han sido parte fundamental en este trayecto llamado vida.

**A mis abuelos paternos y maternos:**

Por que siempre me mostraron que no hay fuerza más grande que la unidad de la familia.

**A toda mi familia:**

Por que gracias a la convivencia que he tenido con cada uno de los miembros de esta maravillosa familia he aprendido mucho.

# AGRADECIMIENTOS



## **A Dios:**

Por que me ha permitido vivir en este tiempo y con las personas que me rodean, me ha puesto pruebas que me han servido para madurar y me da la fuerza para superar las adversidades.

## **A la UNAM:**

Por que es un centro de conocimientos maravillosos y me ha brindado muchas oportunidades y experiencias. En sus instalaciones han pasado muchos momentos felices de mi vida.

## **Al MTI. Omar Mendoza González:**

Por la confianza y el tiempo que dedico a este proyecto, y que en el momento más difícil de este proyecto me ofreció su ayuda.

## **Al M. en C. Marcelo Pérez Medel:**

Por su apoyo que siempre me ha dado como jefe de carrera y como profesor, siempre me apoyo en los momentos más complicados de este proyecto.

## **A la Quím. Laura Mata Montiel:**

Por brindarme su apoyo y su confianza, también por darme consejos que me han sido útiles tanto personal como profesionalmente.

## **Al M. en I. Carlos Omar de la Rosa:**

Por brindarme tu amistad y apoyo, siempre me alentaste a concluir este proyecto. También gracias por la confianza que me has dado.

## **A mis compañeros de la Universidad:**

Que hicieron muy amenos los años de estudio en Aragón y que juntos nos apoyamos para terminar la carrera.



## INTRODUCCIÓN

### CAPÍTULO I REPORTE DEL DIPLOMADO

1.1. Módulo I Fundamentos para el desarrollo orientado al Web. ....	1
1.1.1. Introducción a servicios de Internet .....	2
1.1.2. Tecnología en tres capas.....	3
1.1.2.1. Capa de presentación.....	4
1.1.2.2. Capa de negocio .....	5
1.1.2.3. Capa de datos.....	5
1.1.3. Sistemas operativos .....	6
1.1.4. Elementos de software para un proyecto de desarrollo en WWW .....	7
1.2. Módulo II Análisis y diseño con UML.....	10
1.2.1. Ingeniería de requerimientos .....	11
1.2.2. RUP .....	12
1.2.3. La ingeniería de requerimientos y sus actividades principales.....	12
1.2.4. Técnicas y herramientas .....	13
1.2.5. Enfoque de orientación de objetos.....	13
1.2.6. Herramientas de análisis y diseño con UML.....	17
1.2.7. Diseño de la base de datos.....	20
1.3. Módulo III Introducción a la programación orientada a objetos con Java .....	21
1.3.1. Características de Java.....	22
1.3.2. El lenguaje de programación Java .....	24
1.3.3. La plataforma Java.....	24
1.3.4. Tipos de aplicaciones en Java.....	25
1.3.5. El JDK .....	25
1.3.6. Recomendaciones de programación .....	26
1.3.7. Sintaxis del lenguaje.....	26
1.3.8. Constructores .....	28
1.3.9. Objetos.....	28
1.4. Módulo IV Capa de Usuario.....	29
1.4.1. Proceso de diseño .....	31
1.4.1.1. Entender quién usará el sistema y para qué.....	31
1.4.1.2. Utilizar un estándar o normatividad de la empresa u organización.....	32



1.4.1.3. Bosquejar un diseño .....	32
1.4.1.4. Crear un prototipo.....	33
1.4.1.5. Evaluación con los usuarios .....	33
1.4.1.6. Construcción.....	33
1.4.2. Criterios de usabilidad .....	34
1.4.3. Estructura y organización de las páginas de la interfaz.....	35
1.4.4. HTML.....	35
1.4.5. Estructura de un documento HTML.....	36
1.5. Módulo V Capa de Datos .....	37
1.5.1. Modelos de datos .....	38
1.5.2. Clasificación de los SGBD .....	38
1.5.3. Concepción de una base de datos .....	39
1.5.4. SQL para programadores.....	39
1.5.4.1. Sentencias de selección o consultas .....	40
1.5.4.2. Sentencias CREATE, DROP, INSERT, DELETE y UPDATE .....	41
1.5.5. JDBC API.....	42
1.6. Módulo VI Reglas de Negocio .....	44
1.6.1. Definición de reglas del negocio .....	45
1.6.2. Tipos de reglas de negocio .....	45
1.6.3. Capas en un sistema Cliente/Servidor .....	46
1.6.4. Ubicación de las reglas de negocio.....	48
1.6.5. Arquitectura de aplicaciones Web con Java.....	49
1.6.6. JSP y Servlets .....	50
1.6.7. Modelo Vista Controlador .....	51
1.6.7.1. Técnica de desarrollo de aplicaciones utilizando una herramienta de productividad al desarrollo .....	52
1.6.7.2. Jakarta Struts: MVC en acción. ....	53

## **CAPÍTULO II ANTECEDENTES Y PLANTEAMIENTO DEL PROBLEMA**

2.1. Sistema Farmacéutico Morelos (SFM).....	55
2.2 Entorno histórico de la Farmacia.....	55



2.3. El problema de una organización Farmacéutica.....	59
2.4. Beneficios del proyecto .....	63
2.5. Antecedentes del proyecto.....	65
2.6. Objetivo .....	66
2.7. Alcance .....	66
2.8. Entregables .....	67

## **CAPÍTULO III PROPUESTA Y SOLUCIÓN**

3.1. Especificación de requerimientos.....	69
3.2. Especificaciones funcionales.....	70
3.3. Actores definidos para el sistema .....	71
3.4. Niveles de acceso. ....	73
3.5. Casos de Uso (CU) por área funcional .....	74
3.5.1 Diagramas de caso de uso.....	76
3.6. Requerimientos no funcionales .....	82
3.6. Requerimientos del medio ambiente .....	85

## **CAPÍTULO IV DISEÑO**

4.1. Documentación de Caso de Usos. ....	90
4.2. Diagrama de clases .....	124
4.3. Diagramas de secuencia.....	125
4.4. Arquitectura y diseño .....	129
4.4.1. Componentes .....	131
4.4.2. Modelo de interfase .....	133
4.4.3. Persistencia .....	139
4.4.4. Diseño de la Base de Datos .....	141
4.3. Mapa de navegación .....	151
4.4. Política de diseño.....	166

## **CONCLUSIONES**

## **BIBLIOGRAFÍA**



## INTRODUCCIÓN

A partir de la segunda mitad del siglo XVIII el mundo ha sido testigo de un crecimiento sin precedentes en lo tecnológico, económico y sociocultural. Con la llegada de la Revolución Industrial las organizaciones empezaron a crecer en tamaño y especialmente en complejidad, uno de los problemas más agudos era la asignación de los recursos disponibles.

Con el advenimiento de la revolución de las computadoras, que a partir de 1980 empieza fuertemente con el desarrollo de las computadoras personales, la sociedad adopta el nombre de “la sociedad del conocimiento, la sociedad que maneja la información”.

Actualmente las organizaciones destinan muchos recursos para obtener e interpretar la información que día a día genera la empresa y que es fundamental para mantenerse en competencia con otras instituciones. En consecuencia, es creciente la necesidad de crear soluciones que permitan la manipulación adecuada de la información. Soluciones que puedan maximizar la producción y reduzca lo más posible los costos.

Un elemento que se ha vuelto decisivo para lograr dicho objetivo es la utilización de tecnologías computacionales. Cada empresa es un caso individual, con variables estrictamente únicas y que deben ser analizadas escrupulosamente para la elección de una solución.

La computadora es una máquina electrónica capaz de realizar un gran número de cálculos en segundos y que permite compartir y usar diversos recursos electrónicos, es decir: “El valor de la computadora radica en su increíble velocidad y exactitud. En procesos donde el hombre no se concentra o no pone atención debida a causas de su naturaleza repetitiva, una computadora puede realizarlos durante toda la semana 24 horas del día”<sup>1</sup>.

---

<sup>1</sup> Enid Squire. *La computadora, un auxiliar indispensable*. México, Fondo Educativo Interamericano, 1982, pág 1





El software y hardware deben trabajar en conjunto para formar una solución integral a un problema determinado dentro de un sistema computacional. Las aplicaciones, compradas o desarrolladas deberán permitir manejar el hardware de una manera eficaz, facilitando su uso a los usuarios finales.

En la actualidad, como ya se ha mencionado, el mundo de las empresas ha incrementado su complejidad drásticamente. Éstas, haciendo uso de herramientas tecnológicas buscan crear proyectos que le permitan tener un manejo más controlado de los factores internos y externos que afecten a la organización.

Una de las herramientas que más han ayudado a las organizaciones son los sistemas de información. Un sistema de información es un “sistema automatizado que organiza los datos para producir información”<sup>2</sup>, también permiten conservar la información a través del tiempo y agilizar los procesos repetitivos que se llevan dentro de una empresa. Sin embargo, realizar un adecuado sistema computacional requiere de un gran esfuerzo y de la conjunción de especialistas de diferentes áreas.

El problema más grave al que se enfrentan los desarrolladores de sistemas son las soluciones, es decir, “... no hay soluciones correctas o equivocadas, sino sólo soluciones que son mejores o peores...”<sup>3</sup>. Por ello se hace indispensable tener un buen diseño de proyecto que consiga alcanzar una buena solución al problema que plantea la institución.

De lo anterior, surge la necesidad de tener una mejor preparación profesional. La especialización de los profesionistas del área de las Tecnologías de la Información se vuelve rigurosa ya que esta avanza de manera asombrosa. Por ello, estar actualizado en las metodologías y tecnologías de análisis y desarrollo de sistemas toma gran importancia, el saber usar nuevas tecnologías ayuda a encontrar mejores soluciones a menor costo.

---

<sup>2</sup> Hansen, Gary W; Hansen, James V. *Diseño y Administración de Bases de Datos*. España, Prentice Hall, 1998, pág 11

<sup>3</sup> Bernd Bruegge; Allen H. Dutoit. *Ingeniería de software orientado a objetos*. México, Prentice Hall, 2002, pág ix



El presente trabajo ha surgido ante la necesidad de encontrar una solución particular de un problema real. Por ello, presenta una de las metodologías para el desarrollo de sistemas, es conveniente señalar que se enfocará al modelo orientado a objetos para el desarrollo de un Sistema en Web.

El paradigma orientado a objetos ayuda mucho en la abstracción de la realidad, y al acercarnos a un Sistema computacional que refleje, en gran medida, la realidad de los procesos de una empresa se podrá encontrar una buena solución para ella.

Ahora bien, para tratar de obtener una buena solución a nuestra problemática y documentarla a razón de una exigencia didáctica y metodológica, he proyectado el presente trabajo con el siguiente orden de capitulados.

Para iniciar nos abocaremos al estudio teórico del desarrollo de Sistemas en Web orientado a objetos con Java. En el primer capítulo concentraré todos los módulos que estudié durante el diplomado impartido en la Facultad de Estudios Superiores de Aragón de la Universidad Nacional Autónoma de México cuyo título es: **Desarrollo de Sistemas en Web Orientado a Objetos con Java**, que forma la base teórica.

En este capítulo denominado reporte del diplomado presentaré los fundamentos que me sirvieron de pilares para la formación de un Sistema. El primer pilar concentra los conocimientos sobre Internet y los servicios que ofrece. El segundo se enfoca a la metodología de análisis y desarrollo orientado a objetos. Nuestro tercer pilar es el lenguaje de programación Java, lenguaje seguro y robusto.

Nuestro cuarto pilar se refiere a un patrón de diseño, la tecnología de tres capas: capa de usuario, donde escribiremos los elementos que conforman la interfaz que interactúa con el cliente; de datos, detallaremos el desarrollo de una buena base de datos y, las reglas del negocio, aquí expondremos la importancia y los elementos que conforman a esta capa.



En un segundo capítulo, antecedentes y planteamiento, detallo el problema que me ha dado el pretexto de escribir este trabajo. Presentaré los antecedentes que han llevado a la organización farmacéutica a buscar una solución mediante la creación de un Sistema en Web.

Se plantea la propuesta del proyecto que dará la posible solución al problema de la empresa farmacéutica, describiendo los beneficios, el objetivo, el alcance y los antecedentes de éste. Sin embargo, el estudio del caso no debe considerarse uso exclusivo del presente capítulo, en el desarrollo de Sistemas es común que la comprensión del problema, a detalle, se dé conforme se va obteniendo la solución de ésta.

Por lo que hace al tercer capítulo, una consecuencia del anterior, denominado propuesta y solución, donde presentaremos las especificaciones de los requerimientos de la solución. En este capítulo se empezará la conceptualización de nuestro sistema, se definen los actores que participan y qué procesos realizan, denominando todos los Casos de Uso por área funcional y los niveles de acceso de cada actor, así como las especificaciones no funcionales del sistema. Donde se abordará la seguridad, la usabilidad y el mantenimiento del sistema. En este capítulo solo se mencionan los Casos de Uso ya que, la documentación de los Casos de Uso se encontrará en el siguiente capítulo.

Finalmente en un cuarto capítulo, realizaremos el diseño de la solución entorno a todo lo estudiado anteriormente. Detallaremos cada Caso de Uso que encontramos en el capítulo anterior. Se sentarán las bases para el desarrollo del sistema a través del planteamiento de la arquitectura y diseño de los componentes, modelos de interfaces, persistencia y el diseño de la base de datos.

Se complementará el capítulo con los siguientes documentos: el diccionario de datos, el mapa de navegación y las políticas de diseño. Al término del trabajo reportaré las conclusiones que obtuve, producto de la experiencia al desarrollar esta solución a un problema real. Así, mediante esta acción mostraré la utilización de una metodología de desarrollo de sistemas computacionales orientado a objetos para la Web.



## **CAPÍTULO I REPORTE DEL DIPLOMADO**

### **1.1. MÓDULO I**

#### **FUNDAMENTOS PARA EL DESARROLLO ORIENTADO AL WEB.**

En la actualidad es imposible concebir la falta de una herramienta que, en los últimos años, se ha confirmado como fundamental en cualquier empresa: la computadora. Esa máquina electrónica que acepta datos para procesarlos y dar como resultado información necesaria para tomar decisiones importantes.

Se sabe que una empresa sin comunicación entre sus miembros, es una organización con claras tendencias al fracaso. Por lo que debe existir un medio de comunicación entre los dispositivos electrónicos (impresoras, escáner, fax, etc.) y la computadora misma. Lo que nos lleva a manejar el concepto de red. Un concepto tan añejo como la computadora misma.

Una red: "... es un sistema de transmisión de datos que permite que un cierto número de dispositivos independientes se comuniquen entre sí dentro de un área geográfica ..."<sup>1</sup>, permitiendo la comunicación y la transferencia de archivos. Computadoras y redes, nos dan una amplia gama de posibilidades para resolver muy indistintos problemas. Ahora se necesita saber: ¿cómo? y ¿con qué? aprovecharlas.

Para utilizar estas herramientas de una forma más efectiva se estudió la metodología de desarrollo de sistemas, denominada "Tecnología en tres capas", que constituye una serie de pasos que ayudan a diseñar de manera consistente y de alta calidad Sistemas Web. Paso siguiente se presentó a Unix, un sistema operativo que ha influido en un gran número de programadores, así como también otros sistemas operativos, uno de ellos Linux. Por ultimo se menciona el servidor de Web Apache, el manejador de datos MySQL, y el lenguaje de programación Java.

---

<sup>1</sup> Forouzan, Behrouz A. *Transmisión de datos y redes de comunicación*. España, McGraw-Hill, 2000, pág 355



### 1.1.1. Introducción a servicios de Internet

¿Qué es Internet? Es "... una colección de muchas redes que se comunican entre ellas a través de un único protocolo, TCP/IP."<sup>2</sup>

Internet nació en los EU como proyecto militar, ARPANET, destinado a mantener la comunicación de sus computadoras bajo cualquier ataque enemigo. "Por enero de 1983, todas las computadoras conectadas a ARPANET corrían a través de los protocolos de TCP/IP"<sup>3</sup>.

La manera de comunicación es a través de una serie de pasos y reglas ya definidos (protocolos de red). Uno de ellos es TCP/IP que es un protocolo abierto, independiente de hardware, independiente de software y gratuito.

Hablemos un poco del direccionamiento IP y para qué sirve. Entre las funciones de este protocolo, Internet Protocol (IP), está proveer de direccionamiento en el nivel de red así como la identificación de redes y host IP, asegurándonos una identificación global y única dentro de la red. Estas direcciones IP tienen un formato y no pueden asignarse de manera arbitraria, el organismo encargado de asignar dicha dirección es el interNIC. Debe tenerse en cuenta que si quiere poner un servidor en Internet debe acudir a este organismo.

Internet es una red de redes, donde se observará dos tipos de usuarios: un prestador de servicios y uno que lo solicita. Los servicios más utilizados son: www o World Wide Web (la Web), el correo electrónico o email, los foros de discusión (foros, listas de discusión chat), servicios de transferencia de archivos (FTP), servicio de trabajo compartido en servidores y otros más.

---

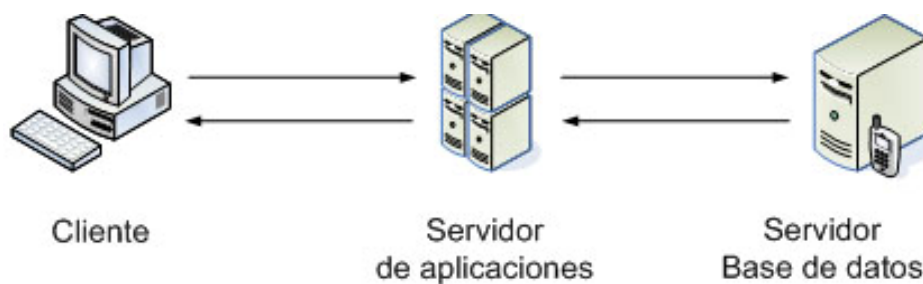
<sup>2</sup> Pike, Mary Ann; Cook, David. La biblia de Internet. España, ANAYA MULTIMEDIA, 1995, pág 78

<sup>3</sup> Atkins, Derek; Buis, Paul. *Internet Security*. United State, New Riders, 1997, pág 8

Estos servicios son de gran utilidad, dependiendo del sistema que se intente desarrollar. El correo electrónico da la posibilidad de establecer otro nivel más de comunicación.

Los foros sirven para saber qué piensa la gente que visita un sitio Web y la transferencia de archivos ayudan a obtener documentos o mandar información en un archivo.

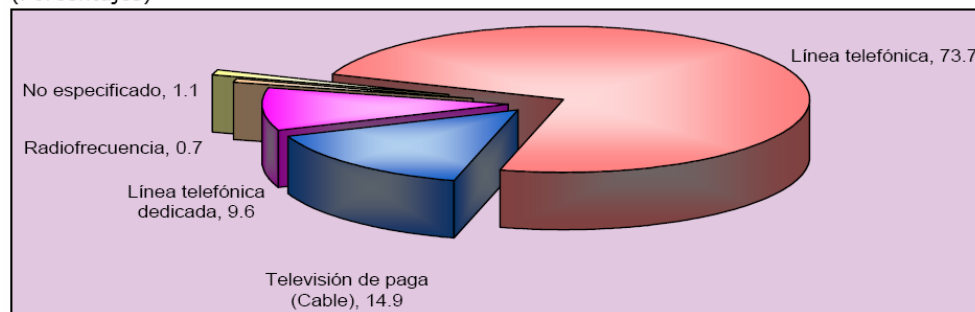
### 1.1.2. Tecnología en tres capas



**Figura 1.1 Tecnología de 3 capas**

Un problema común es intentar atender a miles de usuarios al mismo tiempo por Internet. La mayoría de las personas que se conectan a la Internet lo hacen a través de un ancho de banda<sup>4</sup> muy limitado. Si se ofrece un servicio de consulta se puede observar como se reduce el rendimiento del sistema drásticamente.

**Medio de conexión a Internet, 2005<sup>p\_/\_</sup>**  
(Porcentajes)



Fuente: INEGI. Encuesta Nacional sobre Disponibilidad y Uso de de las Tecnologías de la Información en los Hogares.

p\_/\_: Cifras preliminares correspondientes al mes de junio

**Figura 1.2 Medios de conexión a Internet, 2005<sup>5</sup>**

<sup>4</sup> capacidad de transferencia en KiloBits/segundo

<sup>5</sup> Para ampliar más su información consulte

[http://www.inegi.gob.mx/prod\\_serv/contenidos/espanol/bvinegi/productos/encuestas/especiales/endutih/endutih2005.pdf](http://www.inegi.gob.mx/prod_serv/contenidos/espanol/bvinegi/productos/encuestas/especiales/endutih/endutih2005.pdf)



Al usar una estructura cliente/servidor, que se refiere a: una máquina cliente que proporciona una interfaz a un usuario por medio del cual hará peticiones al servidor y este le responderá a su petición, es decir “una vez establecida la conexión en red, el programa debe lanzar las aplicaciones apropiadas para mandar datos en cierta forma a través del protocolo correcto”<sup>6</sup>.

Un apoyo interesante, a ésta arquitectura, lo ofrece el modelo de tres niveles, que consiste en tratar de liberar, lo más que pueda, al cliente de la lógica empresarial. Esto es, la máquina cliente hará peticiones a través de su interfaz y solo recibirá el resultado de su petición. Esto se logra descargando la mayoría de los procesos en nuestro servidor de aplicaciones o de datos.

La tecnología de tres capas concibe y maneja el uso de éste modelo de tres niveles donde hay una capa de usuario, otra de negocio y una de datos, facilitando el trabajo de creación de un sistema. Con ello se tiene mayor rapidez de respuesta a hacia los clientes, legibilidad del sistema, reutilización de código, facilidad de encontrar errores, mejor mantenimiento del sistema e integridad y seguridad de los datos.

#### **1.1.2.1. Capa de presentación**

La “interfaz de usuario”, aquí no importa cómo acceder a los datos y qué reglas de negocio deberá acatar el sistema, solo como se interactuará con el cliente. La vista o interfaz más utilizada para Internet es el browser, un navegador como netscape, opera, etc.

La tarea principal en esta etapa es conocer el perfil de los usuarios y saber qué desea mostrar la empresa, todo esto para tomar la decisión sobre dónde colocar cada elemento en la pantalla.

Las funciones de la interfaz son cuatro: obtener información del usuario, enviar esa información a la capa de negocio para su validación y procesamiento, recibir de la capa

---

<sup>6</sup> Lender, Rick. *Building Application Servers*. United States, Cambridge University Press, 2000, pág 270.



de negocio los resultados de la interacción de ésta con los datos y, presentar los resultados según como lo solicite el usuario.

### **1.1.2.2. Capa de negocio**

En esta capa se desarrollan las reglas y la lógica de procedimientos necesarios para realizar operaciones del sistema, son el puente entre un usuario y el acceso a datos. La importancia de la capa radica en que aquí se programan todas las actividades operacionales, controles de consistencias, validaciones, cálculos, etc.

La idea es aislar, por completo, toda interacción directa de la base de datos con los usuarios, con la finalidad de conservar seguros los datos y que la información tenga alta integridad. Hay que advertir que esta capa es la más difícil de delimitar, si bien la de usuario queda en la interfaz y la de datos en la comunicación con la Base de Datos, la capa de negocios puede llegar a mezclarse con alguna de las otras dos capas.

Las funciones básicas de la capa de negocio son tres: recibir peticiones de la interfaz, validar e interactuar con los datos para ejecutar las tareas de negocios, es decir, procesos que el sistema automatizó (ejem: Preparación de impuestos por ingreso) y, enviar sus resultados hacia la pantalla.

### **1.1.2.3. Capa de datos**

La capa de datos consiste en realizar la conectividad hacia una Base de Datos Relacional<sup>7</sup>, es decir, el acceso a la información. Los servicios de datos tienen una variedad de formas y tamaños que varían desde el acceso a una Base de Datos como al acceso a un servidor de correo electrónico o a cualquier sistema de archivos.

En la actualidad, el nivel de estandarización de los mecanismos de acceso a los datos es notable. El ODBC es un controlador específico para la base de datos a manejar en el

---

<sup>7</sup> Edgar F. Codd propuso un modelo simple de datos en el que todos ellos se representarían en tablas, esto lo detallare más adelante.





servidor adecuado. Este nos permite, una vez instalado, utilizar sus librerías de objetos para acceder a la información.

Las funciones que debe realizar son: almacenar los datos, recuperar los datos, mantener los datos y cuidar la integridad de los datos. Existen muchos DBMS (Data Base Manager System), los servidores de datos más populares son: SQL Server, Oracle, Postgres SQL, MySQL, entre otros.

### **1.1.3. Sistemas operativos**

Es más fácil definir el término sistema operativo por lo que hacen y no por lo que son. “Un sistema operativo (SO) se puede ver como una colección organizada de software que extiende al hardware y que consta de rutinas de control para operar una computadora y proporcionar un entorno para la ejecución de programas”<sup>8</sup>.

Entre los sistemas operativos más usados encontramos a Unix, desarrollado en los Laboratorios Bell de AT&T y que está escrito con el lenguaje de programación C, sólo subrutinas de alto rendimiento dentro del núcleo del sistema están escritas en ensamblador. Ésta característica permite que el sistema sea muy portable.

Las características más importantes de Unix son: permitir comunicación entre procesos, está escrito en lenguaje de alto nivel, es un sistema multiprogramado, el sistema de archivos tiene estructura de árbol, cuenta con permisos, la gestión de memoria es mediante paginación, etc.

“Unix es un sistema operativo de tiempo compartido, pensado principalmente para el desarrollo de programas, y para entornos de la preparación de documentos. Debido a su fácil transportación, Unix es disponible para una variedad de CPU, desde microprocesadores a supercomputadoras”<sup>9</sup>

---

<sup>8</sup> Milenkovic, Milan. *Sistemas operativos. Conceptos y diseño*. España, McGraw-Hill, 1988, pág 1.

<sup>9</sup> Milenkovic, Milan. *Sistemas operativos. Conceptos y diseño*. España, McGraw-Hill, 1988, pág 521



El núcleo de Unix (Kernel) se considera de tipo monolítico y sus funciones más importantes son: crear procesos, asignación de tiempos de atención y sincronización, asignación de la atención del procesador a los procesos que lo requieran, la administración de espacio en el sistema de archivos, y la supervisión de la transición de datos entre memoria y los dispositivos.

Unix cuenta con el Shell o intérprete de comandos, que además de interpretar comandos también sirve como lenguaje de programación. También tiene procesadores de texto, aplicaciones matemáticas, aplicaciones para generación de gráficos y base de datos. Ofrece la posibilidad de creación de aplicaciones mediante C, C++, Fortran, Java, etc.

Unix da un tratamiento consistente a sus archivos, puede crear y eliminar archivos, permite el crecimiento dinámico de ellos y tiene mecanismos de protección de los archivos, además maneja los dispositivos periféricos como archivos.

#### **1.1.4. Elementos de software para un proyecto de desarrollo en WWW**

Los elementos básicos para un sistema en Web consiste en: tener un medio de comunicación, este nos permitirá entrar a la red Internet; si se va a plantear una interacción por medio de la WWW debemos tener un servidor Web que aloje nuestras páginas Web; si usamos un lenguaje de programación para nuestras reglas del negocio necesitaremos un servidor de aplicaciones que haga nativa la comprensión entre el HTML y nuestro lenguaje y; por ultimo un manejador de datos como MySQL.

Para publicar páginas en Internet se necesita un servidor Web. “Apache es un servidor Web de dominio público para Unix. Éste es una derivación de el NCSA Web server”<sup>10</sup>. El Nacional Center for super Computing Applications Web server cayo en el olvido y gente que trabajo en éste proyecto formo un grupo denominado Apache Group.

---

<sup>10</sup> Ablan, Jerry; Yanoff, Scout. *Web site administrators*. United States, Sams Net, 1996 pág 550 y 575



Su estructura está conformada por módulos, configurados por medio de un archivo de configuración, el `httpd.config`, este archivo se encuentra en la carpeta `Config` dentro de la carpeta de instalación de Apache.

Los módulos del Apache los podemos agrupar tres grupos: de base, de multiproceso y adicionales. Actualmente, encontramos que en muchos sitios se maneja el servidor Web Apache con lenguajes de programación como Java o PHP. Más del 50% de los sitios utiliza Apache y el número sigue en aumento.

“Características de Apache:

- Admite la última versión http
- Es simple
- Traba con CGI's
- Admite servidores virtuales
- Admite la autenticación http
- Integra Perl
- Cuenta con servidor Proxy
- Registros personalizables e información sobre el estado del servidor
- Admite Java Servlets”<sup>11</sup>

Un lenguaje de programación moderno y orientado a objetos es Java. Su capacidad y seguridad han hecho que sea uno de los más usados.

“Si bien Java tiene su base en lenguajes como C y C++, los supera con creces y sería un error pensar que es una simple evolución de éstos”<sup>12</sup>. Java lo podemos utilizar en la capas de presentación, de negocios y de datos.

---

<sup>11</sup> J. Kabir, Mohamed. *La biblia del servidor Apache*. United Status, Anaya Multimedia, 1999, pág 36 y 37

<sup>12</sup> Camacho Fernández, D.; Valls Ferrán, J. M.; García Herrero, J.; Molina López, J. M.; Bueno Rodríguez, E. *Programación, algoritmos y ejercicios resueltos en Java*. España, Prentice-Hall, 2003, pág 1



Para que el servidor Web Apache pueda entender, por así decirlo, a Java, necesita un servlet engine. Éste demonio, servlet engine, lo proporciona Tomcat y permite escuchar peticiones con los Java Servlets Pages (JSP's), estos son parte de la capa de usuario. Es decir, los JSP's construyen las páginas HTML que serán utilizadas para proporcionar la interfaz o respuesta de petición.

Además, anexa la Java Virtual Machine para poder transformar los JSP's en servlets (Server side Entity), estas entidades del lado del servidor serán quienes realicen la lógica del negocio. Existen tres formas de poner un servidor Web con Tomcat:

- Tomcat 100%
- Servidor Híbrido
- Utilizando Apache

Al utilizar Apache server Web se puede realizar de manera más organizada el patrón de diseño Modelo Vista Controlador.

El sistema gestor de base de datos que utilizaré es MySQL. Este cuenta con soporte amplio, es decir, los drivers necesarios para que Java pueda realizar conexiones seguras hacia él. “MySQL es un sistema gestor de base de datos relacional cliente-servidor de coste mínimo ... implementa funcionalidades Web, permitiendo un acceso seguro y sencillo a los datos a través de Internet”<sup>13</sup>.

Los sistemas operativos que soportan la instalación de MySQL son:

- AIX
- FreeBSD
- Mac OS X
- OpenBSD
- Solares 2.5
- Windows

---

<sup>13</sup> Pérez, Cesar. *MySQL para Windows y Linux*. México, Alfaomega, 2004, pág 1 y 7



- Linux

MySQL es un manejador de base de datos inspirado en el Estándar Query Lenguaje (SQL) Server, es pequeño y fácil de usar. Así como también, su costo es mínimo. Al soportar el SQL permite realizar búsquedas organizadas y tan complicadas como se requieran.

## 1.2. MÓDULO II

### ANÁLISIS Y DISEÑO CON UML.

El hombre siempre busca métodos que den solución a diversos problemas de la vida real, valiéndose de sus conocimientos, tecnología y experiencia. La ciencia que combina las matemáticas y la física para transformar recursos a fin de solucionar problemas que surgen por necesidades de la sociedad, se le denomina Ingeniería de Sistemas.

El significado de sistema puede ser variado, dependiendo del contexto, por ejemplo un sistema puede ser "... un procedimiento, un proceso o su control, una red o un paquete de procesamiento de datos basados en la computadora"<sup>14</sup>. Así, podemos decir que un sistema es un conjunto de procedimientos o actividades debidamente ordenadas que se desarrollan conforme a un esquema, todos ellos relacionados entre sí, para alcanzar un fin.

Los sistemas son creaciones complejas, pero muy útiles. Para desarrollar un proyecto de creación de un sistema, la ingeniería del software es de gran ayuda, está sirve para obtener buenas prácticas de desarrollo, así como entender y aplicar diferentes técnicas y herramientas.

---

<sup>14</sup> Wilson, Brian. *Sistemas: Conceptos, Metodología y aplicaciones*. México, Megabyte, 1993, pág 41



Una de las herramientas de análisis y diseño es el Unified Modelling Language(UML), “el lenguaje de modelación unificado es una notación de carácter internacional para describir los diferentes elementos de un análisis y diseño orientado a objetos. El UML define el numero de vistas de un sistema utilizando distintos diagramas como el diagrama de clases y diagramas de interacción”<sup>15</sup>.

El software Poseidón es una herramienta que permite crear diagramas de UML, ayuda a desarrollar diagramas de caso de uso, interacción, secuencia, clases, etc. Este software lo utilizamos durante el diplomado para realizar los diagramas de casos de uso.

### **1.2.1. Ingeniería de requerimientos**

“Los costos de falla en un sistema son muy altos y la ingeniería de requerimientos y procesos de desarrollo de sistemas debe asegurar estos contratiempos haciendo al sistema seguro<sup>16</sup>”. El proceso de desarrollo de software requiere un conjunto de conceptos, una metodología a seguir y un lenguaje propio. Un proceso que permita la definición del problema, su solución y su mantenimiento, a esto le denominaremos el ciclo de vida del software.

Las actividades del desarrollo de ingeniería del software orientado a objetos incluyen la obtención de requerimientos, análisis, diseño del sistema, diseño de objetos e implantación. Estas actividades de desarrollo manejan la complejidad mediante la construcción de modelos del dominio del problema o del sistema. Conocer los requerimientos del sistema sirve para delimitar el alcance de éste.

Uno de los aspectos más importantes de un buen sistema es que sea accesible. Una buena práctica es dejar todo bien documentado, ya que tal vez no seamos los que demos mantenimiento al sistema. Hay que tener presente que los costos de mano de obra pueden llegar a ser costosos.

---

<sup>15</sup> Barclay, K.; Savage, J. *Object-oriented design whit uml and Java*. Great Britain, Elsevier, 2004, pág 17

<sup>16</sup> Sommerville, Ian. *Requeriments engineering: a good practice guide*. Great Britain, Wiley, 1997, pág 255 y 256



También debe tener buena disponibilidad, éste debe ejecutarse en el equipo con que cuente el cliente. El sistema debe ser lo bastante adaptable a otros ambientes de hardware/software, ya que es común encontrar verdaderas ensaladas tecnológicas en las empresas.

Con todo lo anterior se podrá construir un buen sistema, ya que cumplirá la función por la cual fue creado, no hará ni más ni menos. El sistema será útil y aprovechable cuando un usuario diga: *“me ha hecho la vida mas fácil”*.

### **1.2.2. RUP**

*Rational Unified Process*. La meta fundamental de este proceso es asegurar que la producción del software sea de la más alta calidad posible, que el software desarrollado resuelva las necesidades dentro de los recursos determinados por el cliente. RUP nos proporciona un acercamiento disciplinado al asignar tareas y responsabilidades dentro de una organización de desarrollo, define el *quién* esta haciendo *qué*, *cuándo* y *cómo* lo hace para llegar al objetivo.

Las fases para el desarrollo de un sistema son cuatro: inicio (define el alcance del proyecto), elaboración (establecer el plan del proyecto), construcción (desarrollar el sistema), y transición (dar capacitación a los usuarios o clientes que manejaran el sistema).

### **1.2.3. La ingeniería de requerimientos y sus actividades principales**

*Un requerimiento es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo*<sup>17</sup>. Están los requerimientos funcionales, que definen el *¿qué* hace el sistema?, y por otro lado están los requerimientos no funcionales, que indican al sistema como realizar su trabajo: *¿cómo?*, *¿cuándo?*, y *¿cuánto* de *qué?*

---

<sup>17</sup> Esta es una de las definiciones que presenta *The Institute of Electrical and Electronics Engineers* (IEEE)



Un requerimiento debe cumplir características como: necesario, si su omisión provoca una deficiencia en el sistema; conciso, si es fácil de leer y entender; verificable, cuando puede ser cuantificado; consistente, si no es contradictorio con otro requerimiento; no es ambiguo, cuando tiene una sola interpretación; completo, si no necesita ampliar detalles en su redacción.

Es difícil determinar los requerimientos, pero los beneficios son muchos, estos permitirán gestionar las necesidades del proyecto en forma estructurada, permitiendo disminuir costos y evitar retrasos en los proyectos y así realizar productos de alta calidad.

#### **1.2.4. Técnicas y herramientas**

Afortunadamente para los ingenieros en software pueden hacerse de la ayuda de distintas técnicas y herramientas para recabar información de los requerimientos del sistema, sin embargo, la eficacia de estas depende de qué problema sea atacado. Algunas de estas técnicas son: entrevistas y cuestionarios, lluvia de ideas, prototipos y casos de uso.

#### **1.2.5. Enfoque de orientación de objetos**

El enfoque de orientación a objetos “... esto es un nuevo camino de pensamiento sobre los procesos de descomponer problemas y desarrollo de soluciones programadas. El paradigma orientado a objetos (OO) ve un programa como una colección de agentes autónomos, llamados objetos. Cada objeto es responsable de una tarea específica”<sup>18</sup>.

---

<sup>18</sup> Budd, Timothy. *Object-Oriented programming*. United States, Addison-wesley, 1991, pág 15





El paradigma OO se basa en el concepto de **objeto**. “Un objeto es una cosa con la que se puede interactuar: se le puede enviar varios mensajes y éste reacciona ante ellos”<sup>19</sup>, es la abstracción de la realidad, aquello que tiene y guarda un estado, atributos que lo hacen único e indivisible dentro del entorno al que pertenece. Es posible establecer propiedades y atributos de un objeto, lo mismo que su grado de respuesta a estímulos externos (acciones y reacciones a mensajes).

Un objeto son cosas, simples o muy complejas, reales o virtuales. Casi todo lo que nos rodea se puede abstraer en un objeto.



**Figura 1.3 El clásico objeto carro**

Un objeto se abstrae de la realidad en dos aspectos el funcional y de datos. El objeto carro puede realizar acciones de moverse, parar, girar derecha, girar izquierda, ir para atrás, etc.

El cómo lo hace pasa a ser irrelevante, lo importante es que lo sabe hacer, a esto se le llama abstracción funcional. Ahora, nuestro carro tiene características que lo definen como el color, la velocidad, el tamaño, el modelo. La manera en como se almacenan estos datos o los tipos de datos no es importante. Lo relevante es que tiene esos atributos, a esto le llamamos abstracción de los datos.

---

<sup>19</sup> Stevens, Perdita; Pooley, Rob. *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. España, Addison-wesley, 2002, pág 18



Como ya mencionamos, un objeto tiene un comportamiento y un estado. El primero lo definen las operaciones que puede realizar y el segundo lo conforman los datos o atributos del objeto. Un objeto puede mostrar su comportamiento y estado u ocultarlos, según sea conveniente. A esto se le denomina **encapsulación**, nuestro objeto carro muestra una interfaz de otros objetos como el volante que controla la dirección del carro, más sin embargo, no muestra como lo hace.

Ahora la encapsulación nos da el siguiente beneficio, el objeto al mostrar una interfaz y ocultar su implementación, permite realizar cambios en el funcionamiento del objeto sin que afecte la forma de acceder a ellos. “La interfaz pública de un objeto define qué mensajes se aceptan sin importar de dónde vienen”<sup>20</sup>.

Ahora bien, un objeto puede estar conformado por ninguno o varios objetos, que pueden interactuar o no entre sí. Nuestro objeto carro está formado por el objeto acelerador y el objeto motor que con la interacción de ambos hacen que el objeto carro se mueva. Para que suceda esto los objetos utilizan el intercambio de mensajes, una estimulación externa puede hacer que un objeto tenga una contestación, cambie su estado, o reaccione de la manera más apropiada a través de su interfaz (la parte visible del objeto).

Los objetos que se comportan de la misma forma y sus características son muy parecidas se agrupan en tipos de objetos: en una **clase**. Un carro significa cualquier instancia de nuestra clase carro, pero, un carro de golf representa una instancia determinada de la clase carro. Ahora bien, la clase carro debe pertenecer a la clase transporte junto con la clase aviones, la clase trenes, etc.

Por medio de las **clases** es posible asignarle una propiedad a un conjunto de objetos lo que simplifica su control y administración, instar un objeto es generar éste a partir de una clase.

---

<sup>20</sup> Stevens, Perdita; Pooley, Rob. *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. España, Addison-wesley, 2002, pág 20



“Una clase describe un conjunto de objetos que tienen un rol o roles equivalentes en el sistema ... cada objeto pertenece a una clase, y la clase de un objeto es quien determina su interfaz”<sup>21</sup>, una clase tiene un nombre, métodos (funciones) y atributos (datos). La única forma de que se distingan los objetos de una misma clase es a través de sus atributos.

La especialización o generalización de las clases nos lleva a otro concepto, la **herencia**. “La herencia es una forma de reutilización del software en la que se crean nuevas clases a partir de clases ya existentes absorbiendo sus atributos y comportamientos y adornándolos con capacidades que las nuevas clases requieren”<sup>22</sup>. Éste es el mecanismo por el cual se genera la definición de una clase a partir de otra clase ya definida, permitiendo manejar clases colectivamente.

La herencia nos permite definir a los miembros, ocultarlos o añadir nuevos miembros, implementando las relaciones `es_un` (type of) o `es_una_clase_de` (instante of). La herencia es la característica clave en los sistemas orientados a objeto para propiciar la reusabilidad.

En la herencia se habla de la clase padre o base o superclase la cual da todas sus características a nuevas clases creadas a partir de ella, las clases hijas o subclases. Las clases hijas son especializaciones de la clase padre, sin embargo, algunos métodos pueden ser no óptimos, por lo que, se utiliza la sobrecarga de métodos permitiendo a la subclase redefinir funciones heredadas.

Ahora bien, hay cosas que en sí no pueden generar objetos ya que su definición es incompleta. A este caso se le denomina **clase abstracta** que es un conjunto de operaciones y atributos que no describen nada. Sin embargo, en combinación, mediante el polimorfismo, son utilizadas por los objetos para realizar ciertas funciones.

---

<sup>21</sup> Stevens, Perdita; Pooley, Rob. *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. España, Addison-wesley, 2002, pág 21

<sup>22</sup> Deitel, H. M.; Deitel P. J. *Cómo programar en Java*. México, Prentice hall, 1998, pág 326



Con esto tocamos otro concepto importante en el POO<sup>23</sup>, **polimorfismo**. “El polimorfismo nos permite escribir programas de forma general para manejar una amplia variedad de clases interrelacionadas existentes y por especificar”<sup>24</sup>. Cuando un objeto A necesita la ejecución de una función en especial a otro objeto B, al objeto A no le importa y no necesita saber la naturaleza del objeto B dueño del método que requiere.

El modelado orientado a objetos nos permite acercarnos a los términos específicos del dominio del problema, permitiendo construir modelos que forman parte de la aplicación.

Ejemplo: Sistema inscripción:

- Clase alumno
- Clase profesor
- Clase curso
- Clase salón, etc.

### 1.2.6. Herramientas de análisis y diseño con UML

*Lenguaje de Modelado Unificado* surge en 1997 gracias a un grupo de expertos en el análisis y diseño orientado a objetos. La industria del software cuenta ahora con UML. “En realidad, UML es tan expresivo que es improbable no poder utilizarlo con cualquier metodología o proceso de desarrollo orientado a objetos”<sup>25</sup>.

Algunas de las ventajas que nos proporciona son:

1. Los sistemas de software profesionales son diseñados y documentados antes de que sean codificados en cualquier lenguaje. Sabrá exactamente lo que conseguirá, por adelantado.
2. Ya que el diseño del sistema, está antes de crear la primera parte del código, es fácil descubrir el código reutilizable y tratarlo para una mejor eficacia.

---

<sup>23</sup> Paradigma Orientado a Objetos.

<sup>24</sup> Deitel, H. M.; Deitel P. J. *Cómo programar en Java*. México, Prentice hall, 1998, pág 326

<sup>25</sup> Stevens, Perdita; Pooley, Rob. *Utilización de UML en Ingeniería del Software con Objetos y Componentes*. España, Addison-wesley, 2002, pág 61



3. Los lógicos ‘agujeros’ en el diseño del sistema podrán ser descubiertos antes sobre los diagramas del mismo.
4. Las decisiones finales se harán antes de que se encuentre con código mal escrito.
5. Cuando necesite hacer modificaciones en el sistema, será más fácil hacerlo sobre la documentación UML. Hay que recurrir mucho menos a rehacer un nuevo estudio.
6. Si se incorporan nuevos desarrolladores al proyecto, los diagramas UML les permitirá hacerse rápidamente una idea del sistema.

Las herramientas que nos proporciona UML son: casos de uso, diagramas de casos de uso, de iteración, de secuencia, de componentes, de estado, de actividades y de clases. Todas estas herramientas tienen el propósito fundamental de hacer lo más legible las funciones del sistema y servirán para documentar el desarrollo del proyecto.

*Casos de Uso: documental.* “Un caso de uso es la típica interacción entre un usuario y el sistema bajo desarrollo. Esto es usado para capturar algo de la funcionalidad que proveerá el sistema”<sup>26</sup>. Los casos de uso se utilizan durante la obtención de requerimientos y el análisis para representar la funcionalidad del sistema. Con esto se puede extraer el conocimiento fundamental de las funciones de un problema. Es decir, todo lo que debe hacerse, cómo se debe hacer y quién lo debe hacer.

Un caso de uso es una tarea, que un actor necesita ejecutar con la ayuda del sistema, para identificar a los actores y sus metas. Se debe considerar que un actor es cualquier cosa con comportamiento, y preguntarse ¿qué subsistema y personas usarán nuestro sistema? y ¿qué requiere un actor para completar sus requerimientos?

*Diagrama de caso de usos.* Con la ayuda del software Poseidón se pueden crear los diagramas de caso de uso, ya que a medida que avance nuestro proyecto se corre el riesgo de perderse entre tanta información.

---

<sup>26</sup> Barclay, K.; Savage, J. *Object-oriented design whit uml and Java*. Great Britain, Elsevier, 2004, pág 20



Las gráficas que se utilizan en los diagramas son auto explicativo: los muñecos representan a los diversos actores, los óvalos representan a cada caso de uso y hay una línea entre el actor y un caso de uso (si es que el actor puede participar en el caso).

*Diagramas de interacción.* Los diagramas de interacción muestran como pasan los mensajes de un objeto a otro para ejecutar algunas tareas, mostrando el patrón de interacción entre objetos. No necesariamente se hace uso de estos diagramas en todos los proyectos, pero, si resulta que un caso de uso es particularmente complicado entonces los diagramas de interacción pueden ayudar a aclararnos las ideas.

*Diagrama de secuencias.* Los diagramas de secuencia nos servirán para mostrar una interacción ordenada según la secuencia temporal de eventos indicándonos qué mensajes se pasan entre los objetos. Y servirán para formalizar el comportamiento del sistema y para visualizar la comunicación entre objetos.

*Diagrama de componentes.* Un componente debe definir una abstracción precisa con una interfaz bien definida, siendo estas flexibles al cambio de componentes viejos a otros nuevos. Cada uno deberá constar con un nombre que lo distinga de los demás. Estos componentes representarán los aspectos físicos del sistema.

*Diagrama de clase.* Los diagramas de clase los podemos utilizar para describir el sistema desde el punto de vista de objetos, clases, atributos, operaciones y sus asociaciones. Donde las clases son abstracciones que especifican la estructura y el comportamiento común de un conjunto de objetos.

“El diagrama de clases es el principal diagrama para construir un diseño orientado a objetos. Un diagrama de clases describe los tipos de los objetos en el sistema y la relación que existe entre ellos”<sup>27</sup>.

---

<sup>27</sup> Barclay, K.; Savage, J. *Object-oriented design whit uml and Java*. Great Britain, Elsevier, 2004, pág 38



### 1.2.7. Diseño de la base de datos

Para tener una información adecuada y con alta integridad es necesario realizar un buen diseño de la base de datos. El tener un buen sistema administrador de base de datos se vuelve crucial. “El desarrollo de los sistemas administradores de bases de datos se debió a gran parte a dos grandes problemas relacionados con los sistemas de procesamiento de archivos convencionales: la redundancia y la dependencia de datos”<sup>28</sup>.

Anteriormente las bases de datos estaban restringidas por consideraciones físicas, utilizaban punteros físicos o direcciones a disco para relacionar registros en diferentes archivos. El modelo de datos relacional termina, en gran medida, con esa restricción y supera esos problemas.

“El modelo de datos relacional organiza y representa los datos en forma de tablas o relaciones. Relación es un término que viene de la matemática y representa una simple tabla de dos dimensiones, consistente en filas y columnas de datos”<sup>29</sup>.

La base de datos podrá crecer hasta donde los requerimientos pidan, a medida que la base de datos va creciendo se empieza a dificultar la lectura y relación de cada entidad por lo que una solución es utilizar el diagrama entidad/relación (E-R). Este diagrama será una representación gráfica de nuestro modelo de datos relacional.

El proceso de diseño de una base de datos requiere tiempo y paciencia para ir descubriendo poco a poco las relaciones de cada elemento que participa en ella.

---

<sup>28</sup> Beck, Leland L. *Software de sistemas*. México, 1988, pág 380

<sup>29</sup> Hansen, Gary W; Hansen, James V. *Diseño y Administración de Bases de Datos*. España, Prentice Hall, 1998, pág 139



El tener un buen diseño dará las siguientes cualidades: reflejará la estructura del problema en el mundo real, será capaz de representar todos los datos esperados, evitará la redundancia de información, dará un acceso eficaz a los datos y mantendrá la integridad de los datos a través del tiempo.

“Redundancia en los datos es la duplicación de elementos de datos en diferentes archivos. La desventaja más obvia es el espacio de almacenamiento adicional que se requiere”<sup>30</sup>.

Para diseñar un modelo de datos entidad/relación deberá considerar las siguientes fases:

- Identificar las entidades que debe presentar la base de datos
- Determinar las *cardinalidades* de las interrelaciones entre las entidades (uno a uno, uno a muchos y muchos a muchos)
- Dibujar el diagrama Entidad-Relación
- Determinar los atributos de cada entidad y por ultimo definir la clave primaria de cada entidad.

### 1.3. MÓDULO III

#### INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS CON JAVA

“Programar en un lenguaje que soporte el paradigma orientado a objetos beneficia al desarrollador de software al proveer un camino natural hacia un modelo complejo, algo del mundo real”<sup>31</sup>.

Java es un lenguaje orientado a objetos que ha mostrado ser apto para el ámbito de red, su robustez y portabilidad lo hace un lenguaje muy atractivo. No sólo puede servir para

---

<sup>30</sup> Beck, Leland L. *Software de sistemas*. México, 1988, pág 380

<sup>31</sup> Winblad, Ann L.; Edwards, Samuel D. *Object-Oriented Software*. United States, Addison-Wesley, 1990, pág 61





la construcción de interfaz con la capa de negocios, si no también para programar la conexión entre la capa de datos y las reglas de negocio.

Fue diseñado en 1990 por James Gosling, de Sun Microsystems, la reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo pero muy potente y que a la vez fuera capaz de generar código de tamaño muy reducido. Inicialmente Java tenía el nombre de Oak, sin embargo, este nombre ya era ocupado, obligando a Sun Microsystem a adoptar otro nombre: Java.

Al ver que no podría tener mucho éxito con los electrodomésticos su enfoque cambió hacia Internet. “El mercado para dispositivos electrónicos de consumo inteligentes no estaba creciendo con la rapidez que Sun había anticipado ... el proyecto estaba en peligro de ser cancelado”<sup>32</sup>.

Las tres principales razones que llevaron a la creación de Java son:

- necesidad de la comodidad y facilidad de uso por medio de interfaces graficas;
- fiabilidad del código y facilidad de desarrollo para disminuir costos en pruebas y depuración;
- la diversidad de plataformas electrónicas, distintos microprocesadores y la velocidad de los cambios.

### **1.3.1. Características de Java**

Java es un lenguaje de alto nivel que cuenta con las siguientes características: simple, orientado a objetos, interpretado, robusto, seguro, multiplataforma, portable, multihilo y dinámico.

---

<sup>32</sup> Deitel, H. M.; Deitel P. J. *Cómo programar en Java*. México, Prentice hall, 1998, pág 12

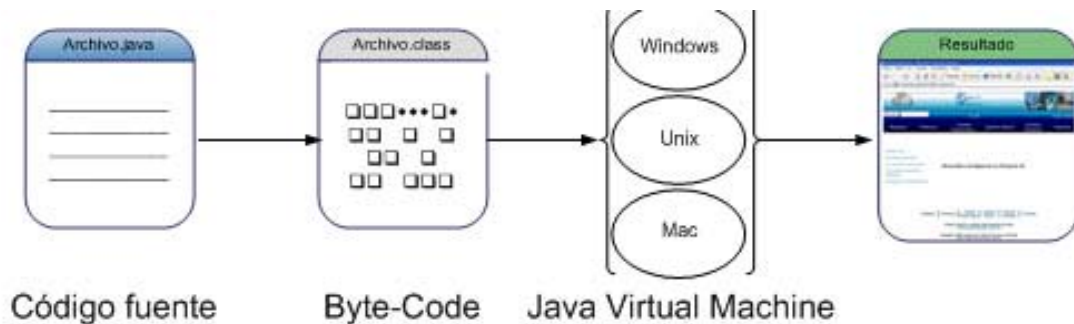


- *Simple.* A diferencia de lenguajes como C, Java elimina complejidades como la aritmética de punteros, no utiliza el comando DELETE y no hay herencia múltiple.
- *Orientado a objetos.* Posee características que hoy día se consideran estándares en los lenguajes OO: objetos, clases, métodos, subclasses, herencia y encapsulamiento.
- *Distribuido.* Java en sí no es distribuido, sino que proporciona las librerías y herramientas para que los programas corran en varias máquinas, interactuando, da la etiqueta al programa de distribuido.
- *Interpretado.* Utiliza una máquina virtual para cada plataforma, este se implementa vía software. El código generado por la compilación de Java (Java Byte Code) es interpretado por la máquina virtual. Así los programas en Java se compilan una vez y se interpretan cada vez que se ejecuta.
- *Robusto.* Java siempre checa los índices al acceder un arreglo, de igual forma revisa los tipos de datos durante la ejecución. Posee un recolector de basura que administra la memoria.
- *Seguro.* El código pasa por muchos tests antes de ejecutarse en un ordenador, para evitar desbordamientos de operandos, conversión ilegal de datos, violación de reglas de acceso y seguridad establecidas. Es seguro, no accede a zonas dedicadas a memoria o sistema, con lo cual evita la interacción de ciertos virus.
- *Multiplataforma.* Los programas en Java pueden ejecutarse en Windows 9x/me/ y /NT, Power/Mac y Unix. Permite una compatibilidad total a nivel de fuentes, bibliotecas y código compilado. Sólo se requiere un intérprete para una plataforma en particular.
- *Multihilo.* Permite muchas actividades simultáneas en un programa, dando un mejor rendimiento interactivo y mejor comportamiento en tiempo real. Aunque estará limitado por el sistema operativo.
- *Dinámico.* Se beneficia todo lo posible de la tecnología orientada a objetos, no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución.

### 1.3.2. El lenguaje de programación Java

“El proceso de compilación y ejecución en Java requiere de la utilización de dos componentes del entorno de desarrollo; por un lado debemos compilar el código java y por otro debemos ejecutar el programa generado”<sup>33</sup>.

La compilación en Java no arroja un archivo ejecutable, después de la compilación se obtiene un archivo en bytecodes listo para ser interpretado y ejecutado por una Máquina Virtual de Java. Cada intérprete es una implementación de la Java Virtual Machine, así el mismo programa escrito en lenguaje Java puede ser ejecutado en Windows, Unix o Mac.



**Figura 1.4 Funcionamiento Java**

### 1.3.3. La plataforma Java

La plataforma Java provee tres elementos: un lenguaje de programación (Java), la máquina virtual de Java y el API de Java. La Java VM es el intérprete que anteriormente hemos mencionado, mientras que la API es una colección de componentes de software listos para usarse.

<sup>33</sup> Camacho Fernández, D.; Valls Ferrán, J. M.; García Herrero, J.; et all. *Programación, algoritmos y ejercicios resueltos en Java*. España, Prentice Hall, 2003, pág 3



El API de Java está agrupado en librerías de clases e interfaces relacionadas, estas librerías son conocidas como paquete. “Java contiene muchas piezas predefinidas llamadas clases que se agrupan mediante subdirectorios del disco en categorías de clases relacionadas entre sí llamadas paquetes. Juntos, estos paquetes se conocen como la interfaz de programación de aplicaciones de Java (Java API, applications programming interface)”<sup>34</sup>.

Observará que es importante saber que uno de esos paquetes es el Java Database Connectivity (JDBC) que permite acceso a un amplio rango de gestores de base de datos.

#### **1.3.4. Tipos de aplicaciones en Java**

Java es un lenguaje de propósito general ya que no fue pensado para resolver algún problema en particular, en general se pueden programar todas las aplicaciones que se producen en otros lenguajes. Java permite construir varios tipos de programas, cada uno con unas características específicas, y que se ejecutan de distintas maneras. Estas son aplicaciones standalone, applets, servlets, JSP's y EJB's.

#### **1.3.5. El JDK**

Java Development Kit, el kit de desarrollo de Java es un conjunto de herramientas, utilidades, documentación y ejemplos para desarrollar aplicaciones Java. El JDK es gratuito y puede ser bajado desde la página oficial de Sun (<http://java.sun.com>). Existen versiones disponibles para varias plataformas. Al instalar el JDK se descargan en la máquina: un compilador, el JVM, un conjunto de clases y otras utilerías. Las herramientas de Java son: java, javac, appletviewer, jar, jdb, javadoc.

---

<sup>34</sup> Deitel, H. M.; Deitel P. J. *Cómo programar en Java*. México, Prentice hall, 1998, pág 166



Al instalarse se deben crear las variables de entorno del sistema como el JAVA\_HOME, CLASSPATH y el PATH, esto se hace con la finalidad que en cualquier ruta del sistema reconozca los comandos de java: java, javac, javadoc, etc.

### **1.3.6. Recomendaciones de programación**

Se debe buscar que los programas sean legibles para cualquier otro programador, esto lo puede lograr definiendo una serie de normas en su forma de programar. Por ejemplo, el nombre de las clases debe iniciar con letra mayúscula, los métodos con letra minúscula, de las constantes todas las letras deben ir con mayúsculas, etc.

Además, se debe tener cuidado en que los nombres de las clases no coincidan con otras ya existentes, que los nombres de las variables no coincidan con palabras reservadas. Considere escribir una sola operación en cada línea y no pase de 80 caracteres e incluya comentarios de cada clase marcando su funcionalidad.

### **1.3.7. Sintaxis del lenguaje**

Java maneja comentarios en sus documentos, estos sirven para describir una línea de código o una clase entera. Hay tres tipos de marca de comentarios: //una línea, /\*segmento de código\*/, /\*\*comentario de documentación\*/.

La estructura de un programa de java es la siguiente:

```
//La aplicación hola mundo
public class Hello1{
    public static void main (String args[]) {
        System.out.println (“Hola mundo”);
    }
}
```

Este pequeño programa se guardara en un archivo con extensión .java, hay que recalcar que el nombre del archivo debe ser el mismo nombre de la clase creada.



Es decir, en el archivo Hello1.java existirá la clase Hello1. Java es un lenguaje sensible a mayúsculas y minúsculas por lo que Hello no será lo mismo que hello.

En el anterior código notará la palabra public, las palabras public y private son las que dan a Java el concepto de encapsulamiento, es decir, una clase public será vista por otras clases, mientras que si tiene private no será visto por nadie protegiendo sus métodos y atributos.

Observará que hay un método main, para que pueda ejecutar las clases que ha programado debe tener una clase principal, en ésta se deberá emplear rigurosamente main ya que sin este no se ejecutarán sus clases.

El lenguaje cuenta con palabras reservadas que no se pueden utilizar como identificadores. Un identificador es una variable, función, clase u objeto. Ejemplo: new, for, continue, while, package.

“Todas las variables que se definen tienen un tipo declarado y este tipo es controlado y comprobado en todas las operaciones y expresiones”<sup>35</sup>. Los tipos básicos son: enteros, coma flotante, caracteres y lógicos.

Una variable debe ser un identificador legal, el tipo con que ésta sea declarada será el tipos de datos que guarde. Algunos tipos de datos son: byte (entero de longitud byte), Int (entero), float (decimal), char (caracter), boolean (valor verdadero o falso), etc., ejem: int suma; (suma guardará números enteros como datos).

Java soporta operadores unarios y binarios. Un operador unario es aquel que solo necesita un operando 1++, el 1 representa nuestro operando y el ++ a nuestro operador. Mientras que un binario es el que requiere dos operandos 2 + 3. El lenguaje maneja operadores lógicos, de asignación y otros.

---

<sup>35</sup> Camacho Fernández, D.; Valls Ferrán, J. M.; García Herrero, J.; et all. *Programación, algoritmos y ejercicios resueltos en Java*. España, Prentice Hall, 2003, pág 6



“El control de la ejecución del programa se realiza en Java utilizando unas pocas sentencias. Estas sentencias son básicamente las mismas que en C/C++”<sup>36</sup>, Java tiene estructuras de control de las cuales tenemos las condicionales (if-else, switch-case), de ciclo (while, do-while, for), para manejo de excepciones (try-catch-finally, throw) y de control general de flujo (break, continue, label:, return).

### 1.3.8. Constructores

Todas las clases de Java utilizan al constructor, éste es el encargado de inicializar los objetos. “Un constructor es un método especial que inicializa las variables de ejemplar de un objeto clase. El método constructor de una clase se invoca automáticamente cuando se crea un objeto de esa clase”<sup>37</sup>.

Un constructor tiene el mismo nombre que la clase: `items = new Vector(10);`. Java soporta la sobrecarga para los nombres de los constructores, es decir, que una clase puede tener varios constructores con el mismo nombre. “Es común tener varios constructores para una clase; esto se logra mediante la sobrecarga de métodos. Los constructores pueden recibir argumentos pero no pueden devolver un valor”<sup>38</sup>.

### 1.3.9. Objetos

“Los seres humanos pensamos en terminos de objetos. Tenemos la maravillosa capacidad de abstracción que nos permite visualizar imágenes en una pantalla como objetos ... Las personas conocen los objetos estudiando sus atributos y observando sus comportamientos. Diferentes objetos pueden tener muchos de los mismos atributos y exhibir comportamientos similares”<sup>39</sup>.

---

<sup>36</sup> Idem, pág 32

<sup>37</sup> Deitel, H. M.; Deitel P. J. *Cómo programar en Java*. México, Prentice hall, 1998, pág 277

<sup>38</sup> Idem

<sup>39</sup> Idem, pág 991



Las clases son la plantilla para la creación de objetos, para utilizarlos debemos declararlos, instanciarlos e inicializarlos. Tenemos un archivo llamado Punto.java el cual contiene a la clase Punto, para crear un objeto de esta clase debemos declararla como: `Punto origen1`; El siguiente paso es instanciarlo y se hace como sigue: `origen1 = new Punto (10,10)`;

Para facilitar la creación de programas en java existen varios softwares de desarrollo como netbeans, eclipse, jcreator, myeclipse, jdeveloper, intelliJ IDEA etc. Estos programas dan un ambiente muy agradable para la programación. Java es un lenguaje que tiene muchas más bondades, aquí traté de exponer las mas importantes.

#### **1.4. MÓDULO IV**

##### **CAPA DE USUARIO.**

A través del diplomado se estudiaron las herramientas que se necesitan para desarrollar un sistema en Web. Estudiamos el lenguaje Java, el Web Server Apache y la metodología con enfoque orientado a objetos para diseñar las aplicaciones de nuestro sistema. Ahora abordaré el tema de las reglas o estándares que se deben tener en cuenta para desarrollar la interfaz de usuario.

Sin duda alguna, cuando las computadoras eran solo dominio de programadores bastaba con tener una línea de comandos. Sin embargo, al evolucionar las aplicaciones y los ordenadores, que pasarán a ser de dominio público, fue necesario crear ambientes agradables a los usuarios.

Así pues, se deben tener en consideración algunas reglas para la creación de un ambiente agradable. La interfaz podemos definirla como el conjunto de componentes empleados por los usuarios para comunicarse con las computadoras.





Cada liga, menú o botón que pongamos en esta pantalla desencadenará una acción en el sistema. En la actualidad el 99% de las interfaces que se han realizado están conformadas por ventanas y un conjunto de controles: botones, cajas de texto, menús desplegables, cajas de diálogos, etc.

Podemos distinguir tres formas de interfaz clásicas de usuario: la interfaz clásica de ventanas, la interfaz integrada en programas de gestión y los navegadores. Elegir una de estas interfaces dependerá del perfil de nuestros clientes y de los usuarios a los que va dirigido. En algunos casos convendrá utilizar las ventanas (por lo habitual que es usar estas interfaces).

Tenemos los siguientes tipos de interfaces:

“ **Interfaces Estáticas:** No tienen cambios y son difíciles de modificar

**Interfaces Dinámicas:** Cambia de acuerdo a los requerimientos “<sup>40</sup>

**“Por su ubicación**

1. Stand-alone. En cada maquina autónoma.
2. Centralizada. En un solo sitio ( Novell ).
3. Distribuida. Internet.

**Por su uso**

Back-end : Soporte, configuración, mantenimiento.

Front-end : Usuario final. “<sup>41</sup>

Ahora es fundamental saber cómo y dónde colocaremos un botón, o dónde colocar una imagen, qué tipo de letra usaremos para una liga o utilizar un menú desplegable. Además no hay que dejar de lado las metodologías que nos ayudarán para desplegar nuestra interfaz.

---

<sup>40</sup>Para ampliar más su información consulte <http://www.itlp.edu.mx/publica/tutoriales/sistsdist2/t53.htm>

<sup>41</sup> Para ampliar más su información consulte <http://www.itlp.edu.mx/publica/tutoriales/sistsdist2/t53.htm>



### **1.4.1. Proceso de diseño**

“Diseñar páginas Web es una actividad compleja. Necesita tiempo para pensar quién va a ver las páginas: cómo ellos la ven y cómo entienden la información, qué computadora utilizan, qué navegador tienen, y qué tan rápido se conectan”<sup>42</sup>.

Basados en los procesos y tareas que los usuarios necesitan realizar, se debe construir un ambiente de trabajo apropiado, es decir, debemos ofrecer lo necesario para que pueda trabajar el cliente.

Para esto se debe tomar en cuenta los siguientes pasos: entender quién usará el sistema y por qué, utilizar un estándar o normatividad de la empresa u organización, bosquejar un diseño, crear un prototipo, evaluarlo con los usuarios y garantizar su construcción.

#### **1.4.1.1. Entender quién usará el sistema y para qué**

Aquí se debe hacer uso de la información que hemos obtenido en el análisis del usuario y tareas (los casos de uso). Se añade, al análisis, la identificación de niveles de usuario que el sistema va a tener y por lo tanto el nivel de acceso, siendo estos dos esenciales para establecer la estrategia de seguridad.

“Los problemas de seguridad son extremadamente complejos porque no solo intervienen problemas tecnológicos, sino también muy complejos comportamientos humanos”<sup>43</sup>.

Por ejemplo, en el caso que trato en este trabajo tenemos los siguientes niveles básicos de usuarios: sólo consulta básica o de área, consulta total, consulta y operaciones básicas o de área, realización de operaciones totales, acceso total, y consulta, operaciones y generación de reporte de área.

---

<sup>42</sup> Ladd, Eric; O’Donell, Jim. *Using HTML 3.2 Java 1.1 and CGI*. United States, Que, 1996, pág 11

<sup>43</sup> Para ampliar más su información consulte <http://webci.com.ar/ayudas/tiposnivelesusuarios.htm>



Son importantes los niveles de usuario, ya que estos nos permitirán evitar la creación de cuentas con privilegios inadecuados, estos niveles de usuario podemos definirlos como los roles. Un rol es lo que un grupo de personas puede hacer en un sistema.

“Se han determinado dos aspectos esenciales en cuanto al establecimiento de los *privilegios*: el primero está referido a las **operaciones permitidas** (*privilegios operativos*) y el segundo a la restricción de los **accesibilidad de ambientes** (*privilegios de accesibilidad*) según los diferentes *niveles* de usuarios definidos”<sup>44</sup>.

En este apartado se desarrolla el Mapa de Navegación General, el cual nos permitirá verificar los procesos que cada nivel de usuario tiene permitido realizar e identificar los procesos que conformarán nuestra interfaz.

#### **1.4.1.2. Utilizar un estándar o normatividad de la empresa u organización**

Es frecuente que algunas organizaciones tengan políticas de diseño para carteles, membretes, documentos, y, tal vez, sistemas o páginas Web. Si es así, se deberá respetar los formatos que la empresa utiliza. Pero en ocasiones, se tendrá que definir los colores, tamaños de letras, logos, frases, etc. Para crear una interfaz de acuerdo a la ideología de la institución, justificando la selección de dicho formato.

#### **1.4.1.3. Bosquejar un diseño**

Este bosquejo se hace en papel, para darnos idea de dónde irá cada elemento y en qué orden se presentarán al usuario. Esto deberá realizarse en cada una de las posibles pantallas que manejemos.

A veces una interfaz contendrá tantos elementos que harán que el usuario llegue a confundirse, aquí es donde debemos tener cuidado ya que lo que buscamos es ayudar, no empeorar la vida del cliente.

---

<sup>44</sup> Idem.



#### **1.4.1.4. Crear un prototipo**

No hay duda que para crear hay que ver. Se trata de crear una vista de lo que será la pantalla, esto sin conexiones a Base de Datos o ejecución de aplicaciones. Esto se hace con la finalidad de corroborar la congruencia de la interfaz con los requerimientos del usuario, ayudándonos a encontrar omisiones y un orden inadecuado de elementos.

#### **1.4.1.5. Evaluación con los usuarios**

“La evaluación es una actividad de vital importancia para cualquier desarrollo de sistemas interactivos que permite comprobar determinados aspectos relacionados con el sistema”<sup>45</sup>. La evaluación debemos hacerla con gente cuyo conocimiento se acerque más a los usuarios reales.

Ellos deben de ser cuestionados en la técnica de “pensamiento en voz alta” la cual consiste en que el usuario vaya narrando todo lo que está pensando al momento de realizar una tarea. Esto sirve para encontrar posibles errores, como un funcionamiento precario del diseño.

#### **1.4.1.6. Construcción**

Una vez que el diseño sea aprobado por el cliente se puede proceder a construir la interfaz, para mi caso usaré el HTML para darle formato al entorno de trabajo en que interactuarán usuarios y sistema. Es conveniente en esta parte del trabajo apoyarse en herramientas como Dreamweaver para la creación de páginas Web.

---

<sup>45</sup> Para ampliar más su información consulte <http://griho.udl.es/i2004/BajarPonencia/5.pdf>



### 1.4.2. Criterios de usabilidad

La usabilidad se refiere a la experiencia de un usuario al interactuar con un sitio Web. No hay que olvidar que los clientes que usan Internet, entran a diversos sitios para buscar información.

“El entorno dentro del cual se inscribe el diseño de una interfaz y la medida de su usabilidad, está dado por tres factores:

1. Una **persona**.
2. Una **tarea**.
3. Un **contexto**.

El diseño de interfaces pertenece a un campo mayor del conocimiento humano, de origen altamente interdisciplinario, llamado **Human Computer Interaction** <sup>46</sup>.

Por ello, debemos tener en cuenta los siguientes puntos:

- Diseño amigable, toda interfaz debe ser amigable, pero más en sistemas Web.
- Congruencia con los procesos, el sistema debe estar estrictamente apegado a los procesos establecidos en el análisis previo.
- Distribución lógica, formularios, botones, listas, tablas, etc. Deben tener un orden que facilite su uso.
- Navegación lógica, el paso entre pantallas debe concordar con los procesos y perfiles del usuario.
- Retroalimentación, uso de mensajes informativos al usuario.
- Consistencia, debe tenerse un formato de diseño y respetarlo durante todo el recorrido.
- Implementación de ayuda, una ayuda no está demás para el buen uso del sistema.

---

<sup>46</sup> Para ampliar más su información consulte <http://sigchi.org/cdg/cdg2.html>



### **1.4.3. Estructura y organización de las páginas de la interfaz**

Es importante tener en cuenta que se debe construir una jerarquía de menús y páginas que parezcan naturales y equilibrados. Una vez más se debe hacer uso de la documentación del análisis, ya que en los Sistemas en Web, la estructura la dan los procesos y generalmente es del tipo vertical con opciones.

Los elementos más utilizados en sistemas Web son encabezados, títulos, imágenes, frames, listas, formularios, tablas, botones, cuadros de diálogo y campos de texto. Esto permitirá saber qué elementos podemos utilizar en el desarrollo de la interfaz.

### **1.4.4. HTML**

Es un lenguaje de marcas que nos sirve para dar formato a la información que deseamos mostrar en un browser (Internet Explorer, Mozilla, etc.). “El lenguaje de marcas de hipertexto (HTML, Hipertext Markup Language) es el lenguaje de marcas que se utiliza actualmente en el World Wide Web”<sup>47</sup>. Este lenguaje permite darle formato a los documentos o archivos que queramos presentar en un navegador de Internet.

Es muy sencillo y fácil de manejar. Sin embargo, por si sólo no puede generar páginas Web dinámicas. Esta desventaja es solucionada por la compatibilidad que ha adquirido HTML, ya que soporta cualquier clase de tags, como las etiquetas especiales para utilizar lenguajes como PHP o Java. A través de los formularios HTML puede mandar peticiones más específicas.

De las etiquetas más importantes están los formularios. Con este conjunto de etiquetas podrá obtener información que el usuario desee ingresar en las: cajas de texto, cajas de selección de opciones, botones de opción, etc.

---

<sup>47</sup> Powell, Thomas. *HTML 4.0 Manual de referencia*. España, McGraw-Hill, 2001, pág 3



“Los formularios se utilizan frecuentemente para tener acceso a la información alojada en la base de datos, como por ejemplo buscar información en un catalogo”<sup>48</sup>. El usuario manda un request (petición) a través de un formulario y el servidor Web debe canalizar esa petición a un controlador para su manejo

En el caso de Java se debe utilizar un server application como Tomcat para hacer nativo Java al HTML. Es decir, no podemos utilizar Java con HTML sin tener un traductor, el servlet engine o Tomcat permite que el servidor Web entienda las etiquetas que utiliza Java.

Así con la combinación de HTML y Java obtendremos la creación de páginas Web dinámicas, que necesitaremos para mostrar los resultados de las consultas hechas a servicios como base de datos, servidor de correos, RMI, etc.

#### **1.4.5. Estructura de un documento HTML**

Cada etiqueta que utiliza HTML tiene un *tag* de <inicio> y un *tag* de </fin> cada etiqueta debe ir encerrada entre pico paréntesis, el documento se puede dividir en dos partes: el encabezado y el cuerpo. Un formato HTML nos permite mostrar la información en listas, en tablas, poner imágenes, tener ligas a otros documentos, usar elementos de formularios, marcos, tipos de letras, colores y etiquetas especiales para usar lenguajes de programación.

```
<HML>
    <HEAD> ..... </HEAD>
    <BODY>
    ....
    </BODY>
</HTML>
```

**Formato de un documento HTML.**

---

<sup>48</sup> Powell, Thomas. *HTML 4.0 Manual de referencia*. España, McGraw-Hill, 2001, pág 380



La etiqueta HTML indica el principio de una página, el *tag* HEAD indica el inicio del encabezado (que sirve para definir diversos valores válidos) y la etiqueta BODY es donde reside la información del documento.

La etiqueta TABLE permite representar y organizar cualquier elemento que se quiera poner en una página Web. Otra forma de organizar grupos de elementos es con el *tag* FRAME, los marcos dividen un browser en verdaderas ventanas individuales. Una tercera opción es la utilización de capas, aunque algunos navegadores no soportan estas etiquetas.

Un sitio está conformado por “n” números de páginas y darle mantenimiento se vuelve muy complicado. Para ello existen herramientas que facilitan la administración de sitios completos, uno de ellos es Macromedia Dreamweaver.

## **1.5. MÓDULO V**

### **CAPA DE DATOS**

“El impacto significativo que ha tenido la información sobre la planificación y la toma de decisiones en las organizaciones ha conducido a un reconocimiento siempre creciente de que la información es un recurso que tiene valor y, por lo tanto, necesita estar organizada y administrada”<sup>49</sup>.

Para esta administración de la información se debe emplear un conjunto integrado de datos que modelen un universo, generalmente compuesto por objetos interrelacionados. Los objetos de un mismo tipo constituyen una entidad que pueden asociarse con otras. Es decir, usar una base de datos.

Mostraré las características de los diferentes Sistemas de Gestión de Base de Datos, así como el lenguaje SQL. Esto servirá para crear consultas más robustas y explotar las bondades que da una base de datos.

---

<sup>49</sup> Gary W. Hanse; James V. Hansen. *Diseño y administración de bases de datos*. pág. 11





Uno de los gestores de base de datos que nos mostraron durante el diplomado, en la capa de datos, es DB2, del cual me enfocaré y mostraré cómo Java puede extraer información de éste manejador de base de datos.

### **1.5.1. Modelos de datos**

“Existen diferentes modelos para interpretar y representar los datos y las relaciones entre ellos. Dichos modelos son una abstracción del mundo real y tratan de representar cómo se agrupan los datos y la relación que existe entre ellos ...”<sup>50</sup>.

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos. La metodología de diseño de datos divide cada modelo en tres esquemas:

- Modelo global (representación grafica);
- Lógico (representa la estructura de datos) y;
- Físico (base de datos).

### **1.5.2. Clasificación de los SGBD**

El criterio principal que se utiliza para clasificar los SGBD es el modelo lógico en que se basan. Son cuatro modelos: el relacional, de red, jerárquico y orientado a objetos, siendo el primero y el último empleados con mayor frecuencia.

Las funciones básicas de estos SGBD son permitir la definición de todos los datos, permitir manipularlos, establecer controles para la seguridad y permitir los accesos concurrentes.

---

<sup>50</sup> Rozic, Sergio E. *Bases de datos y su aplicación con SQL*. MP Ediciones, Argentina, 2004, pág 27



Con ello se puede definir cualquier dato, manipularlo y darle seguridad para conservar una alta integridad a éste y evitar redundancia. “La redundancia de datos es la duplicación de elementos de datos en diferentes archivos. La desventaja mas obvia es el espacio de almacenamiento adicional que se requiere”<sup>51</sup>.

### **1.5.3. Concepción de una base de datos**

Para el desarrollo de una base de datos debe considerar estas tres etapas:

- concepción, es decir, reproducir el mundo real con ayuda de uno de los modelos de datos;
- creación de la BD esquema, se construye la estructura de la base de datos, sin información;
- explotación, cuando los usuarios pueden consultar los datos.

Se debe tener cuidado en la concepción de una base de datos, en las tablas hay que evitar que haya duplicidad de datos. Una mala planeación puede provocar redundancia de datos, pérdida e incoherencia de éstos.

### **1.5.4. SQL para programadores**

*Structured Query Language*. Es un lenguaje de consultas estructurado que se puede utilizar para interactuar con una base de datos. Le permitirá organizar, gestionar y recuperar datos de una base de datos relacional. “El lenguaje SQL es un lenguaje con el cual se escriben todas las acciones u operaciones que se realizan sobre los motores de bases de datos relacionales”<sup>52</sup>.

Este lenguaje ha sido consolidado gracias al Instituto Americano de Normas (ANSI) y por la Organización de Estándares Internacional (ISO) que han hecho de éste un estándar para las bases de datos.

---

<sup>51</sup> Beck, Leland L. *Software de sistemas*. México, 1988, pág 380

<sup>52</sup> Rozic, Sergio E. *Bases de datos y su aplicación con SQL*. MP Ediciones, Argentina, 2004, pág 64



### 1.5.4.1. Sentencias de selección o consultas

En SQL una de las actividades principales que se realizan son las consultas, para ello hace uso del comando SELECT. Esta sentencia es la más potente y compleja instrucción que tiene éste lenguaje, representa la esencia de SQL. SELECT debe combinarse obligatoriamente con el comando FROM y opcionalmente con otras instrucciones como: WHERE, GROUP BY, HAVING, ORDER BY, etc.

Su sintaxis es: SELECT nombre/s\_columna,.. FROM nombre/s\_tabla,.. [alias\_tabla]. Para agregar más nombres de columnas o más nombres de tablas éstas deben separarse por comas. El alias da el nombre completo, según se le quiera dar, a la tabla.

*FROM.* Lista las tablas que contienen los datos a recuperar por la consulta. Si se usan más de una tabla debe tenerse cuidado, ya que los nombres de los campos pueden llegar a coincidir. Para evitarlo se utilizan la siguiente forma: nombre\_tabla.nombrecampo.

*WHERE.* Es un filtro de consultas, es decir que incluya sólo ciertas filas o registros de datos en la consulta sí y solo sí cumplen con los requisitos. Esta sentencia se escribirá después del FROM, SELECT PRECIO FROM MEDICAMENTO WHERE PRECIO < 10.

*ORDER BY.* Ayuda a ordenar el resultado de la consulta dependiendo del campo/s que queramos. Si se omite el campo, por regla tomará el primer campo declarado después del SELECT. Se puede ordenar que sea ascendente o descendente el ordenamiento con DESC y ASC. Por ejemplo:

```
SELECT NOMBRE, PRECIO FROM MEDICAMENTO WHERE PRECIO < 10 ORDER BY NOMBRE ASC
```

*GROUP BY.* Agrupa todas las filas similares y luego produce una fila sumaria de resultados para cada grupo. El resultado de una consulta de 10 personas agrupadas por su sexo podría ser 3 hombres y 7 mujeres.

```
SELECT SEXO, EDAD FROM PERSONA WHERE EDAD = 20 GROUP BY SEXO
```

*HAVING.* Es un filtro para los grupos. Sólo es válida si previamente se ha especificado la cláusula GROUP BY.

```
SELECT SEXO, EDAD FROM PERSONA GROUP BY SEXO, EDAD HAVING EDAD < 30
```



Se puede realizar una consulta con dos sentencias SELECT utilizando el operador UNION. La lista de selección para cada sentencia SELECT debe tener el mismo número de expresiones de columnas con el mismo tipo de datos y en el mismo orden.

```
SELECT CLAVE, NOMBRE FROM MEDICAMENTO UNION SELECT CLAVE, NOMBRE FROM FARMACIA
```

#### **1.5.4.1.1. Funciones de agregación**

Las funciones de agrupamiento pueden ser usadas junto con el SELECT. Estas instrucciones hacen que la consulta arroje un solo valor, los comandos de agrupamiento son SUM (suma total de la columna afectada), AVG (media de la suma total), COUNT (retorna el número de registros con valores no nulos en ese campo), MAX (valor más alto del campo), MIN (valor más bajo).

#### **1.5.4.1.2. Enlace de varias tablas**

Una de las ventajas que dan las bases de datos relacionales es combinar varias tablas, en una consulta, para obtener información. Para realizar éste enlace a cada campo se le agrega el nombre de la tabla de donde procede: PERSONAL.NOMBRE, FARMACIA.NOMBRE.

```
SELECT PERSONAL.NOMBRE, CLAVE.FARMACIA FROM PERSONAL, FARMACIA WHERE PERSONAL.CLV_PERSONAL = FARMACIA.CLV_PERSONAL
```

Con todos estos elementos se pueden armar consultas tan complejas cómo sean necesarias, siempre teniendo cuidado de usar la instrucción adecuada.

#### **1.5.4.2. Sentencias CREATE, DROP, INSERT, DELETE y UPDATE**

SQL maneja instrucciones para crear, borrar, insertar datos y actualizar las tablas. CREATE TABLE, con ésta se pueden insertar más tablas a la base de datos como se requiera. DROP TABLE, borrará por completo el contenido y a la tabla misma. INSERT, podrá con esta sentencia ingresar datos en las tablas. DELETE, podrá borrar



registros de una tabla. UPDATE, actualiza los datos de varios registros. Se debe tener en cuenta el SGDB utilizado, ya que los tipos de datos pueden variar entre uno y otro SGDB.

### 1.5.5. JDBC API

*“Java Data Base Connectivity. Es el estándar de la industria para la conectividad independiente a bases de datos de Java. Con este controlador, Java, puede ejecutar instrucciones SQL”*<sup>53</sup>.

Sus funciones básicas son: establecer una conexión con una base de datos o tener acceso a cualquier fuente de datos tabular, ejecutar sentencias SQL y procesar los resultados de las consultas. Este driver está escrito completamente en Java por lo que es independiente de la plataforma.

Dependiendo de la filosofía de trabajo los drivers JDBC se pueden dividir en cuatro tipos: direct-to-database pure Java driver, driver para middleware de bases de datos, un driver mixto Java-API nativo y el puente JDBC-ODBC. Para establecer, con Java, una conexión hay que seguir dos pasos: cargar el driver y hacer la conexión.

Para cargar los drivers se utiliza una sola línea de código, que consiste en el nombre del controlador contenido en la clase `Class.forName`. Para establecer una conexión se requiere la siguiente línea de código:

```
Connection con = DriverManager.getConnection(uri, "login", "contraseña");
```

La información que requerimos es una cuenta de usuario de la base de datos (login con su contraseña) y la URL representado por la variable uri.

---

<sup>53</sup> Para mas información consultar <http://java.sun.com/docs/books/tutorial/jdbc/basics/index.html>



Una vez establecida la conexión con la base de datos se puede empezar a ingresar datos. Antes, se tiene que crear una línea de comandos, un Statement. Éste objeto es el que envía nuestras sentencias SQL al controlador de la base de datos. La sentencia es:

```
Statement stmt = con.createStatement();
```

Ahora que ya esta creada una línea de comando sólo faltara crear y ejecutar una sentencia SQL. Utilizando uno de los métodos que tiene definido la clase Statement podrá ejecutar cualquier sentencia, ejem:

```
stmt.executeUpdate("CREATE TABLE PERSONA (ID INTEGER)");
```

Para recuperar y realizar consultas, se utilizará la clase ResultSet. La siguiente línea de código muestra como ejecutar un SELECT:

```
ResultSet rs = stmt.executeQuery ("SELECT NOMBRE FROM PERSONA");
```

La clase ResultSet tiene métodos para recuperar los datos de cada columna. Estos son algunos métodos: `getBytes`, `getLong`, `getInt`, `getDate`, `getString` y muchos más. Dependiendo el tipo de dato será que método se deberá utilizar.

La utilización de Java para realizar consultas SQL a las bases de datos le permitirá separar y proteger la integridad de la información. Esto es, ninguna consulta, inserción de datos, creación de tablas será realizada directamente en nuestro manejador de la base de datos.

Con una aplicación Java bien diseñada se puede conectarse a más de una base de datos, sin tener que programar otra aplicación. Con sólo proporcionar el nombre del controlador, una cuenta de acceso y la ruta de la SGBD, obtendrán la conexión a la base de datos que quiera. Tenga en cuenta que deberá contar con el driver adecuado para su BD.



## 1.6. MÓDULO VI

### REGLAS DEL NEGOCIO.

La agrupación de todos los componentes que tienen la funcionalidad de imponer ciertas restricciones o reglas que gobiernan el comportamiento del sistema desarrollado se le conoce como la “capa de negocios” o “lógica empresarial”. “La lógica empresarial se traslada a un servidor el procesamiento orientado a la empresa. El servidor de aplicaciones maneja las conexiones a la base de datos y realiza todas las peticiones de datos a la base de datos”<sup>54</sup>.

“Generalmente se acepta que un servidor de aplicaciones maneja todas las transacciones lógicas y de conectividad que históricamente compartían el cliente y el servidor en un diseño cliente/servidor”<sup>55</sup>.

En todo proceso que lleve a cabo una empresa o institución se pueden encontrar restricciones, limitaciones y reglas. Estas reglas indican a los trabajadores lo que está permitido y lo que no está permitido hacer. Por lo que, para que se cumplan las reglas establecidas por una organización se deben seguir una serie de pasos.

Al desarrollar las reglas de negocio se empieza a determinar como deberá responder la aplicación a las diferentes situaciones. La capa de negocios se encargará de validar y aprobar las diferentes acciones que se lleven en la aplicación por parte del usuario. Por ejemplo, en un sistema de venta por medio de Internet una regla de negocio será no vender más de lo que existe en los almacenes.

Obviamente una petición que se considere inválida generará un error, la aplicación deberá estar preparada para salvar la situación. Se debe contemplar los pasos que lograrán regresar al sistema a un buen funcionamiento y que se cumpla la regla.

---

<sup>54</sup> Macdonald, Michael; Cagle, Kurt. *Visual Basic 6 programación cliente/servidor*. Paraninfo, España, 1999, pág 10

<sup>55</sup> Para más información consultar

[http://www.janium.com/phpws/index.php?module=documents&JAS\\_DocumentManager\\_op=downloadFile&JAS\\_File\\_id=1](http://www.janium.com/phpws/index.php?module=documents&JAS_DocumentManager_op=downloadFile&JAS_File_id=1)



El desarrollo de las reglas de negocios es sin duda la capa mas difícil de acotar, así como debe estar en contacto con la capa de usuario también debe estar en contacto con la capa de datos. Las transacciones de información de la base de datos hacia la pantalla del usuario es controlado por las reglas de negocio.

### **1.6.1. Definición de reglas del negocio**

Dentro de la arquitectura de tres capas una de las tareas más difíciles de establecer son las reglas del negocio. Ya que aquí se encuentran las reglas que gobiernan a toda la aplicación. “Los servicios de negocios son el “puente” entre un usuario y los servicios de datos. Responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo”<sup>56</sup>.

Una petición no se tomará en cuenta si no cumple ciertas reglas, una modificación de los datos o el ingreso de los mismos no se hará si no se toman a consideración las reglas definidas.

Es decir todos los componentes que forman parte de esta capa realizarán la función de una puerta, ésta puerta permitirá o no la utilización de la aplicación. Por lo que, se debe tener muy en cuenta la funcionalidad que debe tener el sistema.

Toda petición tiene como fin una respuesta, de los límites que marquemos dependerá el funcionamiento adecuado de la aplicación. Si una regla es mal diseñada se pondrá en peligro no solo el funcionamiento, sino también, la integridad de la información.

### **1.6.2. Tipos de reglas de negocio**

Dentro de las reglas de negocio se pueden identificar cinco grupos, se puede decir que son cinco tipos de reglas de negocio, que son los siguientes:

---

<sup>56</sup> Para mas información consultar <http://www.geocities.com/trescapas/capadenegocios.htm>





- *Reglas del modelo de datos.* Son todas aquellas que se encargan de controlar que la información básica almacenada para cada atributo o propiedad de una entidad u objeto sea válida. Ejemplo: no hay existencias de producto negativas.
- *Reglas de relación.* Son aquellas que controlan las relaciones entre los datos. Ejemplo: un cliente puede realizar un pedido sólo si este cliente está dado de alta en el sistema.
- *Reglas de derivación.* Son aquellas que a partir de cierta información pueden generar otra. Ejemplo: La suma de todas las ventas nos dan la venta total.
- *Reglas de restricción.* Son aquellas que restringen los datos que el sistema puede contener. Ejemplo: El sueldo de una persona no puede bajar de un mínimo establecido para la obtención de cierta tarjeta de crédito.
- *Reglas de flujo.* Son aquellas que indican qué camino recorre la información y obligan a que sigan solo los caminos válidos.

### **1.6.3. Capas en un sistema Cliente/Servidor**

En una arquitectura Cliente/Servidor existen solo dos capas, el cliente y el servidor. "El sistema cliente / servidor significa, simplemente, que dos (o más) procesos funcionan independientemente de una forma cooperativa"<sup>57</sup>. En relación a los gestores de base de datos hay quienes soportan bastantes reglas de negocio, mediante el uso de llaves primarias, integridad referencial, etc., y quienes solo proporcionan soporte para reglas de negocio.

Usando gestores de base de datos potentes, el programador puede relajarse en el tema de las reglas de negocio, ya que, podemos crear muchas reglas de integridad referencial, en la BD. Así, el gestor de base de datos aceptará o rechazará cualquier inducción de datos según marque la regla.

---

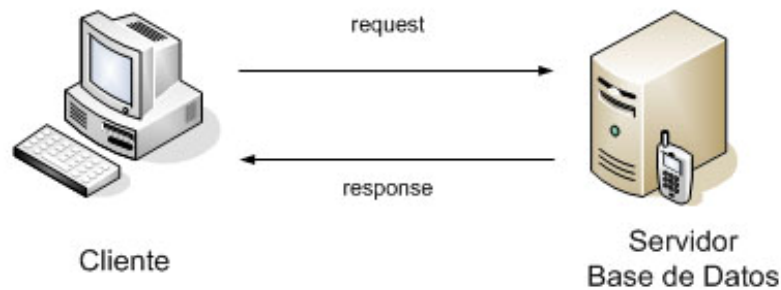
<sup>57</sup> Macdonald, Michael; Cagle, Kurt. *Visual Basic 6 programación cliente/servidor*. Paraninfo, España, 1999, pág 8



Si se utiliza un gestor menos potente, el programador deberá agregar las reglas de negocio en varias líneas de código en los programas que accedan a la base de datos. El problema de programar las reglas de negocio surge cuando esto se hace a través de varias herramientas.

Conforme las reglas son más especializadas, la programación de éstas es inevitable. Por ello, para evitar la duplicidad de cada una de las reglas se requiere que todo el código se encuentre situado en un solo lugar. Además, existe la posibilidad de que un sistema tenga varios gestores de base de datos, asegurar la integridad referencial de la información solo se puede hacer a través de cada aplicación que acceda a estas bases de datos.

Mantenerlo en el gestor de base de datos no es muy viable, por lo que, se debe utilizar una nueva aplicación. La integración de este paso intermedio entre el cliente y la base de datos hará que el sistema Cliente/Servidor pase a ser un sistema de tres capas.



**Figura 1.5 2 capas Cliente / Servidor**



**Figura 1.6 3 capas Cliente / Servidor**



#### 1.6.4. Ubicación de las reglas de negocio

“Las reglas son un elemento clave de la Web Semántica que hace posible la integración, derivación y transformación de los datos desde distintos recursos de forma distribuida, transparente y escalable”<sup>58</sup>.

La ubicación de cada regla se puede simplificar si se toma en cuenta el tipo de regla al que pertenece, utilizando los cinco grupos de reglas. Así, las reglas de modelo de datos se ubicaran tanto en el servidor como en el cliente, al ubicarlas en el servidor se dará mayor robustez a la base de datos y, al ubicarlas en el cliente aseguramos la validación mas rápidamente para el cliente.

Por otro lado, las reglas de relación se ubicarían en el lado del servidor, ya que en el servidor está el gestor de base de datos, es lógico que para mantener una mayor integridad referencial éste tipo de reglas debe estar en el mismo sitio. Cabe aclarar que, si se utilizan más de dos gestores se debe colocar estas reglas en la capa intermedia, esta capa intermedia puede localizarse en la misma máquina donde se encuentra el gestor de datos.

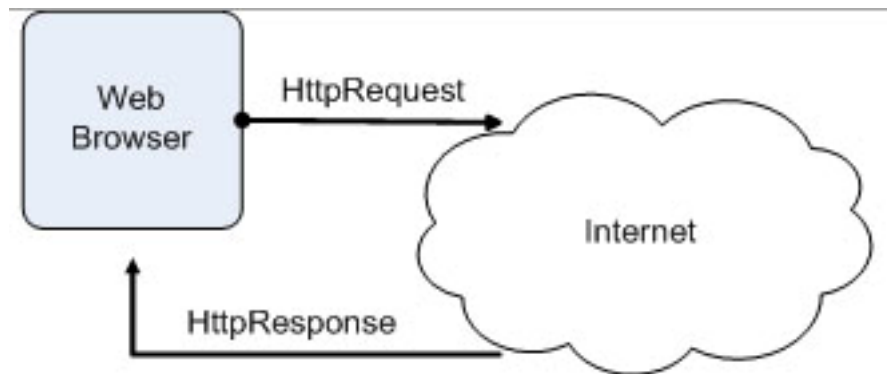
Las reglas de derivación se ubicarán en la capa intermedia o del lado del servidor si cumple con ciertas condiciones, previniendo consultas extremadamente grandes. Es mejor enviar un único resultado que enviar todas las consultas realizar las operaciones y dar un resultado total a través de la red.

Las reglas de restricción lo mas conveniente es ponerlas en la capa intermedia, así se evita alguna complicación con las limitaciones del SQL de los gestores de base de datos. Y las que son casi obvias ubicarlas dentro de la capa intermedia son las reglas de flujo. Mucho dependerá de cada caso, ya que las reglas son únicas, pero, por lo general se siguen las recomendaciones anteriores.

---

<sup>58</sup> Para más información consulte: [http://www.w3c.es/prensa/2005/nota050427\\_RuleLanguage](http://www.w3c.es/prensa/2005/nota050427_RuleLanguage)

### 1.6.5. Arquitectura de aplicaciones Web con Java



**Figura 1.7 Intercambio básico Http<sup>59</sup>**

Cuando se genera una petición de algún usuario en nuestra aplicación Web, se emplean Servlets que atienden y generan nuevos hilos de ejecución para entregar una respuesta a la petición. Para ello se utilizan los métodos `doPost()` o `doGet()` y, para hacer llegar los parámetros a los Servlets se usan el `HttpServletRequest` y el `HttpServletResponse`.

Todo esto lo realiza lo que se llama el Web Container, éste ofrece los siguientes servicios:

- Soporte a comunicaciones.
- Manejo del ciclo de vida de los Servlets
- Soporte a ejecución multithreading.
- Seguridad declarativa.
- Soporte a JSP's.

El Web Container está diseñado para facilitarle el trabajo al programador, es decir, ayuda a que los esfuerzos se encaminen al desarrollo de las reglas de negocio de su aplicación más que escribir código para manejar la ejecución multithreading, la seguridad o bien el acceso a servicios de red. Un ejemplo es Apache Tomcat.

<sup>59</sup> Patzer, Andrew. *JSPEXamples and Best Practices*. Apress, Estados Unidos, 2002, pág 2



### 1.6.6. JSP y Servlets

Una página dinámica permite tener una interactividad enorme con los usuarios, acceder a servicios de base de datos, de servidores de correo, etc. Ésta esta conformada con código HTML y un lenguaje de programación.

Todo inicio con los CGI, pequeñas aplicaciones que se pueden programar con lenguajes como Perl o C, el Common Gateway Interface. Un problema de los CGI es la sobrecarga de procesos “con el tradicional CGI, un nuevo proceso empieza por cada petición HTTP”<sup>60</sup>

Como respuesta del gran gigante del Software, Microsoft, creó ASP (Active Server Page), sin embargo, tardo mucho tiempo en tener versiones estables, junto con el y en paralelo surgió PHP. Otro gigante entró a la competencia, Sun Microsystem con su lenguaje Java crea los Servlets.

Los Servlets se caracterizan por tener incrustado código HTML dentro de código Java, los Servlets son más rápidos y más potentes que un CGI estándar “Los Servlets soportan muchas capacidades que son difíciles o imposibles de realizar con los simples CGI”<sup>61</sup>, son transportables y extensibles. Sin embargo una de los inconvenientes era escribir sentencias casi infinitas de *println()* para el despliegado de un documento HTML.

Con todo lo anterior, la solución llego a través de la creación de los JSP's, Java Servlets Page. Todo lo contrario de los Servlets, un JSP es código Java incrustado en código HTML, esto permitía mezclar HTML con Java conservando toda la potencialidad de los Servlets. “La tecnología JavaServer Pages (JSP) permite mezclar de forma regular, HTML estático con contenido generado dinámicamente”<sup>62</sup>. El problema era que no se metía mucho con el control de flujo y otros problemas de las aplicaciones Web.

---

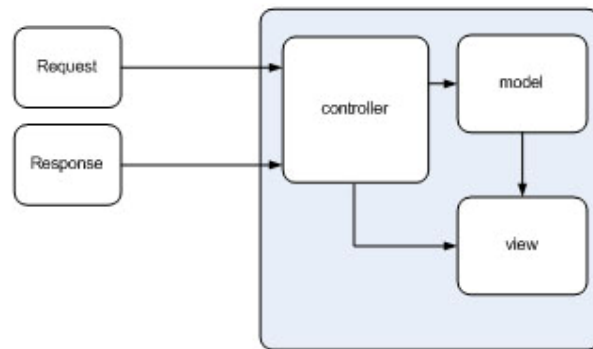
<sup>60</sup> Hall, Marty. *Servlets and Java Server Page*. Prentice Hall, Estados Unidos, 2002, pág 37

<sup>61</sup> Idem, pág 38

<sup>62</sup> Idem, pág 121

Sin embargo, la conjunción de Servlets y JSP hacían una buena solución, mientras los Servlets se utilizaban para el control de flujo, los JSP's se concentraban en la presentación con HTML. A esto se le denominó el Modelo 2 de JSP donde muchos coinciden que sigue el patrón de diseño Modelo-Vista-Controlador de Small Talk.

### 1.6.7. Modelo Vista Controlador



**Figura 1.8** Arquitectura MVC<sup>63</sup>

El MVC fue creado para separar la lógica del negocio de la presentación, esto lo logra intercalando una capa más, así, la vista o capa del usuario es independiente de las reglas del negocio. Éste patrón de diseño el Controlador central es el encargado de dirigir el flujo de la aplicación. Este modelo no es exclusivo de Servlets y JSP. En sistemas Web mientras más limpia sea la relación entre la capa de presentación y la de negocio será más sencilla su implementación.

**Controlador.** Examina y extrae parámetros del objeto request –solicitudes http- y define que servicio se requiere de los que publica el modelo. Invoca objetos del negocio, publicados en el modelo, haciendo llegar los parámetros extraídos del objeto request. Genera un log en la capa Web, mantiene el estado de una sesión.”La parte del controlador del diseño MVC de nivel Web normalmente es un servlet Java”<sup>64</sup>.

<sup>63</sup> Patzer, Andrew. *JSP Examples and Best Practices*. Apress, Estados Unidos, 2002, pág 109

<sup>64</sup> Cavaness, Chuck. *Jakarta Struts*. Anaya multimedia; O’reilly, Madrid, 2005, pág 41



**Modelo.** Contiene las reglas del negocio y el estado del sistema. Una vez que el controlador completa su proceso y selecciona una vista el modelo contendrá toda la información a desplegar.

**Vista.** Despliega la información que contiene el modelo. Una vista no necesita saber nada acerca de la implementación del controlador o de los objetos del negocio. La vista es responsable de la presentación, así como, de obtener la entrada del usuario y enviarla al controlador. “Las vistas dentro del patrón MVC de nivel Web constan por lo general de páginas HTML y JSP”<sup>65</sup>.

#### **1.6.7.1. Técnica de desarrollo de aplicaciones utilizando una herramienta de productividad al desarrollo**

El usar un patrón de desarrollo implica ordenar la forma de trabajo, sobre todo cuando los proyectos son muy grandes, implicando la conjunción de varias personas trabajando en equipo.

Una herramienta de productividad al desarrollo es Struts, el proyecto Struts se inicio en el año 2000. Esta técnica de desarrollo proporciona un marco de trabajo MVC estándar a la comunidad de Java.

Struts es un framework de desarrollo auspiciado por la Apache Software Foundation como parte del proyecto Jakarta. Ésta herramienta se encuentra bajo licencia libre, por lo cual, no tiene costo alguno y es libre de uso. Un framework o marco de trabajo es: “un conjunto de clases e interfaces que cooperan para solucionar un tipo específico de problema de software”<sup>66</sup>.

Un framework es una colección de interfaces que han sido diseñadas para trabajar en conjunto y manejar un tipo particular de problema. En el caso del framework de Struts,

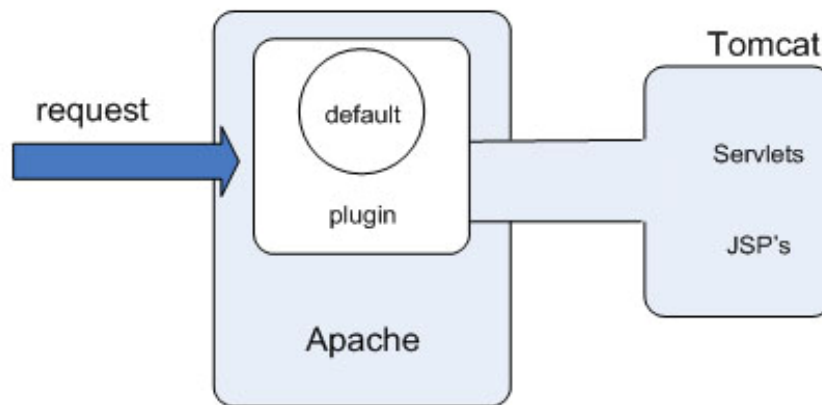
---

<sup>65</sup> Cavaness, Chuck. *Jakarta Struts*. Anaya multimedia; O’reilly, Madrid, 2005, pág 41

<sup>66</sup> Idem, pág 42

esta orientado a solucionar problemas de aplicaciones en Web, ayudando al desarrollo y mantenimiento de aplicaciones complejas.

Hay tres formas de montar un servidor para una aplicación Web con Java, estas formas de montar el servlet engine son: 100% java, servidor hibrido y utilizando Apache con un plugin de Tomcat.



**Figura 1.9 Apache con plugin de Tomcat**

#### **1.6.7.2. Jakarta Struts: MVC en acción.**

Struts tiene tres componentes principales: un servlet controlador (Controlador), páginas JSP (vista) y la lógica de negocio de la aplicación (modelo). Esta API de Java contiene los siguientes componentes:

- ActionServlet. Controla el flujo de ejecución (controlador).
- Action. Se utiliza para acceder a los objetos de negocio. Al recibir una petición este se encargara de proporcionar la ruta de la clase adecuada para resolver dicha petición.
- ActionForm. Para validar la entrada de datos, así el objeto action no realiza los request.
- ActionForward. Se encarga de re-direccionar, a un JSP, la respuesta a la petición
- Struts-config.xml. Contiene la configuración de la aplicación, incluyendo seguridad, acceso a base de datos y mapeo de Servlets.



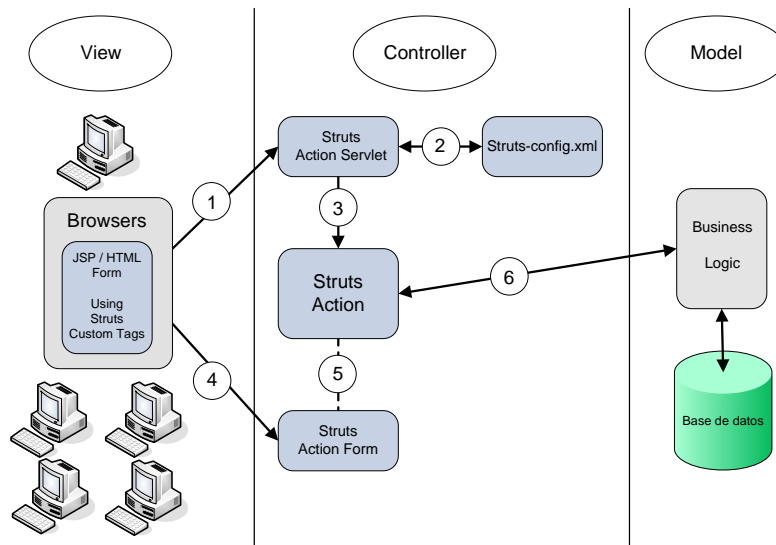


Figura 1.10 Struts en operación<sup>67</sup>

La operación del framework de struts inicia con la petición de un usuario a la aplicación Web, un request es la forma en que se manda una solicitud, ésta es tomada por el componente ActionServlet que a su vez accede al struts-config.xml para saber hacia donde mandar esa petición.

Una vez que sabe que hacer con la petición la manda al ActionForm quien valida la entrada de datos y genera un JavaBean que obtendrá los datos en formato listo para usarlos con cualquier aplicación Java. Después de terminar vuelve al ActionServlet para que éste, basandose en struts-config.xml, lo asocie a un objeto Action. El objeto Action resuelve la petición invocando a los objetos del negocio que sean necesarios por parte del modelo. Para que después, regrese una vez más al ActionServlet y este localice en struts-config.xml la vista o página JSP adecuada para presentar el resultado. Una vez definida la vista se invoca y se presenta el resultado en una página HTML.

En forma muy general así es como trabaja Struts, hay que recalcar el trabajo que realiza el controlador al dirigir todo el flujo de la aplicación, sin duda parte medular lo forma el ActionServlet que junto con el struts-config.xml toman el mando del estado de la aplicación. Un problema clave en el diseño de aplicaciones en Web es retener y validar datos provenientes de un request, con Struts se pueden definir un conjunto de FormBean y almacenar fácilmente los datos de un formulario.

<sup>67</sup> Para más información consulte: <http://www.javafree.org/content/view.jf?idContent=22>



## CAPÍTULO II

### ANTECEDENTES Y PLANTEAMIENTO DEL PROBLEMA

#### 2.1. Sistema Farmacéutico Morelos (SFM)

##### *Sistema para el monitoreo del medicamento de una farmacia*

El SFM es el nombre que, basado en las siglas del Sistema Farmacéutico Morelos, le dimos a la aplicación que resultará en consecuencia del análisis que se ha hecho en este caso en particular. Ésta aplicación trabajará en un entorno Web, utilizando tecnología Java.

El sistema deberá captar el movimiento de la mercancía de la empresa grupo Morelos. Las opciones de movimiento que deberá aceptar serán las altas, bajas y cambios. Las altas, se refiere a la acción que ejecutará la empresa al adquirir mercancía de sus proveedores. Las bajas, será la operación de vender el medicamento a los clientes (mayoritarios y minoritarios) o dar de baja por caducidad al producto. Los cambios, la organización cuenta con más de una sucursal por lo que ésta debe abastecerlas, el cambio significará a qué filial se dirige la mercancía.

El sistema deberá ser capaz de entregar reportes de todos estos movimientos, como también, auxiliar a la empresa con los cortes de caja. Otro servicio que deberá cumplir el sistema es dar información a través de Internet. Esto es, cualquier cliente podrá buscar los productos que ofrece la organización a través de la Web. Por lo que el sistema deberá auxiliar a la organización en referencia a sus descuentos y promociones, haciéndolos automáticos y transparentes.

#### 2.2 Entorno histórico de la Farmacia.

“Hasta ahora clientes; desde ahora, pacientes”

En el siglo XIX en México las farmacias y boticas eran donde se combinaba el arte de manufacturar los medicamentos, el trabajo manual y la ciencia farmacéutica. A finales



de dicho siglo comienza la separación de estas tres actividades buscando la profesionalización de cada una de ellas.

Diagrama de flujo del proceso de venta y su registro que realiza la empresa Grupo Morelos

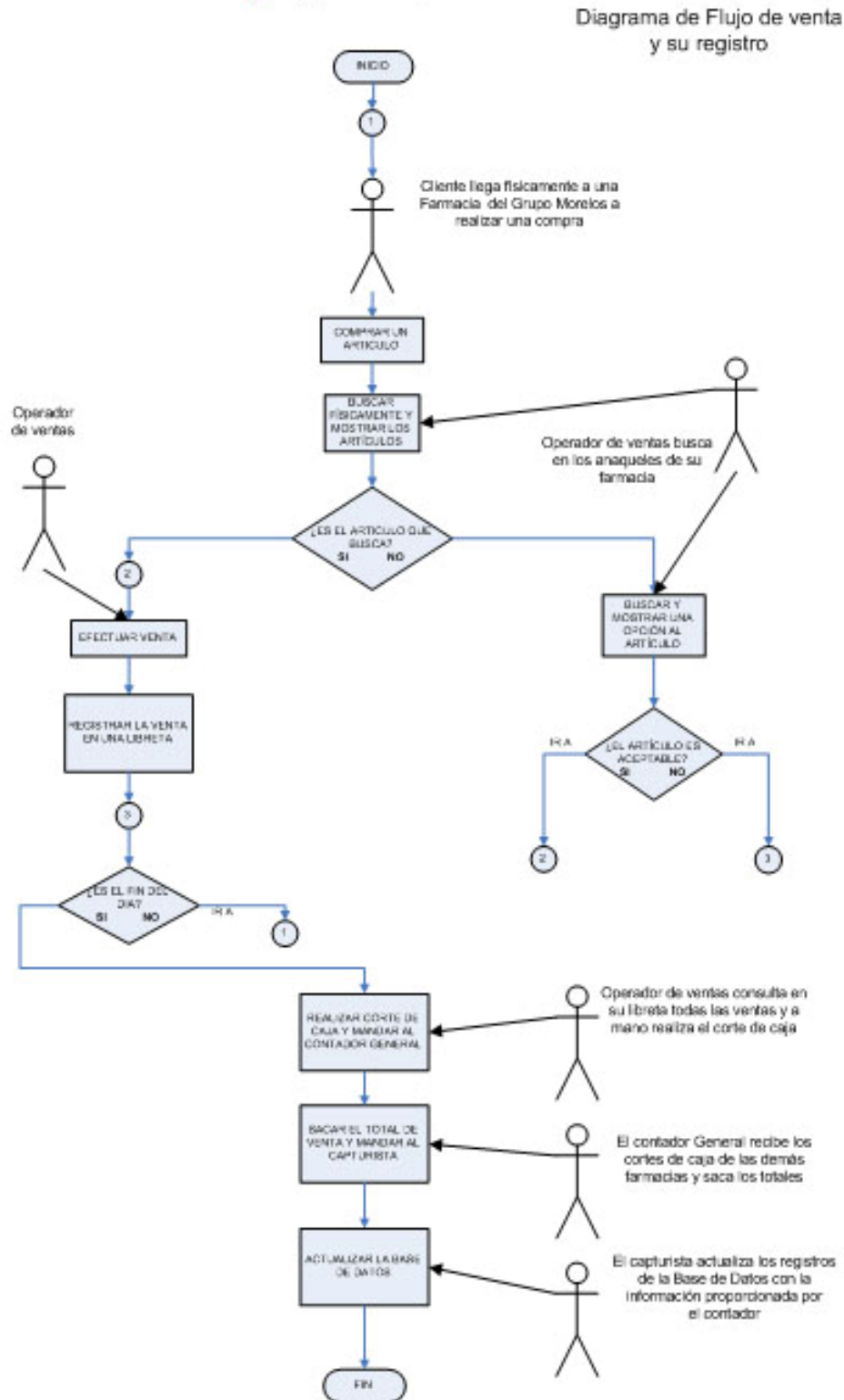


Figura 2.1 Diagrama de Flujo



Así, a finales del siglo XIX la elaboración de los medicamentos estaba asegurada por boticarios y farmacéuticos. Los farmacéuticos eran personas egresadas de la Escuela Nacional de Medicina o de escuelas situadas en el interior de la República Mexicana, mientras los boticarios eran personas de oficio.

La Sociedad Farmacéutica Mexicana estaba conciente de que el avance de la medicina requería más preparación de los farmacéuticos, ya no era tan fácil confiar en boticarios o personas no preparadas para la manufacturación del medicamento. Así, llegaron a desaparecer en gran medida los boticarios, prevaleciendo las farmacias.

En nuestros días es fácil encontrar que las farmacias han empezado a ofrecer nuevos servicios, que antes no necesitaban. En 1984 en el informe Almagro<sup>1</sup> un grupo de farmacéuticos se reunieron para buscar nuevas salidas a su actividad, ya que atravesaba una grave crisis debido a la gran frustración que sentía al ser infrautilizados por la sociedad, a pesar de la gran formación de sus profesionales y nació el concepto de *nueva farmacia*.

Por otro lado, en Estados Unidos amenazaba fuertemente la venta por correo de especialidades farmacéuticas, y surge desde entonces la necesidad de dar respuesta profesional para detener estas prácticas que estaban en contra de la farmacia tradicional.

Esta venta por correo fué la que detonó el cambio de actitud de la farmacia tradicional, se tenía que usar la inventiva para poder detener y hacer volver a los clientes hacia las farmacias.

La OCU, Organización de Consumidores y Usuarios, declaró las siguientes recomendaciones que una nueva farmacia debía tener presente:

1. Los consumidores desean tener una farmacia cerca de su domicilio.
2. Que las farmacias estén bien y siempre abastecidas.
3. Flexibilidad de horarios y guardias que garanticen un buen servicio.

---

<sup>1</sup> Informe Almagro. Una reflexión sobre el presente y futuro de la farmacia española.



4. Comodidad en el acto de comprar, es decir fuera vitrinas protectoras y persianas metálicas.
5. Poder efectuar los pagos con tarjetas.
6. Documento que acreditará la realización de una compra.
7. El consumidor valora muchísimo el trato personalizado del farmacéutico y su papel en las campañas de educación sanitaria.
8. El consumidor valora igualmente los servicios complementarios que ofrece la farmacia.
9. El consumidor acepta las macro farmacias y los productos para farmacéuticos.
10. La OCU acepta que haya un aumento de puntos de venta y que bajen los precios al aumentar la competencia.

Obviamente todos los puntos anteriores están enfocados a los intereses económicos de las farmacias, ya que el reto era tomar de nuevo el mercado de la venta de medicamentos. Entre más se acercara, la farmacia, al cumplimiento de éste decálogo aseguraría tener una buena presencia dentro del mercado.

Pero en sí, ¿Qué es la nueva farmacia?, es una nueva forma de ofrecer servicios. Hasta ahora clientes, desde ahora pacientes. Antes una farmacia se quedaba en lo que hay, lo que se tiene se ofrece. Hoy una farmacia se asegura de lo que puede haber. Tenemos que, la farmacia puede manejar síntomas menores, sigue con la atención de venta de medicamentos, tiene consultas con médicos y se acerca más a la comunidad con información sanitaria.

Hoy en día las farmacias ofrecen servicios que intentan garantizar retener a viejos clientes y captar nuevos posibles clientes. Pero a la vez, surge el problema de manejar grandes cantidades de volúmenes de información, y más aun, si se trata de una cadena de farmacias. El manejar de manera rápida y eficaz toda la información que genera día con día una farmacia es garantizar el cumplimiento del decálogo propuesto por la OCU.

Hoy en día las farmacias ofrecen servicios que intentan garantizar retener a viejos clientes y captar nuevos posibles clientes. Pero a la vez, surge el problema de manejar grandes cantidades de volúmenes de información, y más aun, si se trata de una cadena



de farmacias. El manejar de manera rápida y eficaz toda la información que genera día con día una farmacia es garantizar el cumplimiento del decálogo propuesto por la OCU.

### **2.3. El problema de una organización Farmacéutica.**

El Grupo Comercial Morelos surge en el año de 1996 en la Ciudad de México, su ubicación se encuentra al sur de la ciudad. Su centro de operaciones se encuentra en la delegación política de Xochimilco.

Surge ante la visión comercial de sus fundadores. El grupo Morelos observó que en la delegación Xochimilco existían varias farmacias que solo ofrecían el servicio de la venta de medicamentos. La visión que tiene la organización consiste en adoptar una nueva forma de dar servicio, el servicio que empiezan a implantar es, que en cada filial que abrieran se dará servicio de consultorio médico.

El servicio de consultorio médico consiste en dar un servicio profesional de médicos a bajo costo, tratando de mejorar la calidad de vida de los habitantes locales. La organización se dio cuenta de que la mayor parte de la población local pertenecen a un nivel socioeconómico bajo y que a veces no pueden pagar medicamentos patentados.

Por ello la organización se concentra en la venta de medicamentos genéricos que son tan efectivos como los patentados pero mucho más baratos. Así, la política de la empresa es dar un servicio de calidad a bajo costo y vender medicamentos de buena calidad a bajo costo. Obteniendo con su servicio de consultorio buenos resultados y mejorando la calidad de vida de sus clientes.

**Problemas y desventajas actuales**

	<b>Ventaja</b>	<b>Desventaja</b>
Administración de consultas médicas	Se mantiene un control de la cantidad de consultas generadas por día.	Los registros se llevan de manera manual, las anotaciones se hacen en libreta.
Administración de ventas de producto	Se lleva un registro de las ventas realizadas en cada farmacia.	La anotación de cada venta se registra en libreta, es complicada su recopilación. No se da un seguimiento automático.
Gestión de Reportes de procesos	Se obtiene información actual para la toma de decisiones de la empresa.	Los reportes se generan por períodos largos, depende de que tan rápida sea la recolección de información en cada farmacia.
Abastecimiento de productos	Se obtiene un presupuesto en base a las necesidades de las farmacias.	El abastecimiento es tardío, al llevar sus registros manualmente la información es tardía, se afecta la toma de decisiones.
Manejo de inventarios	Se tiene el control de las existencias.	El conteo se lleva de forma manual, registrándose en libreta, no se lleva un control actualizado de las existencias.

**Tabla 1**

La empresa es actualmente dirigida por el L.C. Marco T. Zavala Gutierrez quien funge como el Director General de la organización. Su sede se encuentra en la siguiente dirección Dienteles 23 Col. Jardines del Sur, Xochimilco D.F.

Actualmente el grupo Morelos cuenta con tres filiales. La Farmacia Morelos 1 es la primer filial que funda la organización, en 1996 esta farmacia abre sus puertas a sus clientes. Esta farmacia funge como el centro de operaciones del grupo Morelos.

Esta farmacia se encuentra trabajando de lunes a sábado manejando los horarios matutino y vespertino. La filial cuenta con los siguientes puestos: personal de mostrador y personal médico.



La Farmacia Morelos 2 es la segunda filial en fundar la organización, esta farmacia nace en marzo del 2002. Al igual que su predecesor trabaja de lunes a sábado manejando el turno matutino y vespertino. Entre el personal que labora dentro de la farmacia esta el personal de mostrador y el personal médico.

La Farmacia Morelos 3 es la ultima farmacia que el grupo Morelos crea, es abierta en diciembre del 2004. Siguiendo con la filosofía del grupo Morelos también cuenta con servicio médico y trabaja un turno matutino y vespertino de lunes a sábado. Cuenta con personal de mostrador y personal médico.

La organización cuenta actualmente con 16 personas empleadas en sus diferentes filiales. Maneja horarios completos, matutino, vespertino. La siguiente lista en la *Tabla 1* muestra los puestos que maneja el grupo Morelos y cuanta gente labora en cada puesto:

<b>Puesto</b>	<b>Numero Personal</b>
Director General	1
Recursos Humanos	1
Capturista	1
Contadora	1
Almacenista	1
Repartidor	1
Personal de Mostrador	6
Personal de Aseo	1
Médicos	4

**Tabla 2**

El siguiente personal trabaja de lunes a viernes de las 9:00 a 19:00 hrs:

- Director General.
- Recursos Humanos.
- Capturista.





- Contador.
- Almacenista.
- Repartidor.

El siguiente personal trabaja de lunes a sábado de las 9:30 a 20:00 hrs:

- Médico.

El siguiente personal trabaja de lunes a sábado de las 9:00 a 20:30 hrs:

- Personal de Mostrador

**Nota:** Solo el personal de mostrador de la farmacia 1 trabaja de lunes a sábado de las 9:00 a las 21:00 hrs.

La organización tiene perfiles específicos al reclutar personal. El personal médico, recursos humanos y contador deben contar con su título universitario. El personal de mostrador, capturista y repartidor deben tener estudios de secundaria o preparatoria. No se les pide que tengan conocimientos de computación, por lo que, se tendrá que capacitar al personal que use el sistema.

Al tratarse de un servicio farmacéutico, la organización ha tenido quejas de falta de medicamentos. Esto a causa del retraso de la información. Hasta el momento la filial que más problemas tiene es la Farmacia Morelos 3, como es de reciente creación tiene pocas ventas y consultas médicas.

La empresa está en pleno crecimiento, cuenta con más de una sucursal. Cada sucursal depende del abastecimiento oportuno de la farmacia matriz. El problema a resolver es el deficiente manejo de sus inventarios.

Cuenta con una base de datos, sin embargo, ésta se encuentra centralizada en su centro de operaciones. Sus reportes de altas y bajas, en las sucursales, las realizan de manera manual, por lo cual, debe ser hasta el final del día la entrega de sus reportes para actualizar la base de datos<sup>2</sup>.

---

<sup>2</sup> Figura 10 Diagrama de Flujo pág 55 de este trabajo



La organización ha concebido la idea de poner un sistema que controle sus reportes de altas y bajas. Sin embargo, aun sigue operando de forma no automatizada. Una de las causas es la falta de recursos tecnológicos y económicos.

Aunque ya cuenta con una base de datos, ésta se encuentra totalmente aislada, sin la posibilidad de comunicación directa entre sus farmacias.

Productos. La empresa cuenta con distintos productos, los cataloga en medicamentos, en material de curación y otros. Los medicamentos cuentan con una fecha de caducidad, por lo que siempre se vende primero lo que se compra. Los productos se compran a diferentes proveedores y, se venden a diferentes clientes.

Consulta de productos. El sistema deberá permitir a los usuarios buscar productos por catálogos, función y sustancia activa, para comprobar que la organización cuenta con ese producto para venderlo, y si no, levantar una orden de compra. Las consultas las pueden hacer todo tipo de usuarios.

Al permitir que los clientes tengan la opción de consultar la información a través del sistema, la farmacia espera dar un nuevo servicio a sus clientes. Tomando en cuenta la necesidad de tener la información exacta por parte de los clientes.

Venta/Compra. Es parte fundamental que se lleve un estricto control de las ventas y las compras de los productos. Este sistema deberá registrar cuando se ha realizado una orden de venta y las existencias sean bajas. Todo esto para advertir una baja de productos y garantizar que siempre estén abastecidas las farmacias.

#### **2.4. Beneficios del proyecto**

El beneficio será la automatización de una parte de la operación de la organización. Al integrar todas las sucursales con el centro de operaciones, la actualización de la base de datos será inmediata.



Todas las filiales tendrán acceso al sistema, podrán usar los servicios de búsqueda y reportes. Así los reportes de cierre de caja serán más rápidos y podrán ser consultados en el momento en que se requiera.

Al tener la información puntual de cada dependencia, el centro de operaciones podrá redirigir los envíos de cada producto, de acuerdo a las necesidades de la sucursal. El beneficio será el abastecimiento acertado a cada farmacia, dando a los clientes la confianza de que siempre encontrarán el producto que necesitan en cada sucursal.

### Beneficios

	Ventaja	Beneficio
Abastecimiento de productos	Al tener un control automatizado y actualizado las existencias en cada farmacia, estas estarán bien abastecidas con el presupuesto aprobado.	Se retendrá a más clientes por la calidad y cantidad de productos. Se captarán más clientes al tener una reputación favorable de existencias.
Manejo de inventarios	Mantener un control riguroso de las existencias reales de cada farmacia.	La empresa podrá abastecer de forma adecuada y oportuna a todas sus filiales evitando la escases de productos.
Administración de ventas de producto	Llevar un registro de ventas automatizado, que refleje los movimientos en los inventarios de cada farmacia.	Facilitar la recopilación de información de ventas de cada farmacia.
Gestión de Reportes de proceso	Generar reportes actualizados de cada farmacia para una toma de decisiones más adecuado.	Obtener reportes confiables y actualizados que permitan un mejor control de todos los procesos llevados por la empresa.

**Tabla 3**

El tener presencia en Internet buscará captar más clientes, así como dar a conocer el nombre de la organización como una empresa creciente. A través de la consulta de las existencias de los medicamentos que la institución maneja, por medio de Internet, se busca dar a los clientes el servicio de “antes de ir a comprar, revisa en que farmacia se encuentra”. Todo con la idea de capturar nuevos clientes y además retenerlos.



La solución propuesta es para la organización grupo Morelos, ya que le ayudará a resolver problemas de automatización internas. Así como, tener una infraestructura tecnología que pueda organizar y mantener mejor la información que genera día con día.

## **2.5. Antecedentes del proyecto**

Hasta ahora el único antecedente a este proyecto en la organización grupo Morelos es la creación de una base de datos para el manejo de las existencias de la farmacia. En sí ninguna sucursal cuenta con tecnologías computacionales, la única que si cuenta, como ya se ha mencionado, es el centro de operaciones.

Sólo se tiene el manejo manual, repetitivo, de las operaciones con respecto a las altas y bajas de sus existencias. Todos los registros se llevan a cabo en una libreta para luego ser ingresados en una base de datos.

El primer problema a resolver será las carencias tecnológicas que tiene la organización, en materia computacional, en cada una de sus sucursales. Después será el observar los procesos repetitivos que se realizan en la farmacia para su automatización. Y por último, resolver la comunicación precaria que tiene el centro de operaciones con sus filiales, es decir, el sistema deberá conectar a todas las sucursales.

Hay que mencionar que para cada problema hay una solución, de que sea mejor o peor dependerá de la identificación correcta de las necesidades del problema. En el mercado existe software que pueden solucionar algunos de estos problemas, una de las limitantes es el idioma, podemos encontrar software de habla inglesa, alemana, francesa, etc.

Estas soluciones pueden resolver algunos problemas pero dejar de lado otras, y otras ofrecen soluciones extras que no se requieren.

Un ejemplo es el sistema de venta anticipada con pantalla táctil AK 0165, sin duda un sistema como éste logrará resolver muchos problemas, sin embargo, el costo de la



tecnología de hardware y software que maneja es sin duda mucho para nuestro problema. También existen las terminales portátiles que nos permitirán solucionar algunos problemas cómo manejar inventarios, realizar reportes, etc. Sin embargo, el hardware es especial, así como la programación de cada Terminal.

El producto que se creará tendrá la ventaja que sólo resolverá lo necesario, ni más ni menos, evitando costos por soluciones que no se necesitan o soluciones faltantes. La tecnología del sistema se adecuará a los recursos de la organización. En un mundo ideal lo mejor sería usar tecnología de punta, sin embargo, muchas instituciones no pueden costear esos productos.

## 2.6. Objetivo

El objetivo del proyecto es hacer una aplicación para detectar los movimientos de la mercancía, para el abastecimiento oportuno de las filiales de la empresa farmacéutica grupo Morelos y dar información a los clientes *online*.

Algunos beneficios y características de la aplicación serán:

- Diseñar un esquema de seguridad de acuerdo con los niveles de usuario. Por cada tipo de operación habrá un nivel de acceso, para garantizar la integridad de la información.
- Hacerla flexible, si hay factores externos o internos que afecten sus reglas de negocio, el sistema será capaz de reconfigurarse sin afectar los demás procesos.
- Permitir el acceso al público en general, a través de la consulta de catálogos.

## 2.7. Alcance

La interfaz del sistema será sencilla y amigable, siendo compatible con los navegadores más usados, mozilla e Internet Explorer, de la red. La interfaz dedicada al público en general contara con “*banners*” publicitarios. Esta aplicación se construirá para que pueda usarse con servidores estándares de Web y de aplicación.



Funcionará con tecnología Java, lo que nos permitirá separar a los usuarios de los datos. Permitiendo lo más posible la integridad de la información. Con la ayuda de Struts pondremos una capa intermedia que nos permita separar los componentes de la vista que se presentara a los usuarios.

La seguridad se basará en la utilización de usuarios y contraseñas, estableciendo niveles de acceso a cada actor. El sistema logrará conectarse a una base de datos, para obtener sus servicios de consulta, de altas y modificaciones. Las páginas Web serán dinámicas, las consultas hechas a la BD se presentarán en formato HTML.

## 2.8. Entregables

Al concluir el proyecto se entregarán al cliente:

- Aplicación Web para “Monitoreo del medicamento de una farmacia”.
- Guía de instalación y configuración.
- Manual para la utilización del sistema.

Configuración de la zona de “*banners*” publicitarios.

## Desventajas y Beneficios

Desventajas	Beneficio
<ul style="list-style-type: none"><li>▪ Existe la posibilidad que el proyecto rebase, en materia de recursos, a la organización.</li><li>▪ Existe un riesgo potencial de seguridad, ya que es una aplicación en red, por lo tanto podría ser hackeada.</li></ul>	<ul style="list-style-type: none"><li>▪ Tener información rápida y confiable de las sucursales.</li><li>▪ Tomar decisiones más certeras al surtir de medicamentos a las filiales.</li><li>▪ Mejorar la operación de cada integrante de cada farmacia.</li><li>▪ Dará un mejor servicio a sus clientes.</li><li>▪ La matriz tendrá comunicación constante con sus filiales.</li><li>▪ Se aumentara las posibilidades del negocio.</li></ul>

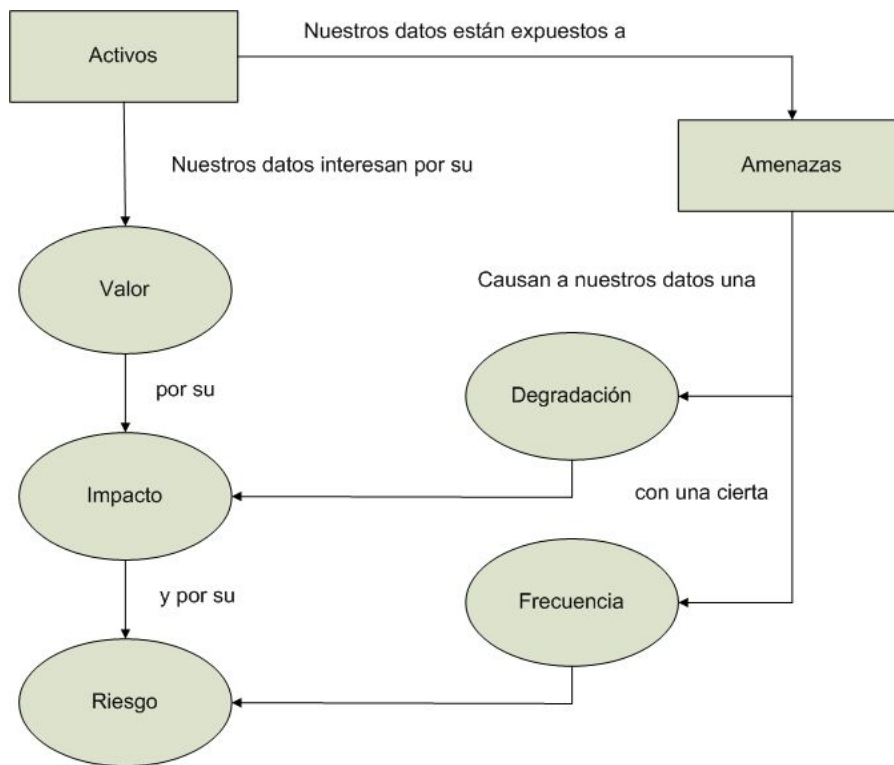
**Tabla 4**

## Riesgo potencial de seguridad



Es un hecho que para la mayoría de las instituciones y empresas lo más importante es mantener integra la información que día a día se produce, en este caso su activo más importante son los datos, la información.

Los activos de la empresa deben ser resguardados de cualquier tipo de amenaza, al ser un Sistema en Web los riesgos pueden ser mayores que en una red local. El siguiente diagrama muestra el riesgo que puede tener los activos.



**Figura 2.2 Riesgo potencial de Seguridad**

Las vulnerabilidades de un Sistema en Web radican en aquellas amenazas que afectan directamente a la plataforma, es decir vulnerabilidades en el Sistema Operativo, en el servidor Web, en el manejador de base de datos y en errores de programación en un sitio Web.

También existen los ataques contra aplicaciones, como la inyección SQL o el secuestro de sesiones. Es importante estar concientes de que existe un riesgo potencial en contra de nuestros activos, pero la configuración adecuada de nuestra plataforma y el cuidado necesario al programar las aplicaciones disminuirán estos riesgos.



## CAPÍTULO III

### PROPUESTA Y SOLUCIÓN

#### 3.1. Especificación de requerimientos

##### *Sistema de monitoreo de medicamento para una farmacia*

El sistema “monitoreo de medicamento” será diseñado para funcionar en una arquitectura de red tipo cliente/servidor. Usando la metodología de desarrollo de “tres capas” se buscará obtener la solución a este problema lo mejor posible.

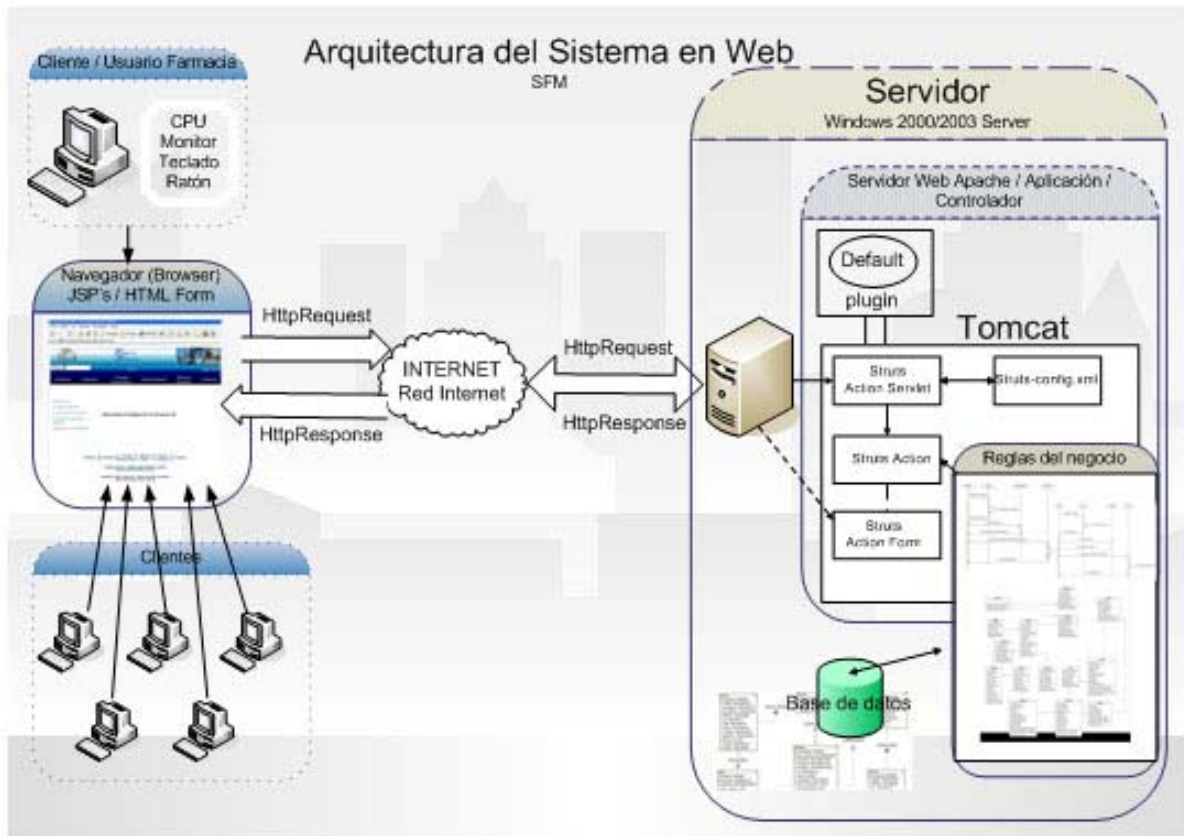
El sistema se dividirá en tres secciones en su desarrollo, la primera será exclusivamente para el usuario. Tratando de encontrar el ambiente más propicio para el trabajo, dentro del sistema, de cada uno de nuestros clientes.

El sistema buscará la personalización en cada una de sus interfaces, basándose en la información recabada, en su base de datos, de cada uno de los usuarios. La interfaz que el sistema empleará para crear sus pantallas interactivas será el browser.

La segunda sección se enfocará en el tratamiento de los datos. El sistema manejará la base de datos por medio de una conexión segura, permitiendo tener una alta integridad de la información a través del tiempo. Para ello utilizará un manejador de base de datos que le permita adecuar conceptos de relación, seguridad de datos, manipulación de datos y accesos concurrentes.

La tercera parte de desarrollo consistirá en definir las reglas del negocio de la empresa, el sistema será capaz de acceder a los datos a través de los módulos programados. Estos módulos serán los que recibirán las peticiones del usuario, aplicarán las reglas de negocio, accederán a los datos, para entregar el resultado al cliente.





**Figura 3.1** Arquitectura del Sistema SFM

El sistema dará a la organización la posibilidad de crear nuevos productos, modificarlos o darlos de baja por caducidad. La creación de reportes generados por la venta de sus productos estará a cargo del sistema. El sistema al encontrar una baja existencial de algún producto en cualquiera de las filiales alertará al operador. La aplicación permitirá la consulta dinámicamente de todos los productos.

### 3.2. Especificaciones funcionales

La necesidad de la organización es controlar de mejor manera los movimientos de sus productos en todas sus filiales. La empresa piensa que al tener la información correcta del estado de sus productos de manera rápida, está podrá surtir de manera mas eficiente a todas sus sucursales.

La solución que proponemos es, realizar un buen diseño de la base de datos e incluir a todas las filiales mediante un sistema computacional. Eliminar la barrera física del



centro de operaciones con cada sucursal mediante una red. Programar módulos que sean lo bastante flexibles y robustos para el manejo de los reportes.

Uno de los objetivos principales del proyecto será diseñar un diagrama entidad/relación que nos ayude a generar una base de datos que permita evitar redundancia de información y que proporcione alta integridad a la información.

Analizar y diseñar una posible solución al problema mediante herramientas UML, basándonos en los casos de uso que genere el análisis. Crear un sistema orientado a objetos para trabajar en arquitectura cliente/servidor con la finalidad de conectar a las filiales con tecnología Java.

### **3.3. Actores definidos para el sistema**

- Operadores de venta.

Es el personal que actualmente se encarga de la venta de medicamentos al público en general, hay al menos un operador en cada filial. Corresponde al personal de mostrador.

1. Es dado de alta en el sistema por el Supervisor de Personal.
2. Su nivel de acceso es: Segundo Nivel
3. Seguridad y privacidad con respecto a otros usuarios del sistema.

- Operadores de compra.

Es el personal que se encargará de registrar las compras de los diferentes productos utilizados por la organización.

1. Es dado de alta en el sistema por el Supervisor General.
2. Su nivel de acceso es: Tercer Nivel
3. Seguridad y privacidad con respecto a otros usuarios del sistema.

- Público en general.

Es el cliente que a través de Internet o de manera directa ha entrado a conocer los productos que la empresa ofrece.



1. Es dado de alta en el sistema por el sistema.
2. Su nivel de acceso es: Primer Nivel.
3. Seguridad y privacidad con respecto a otros usuarios del sistema.

- Operador de reportes.

Es el personal asignado a entregar reportes mensuales, semanales, etc. de las actividades ocurridas. Corresponde al personal Capturista.

1. Es dado de alta en el sistema por el Supervisor de Personal.
2. Su nivel de acceso es: Tercer Nivel.
3. Seguridad y privacidad con respecto a otros usuarios del sistema.

- Supervisor de Personal.

Es el personal asignado al reclutamiento de empleados de la organización, se le asigna valorar a los elementos humanos que entraran a las filas de la empresa. Corresponde al personal de Recursos Humanos.

1. Es dado de alta en el sistema por el Supervisor General.
2. Su nivel de acceso es: Cuarto Nivel.
3. Seguridad y privacidad con respecto a otros usuarios del sistema.

- Supervisor general.

Es la persona que organizara los eventos de abastecimiento de cada sucursal, el abastecimiento de la empresa, de dar de baja productos caducos, de generar ventas mayoritarias. Corresponde al personal Director General.

1. Es dado de alta por el administrador del sistema.
2. Su nivel de acceso es: Quinto Nivel.
3. Seguridad y privacidad con respecto a otros usuarios del sistema.

Al analizar y desarrollar este sistema nos convertimos en los administradores del mismo, ya que hay que dar soporte y mantenimiento del mismo en los primeros meses que sea implantado el sistema.



### 3.4. Niveles de acceso.

El sistema se basa en la seguridad de nombre “usuario/contraseña”, además se utilizan niveles de seguridad para tener acceso a comandos del sistema. Dinámicamente se desplegarán los menús conforme el nivel de acceso.

Primer Nivel (Es el nivel básico).

- Acceso a consultas de productos. Catalogo, por función, sustancia activa, por promoción y avanzada.
- Acceso a consultas de filiales.

Segundo Nivel.

- Acceso a consultas de productos.
- Acceso a los comandos de ventas.
- Acceso a reportes.

Tercer Nivel.

- Acceso a consultas de productos.
- Acceso a los comandos de compras.
- Acceso a los comandos de crear producto, catalogo y artículo.
- Acceso a los comandos de modificar producto, catálogo y artículo.

Cuarto Nivel.

- Acceso a consulta de empleados.
- Acceso a consulta de productos.
- Acceso a los comandos de crear empleado y puesto.
- Acceso a los comandos de modificar empleado y puesto.

Quinto Nivel (Es el nivel más alto).

- Acceso a consulta de productos.
- Acceso a consulta de empleados.
- Acceso a los comandos de crear empleado, puesto, filial, almacén.
- Acceso a los comandos modificar empleado, puesto, filial, almacén.



- Acceso a reportes de compras y ventas.
- Acceso a los comandos de ventas.
- Acceso a los comandos de compras.
- Acceso a los comandos de eliminar.

### **3.5. Casos de Uso (CU) por área funcional**

A continuación se definen las tareas que debe acometer el sistema. Se presentan ordenados tomando en cuenta la función que deben realizar.

#### **A. Control de acceso de usuarios.**

- CU A01 Crear cuenta.
- CU A02 Registrar de usuario.
- CU A03 Solicitar nueva contraseña.
- CU A04 Recordar contraseña.
- CU A05 Consultar perfil.
- CU A06 Actualizar perfil.
- CU A07 Eliminar una cuenta.
- CU A08 Imprimir cuenta.
- CU A09 Validar nivel de acceso.
- CU A10 Iniciar sesión.

#### **B. Control de producto.**

- CU B01 Crear tipo de producto.
- CU B02 Crear catálogo.
- CU B03 Crear artículo.
- CU B04 Actualizar un producto.
- CU B05 Consultar catálogo de artículos.
- CU B06 Actualizar catálogo.
- CU B07 Consultar producto.
- CU B08 Eliminar un catálogo.
- CU B09 Eliminar un producto.



CU B10 Actualizar artículo.

CU B11 Consultar artículo.

CU B12 Eliminar artículo.

C. Control de filiales.

CU C01 Crear Farmacia.

CU C02 Actualizar Farmacia.

CU C03 Consultar Farmacia.

CU C04 Crear almacén.

CU C05 Eliminar Farmacia.

CU C06 Crear puesto.

CU C07 Actualizar puesto.

CU C08 Eliminar puesto.

D. Control de ventas.

CU D01 Crear venta.

CU D02 Registrar orden.

CU D03 Registrar venta.

CU D04 Consultar venta.

CU D05 Cancelar orden.

E. Control de compra.

CU E01 Crear compra.

CU E02 Registrar orden.

CU E03 Registrar compra.

CU E04 Consultar compra.

CU E05 Cancelar orden.

F. Control de monitoreo.

CU F01 Crear monitoreo.

CU F02 Consultar almacén local.

CU F03 Consultar inventario general.

CU F04 Crear mensaje de alerta.

CU F05 Crear reporte de producto.

### 3.5.1 Diagramas de caso de uso

A continuación presento los diagramas de caso de usos, aquí se muestra como se relación un actor definido en el sistema con cada caso de uso.

#### Diagrama de caso de uso para el actor Operador de venta

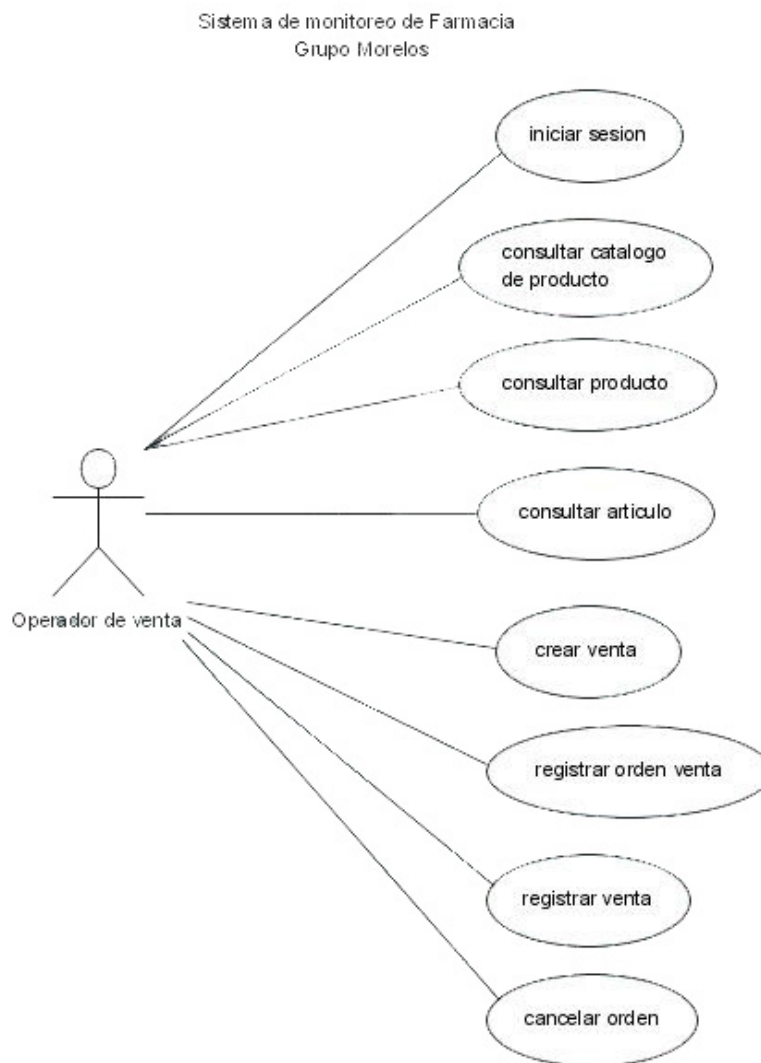
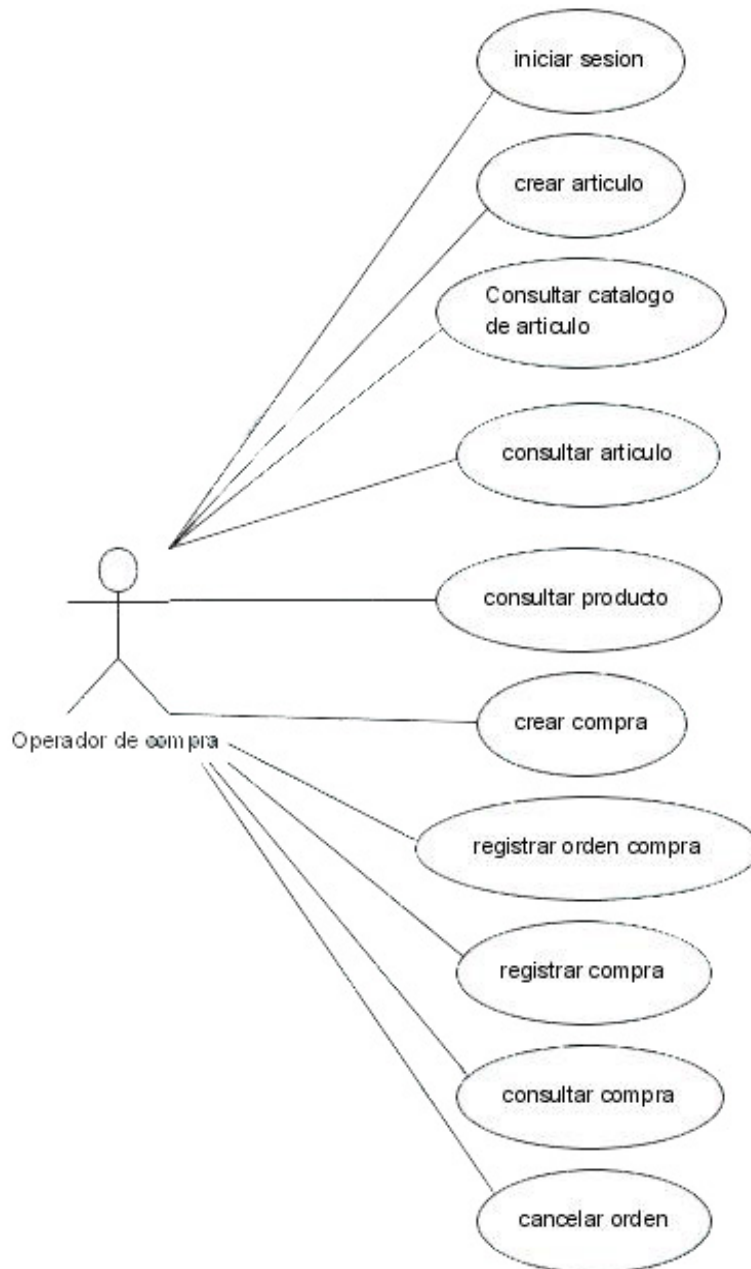


Figura 3.2 Operador de venta

### Diagrama de caso de uso para el actor Operador de compra



**Figura 3.3 Operador de compra**



### Diagrama de caso de uso para el actor Público general

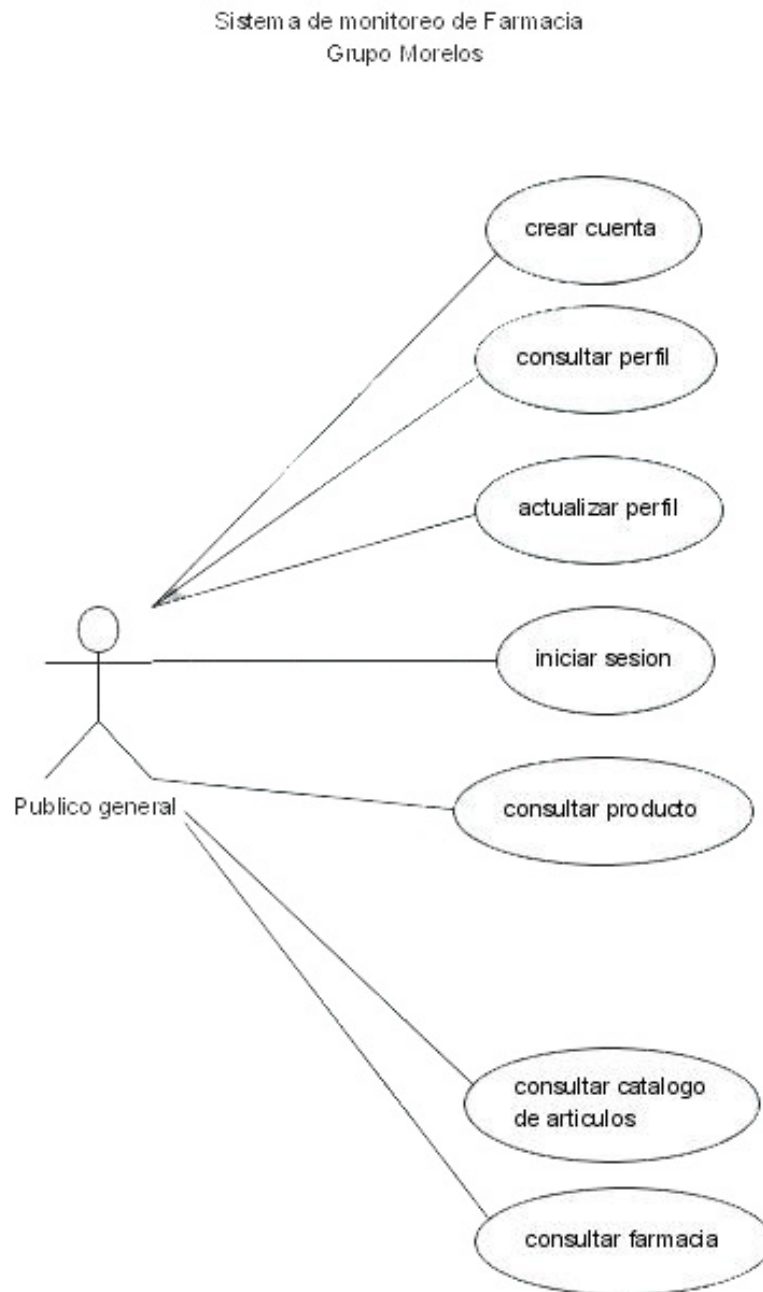


Figura 3.4 Público general



### Diagrama de caso de usos para el actor Operador de reportes

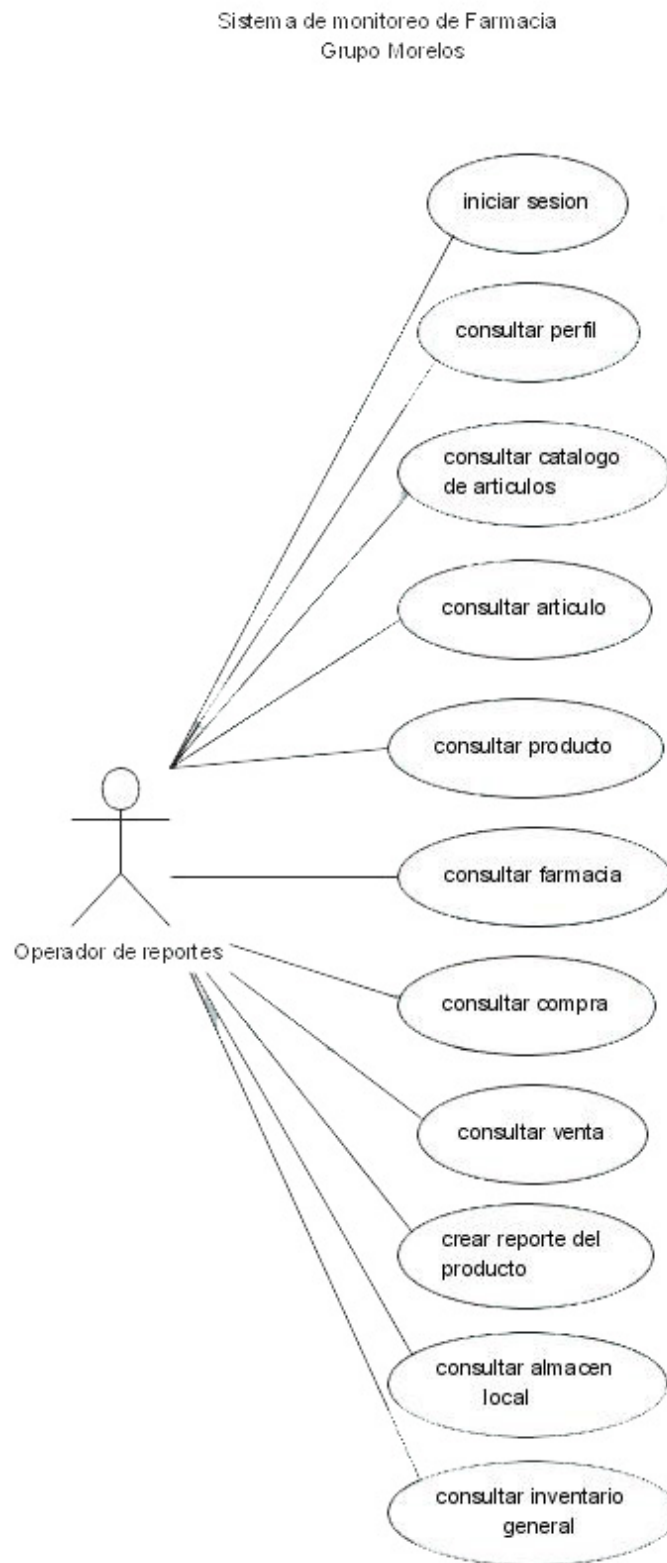


Figura 3.5 Operador de reportes



### Diagrama de caso de usos para el actor Supervisor de personal

Sistema de monitoreo de Farmacia  
Grupo Morelos

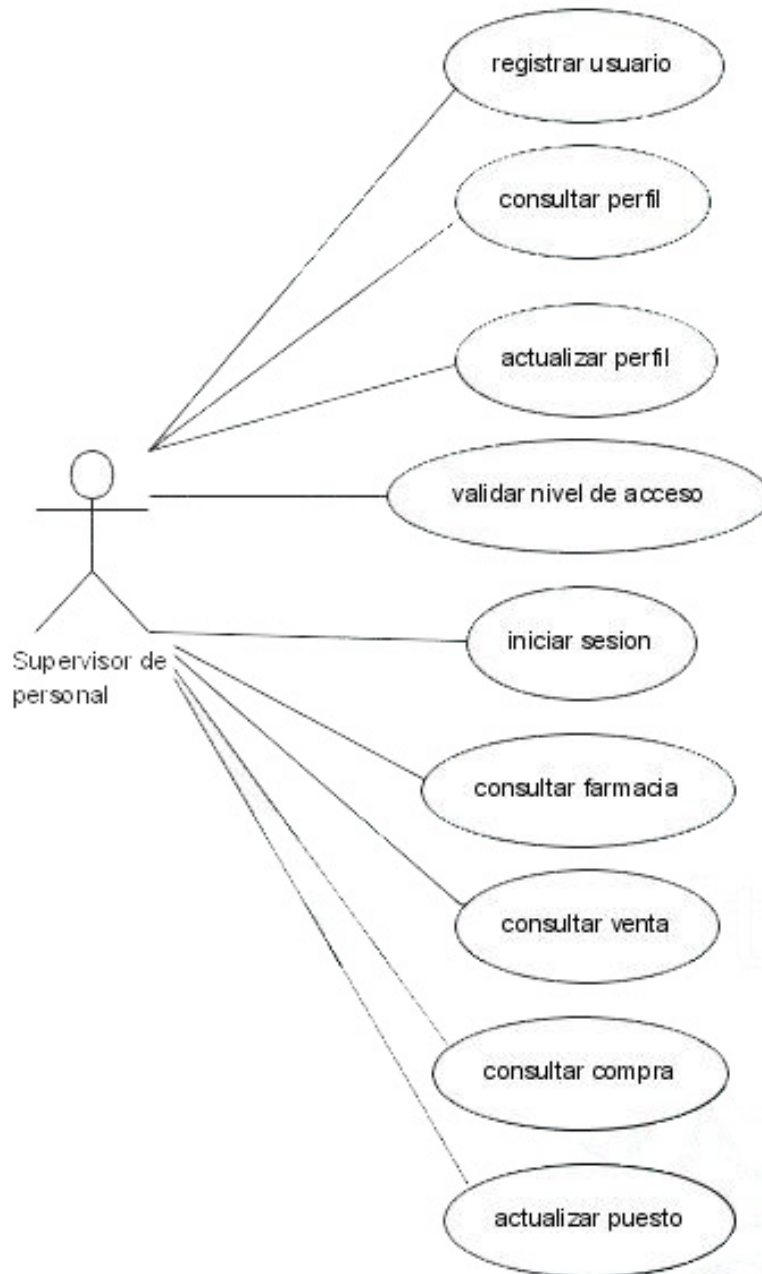
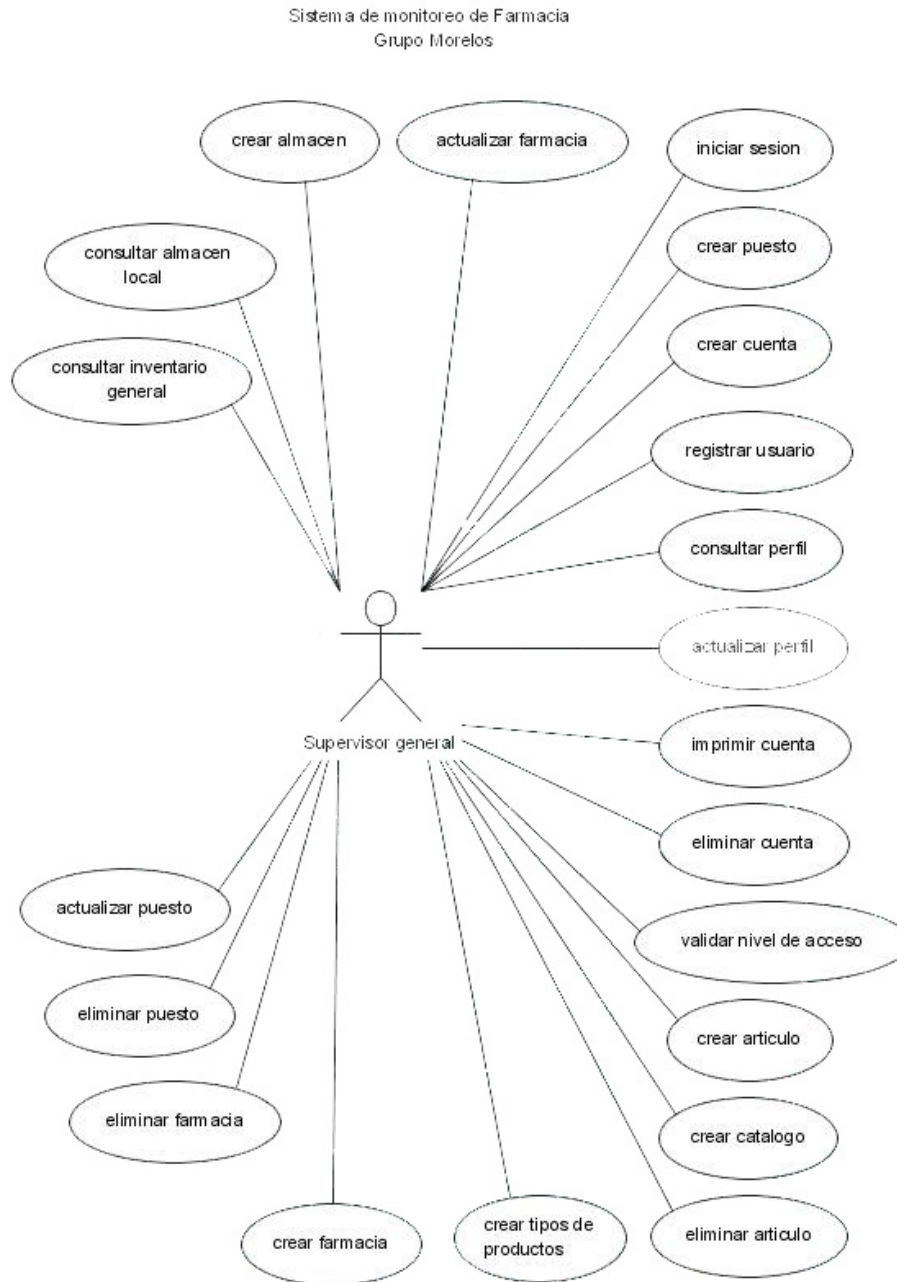


Figura 3.6 Supervisor de personal

## Diagrama de caso de usos del Supervisor general



**Figura 3.7 Supervisor general**

Los casos de uso se han realizado a través del cuestionamiento al personal de las diferentes áreas que conforman a la organización. Más adelante, en el capítulo cuatro “diseño”, se presentara la documentación de cada caso de uso, las descripciones de los casos de uso.



Mientras avanza el ciclo de vida del sistema podrán surgir nuevos casos, por ello se debe tratar de llegar a un acuerdo con la organización para los cambios que deban de solucionarse.

### **3.6. Requerimientos no funcionales**

- La usabilidad del sistema.

Se tiene que muchos de los procesos son repetitivos y realizarlos paso a paso representa la primera dificultad de la eficacia. La experiencia nos dicta que, un empleado, en alguna ocasión de las miles de veces que registra el nombre de un mismo artículo lo escribirá en alómenos una ocasión diferente a los demás, por ello existen varias cajas con el texto predefinido.

La mayoría de la gente esta familiarizada con un ambiente Windows, por lo que, la interfaz del usuario será lo mas posible parecido a una pantalla de ventanas. En este caso también se asume que los usuarios tienen un alto grado de familiaridad con los formatos de las páginas Web.

Se utilizan colores agradables a la vista y que no sea cansina para el usuario. Se mantiene un estilo de letra, definiendo el color y tamaño de cada una de ellas dependiendo de su accionar. Las imágenes serán reducidas de peso (Kb) guardando la mejor calidad posible de la imagen, esto para que los usuarios naveguen más rápido dentro del sitio.

- Seguridad en el sistema.

El acceso al sistema será controlado a través de la utilización y designación de claves. Estas claves están conformadas por el nombre de usuario y contraseña. Se identifican dos tipos de usuarios, en materia de seguridad.



El primero de ellos son los administradores y siendo los segundos los usuarios normales. Un usuario normal puede registrarse en el sistema, escogiendo su nombre de usuario y contraseña. Mientras un administrador será dado de alta por el supervisor en general, asignándole un nombre de usuario y contraseña.

Las contraseñas de los usuarios normales deberán de tener de 3 a 10 caracteres de longitud, para su nombre de usuario no deberá exceder de los 10 caracteres. Las contraseñas de los administradores serán de 8 a 10 caracteres de longitud, para su nombre de usuario no deberá exceder de los 10 caracteres.

Al crear una cuenta, el usuario normal, el sistema le concede un nivel de acceso. El Primer nivel de acceso, limita la cuenta a solo realizar consultas de los artículos existentes que trabaja la empresa.

Al tener un empleado con necesidad de trabajar con el sistema, el supervisor general creara una cuenta y le asignara un nivel de acceso de acuerdo a las actividades que tenga que realizar dentro del sistema.

- Disponibilidad del sistema.

Con las medidas de seguridad, en el sistema, se busca garantizar lo mas posible que los datos estén accesibles en el momento en el que se necesiten. El sistema trabaja en tiempo real lo que hace que la disponibilidad sea un tema de crítica importancia.

Para lograr una disponibilidad aceptable del sistema se deben tener planes de soporte y actualización. Al entregar la aplicación se acuerda entregar documentos que sean de utilidad para el soporte técnico del producto. También se ofrece un acuerdo de soporte del sistema, tratando que nuestra habilidad de proveer soporte técnico sea eficiente y a buen precio.



Las características del sistema están enfocadas a cumplir en un 99% la disponibilidad del sistema al mes, se contempla, que será forzoso detener el sistema para su mantenimiento. Esto es para realizar respaldos de la base de datos, corregir errores, monitorear la aplicación y actualizar el sistema.

Se tiene que tomar en cuenta que las amenazas que potencialmente podrán afectar la disponibilidad del sistema son las amenazas de entorno como fuego, terremoto, inundación, fallo eléctrico y temperatura; accidental como: pérdida de conectividad, error humano, pérdida de personal y fallo de hardware; y deliberadas como el sabotaje de un intruso.

- Mantenimiento y actualización.

El mantenimiento del sistema se realizará de manera efectiva, procurando realizar los cambios necesarios al producto en el tiempo que se requiera. Los cambios en un sistema deben ser para rectificar el funcionamiento, ya que un buen funcionamiento de la aplicación asegura uno de los objetivos fundamentales: retener y atraer a nuevos clientes.

Cada cambio se fundamentará y se documentará detalladamente en nuestro diseño y en la implementación, con la intención de enriquecer el historial del sistema y de la empresa misma. No hay que olvidar que en el diseño del sistema se han considerado los posibles errores que puedan surgir.

La actualización de la aplicación dependerá de las nuevas necesidades que surjan en la interacción de la empresa con sus clientes, y de los nuevos servicios que el grupo Morelos pueda tener en mente. Estas actualizaciones se harán lo mas rápido y eficaz posible tratando de mantener un bajo costo.



### 3.6. Requerimientos del medio ambiente

- Hardware.

El sistema para su funcionamiento necesita que la empresa grupo Morelos tenga una infraestructura computacional en cada una de sus filiales. Además, esta infraestructura debe proporcionar una conexión en red.

El hardware necesario es el siguiente:

1 máquina servidor:

PC, Pentium 4, 2.60GHz de velocidad con 512 de memoria RAM, tarjeta de red 3Com etherlink 100/10 Mb. Con 2 discos duro de 40 Gbytes. Unidad de Floppy, Unidad de CD-writer room.

Un monitor de 17 pulgadas.

Un teclado.

Un “mouse”.

Un no break, que permita mantener 30 minutos al servidor.

Una conexión que le permita entrar a red.

2 máquinas clientes:

PC, Pentium 2 a 350Hz de velocidad con 128 Mb de memoria RAM, tarjeta de red 3com etherlink 100/10 Mb. Con un disco duro de 4 GBytes. Unidad de floppy, Unidad de CD room.

Un monitor de 15 pulgadas.

Un teclado.

Un “mouse”

Un regulador contra altas y bajas de corrientes.

Una conexión que le permita entrar a red.





- Software.

El sistema para su funcionamiento necesita que la empresa grupo Morelos y que los clientes en Internet cuenten con el siguiente software:

En la máquina servidor:

Requiere un sistema operativo

Servidor Web Apache 2.0.

Intérprete de Java.

Kit de desarrollo de Java (JSDK 1.4.2)

Jakarta Tomcat 4.X como mínimo.

Instalar el DBMS DB2 IBM.

Instalar y configurar drivers para acceso a BD DB2 UDB versión 8.X

Un protocolo de red (TCP/IP)

En la máquina cliente:

Requiere de un Sistema Operativo (Windows, Linux)

Browser (IE, Netscape, Mozilla, etc.)

JVM (Java Virtual Machine)

Protocolo de red (TCP/IP)

- Algún API.

El lenguaje Java tiene la gran ventaja de tener proyectos en todo el mundo y, al ser de licencia open source permite reutilizar muchas aplicaciones. En el sistema se utilizan las API's de Struts e IBATIS.

Jakarta Struts se basa en el patrón del Modelo Vista Controlador, este componente nos ayudará a reducir el tiempo de programación en la lógica del negocio. Lo usaremos para representar información a nuestros usuarios.



Los componentes de control de Struts serán los encargados de coordinar las actividades de la aplicación. La recepción de datos del usuario, verificaciones de forma y la selección de un componente están organizados por Struts.

IBATIS es un conjunto de componentes que nos permitirá controlar las persistencias, manejar el concepto de sesión es fundamental para la seguridad del sistema. Este API nos permitirá ahorrar programación para solucionar ese problema.

- Acceso a datos: importación/exportación de datos.

El diseño de la base de datos esta diseñado para evitar la redundancia de información y la integridad de la misma. El manejador de la base de datos estará encargado de verificar una gran parte de la integridad de los datos mediante las relaciones y las reglas de negocio definidas en ellas.

El acceso a los datos esta a cargo de los componentes de la aplicación, la petición la hará el usuario a través de los formularios de las interfaces, estas a su vez serán recogidas por los componentes adecuados validando la información.

Otro componente pedirá la información al manejador de la base de datos para mostrar el resultado de la consulta en una página dinámica.

El registro de los datos se hará a través de la aplicación, una petición de registro se establecerá con un formulario pasando por un componente que recogerá los datos, se validarán los datos y se depositarán en la base de datos.

El sistema deberá almacenar toda la información en una base de datos SQL estándar, esta base de datos solo será modificada por el sistema y en caso necesario por el administrador de la base de datos.



- Acceso a base de datos.

La base de datos solo permite abrir una sesión controlada a través del sistema, el sistema utiliza niveles de acceso que restringen los comandos en nuestra base de datos.

Al intentar acceder al sistema SFM la primera restricción es puesta por la plataforma en la que corre el sistema, obligando al usuario a ingresar un username y password válido. Consecutivamente se obliga a los usuarios a ingresar su nombre de usuario y su contraseña.

Dependiendo del nivel de acceso que le haya proporcionado el supervisor general tendrá acceso al sistema. Una vez iniciado sesión y con los permisos adecuados tendrá acceso a ciertos comandos de modificación a la base de datos.

### **3.7. Tiempos de desarrollo, diagrama de Gantt.**

Con ayuda de un diagrama de Gantt podemos representar los tiempos de análisis y desarrollo del presente proyecto. En el eje horizontal se tiene la escala de tiempo y en el eje vertical se tienen las actividades realizadas en esa escala de tiempo.





## CAPÍTULO IV

### DISEÑO

#### 4.1. Documentación de Caso de Usos.

La documentación que presentamos en esta sección son los casos de uso. Los casos de uso describirán el comportamiento del sistema desde el punto de vista del usuario. En cada caso verá el flujo natural de las acciones que se realizan en el sistema.

También se pretende documentar los posibles fallos que pudieran existir y que ya se han tomado en cuenta para que el sistema funcione adecuadamente. Apoyados en esta herramienta UML se conseguirá planificar las iteraciones del sistema.

Cada caso de uso cuenta con un nombre ya definido, tratando de representar de forma general la función que realiza. El nombre está relacionado con las iniciales CU (caso de uso); con una letra del alfabeto, que indica el área funcional al que pertenece; y un número que lo clasifica.

Cada documento cuenta con su fecha de creación, su autor y a que proyecto está relacionado. Cuenta con un flujo principal y uno o varios flujos alternos. También hace notar que actor es quien realiza la tarea y si algún evento del flujo manda a llamar a otro caso de uso.

Debe considerarse que un actor es cualquier cosa ajena al sistema que se desarrolla y que interactúa con el mismo. Por lo que puede ser una persona, un dispositivo de hardware o un componente del mismo sistema.



Fecha de creación: 2005. Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A01 Crear cuenta</b>	Inicia: A reserva del Actor Supervisor general, supervisor personal. Termina: termina crear cuenta.
Descripción: Permite la creación de una cuenta para un empleado de la organización.	
Actores: Supervisor General, supervisor personal.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	Pide datos personales a su empleado		a	Tipo 1: Es un operador de venta.		
			b	Tipo 2: Es un operador de compra.		
			c	Tipo 3: Es un supervisor personal.		
			d	Tipo 4: Es un operador de reportes.		
2	Inicia sesión en el sistema.	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	a	El supervisor general olvido su contraseña. 1. Se pondrá en contacto con el administrador del sistema.
3		El sistema despliega el menú del sistema SFM.				
4	Selecciona del menú el comando crear.	El sistema despliega las opciones de crear				
5	El actor selecciona crear empleado	El sistema despliega la pantalla con el formulario de alta empleado.				
6	El actor llena el formulario					
7	El actor acepta el formulario	El sistema valida los datos	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos.		
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos.		
8		El sistema le asigna un nivel de acceso, a la nueva cuenta, de acuerdo al puesto del nuevo empleado.				
9		El sistema recoge los datos y los vacía en la BD				
10		El sistema manda un mensaje de aceptación.				
11	El actor recibe una notificación y un reporte del nuevo empleado creado.					
12	Termina crear cuenta					

Precondiciones: El Supervisor deberá estar en sesión, la empresa tendrá un nuevo empleado.

Poscondiciones: Se tendrá un empleado dado de alta en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.



Documento: CU A02 Registrar usuario.

Fecha de creación: 2005. Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A02 Registrar usuario.</b>	Inicia: A reserva del cliente. Termina: termina registrar usuario.
Descripción: Permite la creación de una cuenta para un cliente.	
Actores: público general.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor elige registrarse	El sistema despliega formulario alta				
2	El actor llena el formulario					
3	El actor acepta el formulario	El sistema valida los datos	a	Existe un campo vacío necesario 3. Se pide ingrese los datos del campo 4. Se muestra de nuevo el formulario con los datos validos.		
			b	En un campo los tipos de datos no son los indicados. 3. Se pide ingrese datos validos. 4. Se muestra de nuevo formulario con los datos validos.		
		El sistema consulta a la BD.	c	El nombre de usuario ya existe 1. Se genera una contraseña recomendada y se muestra al usuario. 2. Se pide ingrese una nueva contraseña		
			d	La contraseña o nombre de usuario no es valida. 1. Se pide que deban coincidir los dos campos de la contraseña. 2. Se recomienda no usar caracteres especiales. 3. Se muestra el formulario con los datos validos.		
4		El sistema le asigna un nivel de acceso Primer nivel.				
5		El sistema recoge los datos y los vacía en la BD.				
6		El sistema manda un mensaje de aceptación				
7	El actor recibe una notificación con sus datos de acceso al sistema.					
8	Termina registrar usuario.					

Precondiciones: El cliente deberá estar dispuesto a darse de alta en el sistema.

Poscondiciones: Se tendrá un cliente dado de alta en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU A03 Solicitar nueva contraseña.



Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A03 Solicitar nueva contraseña.</b>	Inicia: A reserva del Actor. Termina: termina solicitar contraseña.
Descripción: Permite la creación de una nueva contraseña para el usuario.	
Actores: Supervisor General, cliente, operador venta, operador reporte, operador compra, supervisor personal.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2	El actor elige cambiar contraseña	El sistema despliega formulario cambio de contraseña.			
3	El actor llena el formulario.				
4	El actor acepta el formulario.	El sistema valida los datos.	a	La contraseña no es valida. 4. Se pide que deban coincidir los dos campos de la contraseña. 5. Se recomienda no usar caracteres especiales. 6. Se muestra el formulario con los datos validos.	
5		El sistema recoge los datos y los vacía en la BD.			
6		El sistema manda un mensaje de aceptación.			
7	El actor recibe una notificación con los datos de su nueva contraseña.				
8	Termina solicitar nueva contraseña.				

Precondiciones: El actor deberá estar en sesión

Poscondiciones: Un actor tendrá nueva contraseña en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU A04 Recordar contraseña.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A04 Recordar contraseña.</b>	Inicia: A reserva del Actor. Termina: termina recordar contraseña.
Descripción: Permite la obtención de la contraseña mediante una palabra clave.	
Actores: Supervisor General, cliente, operador venta, operador compra, supervisor personal, operador reporte.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor elige recordar contraseña	El sistema envía formulario recordar contraseña.			
2	El actor llena formulario				
3	El actor acepta el formulario.	El sistema valida los datos Consulta la BD.	a	No coincide el nombre del usuario y la respuesta clave. 1. Se le pide verifique los datos. 2. Se le presenta de nuevo el formulario recordar contraseña.	

			b	Los caracteres no son	
--	--	--	---	-----------------------	--





				validos. 1. Se le pide verifique sus datos. 2. Se le recomienda no usar caracteres especiales. 3. Se le presenta de nuevo el formulario recordar contraseña.	
4		El sistema consulta la BD y regresa una consulta			
5		El sistema envia un mensaje con los datos del usuario.			
6	El actor recibe una notificación con sus datos de contraseña.				
7	Termina recordar contraseña.				

Precondiciones: El actor tendrá problemas para iniciar sesión.  
 Poscondiciones: El actor recibirá su contraseña.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU A05 Consultar perfil.  
 Fecha de creación: 2005.  
 Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A05 Consultar perfil.</b>	Inicia: A reserva del Actor Supervisor general o de personal. Termina: termina consultar perfil.
Descripción: Permite la consulta de cualquier usuario en el sistema.	
Actores: Supervisor General, Supervisor Personal.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige el comando consultas	El sistema despliega las opciones de consultas			
4	El actor elige consulta empleado	El sistema despliega formulario de búsqueda empleado			
5	El actor llena el formulario				
6	El actor acepta el formulario	El sistema valida los datos	a	Los campos tienen datos inválidos 1. Se pide al usuario verifique los datos 2. Se regresa el formulario búsqueda empleado con datos validos. 3. Se recomienda búsqueda general	
			b	No lleno ningún campo 1. Se pide llene formulario 2. Se regresar formulario búsqueda empleado.	

7		El sistema consulta la	a	No existe el empleado.	
---	--	------------------------	---	------------------------	--



		BD		<ol style="list-style-type: none"> <li>1. Se informa que no existe ese empleado.</li> <li>2. Se regresa formulario de búsqueda empleado.</li> <li>3. Se recomienda busque por puesto.</li> </ol>		
			b	<p>Existe coincidencias de muchos empleados con los datos que ingreso el usuario.</p> <ol style="list-style-type: none"> <li>1. Se muestran todos los empleados.</li> <li>2. Encuentra entre todos los empleados al que busca y lo selecciona.</li> <li>3. Se recomienda nueva búsqueda.</li> </ol>	a	<p>El actor elige que muestre todos los empleados y no encuentra su búsqueda.</p> <ol style="list-style-type: none"> <li>1. Se regresa al formulario de búsqueda de empleado.</li> </ol>
8		El sistema crea un reporte y muestra los resultados				
9	El actor recibe una notificación con todos los datos del empleado.					
10	Termina consultar perfil					

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá un reporte del empleado.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU A06 Actualizar perfil.  
 Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A06 Actualizar perfil.</b>	Inicia: A reserva del Actor Supervisor general o de personal. Termina: termina actualizar perfil.
Descripción: Permite la actualización de un usuario en el sistema.	
Actores: Supervisor General, Supervisor Personal.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige actualizar	El sistema despliega opciones de actualizar			
4	El actor elige actualizar empleado				
5	El actor inicia consultar perfil	El sistema empieza consultar perfil → <b>CU A05 Consultar perfil.</b>	a	El usuario no existe 1. Termina actualizar perfil.	
6		El sistema regresa un formulario con los datos del empleado listo para actualizar			
7	El actor llena el formulario				



8	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos y los originales que resultaron erróneos.		
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos y los originales que resultaron erróneos.		
		El sistema consulta la BD.	a	Los datos claves coinciden con otro usuario 1. Se le pide verificar los datos. 2. Se regresa formulario de actualizar con los datos originales del empleado.		
9		El sistema valida nivel de acceso.				
10		El sistema recoge los datos y los vacía en la BD				
11		El sistema genera un reporte de los cambios hechos				
12	El actor recibe una notificación con los datos del empleado actualizado.					
13	Termina actualizar perfil.					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un empleado actualizado en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU A07 Eliminar una cuenta.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A07 Eliminar una cuenta.</b>	Inicia: A reserva del Actor Supervisor general o de personal. Termina: termina eliminar cuenta.
Descripción: Permite la eliminación de una cuenta en el sistema.	
Actores: Supervisor General, Supervisor Personal.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				
3	El actor elige eliminar	El sistema despliega opciones de eliminar				
4	El actor elige eliminar empleado					



5	El actor inicia consultar perfil	El sistema empieza consultar perfil → <b>CU A05 Consultar perfil.</b>	a	El usuario no existe. 1. Terminar eliminar cuenta		
6		El sistema regresa un formulario con los datos del empleado listo para eliminar				
7	El actor elige eliminar empleado	El sistema manda formulario de confirmación				
8	El actor acepta eliminar	El sistema consulta nivel de acceso	a	El usuario no cuenta con el nivel de acceso. 1. Se le dice que no se puede eliminar a ese usuario. 2. Termina eliminar usuario.		
9		El sistema elimina y regresa un mensaje de eliminación				
10	El actor recibe notificación de eliminación					
11	Termina eliminar cuenta.					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un empleado menos en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU A08 Imprimir una cuenta.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A08 Imprimir una cuenta.</b>	Inicia: A reserva del Actor Supervisor general o de personal. Termina: termina imprimir cuenta.
Descripción: Permite la impresión de una cuenta.	
Actores: Supervisor General, Supervisor Personal.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				
3	El actor elige consultar	El sistema despliega opciones de consultar				
4	El actor elige consultar empleado					
5	El actor inicia consultar perfil	El sistema empieza consultar perfil → <b>CU A05 Consultar perfil.</b>	a	El usuario no existe. 5. Terminar eliminar cuenta		
6	El actor elige imprimir cuenta	El sistema consulta BD y genera un reporte de impresión				
7		El sistema manda a la impresora los datos				
8	Termina imprimir cuenta.					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un reporte impreso de una cuenta.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU A09 Validar nivel de acceso.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A09 Validar nivel de acceso.</b>	Inicia: A reserva del componente de acceso. Termina: termina validar nivel acceso.
Descripción: Valida al usuario si tiene derechos de realizar una operación.	
Actores: Supervisor General, Supervisor Personal, operador de compra, operador de venta, cliente, operador de reporte.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El componente pide el dato puesto empleado	El sistema consulta la BD				
2	El componente pide el dato nivel acceso y lo relaciona con el dato puesto	El sistema consulta la BD				
3	El componente manda un mensaje		a	Invalido 1. No concuerda nivel de acceso con empleado 2. No concuerda Nivel de acceso con puesto. 3. No concuerda nivel de acceso con acción.		
			b	Valido 1. El empleado puede ejecutar la acción. 2. El puesto coincide con el nivel de acceso.		
4	Termina validar nivel de acceso.					

Precondiciones: El actor deberá intentar usar un comando.

Poscondiciones: Se tendrá un actor valido.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU A10 Iniciar sesión.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU A10 Iniciar sesión.</b>	Inicia: A reserva del Actor. Termina: termina iniciar sesión.
Descripción: Permite a un usuario logearse dentro del sistema.	
Actores: Supervisor General, Supervisor Personal, operador de venta, operador compra, cliente, operador reporte.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor elige iniciar sesión	El sistema genera formulario de iniciar sesión				
2	El actor llena el formulario					
3	El actor acepta el formulario	El sistema valida los datos	a	Al actor se le olvida su contraseña. 1. → <b>CU A04 Recordar contraseña.</b> 2. Se le pide darse de alta. 3. Se regresa al formulario de iniciar sesión.		
4		El sistema despliega mensaje de bienvenida				
5	El actor recibe notificación de su estado					
6	Termina iniciar sesión.					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un actor logeado en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.



Documento: CU B01 Crear tipo de producto.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B01 Crear tipo de producto.</b>	Inicia: A reserva del Actor Supervisor general, operador de compra. Termina: termina crear tipo de producto.
Descripción: Permite la creación de un nuevo producto de la organización.	
Actores: Supervisor General, operador de compra.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				
3	El actor elige crear	El sistema despliega opciones de crear				
4	El actor elige crear producto	El sistema despliega formulario de crear producto.				
5	El actor llena formulario					
6	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos.		
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos.		
		El sistema consulta la BD	c	El producto ya existe 1. Regresar formulario 2. Terminar crear tipo producto		
7		El sistema recoge los datos y los vacía en la BD				
8		El sistema manda un mensaje de aceptación.				
9	El actor recibe una notificación y un reporte del nuevo tipo producto creado.					
10	Termina crear tipo producto					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un nuevo tipo de producto en el sistema.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU B02 Crear catálogo.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B02 Crear catálogo.</b>	Inicia: A reserva del Actor Supervisor general, operador compra. Termina: termina crear catálogo.
Descripción: Permite la creación de un nuevo catálogo de la organización.	
Actores: Supervisor General, operador compra.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige crear	El sistema despliega opciones de crear			
4	El actor elige crear catálogo	El sistema despliega formulario de crear catálogo.			
5	El actor llena formulario				
6	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos.	
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos.	
			c	El catalogo ya existe 1. Regresar al formulario 2. Terminar crear catalogo	
7		El sistema recoge los datos y los vacía en la BD			
8		El sistema manda un mensaje de aceptación.			
9	El actor recibe notificación del nuevo catálogo creado				
10	Termina crear catálogo.				

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un nuevo catálogo en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU B03 Crear artículo.



Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B03 Crear artículo.</b>	Inicia: A reserva del Actor Supervisor general, operador de compra. Termina: termina crear artículo.
Descripción: Permite la creación de un nuevo artículo de la organización.	
Actores: Supervisor General, operador compra.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				
3	El actor elige crear	El sistema despliega opciones de crear				
4	El actor elige crear artículo	El sistema despliega formulario de crear artículo.				
5	El actor llena formulario					
6	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos.		
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos.		
			c	El catalogo ya existe 1. Regresar al formulario 2. Terminar crear catalogo		
7		El sistema recoge los datos y los vacía en la BD				
8		El sistema manda un mensaje de aceptación.				

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un artículo nuevo en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU B04 Actualizar un producto.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B04 Actualizar un producto.</b>	Inicia: A reserva del Actor Supervisor general, operador compra. Termina: termina actualizar producto.
Descripción: Permite la actualización de un producto de la organización.	
Actores: Supervisor General, operador compra.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		





2		El sistema despliega el menú del sistema SFM.			
3	El actor elige actualizar	El sistema despliega opciones de actualizar			
4	El actor inicia consultar producto	El sistema empieza consultar producto → <b>CU B07 Consultar producto.</b>	a	El producto no existe 1. Termina actualizar producto.	
5		El sistema regresa un formulario con los datos del producto listo para actualizar			
6	El actor llena el formulario				
7	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos y los originales que resultaron erróneos.	
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos y los originales que resultaron erróneos.	
		El sistema consulta la BD.	a	Los datos claves coinciden con otro producto 1. Se le pide verificar los datos. 2. Se regresa formulario de actualizar con los datos originales del producto.	
8		El sistema valida nivel de acceso.			
9		El sistema recoge los datos y los vacía en la BD			
10		El sistema genera un reporte de los cambios hechos			
11	El actor recibe una notificación con los datos del producto actualizado.				
12	Termina actualizar producto.				

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un producto actualizado en el sistema.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU B05 Consultar catálogo de artículos.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B05 Consultar catálogo de artículos.</b>		Inicia: A reserva del Actor Supervisor general, operador venta, operador compra, operador reporte, cliente. Termina: termina consultar catálogo de artículos.			
Descripción: Permite la consulta de los catálogos de la organización.					
Actores: Supervisor General, operador compra, operador venta, cliente, operador reporte.					
N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor elige consulta	El sistema despliega opciones de consulta			
2	El actor elige consulta catálogo	El sistema despliega formulario de búsqueda catálogo			
3	El actor llena el formulario				
4	El actor acepta el formulario	El sistema valida los datos	a	Los campos tienen datos inválidos 1. Se pide al usuario verifique los datos 2. Se regresa el formulario búsqueda empleado con datos validos. 3. Se recomienda búsqueda general	
			b	No lleno ningún campo 1. Se pide llene formulario 2. Se regresar formulario búsqueda empleado.	
5		El sistema consulta la BD	a	No existe el catálogo. 1. Se informa que no existe ese empleado. 2. Se regresa formulario de búsqueda empleado. 3. Se recomienda busque por general.	
6		El sistema crea un reporte y muestra los resultados			
7	El actor recibe una notificación con todos los datos del catálogo.				
8	Termina consulta catálogo.				

Precondiciones: El actor deberá tener la intención de buscar un catálogo.

Poscondiciones: Se tendrá un reporte de los catálogos en el sistema.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU B06 Actualizar catálogo.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B06 Actualizar catálogo.</b>		Inicia: A reserva del Actor Supervisor general, operador compra. Termina: termina actualizar catálogo.			
Descripción: Permite la actualización de un catálogo de la organización.					
Actores: Supervisor General, operador compra.					
N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige actualizar	El sistema despliega opciones de crear			
4	El actor inicia consultar catálogo	El sistema empieza consultar catálogo → <b>CU B05 Consultar catálogo.</b>	a	El catálogo no existe 1. Termina actualizar catálogo.	
5		El sistema regresa un formulario con los datos del catálogo listo para actualizar			
6	El actor llena el formulario				
7	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos y los originales que resultaron erróneos.	
			b	En un campo los tipos de datos no son los indicados. 3. Se pide ingrese datos validos. 4. Se muestra de nuevo formulario con los datos validos y los originales que resultaron erróneos.	
		El sistema consulta la BD.	a	Los datos claves coinciden con otro catálogo 3. Se le pide verificar los datos. 4. Se regresa formulario de actualizar con los datos originales del catálogo.	
8		El sistema valida nivel de acceso.			
9		El sistema recoge los datos y los vacía en la BD			



10		El sistema genera un reporte de los cambios hechos			
11	El actor recibe una notificación con los datos del catálogo actualizado.				
12	Termina actualizar catálogo.				

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá un catálogo actualizado en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU B07 Consultar producto.  
 Fecha de creación: 2005.  
 Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B07 Consultar producto.</b>	Inicia: A reserva del Actor Supervisor general, operador compra, operador venta, cliente, operador reporte. Termina: termina consultar producto.
Descripción: Permite la consulta de un producto de la organización.	
Actores: Supervisor General, operador compra, operador venta, cliente, operador reporte.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor elige consultar	El sistema despliega opciones de consulta			
2	El actor elige consulta producto	El sistema despliega formulario de búsqueda producto			
3	El actor llena el formulario				
4	El actor acepta el formulario	El sistema valida los datos	a	Los campos tienen datos inválidos <ol style="list-style-type: none"> <li>1. Se pide al usuario verifique los datos</li> <li>2. Se regresa el formulario búsqueda empleado con datos validos.</li> <li>3. Se recomienda búsqueda búsqueda general</li> </ol>	
			b	No lleno ningún campo <ol style="list-style-type: none"> <li>1. Se pide llene formulario</li> <li>2. Se regresar formulario búsqueda empleado.</li> </ol>	
5		El sistema consulta la BD	a	No existe el producto. <ol style="list-style-type: none"> <li>1. Se informa que no existe ese empleado.</li> <li>2. Se regresa formulario de búsqueda empleado.</li> <li>3. Se recomienda busque por puesto.</li> </ol>	
6		El sistema crea un reporte y muestra los resultados			
7	El actor recibe una notificación con todos los datos del producto.				
8	Termina consulta producto.				

Precondiciones: El actor tendrá necesidad de consultar un producto.  
 Poscondiciones: Se tendrá el reporte de los productos en el sistema.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU B08 Eliminar catálogo.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B08 Eliminar catálogo.</b>	Inicia: A reserva del Actor Supervisor general, operador de compra. Termina: termina eliminar catálogo.
Descripción: Permite la eliminación de una catálogo de la organización.	
Actores: Supervisor General, operador compra.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige eliminar	El sistema despliega opciones de eliminar			
4	El actor elige eliminar catálogo				
5	El actor inicia consultar catálogo	El sistema empieza consultar catálogo → <b>CU B05 Consultar catálogo.</b>	a	El catálogo no existe. 1. Terminar eliminar catálogo	
6		El sistema regresa un formulario con los datos del catálogo listo para eliminar			
7	El actor elige eliminar catálogo	El sistema manda formulario de confirmación			
8	El actor acepta eliminar	El sistema consulta nivel de acceso	a	El usuario no cuenta con el nivel de acceso. 1. Se le dice que no se puede eliminar a ese catálogo. 2. Termina eliminar catálogo.	
9		El sistema elimina y regresa un mensaje de eliminación			
10	El actor recibe notificación de eliminación				
11	Termina eliminar catálogo.				

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un catálogo menos en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU B09 Eliminar producto.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B09 Eliminar producto.</b>	Inicia: A reserva del Actor Supervisor general, operador compra. Termina: termina eliminar producto.
Descripción: Permite la eliminación de un producto de la organización.	
Actores: Supervisor General, operador compra.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			



3	El actor elige eliminar	El sistema despliega opciones de eliminar			
4	El actor elige eliminar producto				
5	El actor inicia consultar producto	El sistema empieza consultar producto → <b>CU B07 Consultar producto.</b>	a	El producto no existe. 1. Terminar eliminar producto	
6		El sistema regresa un formulario con los datos del producto listo para eliminar			
7	El actor elige eliminar producto	El sistema manda formulario de confirmación			
8	El actor acepta eliminar	El sistema consulta nivel de acceso	a	El usuario no cuenta con el nivel de acceso. 1. Se le dice que no se puede eliminar a ese producto. 2. Termina eliminar catálogo.	
9		El sistema elimina y regresa un mensaje de eliminación			
10	El actor recibe notificación de eliminación				
11	Termina eliminar producto.				

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá un producto menos en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU B10 Actualizar artículo.  
 Fecha de creación: 2005.  
 Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B10 Actualizar artículo.</b>	Inicia: A reserva del Actor Supervisor general, operador compra, operador venta. Termina: termina actualizar artículo.
Descripción: Permite la actualización de un artículo de la organización.	
Actores: Supervisor General, operador compra, operador venta.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige actualizar	El sistema despliega opciones de crear			
4	El actor inicia consultar artículo	El sistema empieza consultar artículo → <b>CU B11 Consultar artículo.</b>	a	El artículo no existe 1. Termina actualizar artículo.	
5		El sistema regresa un formulario con los datos del artículo listo para actualizar			
6	El actor llena el formulario				



7	El actor acepta formulario	El sistema valida datos.	A	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos y los originales que resultaron erróneos.		
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos y los originales que resultaron erróneos.		
		El sistema consulta la BD.	a	Los datos claves coinciden con otro artículo 1. Se le pide verificar los datos. 2. Se regresa formulario de actualizar con los datos originales del artículo.		
8		El sistema valida nivel de acceso.				
9		El sistema recoge los datos y los vacía en la BD				
10		El sistema genera un reporte de los cambios hechos				
11	El actor recibe una notificación con los datos del artículo actualizado.					
12	Termina actualizar artículo.					

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá un artículo actualizado en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU B11 Consultar artículo.  
 Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B11 Consultar artículo.</b>	Inicia: A reserva del Actor Supervisor general, operador compra, operador venta, cliente, operador reporte. Termina: termina consultar artículo.
Descripción: Permite la consulta de un artículo de la organización.	
Actores: Supervisor General, operador compra, operador venta, cliente, operador reporte.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor elige consulta	El sistema despliega formulario de búsqueda artículo				
2	El actor llena el formulario		a	Elige búsqueda por 1. Sustancia activa 2. Material curación 3. Por función. 4. Búsqueda avanzada.		



3	El actor acepta el formulario	El sistema valida los datos	a	Los campos tienen datos inválidos 1. Se pide al usuario verifique los datos 2. Se regresa el formulario búsqueda empleado con datos validos. 3. Se recomienda búsqueda general		
			b	No lleno ningún campo 1. Se pide llene formulario 2. Se regresar formulario búsqueda empleado.		
4		El sistema consulta la BD	a	No existe el artículo. 1. Se informa que no existe ese empleado. 2. Se regresa formulario de búsqueda empleado. 3. Se recomienda busque por puesto.		
5		El sistema crea un reporte y muestra los resultados				
6	El actor recibe una notificación con todos los datos del artículo.					
7	Termina consulta artículo					

Precondiciones: El actor tendrá necesidad de consultar un artículo  
 Poscondiciones: Se tendrá el reporte de un artículo en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU B12 Eliminar artículo.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU B12 Eliminar artículo.</b>	Inicia: A reserva del Actor Supervisor general, operador compra. Termina: termina eliminar artículo.
Descripción: Permite la eliminación de un artículo de la organización.	
Actores: Supervisor General, operador compra.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige eliminar	El sistema despliega opciones de eliminar			
4	El actor elige eliminar artículo				
5	El actor inicia consultar artículo	El sistema empieza consultar artículo → <b>CU B11 Consultar artículo.</b>	a	El artículo no existe. 1. Terminar eliminar artículo.	
6		El sistema regresa un formulario con los datos del artículo listo para eliminar			





7	El actor elige eliminar artículo	El sistema manda formulario de confirmación			
8	El actor acepta eliminar	El sistema consulta nivel de acceso	a	El usuario no cuenta con el nivel de acceso. 1. Se le dice que no se puede eliminar a ese artículo. 2. Termina eliminar artículo.	
9		El sistema elimina y regresa un mensaje de eliminación			
10	El actor recibe notificación de eliminación				
11	Termina eliminar artículo.				

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá un artículo menos en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU C01 Crear farmacia.  
 Fecha de creación: 2005.  
 Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU C01 Crear farmacia.</b>	Inicia: A reserva del Actor Supervisor general. Termina: termina crear farmacia.
Descripción: Permite la creación de una farmacia de la organización.	
Actores: Supervisor General.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige crear	El sistema despliega opciones de crear			
4	El actor elige crear farmacia	El sistema despliega formulario de crear farmacia.			
5	El actor llena formulario				
6	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos.	
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos.	
		El sistema consulta la BD	c	La farmacia ya existe 1. Regresar formulario 2. Terminar crear farmacia	
7		El sistema recoge los datos y los vacía en la BD			



8		El sistema manda un mensaje de aceptación.			
9	El actor recibe una notificación y un reporte de la nueva farmacia creado.				
10	Termina crear farmacia				

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá una nueva farmacia en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU C02 Actualizar farmacia.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU C02 Actualizar farmacia.</b>	Inicia: A reserva del Actor Supervisor general Termina: termina actualizar farmacia.
Descripción: Permite la actualización de una farmacia de la organización.	
Actores: Supervisor General.	

N	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige actualizar	El sistema despliega opciones de actualizar			
5	El actor inicia consultar farmacia	El sistema empieza consultar farmacia → <b>CU C03 Consultar farmacia.</b>	a	La farmacia no existe 1. Termina actualizar farmacia.	
6		El sistema regresa un formulario con los datos del empleado listo para actualizar			
7	El actor llena el formulario				
8	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos y los originales que resultaron erróneos.	
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos y los originales que resultaron erróneos.	
		El sistema consulta la BD.	a	Los datos claves coinciden con otra farmacia 1. Se le pide verificar los datos. 2. Se regresa formulario de actualizar con los datos originales de la farmacia.	



9		El sistema valida nivel de acceso.			
10		El sistema recoge los datos y los vacía en la BD			
11		El sistema genera un reporte de los cambios hechos			
12	El actor recibe una notificación con los datos de la farmacia actualizada.				
13	Termina actualizar farmacia.				

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá una farmacia actualizada en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU C03 Consultar farmacia.  
 Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU C03 Consultar farmacia.</b>	Inicia: A reserva del Actor Supervisor general, operador compra, cliente, operador venta, operador reporte. Termina: termina consultar farmacia.
Descripción: Permite la consulta de una farmacia de la organización.	
Actores: Supervisor General, operador compra, operador venta, cliente, operador reporte.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor elige consulta farmacia	El sistema despliega formulario de búsqueda farmacia			
2	El actor llena el formulario				
3	El actor acepta el formulario	El sistema valida los datos	a	Los campos tienen datos inválidos 4. Se pide al usuario verifique los datos 5. Se regresa el formulario búsqueda empleado con datos validos. 6. Se recomienda búsqueda general	
			b	No lleno ningún campo 3. Se pide llene formulario 4. Se regresar formulario búsqueda empleado.	
4		El sistema consulta la BD	a	No existe la farmacia. 4. Se informa que no existe ese empleado. 5. Se regresa formulario de búsqueda empleado. 6. Se recomienda busque por puesto.	
5		El sistema crea un reporte y muestra los resultados			
6	El actor recibe una notificación con todos los datos de la farmacia.				
7	Termina consulta farmacia.				

Precondiciones: El actor tendrá necesidad de consultar datos de una farmacia.  
 Poscondiciones: Se tendrá un reporte de una farmacia del sistema.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU C04 Crear almacén.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU C04 Crear almacén.</b>	Inicia: A reserva del Actor Supervisor general. Termina: termina crear almacén.
Descripción: Permite la creación de un nuevo almacén de la organización.	
Actores: Supervisor General.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige crear	El sistema despliega opciones de crear			
4	El actor elige crear almacén	El sistema despliega formulario de crear almacén.			
5	El actor llena formulario				
6	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos.	
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos.	
		El sistema consulta la BD	c	El almacén ya existe 1. Regresar formulario 2. Terminar crear almacén	
7		El sistema recoge los datos y los vacía en la BD			
8		El sistema manda un mensaje de aceptación.			
9	El actor recibe una notificación y un reporte del nuevo almacén creado.				
10	Termina crear almacén				

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un nuevo almacén en el sistema.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU C05 Eliminar farmacia.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU C05 Eliminar farmacia.</b>	Inicia: A reserva del Actor Supervisor general. Termina: termina eliminar farmacia.
Descripción: Permite la eliminación de una farmacia de la organización.	
Actores: Supervisor General.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				
3	El actor elige eliminar	El sistema despliega opciones de eliminar				
4	El actor elige eliminar farmacia					
5	El actor inicia consultar farmacia	El sistema empieza consultar farmacia → <b>CU C03 Consultar farmacia.</b>	a	La farmacia no existe. 1. Terminar eliminar farmacia.		
6		El sistema regresa un formulario con los datos de la farmacia lista para eliminar				
7	El actor elige eliminar farmacia	El sistema manda formulario de confirmación				
8	El actor acepta eliminar	El sistema consulta nivel de acceso	a	El usuario no cuenta con el nivel de acceso. 1. Se le dice que no se puede eliminar a esa farmacia. 2. Termina eliminar farmacia.		
9		El sistema elimina y regresa un mensaje de eliminación				
10	El actor recibe notificación de eliminación					
11	Termina eliminar farmacia.					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá una farmacia menos en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU C06 Crear puesto.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU C06 Crear puesto.</b>	Inicia: A reserva del Actor Supervisor general, supervisor personal. Termina: termina crear puesto.
Descripción: Permite la creación de un puesto en la organización.	
Actores: Supervisor General, supervisor personal.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				



3	El actor elige crear	El sistema despliega opciones de crear			
4	El actor elige crear puesto	El sistema despliega formulario de crear puesto.			
5	El actor llena formulario				
6	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos.	
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos.	
		El sistema consulta la BD	c	El puesto ya existe 1. Regresar formulario 2. Terminar crear puesto	
7		El sistema recoge los datos y los vacía en la BD			
8		El sistema manda un mensaje de aceptación.			
9	El actor recibe una notificación y un reporte del nuevo puesto creado.				
10	Termina crear puesto				

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá un puesto nuevo en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU C07 Actualizar puesto.  
 Fecha de creación: 2005.  
 Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU C07 Actualizar puesto.</b>	Inicia: A reserva del Actor Supervisor general, supervisor personal. Termina: termina actualizar puesto.
Descripción: Permite la actualización de un puesto en la organización.	
Actores: Supervisor General, supervisor personal.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige actualizar	El sistema despliega opciones de actualizar			
4	El actor elige actualizar empelado				
5	El actor inicia consultar puesto	El sistema empieza consultar perfil	a	El puesto no existe 1. Termina actualizar puesto.	



6		El sistema regresa un formulario con los datos del puesto listo para actualizar			
7	El actor llena el formulario				
8	El actor acepta formulario	El sistema valida datos.	a	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo 2. Se muestra de nuevo el formulario con los datos validos y los originales que resultaron erróneos.	
			b	En un campo los tipos de datos no son los indicados. 1. Se pide ingrese datos validos. 2. Se muestra de nuevo formulario con los datos validos y los originales que resultaron erróneos.	
		El sistema consulta la BD.	a	Los datos claves coinciden con otro puesto 1. Se le pide verificar los datos. 2. Se regresa formulario de actualizar con los datos originales del puesto.	
9		El sistema valida nivel de acceso.			
10		El sistema recoge los datos y los vacía en la BD			
11		El sistema genera un reporte de los cambios hechos			
12	El actor recibe una notificación con los datos del puesto actualizado.				
13	Termina actualizar puesto				

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá un puesto actualizado en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU C08 Eliminar puesto.  
 Fecha de creación: 2005.  
 Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU C08 Eliminar puesto.</b>	Inicia: A reserva del Actor Supervisor general. Termina: termina eliminar puesto.
Descripción: Permite la eliminación de un puesto de la organización.	
Actores: Supervisor General.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				



3	El actor elige eliminar	El sistema despliega opciones de eliminar			
4	El actor elige eliminar puesto				
5	El actor inicia consultar puesto	El sistema empieza consultar puesto	a	El puesto no existe. 1. Terminar eliminar puesto.	
6		El sistema regresa un formulario con los datos del puesto listo para eliminar			
7	El actor elige eliminar puesto	El sistema manda formulario de confirmación			
8	El actor acepta eliminar	El sistema consulta nivel de acceso	a	El usuario no cuenta con el nivel de acceso. 1. Se le dice que no se puede eliminar a ese puesto. 2. Termina eliminar puesto.	
9		El sistema elimina y regresa un mensaje de eliminación			
10	El actor recibe notificación de eliminación				
11	Termina eliminar puesto.				

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá un puesto menos en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU D01 Crear venta.  
 Fecha de creación: 2005.  
 Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU D01 Crear venta.</b>	Inicia: A reserva del Actor Supervisor general, operador venta. Termina: termina crear venta.
Descripción: Permite la creación de una venta a un cliente.	
Actores: Supervisor General, operador venta, cliente.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>	
2		El sistema despliega el menú del sistema SFM.			
3	El actor elige crear	El sistema despliega opciones de crear			
4	El actor elige crear venta	El sistema despliega formulario de crear venta.			
5	El actor llena el formulario	El sistema ofrece cajas de texto			El actor especifica todos los artículos y cantidad.
6	El actor acepta el formulario	El sistema consulta la BD	a	Existencias bajas 1. Se recomienda cierta cantidad del artículo. 2. Se recomiendan artículos similares.	
			b	No hay existencias 1. Se recomiendan artículos similares. 2. Termina venta.	
7		El sistema regresa formato de compra			





8	El actor espera confirmación del cliente.				
9	El actor registra la orden	El sistema lanza registrar orden → <b>CU D02 registrar orden.</b>			
10	Termina crear venta				

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se iniciara el proceso de una venta en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU D02 Registrar orden.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU D02 Registrar orden.</b>	Inicia: A reserva del componente del sistema. Termina: termina registrar orden.
Descripción: Permite el registro de una operación de venta en el sistema.	
Actores: Supervisor General, operador venta, componente del sistema.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El componente recibe orden de registrar orden.					
2	El componente pide al sistema vaciar datos a la BD	El sistema valida datos y lanza un mensaje de registro de orden				
3	Termina registrar orden					

Precondiciones: El actor deberá creado una venta.

Poscondiciones: Se tendrá un registro de venta en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU D03 Registrar venta.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU D03 Registrar venta.</b>	Inicia: A reserva del Actor Supervisor general, operador venta. Termina: termina registrar venta.
Descripción: Permite la confirmación de una orden de venta.	
Actores: Supervisor General, operador venta, cliente.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El componente recibe orden de confirmación.					
2	El componente pide al sistema cambiar estatus de orden venta	El sistema lanza un mensaje con el nuevo estado de la orden	a	Cancelado		
			b	Confirmado 1. Modificar existencias		
3	Termina registrar orden					

Precondiciones: El actor deberá estar la orden de venta.

Poscondiciones: Se tendrá una venta confirmada o cancelada.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU E01 Crear compra.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU E01 Crear compra.</b>	Inicia: A reserva del Actor Supervisor general, operador compra. Termina: termina crear compra.
Descripción: Permite la creación de una compra de la organización.	
Actores: Supervisor General, operador compra.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				
3	El actor elige crear	El sistema despliega opciones de crear				
4	El actor elige crear compra	El sistema despliega formulario de crear compra.				
5	El actor llena el formulario	El sistema ofrece cajas de texto				El actor especifica todos los artículos y cantidad.
6		El sistema regresa formato de compra				
7	El actor registra la orden	El sistema lanza registrar orden → <b>CU E02 registrar compra.</b>				
8	Termina crear venta					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá el inicio de proceso de compra en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU E02 Registrar orden.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU E02 Registrar orden.</b>	Inicia: A reserva del componente del sistema. Termina: termina registrar orden
Descripción: Permite el registro de una orden de compra en el sistema	
Actores: Supervisor General, operador compra, componente del sistema.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El componente recibe orden de registrar orden.					
2	El componente pide al sistema vaciar datos a la BD	El sistema valida datos y lanza un mensaje de registro de orden				
3	Termina registrar orden					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá un registro de orden en el sistema.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU E03 Registrar compra.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU E03 Registrar compra.</b>	Inicia: A reserva del Actor Supervisor general, operador compra. Termina: termina registrar compra.
Descripción: Permite la confirmación de una orden de compra.	
Actores: Supervisor General, operador compra.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El componente recibe orden de confirmación.					
2	El componente pide al sistema cambiar estatus de orden compra	El sistema lanza un mensaje con el nuevo estado de la orden	a	Cancelado		
			b	Confirmado 1. Modificar existencias		
3	Termina registrar orden compra					

Precondiciones: El actor deberá estar en sesión.

Poscondiciones: Se tendrá una orden de compra confirmada o cancelada en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU F01 Crear monitoreo.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU F01 Crear monitoreo.</b>	Inicia: A reserva del Actor Supervisor general. Termina: termina crear monitoreo.
Descripción: Permite la definición de cómo se llevara a cabo un monitoreo.	
Actores: Supervisor General.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor inicia sesión en el sistema	Introduce su nombre de usuario y su contraseña → <b>CU A10 Iniciar sesión.</b>	a	Al actor se le olvida su contraseña. → <b>CU A04 Recordar contraseña.</b>		
2		El sistema despliega el menú del sistema SFM.				
3	El actor elige crear	El sistema despliega opciones de crear				
4	El actor elige crear monitoreo	El sistema despliega formulario de crear monitoreo.				
5	El actor llena el formulario					
6	El actor acepta formulario	El sistema valida los datos	a	Los campos tienen datos inválidos 1. Se pide al usuario verifique los datos. 2. Se regresa el formulario con los datos válidos y limpiando los datos fallidos		
			b	No relleno ningún campo 1. Se pide al usuario rellene el formulario. 2. Se regresa el formulario de creación de monitoreo.		



			c	Existe un campo vacío necesario 1. Se pide ingrese los datos del campo. 2. Se muestra de nuevo el formulario con los datos validos.		
7		El sistema recoge los datos y los vacía en la BD				
8		El sistema manda un mensaje de confirmación de creación de monitoreo				
9	Termina la creación del monitoreo del sistema					

Precondiciones: El actor deberá estar en sesión.  
 Poscondiciones: Se tendrá monitoreo configurado en el sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.  
 Documento: CU F02 Consultar almacén local.  
 Fecha de creación: 2005.  
 Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU F02 Consultar almacén local.</b>	Inicia: A reserva del Actor componente del sistema. Termina: termina consultar almacén local.
Descripción: Permite al sistema consultar los almacenes de las filiales de la organización.	
Actores: componente del sistema.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos	Excepciones
1	El actor llama al componente consulta del sistema, pasando los parámetros (almacén)	El sistema consulta la base de datos y revisa los inventarios del almacén	a	El almacén no existe 1. El sistema despliega una ventana de error 2. El sistema recomienda revisar los almacenes en existencia 3. Termina consultar almacén local	
2		El sistema devuelve un estado del almacén.			
3		El sistema devuelve información sobre el almacén			
4	El actor componente recibe la información				
5	El actor manda a llamar al componente existencias del sistema y le pasa la información recibida.	El sistema procesa la información y devuelve un estado.	a	La existencia del producto es baja 1. El sistema llama → <b>CU F05 Crear reporte producto</b> 2. El sistema llama → <b>CU F04 Crear mensaje alerta.</b>	
			b	La existencia del producto es normal 1. El sistema llama → <b>CU F05 Crear reporte producto</b>	
6		El sistema lanza una ventana con los mensajes de alerta			
7	Termina consulta almacén local				

Precondiciones: El actor iniciara un monitoreo.  
 Poscondiciones: Se iniciara un monitoreo en el sistema.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU F03 Consultar inventario general.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU F03 Consultar inventario general.</b>	Inicia: A reserva del Actor componente del sistema. Termina: termina consultar inventario general.
Descripción: Permite la consulta general de un artículo en el sistema.	
Actores: componente del sistema.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor llama al componente consulta del sistema, pasando los parámetros (todos)	El sistema consulta la base de datos y revisa los inventarios.				
2		El sistema devuelve la información sobre los inventarios				
3	El actor componente recibe la información					
4	El actor manda a llamar al componente existencias del sistema y le pasa la información recibida	El sistema aplica los criterios del monitoreo a la información recibida	a	La existencia del producto es baja 3. El sistema llama → <b>CU F05</b> <b>Crear reporte producto</b> 4. El sistema llama → <b>CU F04</b> <b>Crear mensaje alerta.</b>		
			b	La existencia del producto es normal 2. El sistema llama → <b>CU F05</b> <b>Crear reporte producto</b>		
6		El sistema lanza una ventana con los mensajes de alerta.				
7	Termina consulta inventario					

Precondiciones: Se habrá iniciado un monitoreo del sistema.

Poscondiciones: se tendrá un monitoreo del sistema.

Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU F04 Crear mensaje de alerta.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU F04 Crear mensaje de alerta.</b>	Inicia: A reserva del componente del sistema. Termina: termina crear mensaje de alerta.
Descripción: Permite la creación de un mensaje que se mostrara al supervisor general, operador de compra, operador venta.	
Actores: Supervisor General, operador compra, operador venta, componente del sistema.	

n	Flujo principal de eventos ACTOR	Flujo Principal		Flujos Alternos		Excepciones
1	El actor componente llama al componente consulta del sistema con los parámetros (producto)	El sistema consulta la base de datos y devuelve la información				
2	El actor componente recibe la información					
3	El actor componente le da un valor alerta al estado del producto	El sistema modifica en la BD el nuevo estado del producto				
10	Termina crear					

Precondiciones: El actor estará monitoreando el sistema.

Poscondiciones: Se tendrá un mensaje de advertencia.



Proyecto: Sistema para el monitoreo de medicamentos de una organización farmacéutica.

Documento: CU F05 Crear reporte de producto.

Fecha de creación: 2005.

Autor: Xavier Augusto Olivares Chávez – [xav\\_16050@yahoo.com](mailto:xav_16050@yahoo.com) © 2005 Reservado todos los derechos.

<b>CU F05 Crear reporte de producto.</b>	Inicia: A reserva del componente del sistema. Termina: termina crear reporte del producto.
Descripción: Permite la creación de un reporte de los productos con bajas existencias en el sistema.	
Actores: componente del sistema, supervisor general, operador de compra, operador de venta.	

n	Flujo principal de eventos ACTOR	Flujo Principal	Flujos Alternos	Excepciones
1	El actor componente llama al componente consulta del sistema con los parámetros (producto)	El sistema consulta la BD y devuelve la información		
2	El actor recibe la información y llena un formulario			
3	El actor llama al componente historial del sistema y pasa la información del formulario	El sistema valida los datos y registra la fecha de levantamiento de reporte		
4		El sistema escribe en la BD		
5	Termina crear reporte de producto			

Precondiciones: El actor terminara el monitoreo del sistema.

Poscondiciones: Se tendrá un reporte de los artículos de baja existencia.



### 4.2. Diagrama de clases

Después de escribir los casos de uso se tiene material suficiente para empezar con el desarrollo de las clases que van a conformar al sistema, estos objetos son creados a partir de los casos de usos antes descritos.

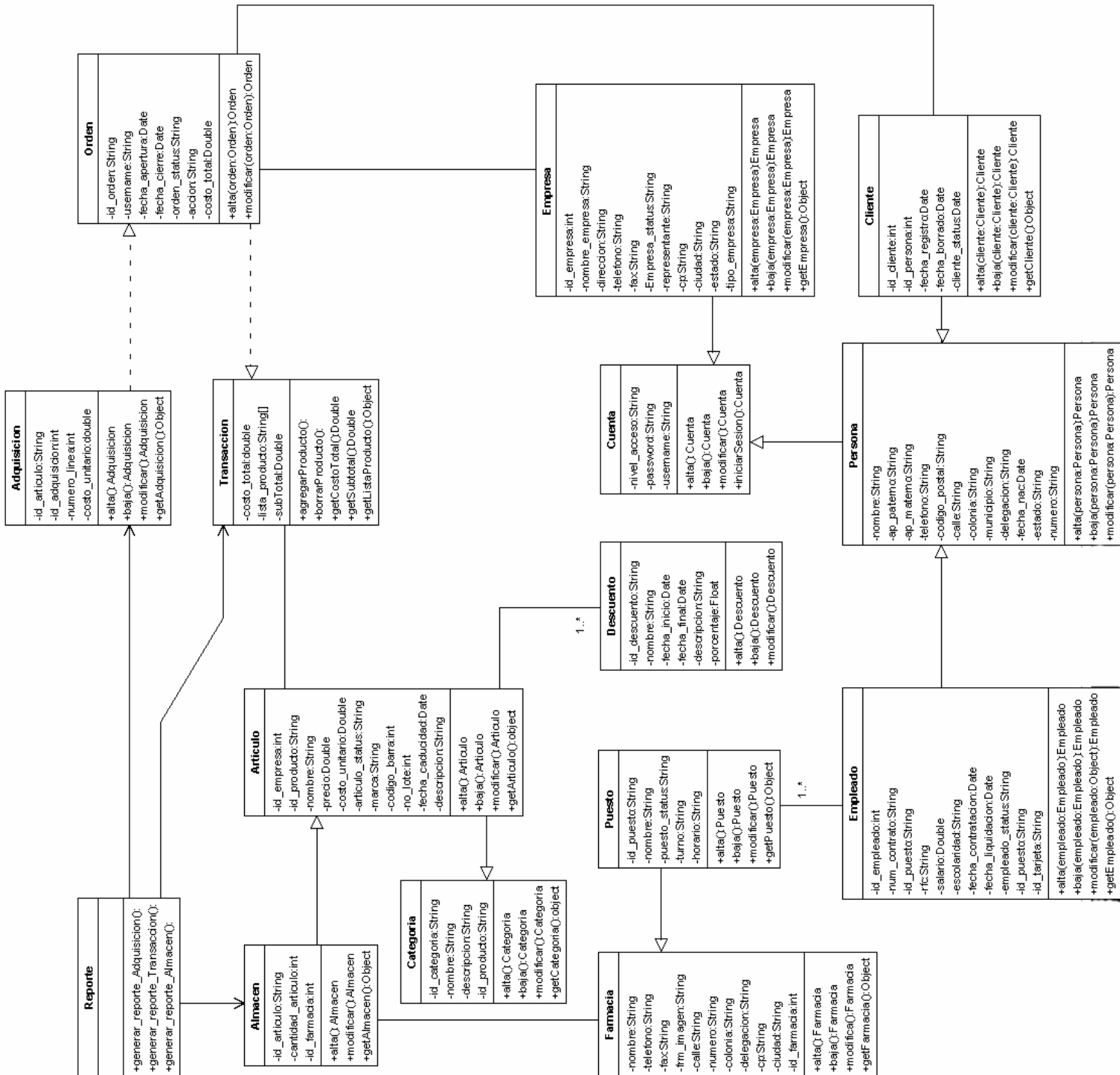


Figura 4.1 Diagrama de clases



### 4.3. Diagramas de secuencia

Después de determinar los objetos que estarán dentro del sistema se pueden realizar los diagramas de secuencia. Ellos nos mostraran como interactúan los objetos entre si. En las siguientes figuras se muestran los diagramas de secuencia.

Este diagrama representa la secuencia que se realiza para lograr una compra de producto.

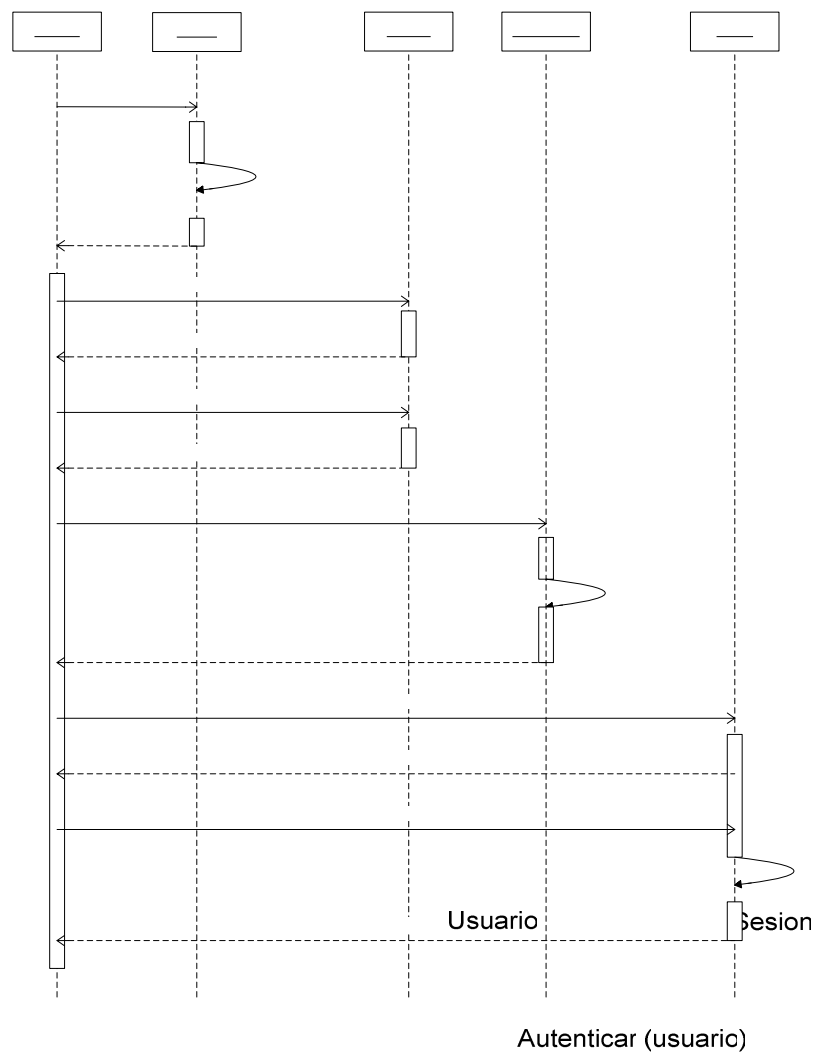


Figura 4.2 Diagrama de secuencia de Compra

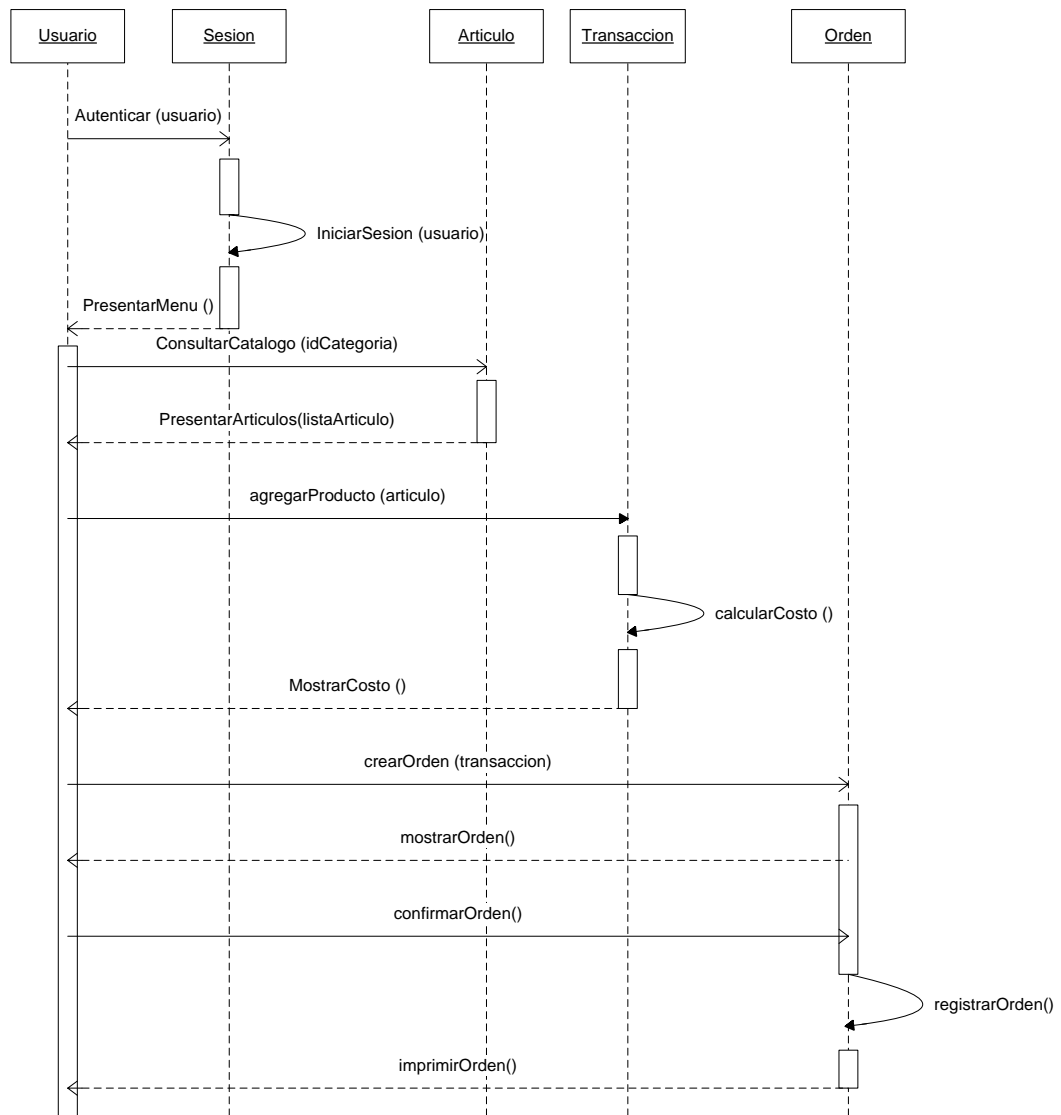
IniciarSesion(usuario)

PresentarMenu()





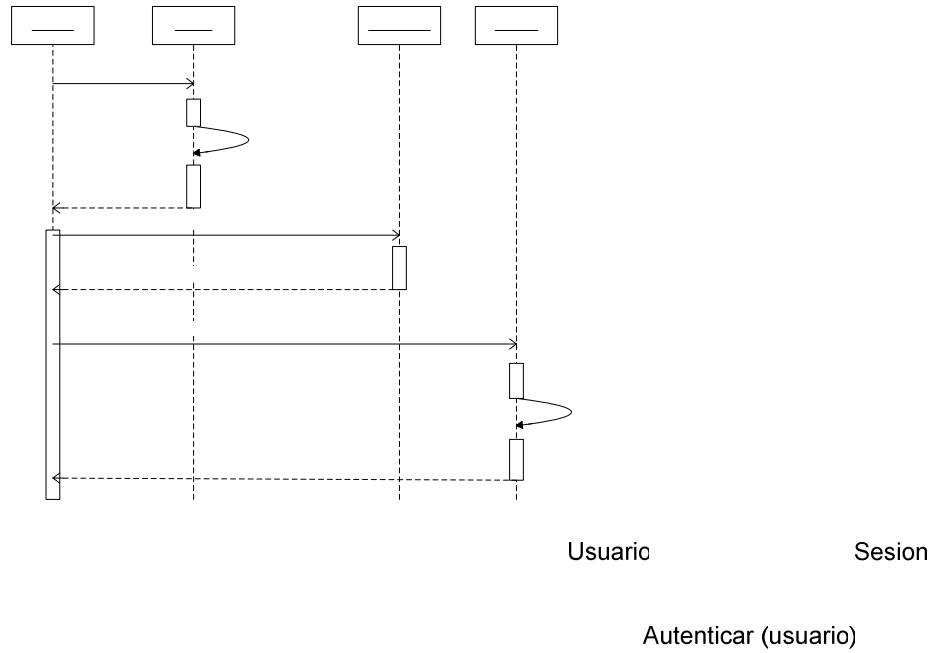
El siguiente diagrama de secuencia es el de Venta.



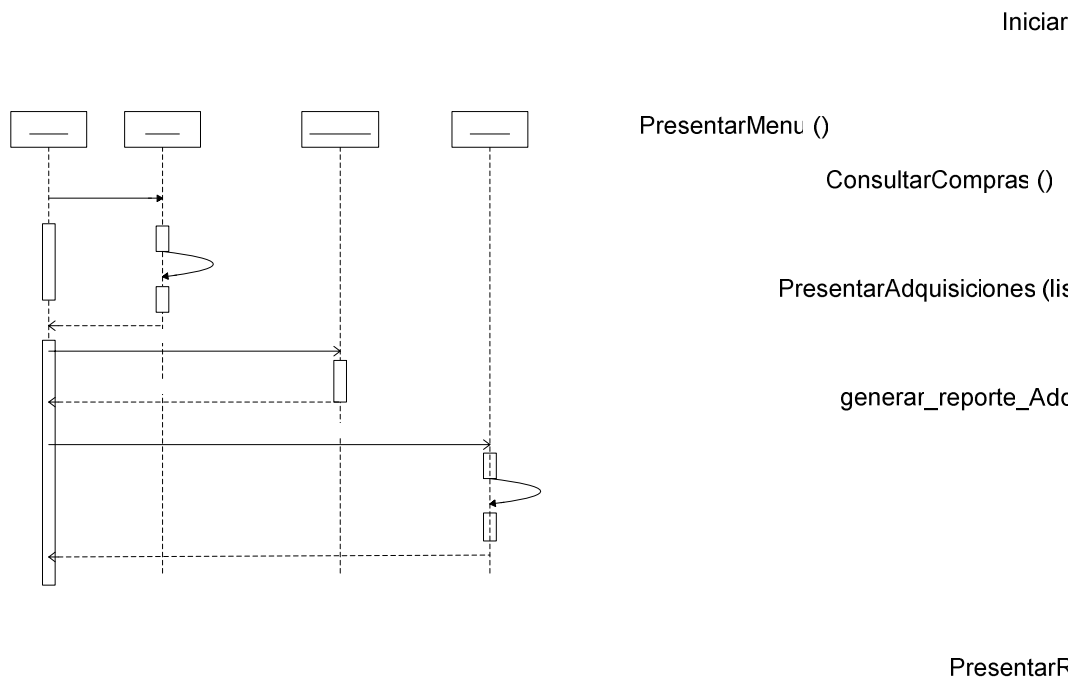
**Figura 4.3 Diagrama de secuencia de Venta**



Las siguientes figuras muestran los diagramas de secuencia de Reporte de compra y Reporte venta.



**Figura 4.4 Diagrama de secuencia de Reporte de compra**



**Figura 4.5 Diagrama de secuencia de Reporte de venta**



El Reporte de almacén tiene el siguiente diagrama de secuencia.

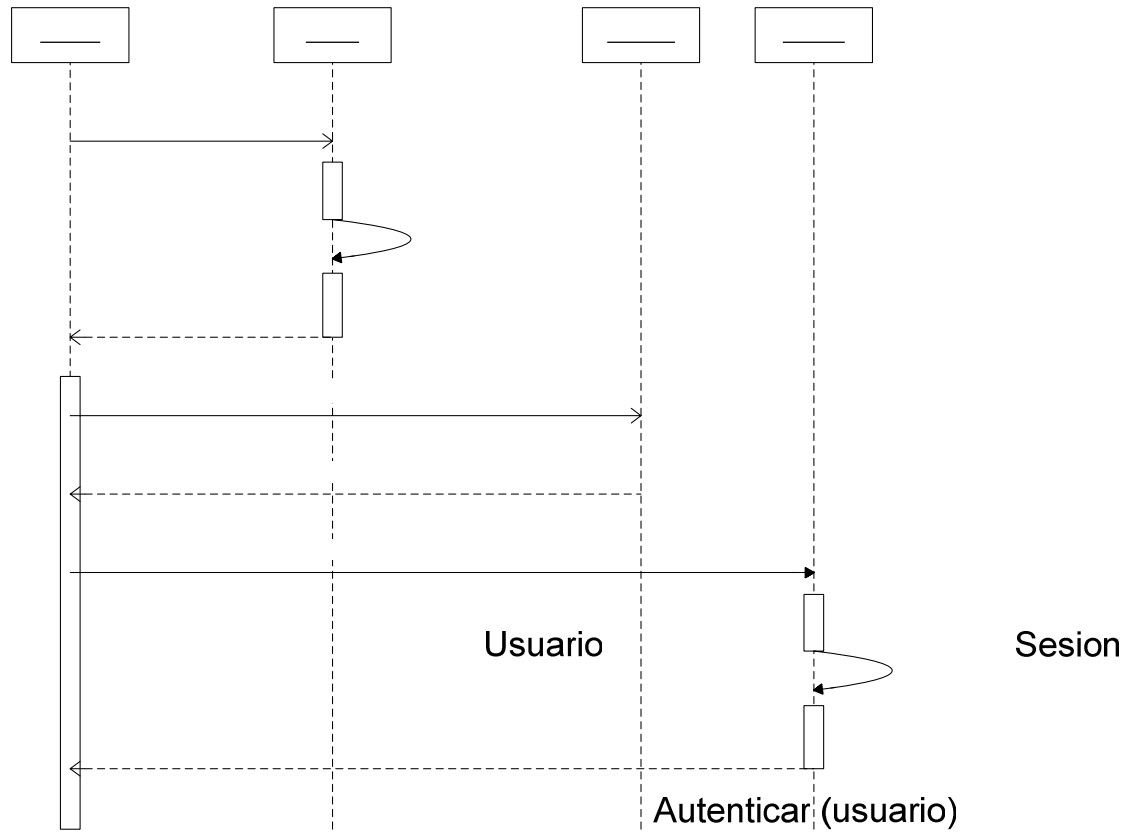


Figura 4.6 Diagrama de secuencia Reporte almacén

PresentarMenu ()

ConsultarAlma

PresentarAlmace

generar\_rep



#### 4.4. Arquitectura y diseño

Documentos relacionados:

- Diseño de la Base de datos
- Mapa de navegación
- Políticas de diseño

El sistema que se realizará para el grupo Morelos tiene una plataforma específica en la cual trabajará. Se tienen dos clasificaciones, la primera, que es donde se desempeñara el sistema y la segunda, que es donde se accederá al sistema.

El sistema trabajará basándose en la arquitectura de red cliente/servidor, este sistema en Web utiliza la metodología de “tres capas”. Con esto se pretende separar los tres componentes principales de un sistema en Web.

Por ello las características de un equipo que se conectará al sistema mediante la red Internet necesita la siguiente plataforma de software:

1 sistema operativo: Windows 9x, ME, XP, NT; Linux; Unix; MAC OS.

1 navegadores de red: Internet Explorer, Netscape, Mozilla, Opera, Copernic, Konqueror.

1 Java Virtual Machine.

1 conexión a red Internet.

Se debe tener en cuenta que el sistema deberá correr en diversos sistemas operativos con la ayuda de la máquina virtual de Java. Sin embargo, existe la posibilidad que en cada navegador funcione diferente la interfaz. Por ello es importante recalcar el uso de la JVM y considerar el funcionamiento de cada navegador.

Con respecto a la maquina destinada a nuestro sistema deberá considerarse los siguientes aspectos: Primero, en que plataforma se va a trabajar, instalación y configuración del software que requiere el sistema; Segundo, que arquitectura o



metodología se esta utilizando para resolver el problema; Y tercero, que objetivos nos da la arquitectura antes mencionada.

Los aspectos más importantes de la arquitectura del sistema son:

Requiere como servidor Web Apache 2.0.

Intérprete de Java.

Kit de desarrollo de Java (JSDK 1.4.2)

Jakarta Tomcat 4.X como mínimo.

DBMS DB2 IBM.

Instalar y configurar drivers para acceso a BD DB2 UDB versión 8.X

Para desarrollar el código se utilizara Eclipse 3.

Para desarrollar las herramientas UML Poseidon CE 2.5

Para desarrollar el diagrama entidad-relación DBDesigner 4.

La metodología de desarrollo de software, como ya se ha mencionado, será con un esquema de tres capas.

Cliente. Estará encargada de la interacción en el browser.

Datos. Acceso a la BD relacional para objetos persistentes.

Reglas de negocio. Todos los componentes de software.

Estrictamente hablando, las reglas de negocios, se realizan a través de los servidores de aplicaciones. Un servidor de aplicaciones es muy complejo y difícil de configurar, por ello, la parte de reglas de negocio, de nuestro esquema del sistema, haremos uso del Web Container.

Se utilizará el siguiente patrón de diseño: Modelo Vista Controlador (MVC). Utilizando la infraestructura y el conjunto de clases que nos da Struts obtendremos más rapidez de desarrollo de nuestro sistema y nos da la ventaja de reutilizar el código.

Para el control de sesiones, esto es, para que el sistema pueda mantener el concepto de sesión, se utilizara IBATIS. El Web por naturaleza es un *state-less*, es decir, al



navegador no le interesa mantener una conexión permanente. Con la combinación de Struts e IBATIS el sistema logrará crear un contexto de aplicación.

Este tipo de arquitectura busca los siguientes objetivos:

- Liberar a las máquinas clientes de la mayor parte de los procesos que lleva acabo el sistema.
- Lograr una mayor usabilidad, que los usuarios se sientan cómodos en la interfaz del sistema.
- Facilidad de mantenimiento y actualización del sistema.
- Integridad de la información de la empresa, evitando el contacto directo entre el usuario y la base de datos.
- Seguridad, control de acceso al sistema a través de niveles de seguridad.
- El sistema será quien interactúe con la base de datos, protegiendo los datos.
- Lograr la reutilización de código.
- Aumentar la productividad del sistema.

#### **4.4.1. Componentes**

A continuación se listan los componentes de la aplicación:

Componentes de presentación.

- Gestión de control de acceso.
- Generación de usuario/contraseña para público en general.
- Presentación de consultas de productos.
- Presentación de consultas empleados, ventas y compras para la empresa.
- Generación de cotizaciones.
- Consultar perfil de público en general y algunos empleados.
- Generación de usuario/contraseña para empleados de la empresa.
- Presentación del estado de la compra o venta.



#### Componentes del negocio.

- Buscador de artículos.
- Monitoreo de artículos.
- Buscador de empleados.
- Buscador de ventas.
- Buscador de compras.
- Buscador de perfil.
- Buscador de filial.
- Distribuidor de los artículos.

#### Componentes de datos.

- Recuperar contraseña del usuario.
- Crear compra.
- Crear venta.
- Crear catalogo.
- Crear producto.
- Crear artículo.
- Crear filial.
- Crear almacén.
- Crear acceso al usuario.
- Actualizar artículo.
- Actualizar catalogo.
- Actualizar producto.
- Actualizar filial.
- Actualizar almacén.
- Crear registro de sesión.
- Listar los artículos de las filiales.



#### 4.4.2. Modelo de interfase

A continuación, en la *Tabla 3* se muestra las interacciones más comunes del sistema, el menú principal de todo el sistema.

Interacción	Propósito	Contenido/Respuesta/Comportamiento
Página principal	Da las diferentes opciones que tiene un usuario dentro del sitio.	
<ul style="list-style-type: none"><li>• Productos</li></ul>	Despliega las diferentes opciones de búsqueda de artículos que maneja la empresa	<ul style="list-style-type: none"><li>a) Catálogos.</li><li>b) Búsqueda por promociones.</li><li>c) Productos nuevos.</li><li>d) Búsqueda avanzada.</li></ul>
<ul style="list-style-type: none"><li>• Cotización</li></ul>	Despliega las dos formas de obtener una cotización de la empresa.	<ul style="list-style-type: none"><li>a) Cotización asíncrona</li><li>b) Cotización.</li></ul>
<ul style="list-style-type: none"><li>• Quienes somos</li></ul>	Despliega las diferentes opciones que describen a la empresa.	<ul style="list-style-type: none"><li>a) Historia de la organización.</li><li>b) Filosofía y camino.</li><li>c) Servicios.</li><li>d) Directorio.</li></ul>
<ul style="list-style-type: none"><li>• Nuestras farmacias</li></ul>	Despliega el menú que contiene la descripción de cada una de las filiales del grupo Morelos.	<ul style="list-style-type: none"><li>a) Farmacia Morelos 1.</li><li>b) Farmacia Morelos 2.</li><li>c) Farmacia Morelos 3.</li><li>d) Farmacia Franquicias.</li></ul>
<ul style="list-style-type: none"><li>• Contactar</li></ul>	Despliega el menú de las maneras en que el sitio se comunica con los navegantes.	<ul style="list-style-type: none"><li>a) Formulario contactar.</li><li>b) Foro.</li></ul>





<ul style="list-style-type: none"> <li>SFM</li> </ul>	<p>Despliega el menú de el Sistema de la Farmacia Morelos, de acuerdo al nivel de acceso tendrá más comandos que usar.</p>	<ul style="list-style-type: none"> <li>a) Crear.</li> <li>b) Actualizar.</li> <li>c) Eliminar.</li> <li>d) Venta.</li> <li>e) Compra.</li> <li>f) Productos.</li> </ul>
<ul style="list-style-type: none"> <li>Iniciar sesión</li> </ul>	<p>Dar acceso a usuarios validos que deseen usar el sistema, así como registrar a usuarios nuevos o recordar contraseñas.</p>	<ul style="list-style-type: none"> <li>a) Iniciar sesión.</li> <li>b) Alta clave.</li> <li>c) Recordar clave.</li> </ul>

**Tabla 5**

A continuación se muestra el bosquejo de cómo quedaría la algunas pantallas. Se ha anexado a este documento el mapa de navegación, este mapa le ayudará a ver como es la secuencia de pantallas dentro del sistema en Web.



Sabias que ...  
 La oferta del mes.  
 Lo que mas se busca.  
 Consulta a nuestros medicos.  
 Animate a contactarnos.

*Bienvenido a la Pagina de La Farmacia M.*

[Productos.](#) | 
 [Cotización.](#) | 
 [Consulta Farmaceutica.](#) | 
 [¿Quienes Somos?.](#) | 
 [Nuestras Farmacias.](#) | 
 [Contactar.](#)

Si deseas aportar o comentar algo mandanos un correo.  
[webmaster@fmorelos.com.mx](mailto:webmaster@fmorelos.com.mx)

Copyright © 2005 CenterX Inc. Todos los derechos reservados.  
 Pagina elaborada por dicha empresa.

**Figura 4.7 Pagina principal Web**

haga clic aquí para acceder a ella'. The registration form contains the following fields: 'Nombre' (Adrian), 'Apellido Paterno' (Hernandez), 'Apellido Materno' (Lopez), 'Número de registro' (adriani1957, with a note 'mínimo 10 caracteres'), 'Elija una contraseña' (password field), 'Vuelva a introducir la contraseña' (confirm password field), 'Dirección de correo electrónico' (adri\_1957@hotmail.com), 'Dirección' (Av. Juarez), 'Cobada' (Bo. San pedro), 'CP' (empty), 'Delegación' (Xochimilco), 'Estado o Ciudad' (DF), and 'Teléfono' (23456712). An 'Aceptar' button is at the bottom of the form. Below the form, there is a note: 'Debe llenar todos los campos para que sea aceptada su solicitud.' and a navigation bar with links for 'Productos', 'Cotización', 'Consulta Farmacéutica', '¿Quiénes Somos?', 'Nuestras Farmacias', and 'Contactar'. At the bottom, there is a footer with contact information: 'Si deseas aportar o comentar algo mandanos un correo: [webmaster@farmorelos.com.mx](mailto:webmaster@farmorelos.com.mx)', 'Copyright © 2005 Centex Inc. Todos los derechos reservados.', and 'Página elaborada por dicha empresa.'"/>

Figura 4.8 Página alta cliente

webmaster@farmorelos.com.mx', 'Copyright © 2005 Centex Inc. Todos los derechos reservados.', and 'Página elaborada por dicha empresa.'"/>

Figura 4.9 Página nuestras farmacias



The screenshot shows the product page for Buscapina. At the top, there is a navigation bar with the Farmacia Morelos logo and a search bar containing 'buscapina'. Below the navigation bar, there are several menu items: 'Productos', 'Cotización', 'Consulta Farmacéutica', '¿Quiénes Somos?', 'Nuestras Farmacias', and 'Contactar'. The main content area features a search bar with 'buscapina' entered and a list of search results. The results include 'Buscapina' (Ciprolax) and 'Aspirina Protect'. Below the search results, there is a navigation bar with the same menu items as above. At the bottom, there is a copyright notice: 'Copyright © 2005 Centrix Inc. Todos los derechos reservados. Página elaborada por dicha empresa.'

Figura 4.10 Página productos

The screenshot shows the login page of the Farmacia Morelos website. At the top, there is a navigation bar with the Farmacia Morelos logo and a search bar. Below the navigation bar, there are several menu items: 'Productos', 'Cotización', 'Consulta Farmacéutica', '¿Quiénes Somos?', 'Nuestras Farmacias', and 'Contactar'. The main content area features a login form with the following fields: 'Login ID:' and 'Contraseña:'. Below the login form, there is a button labeled 'Iniciar sesión'. To the left of the login form, there is a section titled 'La farmacia morelos pone a tu servicio:' with a list of services. Below the login form, there is a link labeled 'Ayuda para iniciar sesión. Olvidaste tu contraseña.' At the bottom, there is a copyright notice: 'Copyright © 2005 Centrix Inc. Todos los derechos reservados. Página elaborada por dicha empresa.'

Figura 4.11 Página de login



The screenshot shows the Farmacia Morelos website interface. At the top, there is a navigation bar with a search box, the logo 'FARMACIA MORELOS S.F.M.', and links for 'Iniciar Sesión', 'Productos', 'Cotización', 'Consulta Farmaceutica', '¿Quiénes Somos?', 'Nuestras Farmacias', and 'Contactar'. Below the navigation bar, there is a section titled 'Farmacia Morelos. Productos' with a 'Sabías que ...' and 'Ayuda:' section. The main content area features an alphabetical index from A to Z. Under the letter 'A', a list of active substances is displayed, each with a link to its product page. The list includes: AB, AB ANTITUSIVO, ABACTEN, ABACTEN FORTE, ABALGIN Gotas, ABALGIN Supositorio Adulto-Supositorio Infantil, ABECIDIN A C D, ABILIFY, ABINOL, ABIOL EX, ABILILAR, ACAMOL, ACAMOL COMPUUESTO, ACANTEX, ACASUMUL, ACCOLATE, ACCU-CHEK ACTIVE GLUCOSE, ACCU-CHEK ACTIVE KIT, ACCU-CHEK GO CINTAS REACTIVAS, ACCU-CHEK GO KIT, ACCU-CHEK MULTICLIX, ACCU-CHEK MULTICLIX LANCETAS, ACCU-CHEK SENSOR COMFORT, ACCU-CHEK SENSOR KIT, and ACCU-CHEK SOFTCLIX. To the right of this list, the corresponding laboratory names are listed: SAVAL, ANDROMACO, CHILE, PASTEUR, BRISTOL, RECALCINE, ANDROMACO, PHARMA INVESTI, VOLTA, ROCHE, PHARMA INVESTI, ASTRA ZENECA, and ROCHE. Below the list, there is a 'Mas resultados ...' section with a 'Siguiente' link. At the bottom of the page, there is a copyright notice: 'Copyright © 2005 CenterX Inc. Todos los derechos reservados. Pagina elaborada por dicha empresa.'

Figura 4.12 Página de productos por lista alfabética

The screenshot shows the detailed product page for ACANTEX on the Farmacia Morelos website. The page layout is consistent with the previous screenshot, but the main content area is filled with detailed information about the product. At the top, the product name 'ACANTEX' is prominently displayed. Below it, there is a section for 'INDICACIONES' (Indications), which lists various conditions such as 'Cefaleas', 'Migrañas', 'Dolor de cabeza', 'Dolor de dientes', 'Dolor de nervios', 'Dolor de espalda', 'Dolor de articulaciones', 'Dolor de estómago', 'Dolor de garganta', 'Dolor de ojos', 'Dolor de oídos', 'Dolor de nariz', 'Dolor de senos paranasales', 'Dolor de cuello', 'Dolor de hombros', 'Dolor de codos', 'Dolor de muñecas', 'Dolor de manos', 'Dolor de pies', 'Dolor de tobillos', 'Dolor de rodillas', 'Dolor de caderas', 'Dolor de codos', 'Dolor de hombros', 'Dolor de codos', 'Dolor de muñecas', 'Dolor de manos', 'Dolor de pies', 'Dolor de tobillos', 'Dolor de rodillas', 'Dolor de caderas'. The page also includes sections for 'Contraindicaciones', 'Precauciones', 'Efectos secundarios', 'Interacciones', 'Condiciones de almacenamiento', 'Fecha de caducidad', 'Fecha de elaboración', 'Lote', 'Número de registro', 'Número de autorización', 'Número de inscripción', 'Número de licencia', 'Número de registro', 'Número de autorización', 'Número de inscripción', 'Número de licencia'. At the bottom of the page, there is a copyright notice: 'Copyright © 2005 CenterX Inc. Todos los derechos reservados. Pagina elaborada por dicha empresa.'

Figura 4.13 Página descripción de un producto



Nombre:

ID artículo:  ID empresa:  ID descuento:

ID producto:

Precio:  Costo Unitario:  Estado:

Código de barra:  Num. Lote:  Fecha Caducidad:

Descripción 1:

Descripción 2:

Descripción 3:

Descripción 4:

Crear en Base de Datos Limpiar Cancelar

CREAR LIMPIAR CANCELAR

Productos Cotización Consulta Farmacéutica ¿Quiénes Somos? Nuestras Farmacias Contactar

Si deseas aportar o comentar algo mandanos un correo. [webmaster@farmos.com.mx](mailto:webmaster@farmos.com.mx)

Copyright © 2005 Consoft Inc. Todos los derechos reservados. Página elaborada por dicha empresa.

Figura 4.14 Página crear artículo

Nombre:

Correo e-mail:

Lista de Productos:

Subir archivos:  Examinar

Enviar Formularios:  Enviar

Productos Cotización Consulta Farmacéutica ¿Quiénes Somos? Nuestras Farmacias Contactar

Si deseas aportar o comentar algo mandanos un correo. [webmaster@farmos.com.mx](mailto:webmaster@farmos.com.mx)

Copyright © 2005 Consoft Inc. Todos los derechos reservados. Página elaborada por dicha empresa.

Figura 4.15 Página de cotizaciones



#### 4.4.3. Persistencia

El diseño de la base de datos se ha basado en el modelo entidad - relación, esta metodología nos permite representar la información en tablas, cada tabla define una entidad única que tiene atributos. Cada registro es una tupla, es un conjunto de atributos, que nos da información con la cual se pueden tomar decisiones.

Cada entidad se encuentra relacionada con otras entidades reforzando la integridad de datos, sin embargo hay que tomar en cuenta que se debe tener cuidado al establecer dichas relaciones. En el siguiente análisis para el diseño de la base de datos que requiere el sistema se procuró evitar redundar en la información.

El sistema y el administrador de la base de datos serán los únicos responsable de modificar la información de la base de datos. El sistema, mediante los niveles de acceso, buscará limitar las acciones de cada empleado hacia la base de datos.

El usuario nunca se registrará o abrirá alguna sesión directa con el manejador de la base de datos, el sistema será el encargado de abrir y mantener una sesión bien definida con el manejador de la base de datos.

El cliente que navega por la red solo podrá acceder a la base de datos a través de la aplicación, y esta limitado a solo obtener información de los productos, servicios y empleados que radican dentro de la empresa grupo Morelos.

El operador de mostrador, empleado que tiene presencia en cada filial de la empresa, encargado de la venta de los productos a los clientes accederá mediante el sistema a la base de datos. Este personaje esta limitado a realizar consultas de los productos y realizar las ventas de los mismos, también tiene la facultad de cancelar una orden de venta.

El operador de compra, empleado que se encuentra en el centro de operaciones del grupo Morelos, encargado de la compra de los productos a los proveedores de la empresa accederá a la base de datos mediante el sistema. Estará limitado a las acciones



de consulta de producto, actualizar mediante la compra la existencia de productos, de modificar productos y cancelar ordenes de compra.

El supervisor general encargado del manejo de todos los elementos del sistema, accederá a la base de datos mediante la aplicación. Sus acciones son crear: filial, empleados, proveedores, productos y catálogos; actualizar filial, empleados, proveedores, productos y catálogos; eliminar filial, empleados, proveedores, productos y catálogos.

El supervisor general será el encargado de dirigir a los demás elementos de la empresa que interaccionan con el sistema. El sistema mediante su aplicación de monitoreo le dará aviso de baja de existencia. Por ello se debe considerar al sistema como un actor que puede consultar a la base de datos.

El administrador de la base de datos será el encargado de realizar el mantenimiento de la BD. Planeará en que determinado tiempo se deberá realizar un respaldo de la información. Tendrá contacto directo con el administrador del sistema para revisar los posibles fallos de la aplicación, todo esto con la finalidad de mantener al 100% la integridad de la información.

El resultado del diseño del diagrama entidad – relación para la creación de la base de datos arroja 16 tablas, que son las siguientes:

1. Persona
2. Contrato
3. Empleado
4. Puesto
5. Farmacia
6. Cliente
7. Empresa
8. Cuenta
9. Categoría
10. Producto
11. Artículo



12. Almacén
13. Tarjetapago
14. Descuento
15. Orden
16. Detalle\_orden

El diccionario de datos lo podrá encontrar a partir de la página 136, donde están definidos los atributos de cada entidad, el diagrama entidad – relación lógico, donde se muestran las relaciones de cada entidad y, el diagrama entidad – relación físico.

#### **4.4.4. Diseño de la Base de Datos**

En este documento encontrará el análisis y desarrollo de la base de datos que consideramos mas apropiada para el manejo de los datos de la empresa. Debemos mencionar que aquí no se hace la recomendación de que manejador de base de datos es la apropiada.

Encontrará en primer lugar la documentación que corresponde al diagrama entidad-relación, esta será presentada de manera lógica. Con este documento podrá distinguir las diferentes entidades necesarias, la relación que guardan entre ellas, y los atributos de cada una de ellas.

La siguiente referencia será el diagrama entidad – relación, su formato será de manera física. Con esta información tendrá una introducción global de las propiedades de cada atributo perteneciente a una entidad.





Por ultimo presentamos a detalle cada tabla que conforman a la base de datos, este documento es el Diccionario de datos. Cada tabla es una entidad, cada entidad posee atributos que la hacen única, cada entidad contiene una llave primaria y puede tener una o más llaves foráneas.

Cada atributo de las entidades tienen propiedades, estas son: nombre del campo, tipo de datos, tipo clave, restricciones de modelo, integridad referencial y en que parte del sistema se utiliza. Dependiendo del tipo de datos se desplegarán más propiedades de cada atributo.

Se tratara de abarcar cada entidad en una sola hoja, si esta entidad tuviera tantos atributos que sobrepasara el formato de una hoja se continuara en la siguiente. La notación para cada tabla es la siguiente:

Nombre Tabla: cliente\_1                      Proyecto: Sistema de Monitoreo.

Al observar que el nombre de la tabla contiene un guión sabrá que a esta tabla le corresponden más de una hoja, el número seguido del guión le indicara en que hoja de la tabla se encuentra. La información se le presentará mediante una tabla, esto para facilitar su lectura.



### Diagrama Base de Datos

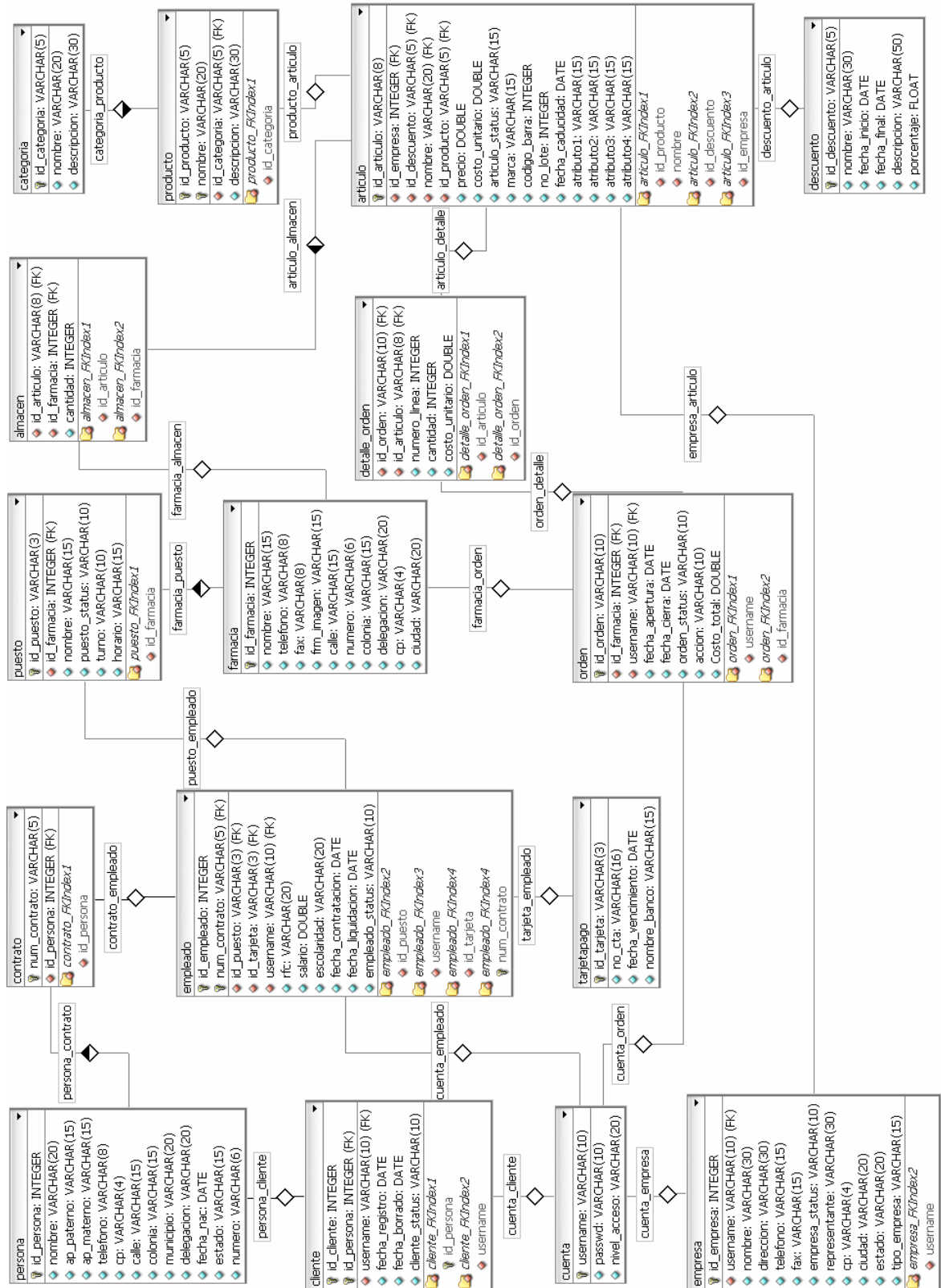


Figura 4-16



Nombre Tabla: <b>persona</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	Id_persona	primaria	integer		si	no		
2	Nombre		varchar	20		no		
3	Ap_patermo		varchar	15		no		
4	Ap_materno		varchar	15		no		
5	Telefono		varchar	8		si		
6	Cp		varchar	4		si		
7	Calle		varchar	15		si		
8	Colonia		varchar	15		si		
9	Municipio		varchar	20		si		
10	delegación		varchar	20		si		
11	estado		varchar	15		no		
12	fecha_nac		Date			si		
13	numero		varchar	6		si		

**Tabla 6**

La tabla **persona**, *Tabla 6*, es donde se encuentran los datos generales de un usuario. La persona puede ser un cliente o un empleado de la organización, por ello guarda relación con las tablas cliente y mediante la tabla contrato con la del empleado.

Mediante las reglas de negocio, escritas en el código de los componentes, se harán obligatorios algunos atributos. Esto dependerá del componente de registro, ya sea para un cliente o para un empleado.

Nombre Tabla: <b>contrato</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
N	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	num_contrato	primaria	varchar	5	si	no		
2	id_persona	foránea	integer			no		

**Tabla 7**

La tabla **contrato**, *Tabla 7*, es la que relaciona a persona con empleado. Una persona puede tener más de un contrato en la organización, pero el contrato solo dará por resultado a un empleado.



Nombre Tabla: <b>cliente</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
N	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_cliente	primaria	integer		si	no		
2	id_persona	primaria	integer			no		
3	username	foránea	varchar	10		no		
4	fecha_registro		Date			no		
5	fecha_borrado		Date			si		
6	cliente_status		varchar	10		no	Activo	

**Tabla 8**

La tabla **cliente**, *Tabla 8*, guarda los datos de un usuario registrado de Internet como cliente. Así la tabla persona en conjunto con cliente forman a un usuario registrado de Internet.

Nombre Tabla: <b>cuenta</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	Username	primaria	varchar	10	no	no		
2	Passwd		varchar	10		no		
3	nivel_acceso		varchar	20		no	Primer nivel	

**Tabla 9**

La tabla **cuenta**, *Tabla 9*, guarda los datos de todos los usuarios que pueden tener acceso al sistema. Siendo el primer nivel el acceso básico, este nivel lo obtienen todos los clientes. El encargado del username y el passwd de un empleado de la organización estarán a cargo del administrador de cuentas.

Nombre Tabla: <b>tarjetapago</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_tarjeta	primaria	varchar	3	si	no		
2	no_cta		varchar	16		no		
3	fecha_vencimiento		Date			no		
4	nombre_banco		varchar	15		no		

**Tabla 10**



La tabla **tarjetapago**, *Tabla 10*, es donde se guarda la información de la tarjeta bancaria de un empleado. La organización paga el salario de sus empleados a través de la correspondiente cuenta de banco de cada empleado.

Nombre Tabla: <b>empleado</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_empleado	primaria	integer		si	no		
2	num_contrato	primaria(fk)	varchar	5		no		
3	id_puesto	foránea	varchar	3		no		
4	id_tarjeta	foránea	varchar	3		no		
5	username	foránea	varchar	10		no		
6	rfc		varchar	18		no		
7	salario		double			no		
8	escolaridad		varchar	20		no		
9	fecha_contratacion		Date			no		
10	fecha_liquidacion		Date			si		
11	empleado_status		varchar	10		no	Activo	

**Tabla 11**

La tabla **empleado**, *Tabla 11*, es la que guarda la información de la persona que ha sido contratada por la organización. Se relaciona con las siguientes tablas: contrato, puesto, tarjetapago y cuenta. Dependiendo del tipo de empleado se le asignará un nivel de acceso al sistema. Se considera a un empleado recién contratado como activo.

Nombre Tabla: <b>empresa</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_empresa	primaria	integer		si	no		
2	username	foránea	varchar	10		no		
3	nombre		varchar	30		no		
4	direccion		varchar	30		no		
5	telefono		varchar	15		no		
6	fax		varchar	15		si		
7	empresa_status		varchar	10		si	Activo	
8	representante		varchar	30		no		
9	cp		varchar	4		no		
10	ciudad		varchar	20		si	Distrito Federal	
11	estado		varchar	20		si		
12	tipo_empresa		varchar	15		no	Proveedor	

**Tabla 12**



La tabla **empresa**, *Tabla 12*, define a un cliente como una organización que compra a mayoreo, como también define a la empresa que resulta ser proveedor del grupo Morelos.

La tabla empresa se relaciona con la tabla cuenta, ya que como cliente puede registrarse en el sistema. Y se relaciona con la tabla artículo, representando a la empresa proveedora de ese único producto.

Nombre Tabla: <b>puesto</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_puesto	primaria	varchar	3	si	no		
2	id_farmacia	foránea	integer			no		
3	nombre		varchar	15		no		
4	puesto_status		varchar	10		no	Vacante	
5	turno		varchar	10		no		
6	horario		varchar	15		no		

**Tabla 13**

La tabla **puesto**, *Tabla 13*, relaciona a las tablas empleado y farmacia. Indica el empleado en que farmacia labora, el turno y el horario.

Nombre Tabla: <b>almacen</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_articulo	foránea	varchar	8		no		
2	id_farmacia	foránea	integer			no		
3	cantidad		integer			no		

**Tabla 14**

La tabla **almacen**, *Tabla 14*, es el inventario, las existencias de cada artículo que tiene la empresa. Organizada a través de cada filial, esto es para saber cuantos artículos de un tipo tiene una filial.

Para ello almacen se relaciona con las tablas articulo y farmacia.



Nombre Tabla: <b>farmacia</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_farmacia	primaria	integer		si	no		
2	nombre		varchar	15		no		
3	telefono		varchar	8		no		
4	fax		varchar	15		si		
5	frm_imagen		varchar	15		no		
6	calle		varchar	15		no		
7	numero		varchar	6		no		
8	colonia		varchar	15		no		
9	delegacion		varchar	20		no		
10	cp		varchar	4		no		
11	ciudad		varchar	20		no	Distrito Federal	

**Tabla 15**

La tabla **farmacia**, *Tabla 15*, contiene los datos de cada filial que tiene el grupo Morelos. Se relaciona con las tablas puesto, almacen y orden.

Mediante la tabla almacen se relacionan los artículos con cada una de las filiales.

Nombre Tabla: <b>categoria</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_categoria	primaria	varchar	5	si	no		
2	nombre		varchar	20		no		
3	descripcion		varchar	30		no		

**Tabla 16**

La tabla **categoria**, *Tabla 16*, contiene la información de las clases de productos que maneja la empresa. Con esta información se cataloga un producto.

La tabla se relaciona con la siguiente tabla producto.

Nombre Tabla: <b>producto</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_producto	primaria	varchar	5	si	no		
2	nombre	primaria	varchar	20		no		
3	id_categoria	foránea	varchar	5		no		
4	descripcion		varchar	30				

**Tabla 17**



La tabla **producto**, *Tabla 17*, engloba a los artículos de acuerdo con sus detalles más específicos. Se relaciona con la tabla artículo.

Nombre Tabla: <b>artículo</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_articulo	primaria	varchar	8	si	no		
2	id_empresa	foránea	integer			no		
3	id_descuento	foránea	varchar	5		no		
4	nombre	foránea	varchar	20		no		
5	id_producto	foránea	varchar	5		no		
6	precio		Double			no		
7	costo_unitario		Double			no		
8	articulo_status		varchar	15		no	Activo	
9	marca		varchar	15		no		
10	codigo_barra		integer			no		
11	no_lote		integer					
12	fecha_caducidad		Date			no		
13	atributo1		varchar	15				
14	atributo2		varchar	15				
15	atributo3		varchar	15				
16	atributo4		varchar	15				

**Tabla 18**

La tabla **artículo**, *Tabla 18*, guarda todos los datos de cada uno de los productos que vende la empresa. Se relaciona con las siguientes tablas: almacen, detalleorden, descuento y con empresa.

Se toma en consideración que un artículo puede estar con un descuento.

Nombre Tabla: <b>descuento</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
N	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_descuento	primaria	varchar	5	si	no		
2	nombre		varchar	30		no	No hay descuento	
3	fecha_inicio		Date			no	01/01/2005	
4	fecha_final		Date			no	31/12/9999	
5	descripcion		varchar	50		si	En este producto no hay descuento	
6	porcentaje		Flota			no	.0	

**Tabla 19**





La tabla **descuento**, *Tabla 19*, es donde se maneja la información de las promociones que pueda tener la organización, se relaciona con la tabla articulo para definir en el producto una promoción.

Nombre Tabla: <b>detalle_orden</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
N	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_orden	foránea	varchar	10		no		
2	id_articulo	foránea	varchar	8		no		
3	numero_linea		integer			no		
4	cantidad		integer			no		
5	costo_unitario		Double			no		

**Tabla 20**

La tabla **detalle\_orden**, *Tabla 20*, es donde tiene la información de cada artículo que se vende y en que cantidad, ligando a el articulo a un numero de orden determinado.

Nombre Tabla: <b>orden</b>		Proyecto: Sistema de monitoreo de los medicamentos de un organización farmacéutica.				Autor: Xavier Augusto Olivares Chávez.		Fecha:13/05/2005
n	Nombre campo	Llave	Tipo dato	Longitud	Auto incrementar	Nulo	Valor predeterminado	
1	id_orden	primaría	varchar	10	si	no		
2	id_farmacia	foránea	integer			no		
3	username	foránea	varchar	10		no		
4	fecha_apertura		Date			no		
5	fecha_cierre		Date			si		
6	orden_status		varchar	10		no	Pendiente	
7	accion		varchar	10		no	Venta	
8	costo_total		Double			no		

**Tabla 21**

La tabla **orden**, *Tabla 21*, es donde se guardan todas las órdenes ya sean compras o ventas de artículos con que negocia la empresa grupo Morelos. Se relaciona con la tabla cuenta para establecer un control de quien hace una orden. Y con la tabla farmacia para saber de que farmacia proviene la orden.




### 4.3. Mapa de navegación

Interacción de las diferentes pantallas del sistema.

Este documento integra de manera puntual las interacciones más comunes del sistema. Cada bloque representa una pantalla del sistema. Se vera la estructura de todo el sitio en Web, la interfaz elegida es el browser.

La primera parte constará de los elementos mas generales, así, poco a poco se ira desglosando el mapa de navegación. Se tratará, para la comodidad del lector, que cada sección abarque una hoja. Si una hoja no alcanzará para contener un diagrama se utilizara la siguiente notación.

Si el diagrama continua, la línea que une un bloque a otro continuará al final de la línea se pondrá un símbolo como el siguiente:  , en la siguiente hoja solo tendrá que localizar el numero que corresponde a la línea y seguir con su lectura. Esta táctica se utilizará si el documento llegara a cortarse de arriba hacia abajo o viceversa y de izquierda a derecha o viceversa.

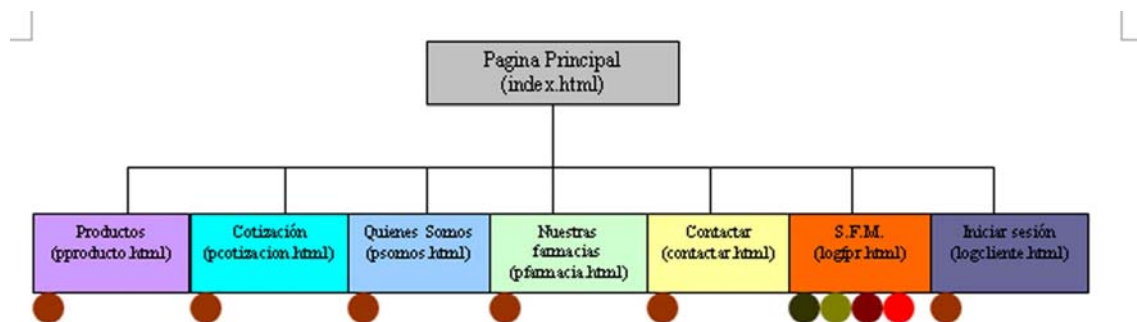
Para cada bloque principal se utilizara un color en especifico, este le permitirá hacer referencias mas rápidas a cualquier pantalla del sistema que le interese. El mapa de navegación empezará con el bloque que tiene por nombre PAGINA PRINCIPAL, cada bloque al igual que éste constara de un nombre representativo y en la parte de abajo se escribirá el nombre físico del archivo, por ejemplo principal.html.

Este documento mostrará los procesos que cada nivel de usuario tiene permitido realizar e identificar las secciones que conforman la interfaz del sistema. Algunos bloques dan como resultado a muchas opciones de pantalla, que todas ellas comparten el mismo funcionamiento.



Para cada sección, de ser necesario, contarán con anexos que explicaran mas a detalle a esos bloques.

- Primer Nivel
- Segundo Nivel.
- Tercer Nivel.
- Cuarto Nivel.
- Quinto Nivel.



**Figura 4.17** Página principal

En la figura 25 se observa que algunos nombres empiezan con la letra “p”, esta letra indica que es la página principal de la sección. En algunos casos como el bloque iniciar sesión no se empleo la palabra por comodidad. Rápidamente damos la descripción de cada bloque.

Productos. Es el área que corresponde a las consultas de todos los productos, se tendrá opciones como búsqueda por catálogos, productos, etc.

Cotización. Es un servicio nuevo de la farmacia, dar cotizaciones de venta al mayoreo y menudeo. Usuarios que estén registrados podrán usarlo.

Quienes somos. Es la descripción de la organización, su historia, su filosofía y quienes la conforman

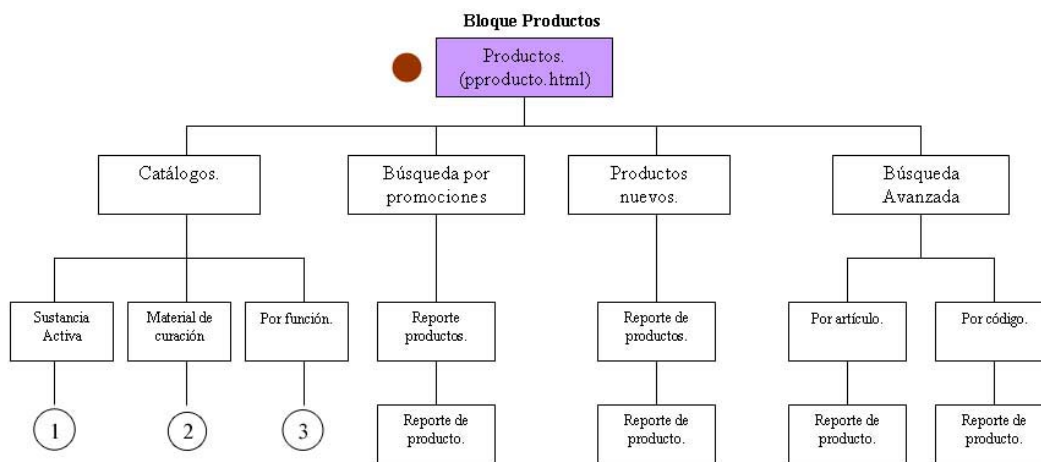
Nuestras farmacias. En este apartado se dan las direcciones físicas y medios de comunicación con cada farmacia y la organización misma.



Contactar. Se da la dirección física de la matriz y medios de comunicación. Así como la posibilidad de opinar acerca del sitio.

S.F.M. Esta área es exclusiva de los empleados de la organización farmacéutica, aquí es donde inician su sesión del sistema todos los operadores.

Iniciar sesión. Se utiliza para tener clientes de Internet registrado, el registro es para dar servicios extras.



**Figura 4.18** Página Bloque productos

Rápidamente damos la descripción de cada bloque que se encuentra en la figura 26. El bloque de catálogos contiene a los productos de la empresa. La búsqueda se puede realizar por sustancia activa, material de curación y función.

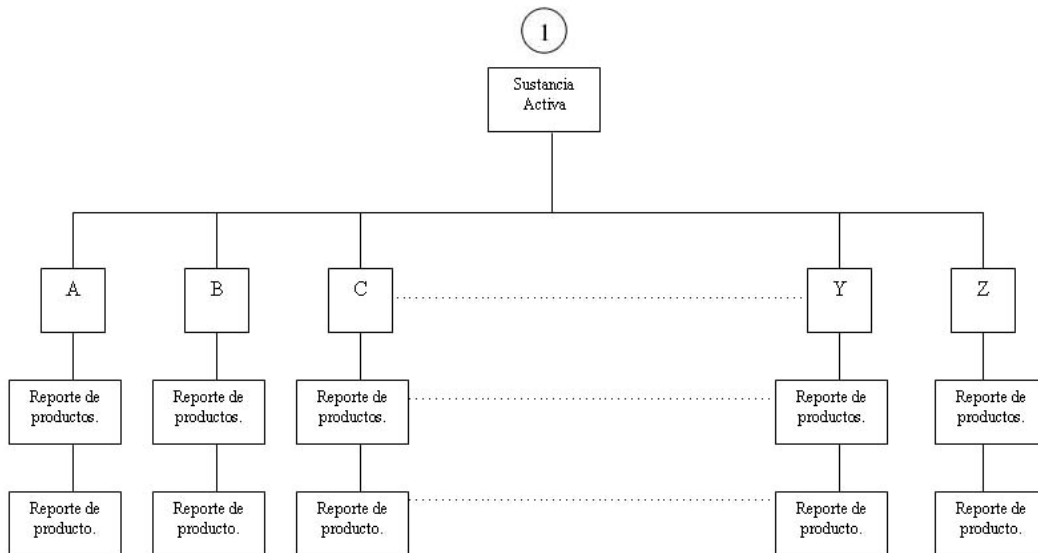
El bloque búsqueda por promociones da por resultado una pantalla con todos los productos que estén en promoción, para elegir a detalle a uno.

El bloque productos nuevos entrega por resultado a los productos que se catalogan como nuevos en la organización, para consultar un producto.

El bloque búsqueda avanzada da la opción de buscar por el código del producto o por artículo.

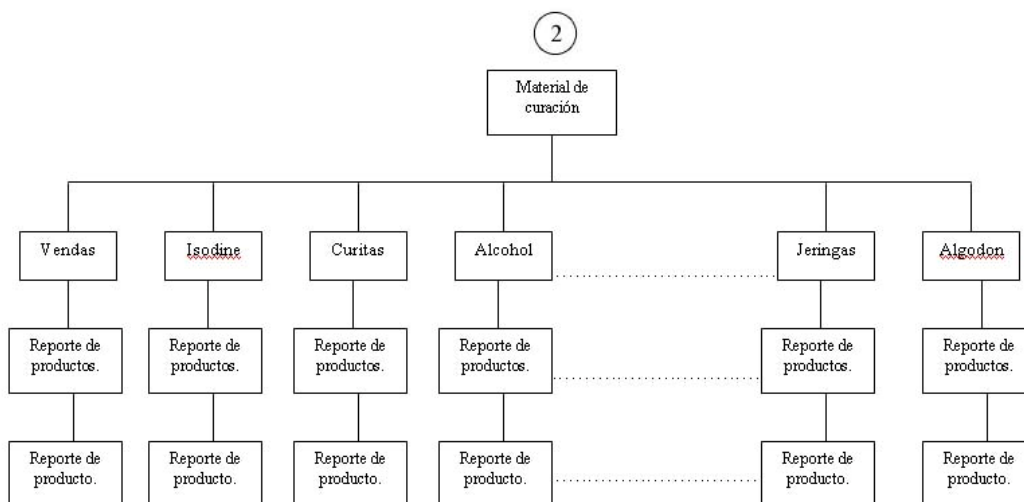


- El bloque por código es una consulta mediante el código único de un producto y entregar un reporte de producto.
- El bloque por artículo es una consulta que requiere el ingreso de varios atributos del producto.



**Figura 4.19** Página Sustancia activa

Rápidamente damos la descripción de cada bloque de la figura 27. El bloque A da por resultado todos los productos que contenga una sustancia activa que su nombre empiece por la letra A, para poder consultar un producto a detalle. Los demás bloques son iguales al bloque A, desde la A – Z.



**Figura 4.20** Página Material de curación

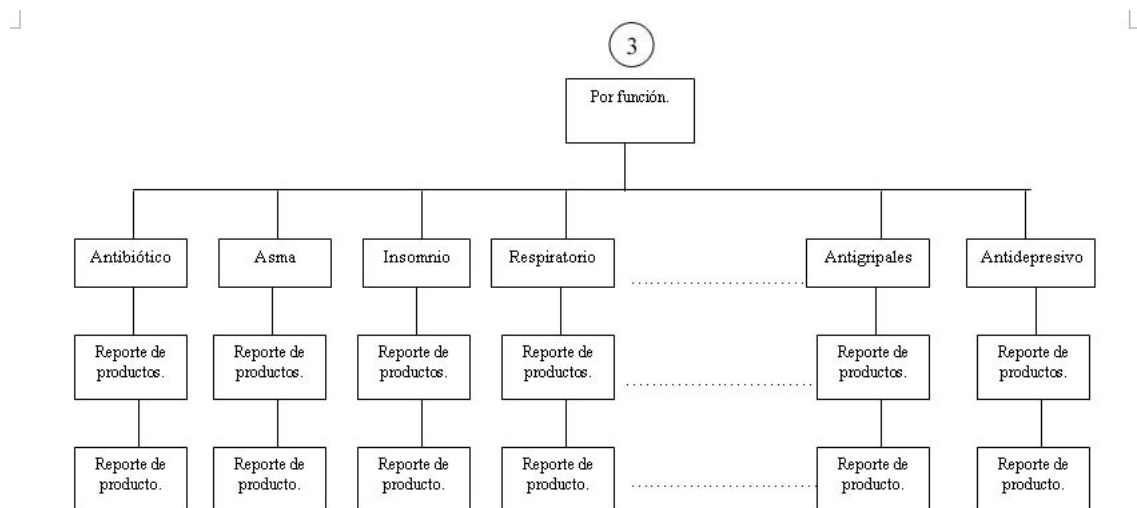


Rápidamente damos la descripción de cada bloque que se encuentra en la figura 28. El bloque de vendas es la pantalla que muestra todos los productos catalogados como vendas, de aquí podrá elegir un producto para ver sus detalles.

El bloque de Isodine es la pantalla que muestra todos los productos catalogados como vendas, de aquí podrá elegir un producto para ver sus detalles.

Los demás bloques funcionan igual que los dos anteriores, dependiendo del nombre del bloque serán los productos que muestre.

**Nota: Ver el Anexo A, pág 148.**



**Figura 4.21 Página por función**

Rápidamente damos la descripción de cada bloque de la figura 29. El bloque de Antibiótico es la pantalla que muestra todos los productos catalogados como vendas, de aquí podrá elegir un producto para ver sus detalles.

El bloque de Asma es la pantalla que muestra todos los productos catalogados como vendas, de aquí podrá elegir un producto para ver sus detalles.

Los demás bloques funcionan igual que los dos anteriores, dependiendo del nombre del bloque serán los productos que muestre.

**Nota: Ver el Anexo B, pág 149.**



## Anexo A

El material de curación en sí no se puede considerar como medicamento, sin embargo, resulta esencial para aplicar muchos de ellos. Es por lo tanto un complemento de varios medicamentos, y resulta esencial manejarlo con el mismo cuidado que los medicamentos.

Así que la empresa Grupo Morelos ha catalogado el material de curación de la siguiente forma:

- Vendas
- Isodine
- Jabon quirurgico
- Gasas
- Micropore
- Abatelenguas.
- Curitas
- Mertiolate.
- Alcohol.
- Algodón.
- Termómetro.
- Jeringas.
- Agua oxigenada.

Cada clasificación representa en el mapa de navegación un bloque, estos bloques se crean dinámicamente. Cada bloque al seleccionarlo da por resultado una lista de productos pertenecientes a la clasificación del bloque, dando la posibilidad de consultar un artículo en específico.



## Anexo B

La clasificación del medicamento por función se refiere a organizar en grupos a los medicamentos que sirvan para combatir un mismo síntoma. La empresa tiene, por experiencia, que la clasificación por función es una herramienta bien vista por los clientes. Cuando un cliente no encuentra el medicamento que busca, tiene la opción de pedir al vendedor que le muestre los medicamentos que combaten la misma enfermedad.

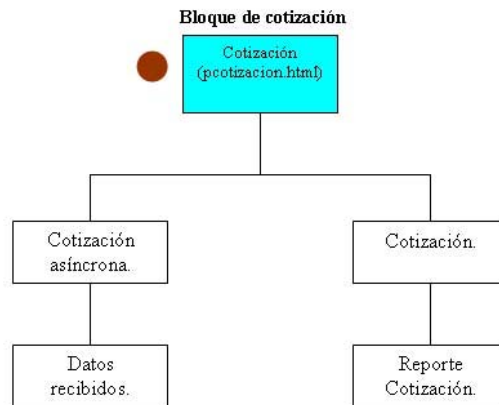
El Grupo Morelos utiliza la siguiente clasificación de medicamentos por función:

- Antigripales
- Antiespasmodicos
- Suplementos
- Antihipersensitivos
- Antidepresivos
- Antiinflamatorios
- Antibioticos
- Hipoglucemiontes
- Cardiológico
- Antipirecticos
- Asma
- Oftalmologicas
- Respiratorios
- Perdida de peso
- Insomnio
- Antihistaminicos
- Control de natalidad
- Impotencia sexual
- Cuidado de la piel

Cada clasificación representa en el mapa de navegación un bloque, estos bloques se crean dinámicamente. Cada bloque al seleccionarlo da por resultado una lista de



productos pertenecientes a la clasificación del bloque, dando la posibilidad de consultar un artículo en específico.



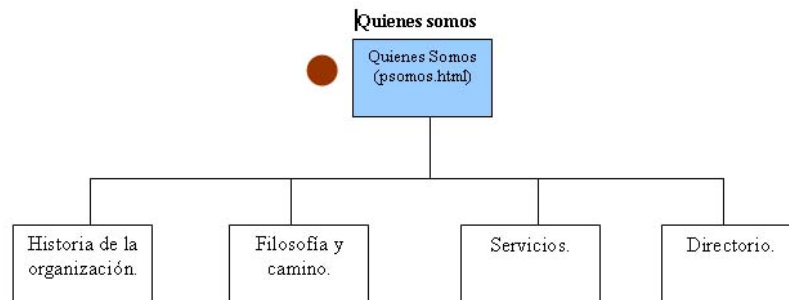
**Figura 4.22** Página Cotización

Rápidamente damos la descripción de cada bloque de la figura 30. El bloque de cotización asíncrona permite al cliente mediante un formulario escribir una lista de productos o bien subir un archivo de texto con su lista, para que en un tiempo determinado se le envíe un e-mail con la cotización.

- a) El bloque datos recibidos es una pantalla que se le muestra al usuario cuando la transacción de información ha resultado satisfactoria.

El bloque cotización es un nuevo servicio que ofrece la organización a sus clientes registrados, permite obtener cotizaciones rápidas.

- b) El bloque reporte cotización es el resultado del cálculo de los artículos que necesita el cliente.



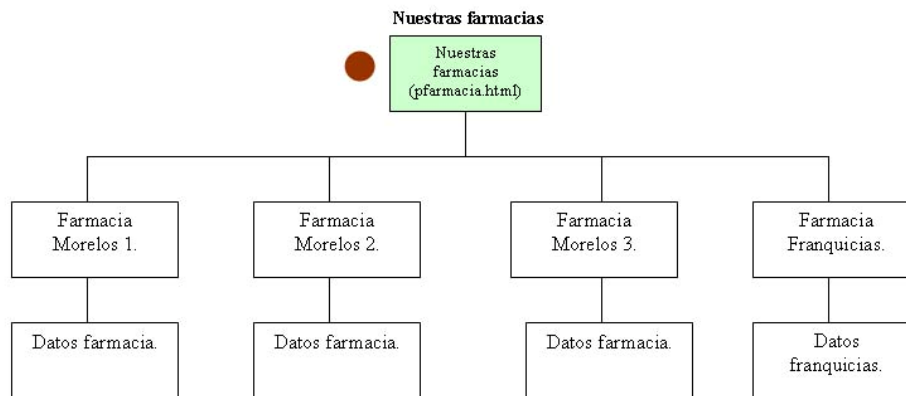
**Figura 4.23** Página Quienes somos

Rápidamente damos la descripción de cada bloque que se halla en la figura 31. El bloque de historia de la organización es la pantalla que contendrá los datos históricos de la empresa y de cada una de las farmacias que pertenecen a ella.

El bloque Filosofía y camino es la página que muestra los fundamentos con que se rige la empresa, así como las metas a lograr a futuro.

El bloque de servicios da información de los servicios que presta la organización, mediante sus filiales, al público. Servicios que se dan tanto de manera directa como en su sitio en Web.

El bloque directorio contiene los nombres del personal que labora para la empresa, en que filial trabaja y el puesto que desempeña.

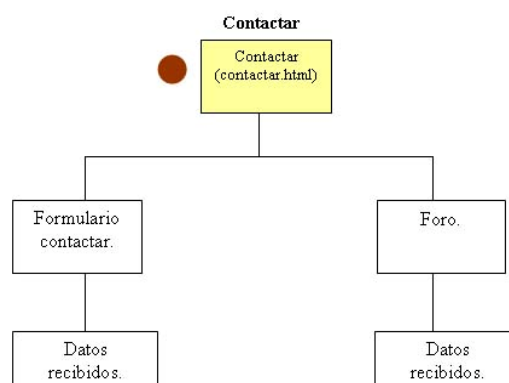


**Figura 4.24** Página Nuestras farmacias

Rápidamente damos la descripción de cada bloque que se encuentran en la figura 32. En el bloque Farmacia Morelos 1 contiene la información de la filial que tiene por nombre Morelos1, esto es, la dirección física de donde se encuentra y su número telefónico. Al escoger la liga da por resultado el bloque Datos farmacia.

Los dos siguientes bloques Farmacia Morelos 2 y Farmacia Morelos 3 trabajan de la misma forma que el bloque anterior.

El bloque de farmacias franquicias se refiere a mostrar los datos de aquellas farmacias que han comprado el nombre y la forma de trabajo en una farmacia desarrollada por Grupo Morelos.



**Figura 4.25** Página Contactar

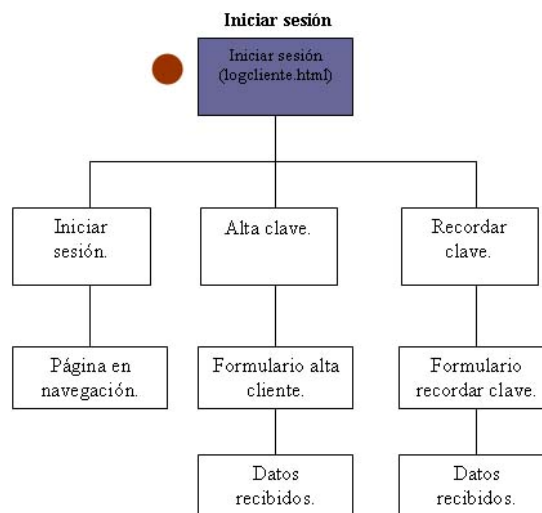


Rápidamente damos una descripción de cada bloque de la figura 33. El bloque formulario contactar es una pantalla que pretende capturar información básica del cliente que desea contactar a la organización, y cuenta con una caja de texto que servirá para que el usuario escriba sus preguntas o dudas.

- a) El bloque datos recibidos es una pantalla que se le muestra al usuario cuando la transacción de información ha resultado satisfactoria.

El bloque Foro es una pantalla que recaba información de cómo el cliente ha visto el sitio Web, utilizando un cuestionario simple. Hay que recalcar que la empresa se reserva el derecho de publicar encuestas según su criterio.

- b) El bloque datos recibidos es una pantalla que se le muestra al usuario cuando la transacción de información ha resultado satisfactoria.



**Figura 4.26** Página Iniciar sesión

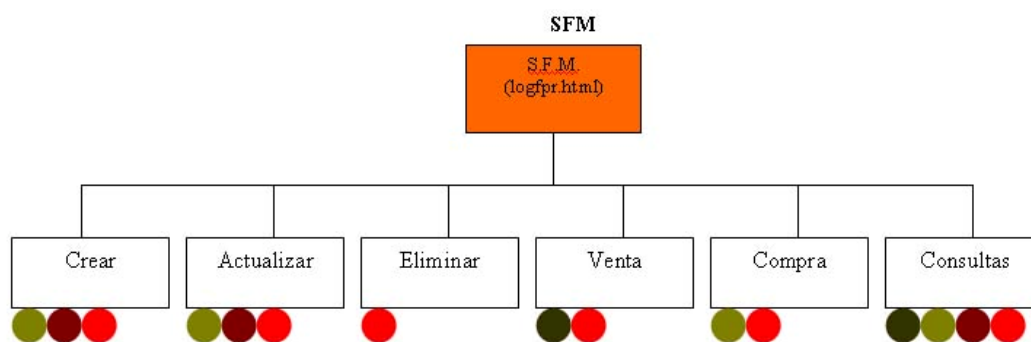
Rápidamente damos la descripción de cada bloque de la figura 34. El bloque iniciar sesión es el formulario que le pide al cliente su nombre de usuario y la contraseña para iniciar sesión.

- a) El bloque página de navegación se refiere a la página en que estaba el cliente, puede ser: pagina principal, cotización, foro.

El bloque alta clave es donde un usuario podrá registrarse y darse de alta en el sistema.

- b) El bloque formulario alta cliente es la encuesta para recabar información del cliente que quiere registrarse en el sistema.

El bloque recordar clave es donde se entrega un cuestionario que al contestar el cliente correctamente recibirá por e-mail su contraseña.



**Figura 4.27** Página SFM

Rápidamente damos la descripción de cada bloque que se encuentra en la figura 35. El bloque crear contiene los comandos para crear puesto, empleado, filial, almacén, catálogo, producto y artículo.

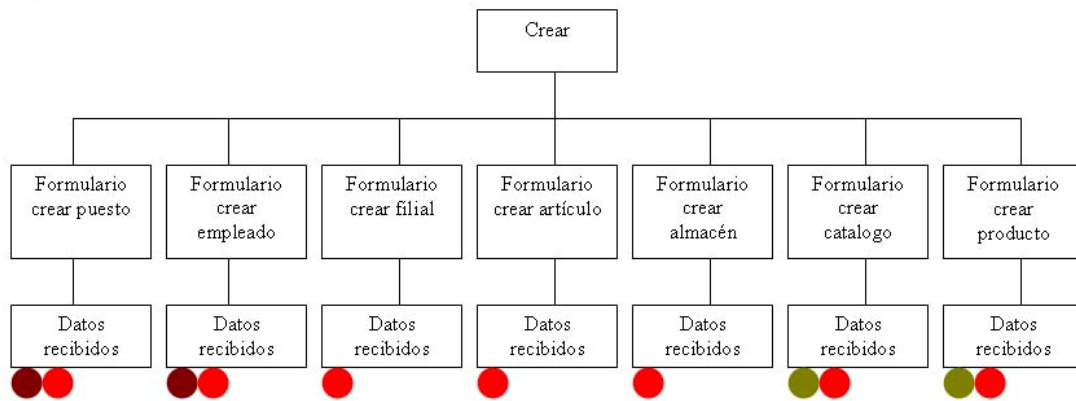
El bloque actualizar contiene los comandos para actualizar puesto, empleado, filial, almacén, catálogo, producto y artículo.

El bloque eliminar contiene los comandos para eliminar puesto, empleado, filial, almacén, catálogo, producto y artículo.

El bloque venta contiene los comandos de ventas.

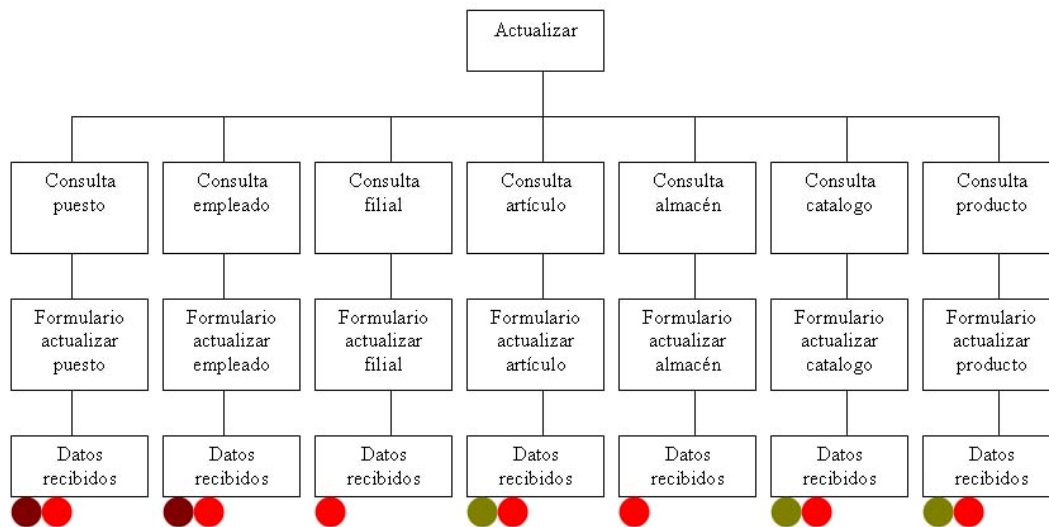
El bloque compra tiene los comandos de compra.

El bloque productos es una búsqueda avanzada.



**Figura 4.28** Página Crear

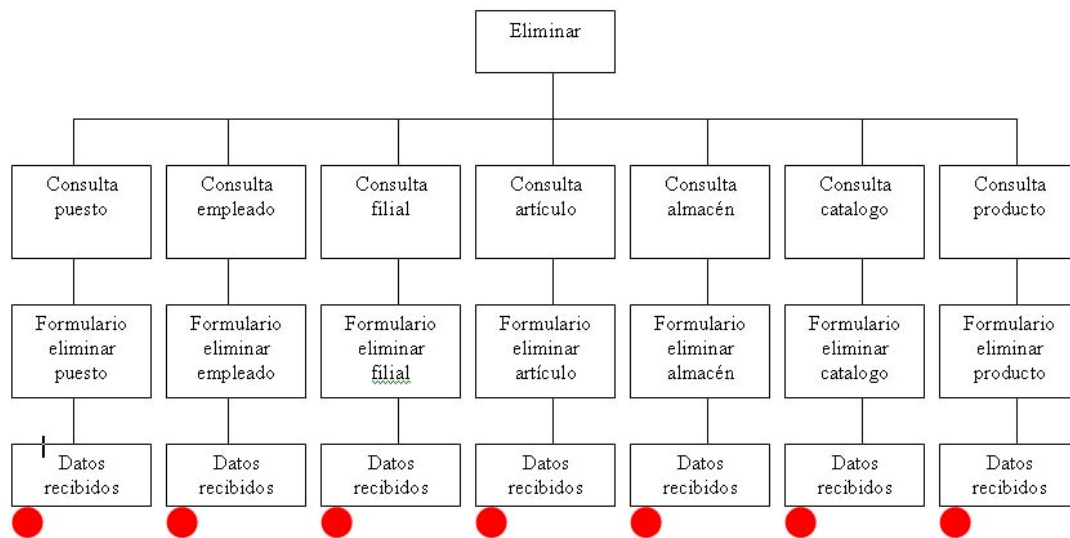
Rápidamente damos la explicación de cada bloque que se presenta en la figura 36. Los bloques formulario crear puesto, empleado, filial, artículo, etc. despliegan un formulario específico que requiere cada captura. Una vez introducidos los datos se presenta una página de datos recibidos, dependerá el resultado si los datos están bien o mal.



**Figura 4.29** Página Actualizar

Rápidamente se da una descripción de los bloques que se detallan en la figura 37. El bloque consulta puesto, empleado, filial, etc., es el encargado de buscar el elemento que se desea actualizar. Una vez encontrado el elemento se entrega una página de formulario actualizar puesto, empleado, filial, etc., para que se proceda a escribir los nuevos datos.

El bloque datos recibidos entregará una página que le avisara al usuario si esta bien o mal sus datos ingresados.



**Figura 4.30** Página Eliminar

Rápidamente damos una explicación de cada bloque presentado en la figura 38. El bloque consulta puesto, empleado, filial, etc., se encarga de buscar el elemento que se quiere borrar. Una vez que se encuentra se despliega en la pantalla el formulario eliminar puesto, empleado, filial, etc., para proceder con la eliminación del elemento.

El bloque datos recibidos es la pantalla que indica que se ha borrado ó no el elemento.



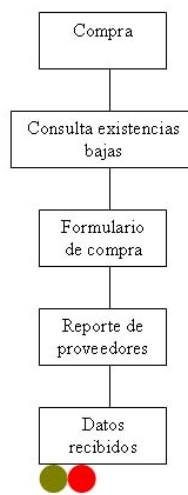
**Figura 4.31** Página Venta

Rápidamente se da la descripción de cada bloque de la figura 39. El bloque formulario venta es la pantalla donde el usuario ingresa los datos de los artículos candidatos para una venta.



El bloque de datos recibidos despliega una ventana informativa de existencia suficiente o artículo inexistente.

El bloque confirmar venta es la pantalla que le sirve al usuario para cerrar una venta o cancelarla.



**Figura 4.32** Página Compra

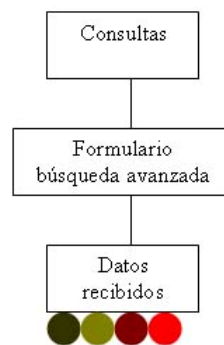
Rápidamente se da una descripción de cada bloque que se encuentra en la figura 40. El bloque consulta bajas existencia despliega un reporte de los productos que el sistema a detectado con baja existencia en los almacenes de cada farmacia.

El bloque formulario de compra es donde se agregan los productos que han sido aceptados para su compra.

El bloque reporte de proveedores despliega en la pantalla un reporte de los proveedores de cada artículo candidato a comprar.

El bloque datos recibidos despliega una pantalla que informa los datos de la orden.





**Figura 4.33** Página Consultas

Rápidamente se da una descripción de cada bloque que presenta la figura 41. El bloque Formulario búsqueda avanzada es una pantalla que recoge datos específicos de un artículo.

El bloque datos recibidos despliega una pantalla con todos los datos del artículo, el almacén y la farmacia en que se encuentra.

#### **4.4. Política de diseño**

El diseño de la interfaz será realizado en base al formato de HTML, los elementos utilizados para el sistema son los siguientes:

- Encabezados.
  - Títulos.
  - Imágenes.
  - Listas.
  - Formularios.
  - Botones.
  - Cajas de texto.
- 
- Antecedentes de diseño.

El grupo Morelos ha utilizado algunas herramientas propagandísticas como volantes, bolsas, tarjetas de presentación y letreros de publicidad. En cada uno de ellos se



encuentra que no hay un patrón de diseño, los bocetos han sido dejados a la consideración de las empresas que les proporcionan estos objetos publicitarios.

La única recomendación de diseño que hacen es sobre el nombre “Farmacia Morelos”, este nombre tiene las siguientes características:

- El color de la letra será de azul.
- La palabra Farmacia deberá ir sola en un renglón.
- La palabra Morelos deberá ir sola en un renglón abajo.
- La letra F de la palabra farmacia debe ser más grande y tener un estilo diferente.
- La letra M de la palabra Morelos debe ser más grande y tener un estilo diferente.

La letra F de la palabra farmacia y la M de la palabra Morelos deberán coincidir lo más posible.

Para la creación de los letreros publicitarios la organización fué específica en la utilización de colores suaves, combinando de tonos de azul y blanco. Esto combinando e integrando el nombre “Farmacia Morelos”.

- Patrones de diseño.

Para la interfaz se consideran las recomendaciones del grupo Morelos. Se realizará una interfaz que combine con los colores azul y blanco. La presentación será lo mas sobria posible, ya que se trata de una organización seria. Las imágenes estarán realizadas de manera que los colores combinen. Las representaciones de formularios serán con una estructura que tratara de conservarse a lo largo del sitio.

Las letras en un documento deben tener un formato que represente su importancia o función, por ello se establecen diferentes tipos de letras dentro de la interfaz.

- Título.
- Menú principal
- Texto normal.
- Letra de formulario.



- Correo.
- Menú secundario.
- Login.
- Letritas.
- Liga normal.
- Sub menú.

*Título.* El título es la palabra que determina el tema de cada pantalla, esta resaltado de tal forma que sea una palabra fácilmente identificable.

La letra es de tipo: Times New Roman con un tamaño de 18px, el estilo de letra es Italia y remarcado. El color de la letra es negro.

*Menú principal.* Como su nombre lo dice, son las acciones más importantes que puede realizar un usuario. La importancia de resaltar estas palabras y que se identifiquen como ligas a otras páginas es fundamental. En cada pantalla estarán presentes.

La letra es de tipo: Verdana con un tamaño de 14px, es de estilo normal y remarcado, el color es un tono de gris suave.

*Texto normal.* La mayor parte de la información, como en cualquier documento, esta definido de una forma estándar, hemos elegido el siguiente diseño pensando en la comodidad de los usuarios.

La letra es de tipo: Times New Roman con un tamaño de 12px, es de estilo normal sin ningún efecto, el color es de un tono de gris fuerte.

*Letra de Formulario.* Este tipo de letra se utilizará en todos los formatos en que se pidan datos, fácilmente el usuario reconocerá que la palabra corresponde a un formulario.

La letra es de tipo: Time New Roman con un tamaño de 12px, es de estilo normal y remarcado, el color de la letra es negro.



*Correo.* El correo es la forma mas usual de comunicarse entre los cibernautas, por ello se escoge este tipo de letra que distinga fácilmente las direcciones de correo electrónico. Es de gran importancia que los usuarios sepan con quien pueden ponerse en contacto.

La letra es de tipo: Verdana con un tamaño de 10px, es de estilo normal, remarcado y subrayado. El color de la letra es de un tono azul fuerte.

*Menú secundario.* Las acciones secundarias de cada liga del menú principal estarán representadas por este tipo de letra. Este tipo de letra esta hecho para llamar la atención de una forma más discreta pero atrayente.

La letra es de tipo: Verdana con un tamaño de 14px, es de estilo normal y sin efectos, el color de la letra es de un tono azul claro.

*Login.* La única excepción, de tipo de letra, en el menú principal es el inicio de sesión, se ha cambiado el tipo de letra en representación de la importancia de iniciar una sesión. El motivo es hacer mucho más seductor.

La letra es de tipo: Times New Roman con un tamaño de 14px, es de estilo normal y remarcado, el color de la letra es negro.

*Letritas.* En diversos documentos se encuentran pequeñas notas o algunos comentarios, por ello la utilización de una letra chica que llame la atención y que se integre en toda la pantalla es necesario. Este tipo de letra se usa para dejar comentarios al final de la página.

La letra es de tipo: Verdana con un tamaño de 9px, es de estilo normal y remarcado, el color de la letra es negro.

*Liga normal.* Las ligas que no pertenezcan a una acción definida serán representadas por este tipo de letra.

La letra es de tipo: Verdana con un tamaño de 12px, es de estilo normal y remarcado, el color de la letra es de un tono azul fuerte.



*Sub menú.* Una página Web esta limitada horizontalmente, sin embargo, verticalmente el documento se puede extender demasiado que el menú principal se pierda, para ello al final del documento se pone un menú de las mismas características que el principal.

Este menú esta representado por el tipo de letras de Sub menú.

La letra es de tipo: Times New Roman con un tamaño de 9ptos, es de estilo normal y remarcado, el color de la letra es de azul marino fuerte

*Artículos.* Será el tipo de letra que se use para la representación de las consultas de información.

La letra es de tipo; Times New Roman con un tamaño de 12px, es de estilo normal y remarcado, el color de esta letra es de un tono gris fuerte.

- Imágenes.

Las imágenes utilizadas para detallar artículos o dar presentación a la pagina deben ser lo menos pesadas posibles, pero conservando la calidad de imagen. La imagen es un aspecto muy importante para dar a conocer los productos.

- Se trabajará con imágenes de tamaño de 100x100px para presentar a los artículos.
- Se trabajará con imágenes de tamaño de 150x75px para las imágenes en el encabezado.
- Se trabajará con imágenes de tamaño de 400x200px para las imágenes de vistas.
- Se trabajará con imágenes de tamaño de 400x400px para imágenes que funcionen como mapas sensibles.

En materia de banners publicitarios se seguirán tres tamaños estándares que son: 120x60px, 120x90px y 234x60px.



## CONCLUSIONES

Los paradigmas de programación son vistos como una filosofía de creación de software, cada uno de ellos tiene ventajas y desventajas. La experiencia dicta cual será el más adecuado en los diferentes casos, que, a lo largo de nuestro desempeño profesional enfrentaremos.

Durante la realización de cada una de las partes de este trabajo he obtenido muy gratas experiencias, aplicando los conocimientos obtenidos en el diplomado “Desarrollo de sistemas en Web con Java”. De ahí que tenga las siguientes conclusiones.

Las tecnologías computacionales ayudan a las empresas a ser más productivas, sin embargo, es importante tomar en cuenta las posibilidades económicas de cada caso. El realizar un análisis para un sistema sin tomar en cuenta el aspecto económico es arriesgar la conclusión de un proyecto.

El realizar un sistema requiere del trabajo en conjunto de muchos especialistas. Se requiere de un especialista en administración de bases de datos (DBA), un administrador del sistema operativo del servidor y, en nuestro caso, un analista y desarrollador de sistemas.

La especialización del equipo de trabajo enriquece a todos los miembros del equipo así como al producto final. Como egresados de la Universidad Nacional Autónoma de México campus “Aragón” tenemos el deber de buscar la excelencia en los trabajos que realicemos.

En México, la utilización de Internet como medio de difusión ha ido incrementándose año con año<sup>1</sup>. Al respecto concluimos que la utilización de un medio como Internet, que se ha convertido en una forma de difusión de distintos servicios, no debe desaprovecharse.

---

<sup>1</sup> Para mas información

<http://www.inegi.gob.mx/est/contenidos/espanol/rutinas/ept.asp?t=tin136&c=5561>



Una de las grandes ventajas que tiene Internet es que puede conectarse todo el público. Por lo que se puede encontrar gente maliciosa que busca, en diversos grados de agresión, perjudicar a los diversos usuarios de esta vía de comunicación.

Los ataques que puede realizar un individuo en la Internet pueden exponer a cualquier usuario (empresas, instituciones, gobierno, personas, etc.) a un fraude monetario, una modificación o pérdida de datos, una interrupción de sus servicios tecnológicos, tiempo perdido, etc.

Por ello es primordial proteger y considerar todos los posibles peligros en que se vera comprometido el sistema. Considerar en que entorno va a trabajar y establecer lo mejor posible las políticas de seguridad dentro del sistema.

La elección, instalación y configuración del software es parte primordial, así como, la plataforma en que se realizara el sistema deberá ser producto del previo análisis del problema en cuestión.

La arquitectura de tres capas permite que la fuente de información sea eficiente y que trabaje lo mejor posible, realizar procesos o cálculos sistematizados evitando lo más posible el error humano. Por ello se debe de poner empeño por definir lo mejor posible los límites de cada capa, ya que de no ser así el sistema podría quedar comprometido, la arquitectura de tres niveles por si misma genera un nivel de seguridad.

A medida que se toman más en serio la ingeniería del software se tienen mejores herramientas de análisis y desarrollo. Así la combinación de diseño basado en UML que es orientado a objetos y las mejores prácticas que describe el RUP se pueden obtener resultados más claros y concisos.

El lenguaje UML (Unified Modeling Language) ofrece un modelo de desarrollo que permite construir y documentar todos los elementos que conforman a un sistema basado u orientado a objetos.



Permitiendo llevar el proceso de desarrollo de manera ordenada y clara, por ello se debe poner énfasis en la utilización de este modelo de desarrollo. El comprender y saber utilizar este tipo de herramientas aumenta la productividad y la calidad de la documentación y del producto final, el sistema.

La programación orientada a objetos permite mantener un orden y facilita la lectura de código a otros desarrolladores de software. Permitiendo que la eficacia del trabajo en equipo sea maximizada y que no se tengan mayores obstáculos que el entendimiento natural del ser humano.

El lenguaje de programación Java es de los más robustos para sistemas en Web. Este lenguaje a través del tiempo se ha ido fortaleciendo y ha tenido gran aceptación en el mercado industrial de la creación del software.

Al construir una aplicación computacional que permita la automatización de procesos repetitivos de una organización he aprendido lo difícil que es esclarecer los objetivos fundamentales que debe cumplir un sistema.

La comunicación con los diferentes actores, que serán quienes utilicen el sistema creado, es fundamental. Ellos son realmente los que aprobarán o desecharán la solución propuesta.

Establecer como verdades cada uno de los objetivos que el cliente busca que cumpla el sistema, mantener claros los objetivos desde el principio hasta el final es un punto que debe tomarse con cuidado, el no hacerlo podría complicar el desarrollo del sistema.



**INDICE DE IMAGENES**

<b>Figura</b>	<b>Nombre</b>	<b>Página</b>
Figura 1.1	Tecnología de 3 capas	3
Figura 1.2	Medios de conexión a Internet, 2005	3
Figura 1.3	El clásico objeto carro	14
Figura 1.4	Funcionamiento Java	24
Figura 1.5	2 capas Cliente/Servidor	47
Figura 1.6	3 capas Cliente/Servidor	47
Figura 1.7	Intercambio básico Http	49
Figura 1.8	Arquitectura MVC	51
Figura 1.9	Apache con plugin de Tomcat	53
Figura 1.10	Struts en operación	54
Figura 2.1	Diagrama de flujo	56
Figura 2.2	Riesgo potencial de seguridad	68
Figura 3.1	Arquitectura del Sistema SFM	70
Figura 3.2	Operador de venta	76
Figura 3.3	Operador de compra	77
Figura 3.4	Público General	78
Figura 3.5	Operador de reportes	79
Figura 3.6	Supervisor de personal	80
Figura 3.7	Supervisor general	81
Figura 4.1	Diagrama de clases	124
Figura 4.2	Diagrama de secuencia de Compra	125
Figura 4.3	Diagrama de secuencia de Venta	126
Figura 4.4	Diagrama de secuencia de Reporte de compra	127
Figura 4.5	Diagrama de secuencia de Reporte de venta	127
Figura 4.6	Diagrama de secuencia de Reporte almacén	128
Figura 4.7	Página principal Web	134
Figura 4.8	Página alta cliente	135
Figura 4.9	Página nuestras farmacias	135
Figura 4.10	Página productos	136



<b>Figura</b>	<b>Nombre</b>	<b>Página</b>
Figura 4.11	Página de login	136
Figura 4.12	Página de producto por lista alfabética	137
Figura 4.13	Página descripción de un producto	137
Figura 4.14	Página crear articulo	138
Figura 4.15	Página de cotizaciones	138
Figura 4.16	Diagrama Base de Datos	143
Figura 4.17	Página principal	152
Figura 4.18	Página Bloque de productos	153
Figura 4.19	Página Sustancia activa	154
Figura 4.20	Página Material de curación	154
Figura 4.21	Página por función	155
Figura 4.22	Página Cotización	158
Figura 4.23	Página Quienes somos	159
Figura 4.24	Página Nuestras Farmacias	160
Figura 4.25	Página Contactar	160
Figura 4.26	Página Iniciar sesión	161
Figura 4.27	Página SFM	162
Figura 4.28	Página Crear	163
Figura 4.29	Página Actualizar	163
Figura 4.30	Página Eliminar	164
Figura 4.31	Página Venta	164
Figura 4.32	Página Compra	165
Figura 4.33	Página Consultas	166



## BIBLIOGRAFÍA

1. A. FOROUZAN, Behrouz; Transmisión de datos y redes de comunicación. MCGRAW-HILL, España, 2002.
2. ABLAN, Jerry; YANOFF, Scott. Web Site Administrators. SAMS NET, EU, 1996
3. ATKINS, Derek; BUIS, Paul; et al. Internet security professional reference. NEW RIDERS, United States of America, 1997.
4. BARCLAY, Kenneth y SAVAGE, John. Object-oriented design with UML and JAVA. ELSEVIER, BUTTERWORTH-HEINEMANN, Inglaterra, 2004.
5. BUDD, Timothy. Object – Oriented programming. ADDISON-WESLEY, EU, 1991
6. CAGLE, Kurt; MACDONALD, Michael. Visual Basic 6 Programación cliente / servidor. PARAINFO, España, 1999.
7. CAMACHO FERNÁNDEZ, David; VALLS FERRÁN, José María; GARCÍA HERRERO, Jesús; MOLINA LÓPEZ, José y BUENO RODRÍGUEZ, Enrique. Programación, Algoritmos y Ejercicios Resueltos en JAVA. PEARSON EDUCACIÓN, S.A., Madrid, 2003.
8. FEGHHI, Jalal. Web developer's guide to JAVA BEANS. CORIOLIS GROUP BOOKS, United States of America, 1997.



9. J. KABIRG, Mohammed. La biblia del servidor Apache. ANAYA MULTIMEDIA, United States of America, 1998.
10. L. WINBLAD, Ann; D. EDWARDS, Samuel; R. KING, David. Object – Oriented Software. ADDISON-WESLEY, EU, 1990
11. LADD, Eric; O'DONELL, Jim. Using HTML, Java and CGI. QUE, EU, 1996
12. LEANDER, Rick. Building Application Servers.CAMBRIDGE UNIVERSITY, EU, 2000.
13. MILENKOVIC, Milan. Sistemas Operativos. Conceptos y Diseño. MC GRAW-HILL, España, 1988.
14. PÉREZ, Cesar. MySQL para Windows y Linux. ALFAOMEGA, México, 2004
15. PIKE, Mary Ann; COOK, David. La biblia de Internet. ANAYA MULTIMEDIA, España, 1995.
16. POWELL, Thomas. HTML 4 Manual de Referencia. MCGRAW-HILL, España, 2001.
17. SOMMERVILLE, Ian; SAWYER, Pete. Requeriments Engineering. WILEY, Great Britain, 1997
18. SQUIRE, Enid. La computadora, un auxiliar indispensable. FONDO EDUCATIVO INTERAMERICANO, México, 1982.



19. STEVENS, Perdita y POOLEY, Rob. Utilización de UML en ingeniería del software con objetos y componentes. PEARSON EDUCACIÓN, S.A., Madrid, 2002.
20. WILSON, Brian. Sistemas: Conceptos, Metodología y aplicaciones. MEGABYTE, México, 1993.

**Direcciones URL:**

21. Antelo Jiménez, Erick. Diseño de aplicaciones en tres capas con Windows DNA. <http://www.geocities.com/trescapas/capadenegocios.htm>
22. CIFH. Niveles, privilegios y tipos de usuario. <http://webci.com.ar/ayudas/tiposnivelesusuarios.htm>
28. Goetten Junior, Vicente. Desmitificando o Framework Jakarta Struts. <http://www.javafree.org/content/view.jf?idContent=22>
23. GRANOLLERS, T; PERDRIX, F; LORÉS, J. Incorporación de Usuarios en la Evaluación de la Usabilidad por Recorrido Cognitivo. <http://griho.udl.es/publicacions/2004/Recorrido-Cognitivo-INTERACCION.PDF>.
21. INSTITUTO TECNOLÓGICO DE LA PAZ. Tutorial de sistemas Distribuidos II. <http://www.itlp.edu.mx/publica/tutoriales/sistsdist2/t53.htm>



26. JANIUM TECHNOLOGY. Aplicaciones basadas en WEB Internet / Intranet.  
[http://www.janium.com/phpws/index.php?module=documents&JAS\\_DocumentManager\\_op=downloadFile&JAS\\_File\\_id=1.](http://www.janium.com/phpws/index.php?module=documents&JAS_DocumentManager_op=downloadFile&JAS_File_id=1)
24. MERCOVICH, EDUARDO. Ponencia sobre diseño de interfaces y usabilidad.<http://planeta.gaiasur.com.ar/infoteca/siggraph99/disenio-de-interfaces-y-usabilidad.html#9.3>
25. SUN MICROSYSTEM. JDBC basics. <http://java.sun.com/docs/books/tutorial/jdbc/basics/index.html>