



**BENEMÉRITA UNIVERSIDAD
AUTÓNOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN**

**SISTEMA DE ADQUISICIÓN Y
DISTRIBUCIÓN DE ACCESORIOS Y
EQUIPOS CELULARES**

TESIS

**QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN**

PRESENTA:

GRACIELA FABIOLA JIMÉNEZ LLANOS.

ASESORA:

DRA. DARNES VILARIÑO AYALA.

PUEBLA, PUE.

DICIEMBRE 2009.

RESUMEN.

El presente proyecto tiene como finalidad la creación de un Sistema vía web para la Adquisición y Distribución de Equipos Celulares, desarrollado por la necesidad de la Empresa de contar con un sistema consistente desde el lanzamiento y trazado de la transacción por parte del representante del Fabricante de los accesorios o equipos celulares, hasta el almacenamiento de los datos en algún Almacén de una sucursal.

Para llevar a cabo la creación de este proyecto se utilizó la metodología basada en el Ciclo de Vida de Cascada el cual presenta las siguientes fases: Requerimientos, Análisis, Diseño, Implementación y Pruebas.

El análisis de requerimientos se utiliza para identificar la complejidad del problema y nos permite identificar cada una de las funciones que realizará el Sistema de forma detallada. Para una mejor comprensión de estas funciones se utilizó una herramienta que permite manipular elementos en UML la cual nos permitió hacer diagramas de Casos de Uso, para así poder representar de una manera más sencilla estas funciones.

En la fase de diseño se desarrolló una base de datos basada en el modelo Entidad-Relación, la cual facilita la comprensión de todos los elementos de la base de datos como lo son las entidades, los atributos y las relaciones que existen entre ellos.

Para la fase de la implementación del sistema se utilizaron diferentes herramientas como: ORACLE para la manipulación y gestión de la Base de Datos, y como herramienta de implementación se empleó el IDE de IBM Rational Application Developer por estándares de la Empresa.

Para implementación de la aplicación y por estandarización de aplicaciones en la empresa se utilizó el framework de Struts2, basado en el patrón MVC (*model, view, controller*), lo que nos permitió conocer además la interacción con otros frameworks utilizados, tal es el caso de Ibatis y Spring para algunas etapas de la implementación.

Después de este proceso se procedió a realizar las pruebas pertinentes para comprobar que el Sistema cumple con las funciones antes especificadas de una manera correcta y eficiente, desde la perspectiva de cada operador del sistema, y las tareas asociadas a él.

AGRADECIMIENTOS:

A mi pequeña familia por su apoyo incondicional, sus consejos, el valor y el amor que me han dado en el transcurso de mi vida.

A mis profesores por lo que me han enseñado, corregido y demostrado durante este período de trabajo.

A mis amigas que han estado conmigo en las buenas y en las malas, y aquellos con los que he compartido esta etapa; porque me han dado un pequeño espacio en sus vidas.

I NDICE

A. INTRODUCCIÓN.....	2
1.- Marco Teórico.....	4
1.1 Ingeniería de Software.....	4
1.2 Modelo de Cascada.....	5
1.3. UML.....	6
1.3.1 Diagramas UML.....	7
1.4 Base de Datos.....	8
1.4.1 Sistema de Gestión de Base de Datos.....	9
1.4.2 Abstracción de Datos.....	9
1.4.3 Esquemas.....	10
1.4.4 Independencia de los Datos.....	10
1.4.5 Modelos de Datos.....	11
1.4.6 Modelo Entidad Relación.....	11
1.4.7 Modelo Relacional.....	12
1.4.7.1 Normalización o Formas Normales.....	15
1.5 Modelo Cliente Servidor.....	16
1.6 Plataforma.....	17
1.6.1 Microsoft Visio.....	17
1.6.2 Oracle.....	17
1.6.3 TOAD.....	17

1.6.4 IBM Websphere Application Server (WAS).....	18
1.6.5 IBM Rational Application Developer (RAD).....	18
1.6.6 STRUTS.....	18
2.- Análisis del Sistema.....	21
2.1 Planteamiento del Problema.....	21
2.1.1 Antecedentes.....	21
2.1.2 Descripción del Problema.....	21
2.2 Especificación de Requerimientos.....	25
2.3 Modelo de Casos de Uso.....	28
2.3.1 Especificación de Casos de Uso.....	31
3.- Diseño del Sistema.....	43
3.1 Diseño Conceptual.....	43
3.2 Diseño Lógico.....	48
3.2.1 Normalización de Base de Datos del Sistema.....	49
3.3 Diseño Físico.....	50
3.4 Diccionario de Datos.....	51
3.5 Diseño de la Interfaz del Sistema.....	56
4.- Implementación y Validación del Sistema.....	58
4.1 Implementación del sistema.....	58
4.2 Interfaz del Sistema.....	62
4.2.1 Interfaz de Control de Acceso.....	63
4.2.2 Interfaz Área de Trabajo Personalizada por Operador.....	63
4.2.3 Interfaz Genera Propuesta.....	64
4.2.4 Interfaz Dictaminar Propuesta.....	65

4.2.5 Interfaz Primera Carga.....	66
4.2.6 Interfaz Aceptar Primera Carga.....	67
4.2.7 Interfaz para Realizar Transferencias.....	67
4.2.8 Interfaz de Consultas.....	68
Conclusiones y recomendaciones.....	71
Bibliografía.....	72

ÍNDICE DE FIGURAS

Fig. 1.1	Diagrama de Ciclo de Vida del Software.....	5
Fig. 1.2	Modelo de Cascada.....	5
Fig. 1.3	Representación Gráfica de elementos del modelo Entidad Relación.....	11
Fig. 1.4	Manejo de Datos.....	13
Fig. 1.5	Patrón MVC.....	18
Fig. 1.6	Framework Struts.....	19
Fig. 1.7	Framework Struts 2.....	20
Fig. 2.1	Caso de Uso – Sistema General.....	29
Fig. 2.2	Casos de Uso que controla el actor Representante del Fabricante.....	30
Fig. 2.3	Casos de Uso que controla el actor Persona de Inventario.....	30
Fig. 2.4	Casos de Uso que controla el actor Gerente de Sucursal.....	30
Fig. 2.5	Casos de Uso que controla el actor Administrador.....	31
Fig. 3.1	Diagrama Entidad-Relación.....	47
Fig. 3.2	Diagrama generado por TOAD.....	50
Fig. 3.3	Interfaz Inicio.....	56
Fig. 3.4	Interfaz Principal.....	57
Fig. 3.5	Interfaz de área de Trabajo para cada Operador.....	57
Fig. 4.1	Organización del framework – archivo web.xml.....	58
Fig. 4.2	Parte declarativa – archivo web.xml.....	59
Fig. 4.3	index.jsp – Código referenciando action.....	59
Fig. 4.4	struts.xml – Asignación de clase action.....	60
Fig. 4.5	Implementación de un objeto POJO.....	60
Fig. 4.6	Elementos que forman un DAO.....	60
Fig. 4.7	Código que constituye a la Interfaz DAO.....	61
Fig. 4.8	Implementación del componente DAO.....	61
Fig. 4.9	Declaración del archivo que contiene el mapeo.....	61
Fig. 4.10	Mapeos entre objetos parámetros u objetos de entrada o salida.....	62
Fig. 4.11	Interfaz de Inicio.....	64
Fig. 4.12	Interfaz Principal.....	64
Fig. 4.13	Interfaz Genera Propuesta.....	65
Fig. 4.14	Interfaz Dictamina Propuesta.....	66
Fig. 4.15	Interfaz Genera Primera Carga.....	66
Fig. 4.16	Interfaz Acepta Primera Carga.....	67
Fig. 4.17	Interfaz de Transferencias.....	68
Fig. 4.18	Interfaz Consulta Propuesta.....	69
Fig. 4.19	Consulta de Accesorio y Equipos Celulares.....	69
Fig. 4.20	Interfaz para Consultar Transferencias.....	70

ÍNDICE DE TABLAS

Tabla 3.1	Rol.....	51
Tabla 3.2	Operador.....	52
Tabla 3.3	Rep_Fabricante.....	52
Tabla 3.4	Persona_Inventario.....	52
Tabla 3.5	Gerente_Suc.....	53
Tabla 3.6	Propuesta.....	53
Tabla 3.7	Detalle_Propuesta.....	54
Tabla 3.8	Dictamen.....	54
Tabla 3.9	Sucursal.....	54
Tabla 3.10	Almacén.....	55
Tabla 3.11	Primera Carga.....	55
Tabla 3.12	Articulo.....	55
Tabla 3.13	Transferencia.....	56

I NTRODUCCIÓN

Hoy en día la utilización de la tecnología abarca necesidades básicas que van desde la alimentación, vestimenta, vivienda, incluyendo las artes, el deporte, la fabricación de armas, el cuidado del medio ambiente y también se ve en algunas ocasiones como símbolo de estatus.

Tecnologías hay muchas y diferentes entre sí pero las tecnologías de las telecomunicaciones, han registrado grandes progresos a partir de los primeros satélites de comunicaciones. Ha aumentado la velocidad y la memoria de las computadoras, disminuyendo sustancialmente su tamaño. La aparición de circuitos integrados y la invención de los teléfonos celulares; ha sido otro paso importante, convirtiéndose éste último en una herramienta primordial para las personas.

El sistema es para una empresa dedicada al rubro de la telefonía celular y requiere que todos sus sistemas generen transacciones en línea, con la característica de que las transacciones se ejecuten en el menor tiempo posible y que sean consistentes para garantizar información confiable y veraz dentro de su calidad de servicio.

La tecnología web nos permite agilizar los procesos, mejorar la productividad y aumentar la eficiencia tanto del trabajo como de cualquier aspecto de nuestra vida diaria, haciendo cada tarea más sencilla para el usuario.

En este aspecto la empresa requiere de un sistema vía web que controle el proceso de adquisición y distribución de accesorios y equipos celulares desde la oferta propuesta por el Fabricante.

Actualmente no se tiene un sistema con estas características, además de que la idea del sistema es poder concentrar los artículos en un almacén central para que a partir de éste puedan ser trasladados a un almacén de una sucursal.

Por lo anterior se decidió realizar un análisis de los requerimientos y diseñar un sistema que permita sustentar toda la información necesaria de manera adecuada.

El sistema desarrollado como parte de este trabajo, considera una etapa de implementación y un período de pruebas, en aras de garantizar la integridad y consistencia de la información almacenada a través de todo el flujo.

En base a lo anterior nuestros objetivos son:

OBJETIVOS GENERALES.

Proveer un sistema eficiente, confiable, integral y consistente desde el lanzamiento y trazado de la transacción hasta el almacenamiento de los datos.

Satisfacer la necesidad que se tiene actualmente para concentrar los equipos y accesorios en un almacén central y que se distribuyan sólo bajo demanda de la zona en la que se encuentra la sucursal.

OBJETIVOS ESPECÍFICOS.

La generación de propuestas para la adquisición de accesorios y equipos celulares por parte del fabricante.

El proceso de análisis para aceptar o rechazar dicha propuesta.

La carga inicial o transferencia inicial de los artículos (accesorios y equipos celulares), involucrados en la propuesta hacia un Almacén Central.

La distribución de accesorios y equipos celulares realizada previamente por selección, entre almacenes de diferentes sucursales o centros de atención personalizada.

El registro de los movimientos de entrada y salida de los artículos desde un almacén central a cualquier almacén de Sucursal.

El documento de tesis presenta la siguiente estructura:

Capítulo 1.- Parte del trabajo de tesis dedicado a conceptos, metodologías, paradigmas y, fundamentación de las herramientas propuestas a emplearse en el proceso de implementación del sistema.

Capítulo 2.- Abarca la comprensión y explicación del flujo lógico del sistema para generar los requerimientos, casos de uso y diagramas requeridos para el sistema.

Capítulo 3.- Sección dedicada al diseño del sistema, desde el modelo conceptual de la Base de Datos hasta el diseño de las interfaces propias del sistema.

Capítulo 4.- Parte del trabajo tesis dedicado a la implementación y detalle de la misma.

Conclusiones y Recomendaciones.- Justificación del cumplimiento de los objetivos planteados.

Capítulo 1. MARCO TEÓRICO.

Capítulo que comprende conceptos, metodologías, paradigmas y, fundamentación de las herramientas propuestas a emplearse en las fases de análisis, diseño e implementación del sistema.

1.1 INGENIERÍA DEL SOFTWARE.

Un *sistema* consiste en una combinación de recursos integrados de tal forma que cumplan una función específica, en respuesta a una necesidad designada de un usuario; dichos recursos pueden ser seres humanos, materiales, equipos, software, instalaciones, datos, etc.

El *software* son los programas o conjuntos de programas, la documentación y la configuración de los datos asociados a un sistema y se caracteriza por ser un elemento lógico, desarrollado, confiable, eficiente y reutilizable.

La *ingeniería del Software* es una disciplina encargada de todos los aspectos relacionados con la producción de software, comenzando en sus etapas más tempranas y que van desde la especificación del sistema, hasta el mantenimiento del mismo; tras su puesta en producción, esto involucra el uso apropiado de todas las herramientas, métodos, procesos y teorías para solucionar los problemas que se estiman o aparecen durante el proceso.

El *proceso de Software o ciclo de vida del Software* es el conjunto de actividades para producir un producto de software de alta calidad.

Las actividades involucradas son:

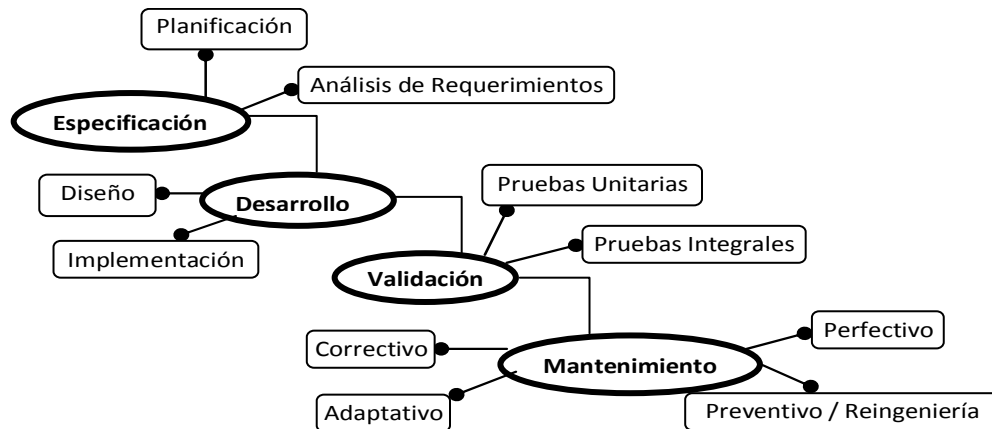
Especificación.- Comprende una etapa de Planificación y Análisis de Requerimientos que establecen la lógica del software, su fragmentación y sus restricciones en su operación.

Desarrollo.- Diseño y programación.

Validación.- Etapa de pruebas.

Evolución o Mantenimiento.- Modificación para adaptación a cambios solicitados .

Las cuales se pueden apreciar en la Fig. 1.1.



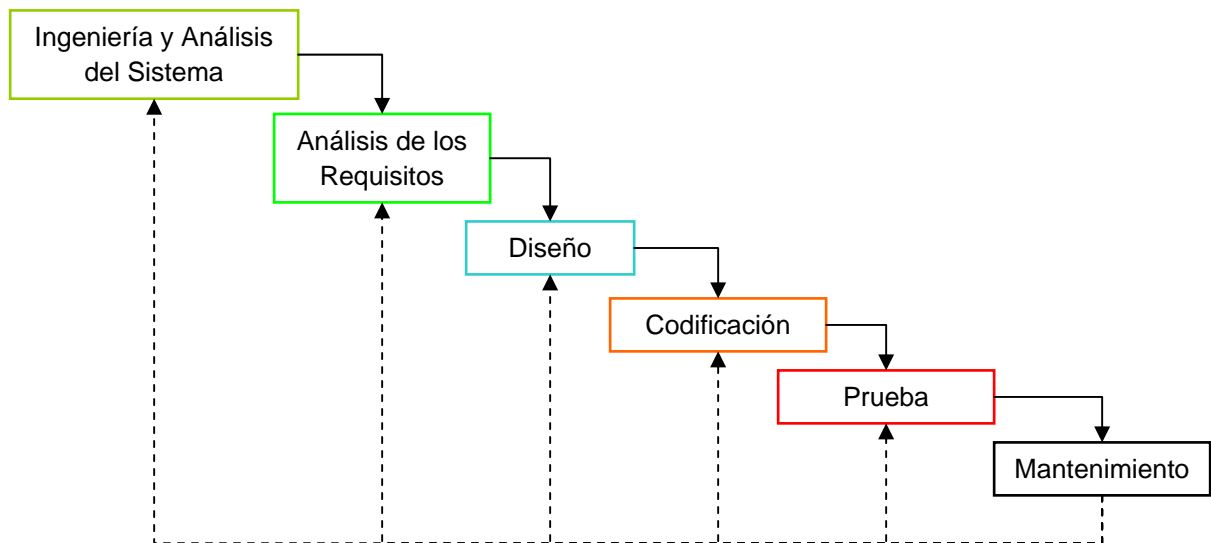
[Fig.1.1 Diagrama de ciclo de vida]

1.2 MODELO DE CASCADA.

Llamado algunas veces “ciclo de vida básico” o “modelo lineal secuencial” es el paradigma más antiguo y más extensamente utilizado en la ingeniería de software.

Sugiere un enfoque secuencial para el desarrollo del software que comienza en un nivel de sistemas progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

El Modelo de Cascada que se muestra gráficamente en la Fig. 1.2 comprende las siguientes actividades:



[Fig.1.2 Modelo de Cascada]

Análisis de los requisitos del software. El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Para comprender la naturaleza del (los) programa(s) a

construirse, el ingeniero (<<analista>>) del software debe comprender el dominio de información del software, así como la función requerida, comportamiento, rendimiento e interconexión.

Diseño. El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso de diseño traduce requisitos en una representación del software donde se pueda evaluar su calidad antes de que comience la codificación.

Generación de código. El diseño se debe traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente.

Pruebas. Una vez que se ha generado el código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos.

Mantenimiento. El software indudablemente sufrirá cambios después de ser entregado al cliente (una excepción posible es el software empotrado). Se producirán cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios de su entorno externo (por ejemplo: se requiere un cambio debido a un sistema operativo o dispositivo periférico nuevo), o porque el cliente requiere mejoras funcionales o de rendimiento. El soporte y mantenimiento del software vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo [1].

1.3 UML.

UML es una notación estándar que nos permite trabajar en la visualización, especificación y, modelar los artefactos ejecutables de un sistema de software; cabe resaltar que no implica en sí una programación.

Una de las metas principales de UML es avanzar en el estado de la integración de las partes del sistema proporcionando herramientas de operabilidad para el modelado visual de objetos.

La notación de UML proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica del sistema.

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✓ *Visualizar:* UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

-
- ✓ **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
 - ✓ **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
 - ✓ **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión [2].

1.3.1 DIAGRAMAS UML.

Los Diagramas de UML tienen su propio objetivo, algunos de ellos se describen a continuación *sólo para fines informativos*, recalcando que para este proyecto específicamente utilizamos el Diagrama de Casos de Uso:

Diagrama de Casos de Uso.

Muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción con el usuario final.

Los elementos que lo componen son: *actores, casos de uso y relaciones entre casos de uso.*

Diagramas de Interacción.

Muestran un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: *Diagramas de Secuencia y Diagramas de Colaboración.*

Los **Diagramas de Secuencia** muestran una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

Los **Diagramas de Colaboración** muestran una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia.

Diagramas de Actividades.

Los *Diagramas de Actividades* sirven fundamentalmente para modelar el flujo de control entre actividades.

La idea es generar una especie de diagrama Pert, en el que se puede ver el flujo de actividades que tienen lugar a lo largo del tiempo, así como las tareas concurrentes que pueden realizarse a la vez.

El Diagrama de Actividades sirve para representar el sistema desde otra perspectiva, y de este modo complementa a los anteriores diagramas vistos.

Gráficamente un Diagrama de Actividades será un conjunto de arcos y nodos.

Desde un punto de vista conceptual, el diagrama de actividades muestra cómo fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecución de un proceso más complejo.

Por este motivo, en un diagrama de actividades aparecerán acciones y actividades correspondientes a distintas clases. Colaborando todas ellas para conseguir un mismo fin.

Un diagrama de Actividades contiene: *Estados de Actividad, Estados de Acción, Transiciones y Objetos.*

La representación de un estado de actividad y un estado de acción es un rectángulo con las puntas redondeadas, en cuyo interior se representa bien una actividad o bien una acción.

Un *estado de acción* es un elemento atómico que corresponde a una acción simple.

Un *estado de actividad* se puede descomponer en más sub-actividades representadas a través de otros diagramas de actividades, también se caracteriza por presentar acciones de entrada (entry) y de salida (exit) [3].

1.4 BASES DE DATOS

Una **Base de Datos** es un conjunto de datos almacenados entre los que existen relaciones lógicas y, ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. Donde los datos se interpretan como hechos conocidos que pueden registrarse y que tienen un significado implícito.

Una Base de Datos tiene las siguientes propiedades implícitas:

- Representa algún aspecto del mundo real, en ocasiones llamado *minimundo* o universo de discurso.

- Toda Base de Datos se diseña, construye y puebla con datos para un propósito específico

Una característica fundamental del enfoque de Bases de Datos es que el sistema no sólo contiene la Base de Datos misma, sino también una definición o descripción completa de la Base de Datos. Esta información se almacena en el catálogo del sistema, que contiene información como la estructura de cada archivo, el tipo y formato de almacenamiento de cada elemento de información almacenada en el catálogo se denomina metadatos.

Ventajas de las Bases de Datos.

- Control de la redundancia.

- Menos uso de espacio.

- Restricción de accesos.

- Almacenamiento persistente de objetos.

- Múltiples interfaces.

- Relaciones complejas entre datos.

- Restricciones de integridad.

- Copias de seguridad y recuperación.

La generación y el mantenimiento de las Bases de Datos pueden ser manuales o mecánicos; por ejemplo el catálogo en tarjetas de una biblioteca es un ejemplo de Base de Datos que se puede crear y mantener manualmente. Las Bases de Datos computarizadas se pueden crear y mantener con un grupo de programas de aplicación escritos específicamente para esa tarea, o bien mediante un sistema de gestión de Bases de Datos.

1.4.1 SISTEMA DE GESTION DE BASE DE DATOS.

Un **Sistema de Gestión de Base de Datos** SGBD (*Database Management System: DBMS*) es un sistema de software de propósito general que facilita el proceso de definir, construir y manipular Bases de Datos para diversas aplicaciones además de proporcionar acceso controlado a la misma. Para definir una Base de Datos hay que especificar los tipos de datos, las estructuras y las restricciones de los datos que se almacenarán en ella.

Construir una Base de Datos es el proceso de guardar los datos mismos en algún medio de almacenamiento controlado por el SGBD.

En la manipulación de una Base de Datos intervienen funciones como consultar la Base de Datos para obtener datos específicos, actualizar la Base de Datos para reflejar cambios en *minimundo* y generar informes a partir de los datos.

El software del SGBD no está escrito para una aplicación de base de datos específica, así que tiene que consultar un catálogo para conocer la estructura de los archivos de una Base de Datos en particular.

El principal criterio que suele utilizarse para clasificar los SGBD es el modelo de Datos en que se basan. Los modelos de datos empleados con mayor frecuencia en los SGBD comerciales actuales son el relacional, el de red y el jerárquico; el más reciente se basa en modelos orientados a objetos [4].

Tipos de manejadores de bases de datos o SGBD.

Algunos tipos de base de datos, se describen los siguientes:

MySql: Es un manejador de base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.

PostgreSql y Oracle: Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y son utilizados en intranets y sistemas de gran calibre.

Access: Es un manejador de base de datos desarrollada por Microsoft. Esta base de datos, debe es creada bajo el programa access, el cual crea un archivo .mdb.

Microsoft SQL Server: Es un manejador de base de datos más potente que Access desarrollado por Microsoft. Se utiliza para manejar grandes volúmenes de información.

1.4.2 ABSTRACCIÓN DE DATOS

Uno de los propósitos de un Sistema de Bases de Datos es proporcionar a los usuarios una visión *abstracta* de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

Existen tres niveles de abstracción:

- ✓ *Nivel físico*: el nivel más bajo de abstracción describe cómo se almacenan realmente los datos. En el nivel físico se describen a detalle las estructuras de datos complejas de bajo nivel.
- ✓ *Nivel lógico*: el siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. Los administradores que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.
- ✓ *Nivel de vistas*: el nivel más alto de abstracción describe sólo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. El sistema puede proporcionar muchas vistas para la misma base de datos [5].

1.4.3 ESQUEMAS.

La colección de información almacenada en la base de datos en un momento particular se denomina un ejemplar de la base de datos. El diseño completo de la base de datos se llama esquema.

Para que el sistema sea útil, debe recuperar los datos eficientemente. Como muchos usuarios de sistemas de bases de datos no están familiarizados con computadoras, los desarrolladores esconden la complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema.

Los esquemas son raramente modificados, si es que lo son alguna vez. Existen diferentes tipos de esquemas según los niveles de abstracción: físico, lógico y de vistas que se corresponden con los niveles físico, lógico y de vistas mencionados anteriormente.

1.4.4 INDEPENDENCIA DE LOS DATOS.

La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella. Existen dos niveles de independencia de datos.

Independencia física de datos: Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación. Las modificaciones en el nivel físico son ocasionalmente necesarias para mejorar el funcionamiento.

Independencia lógica de datos: Capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación. Las modificaciones en el nivel lógico son necesarias siempre que la estructura lógica de la base de datos se altere.

La independencia de datos lógica es más fácil de proporcionar que la independencia de datos física, ya que los programas de aplicación son fuertemente dependientes de la estructura lógica de los datos a los que ellos acceden.

El concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos. La independencia de datos implica un divorcio entre programas y datos [5].

1.4.5 MODELOS DE DATOS

Modelo de datos: es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. Los diferentes modelos de datos se clasifican en tres grupos diferentes:

- Modelos lógicos basados en objetos.
- Modelos lógicos basados en registros.
- Modelos físicos.

1.4.6 MODELO ENTIDAD-RELACION

Este modelo se obtiene en tiempo de diseño de la base de datos. Fue propuesto por Peter Chen en 1976 y desde entonces se viene utilizando de una forma muy global. Se caracteriza por utilizar una serie de símbolos y reglas para representar los datos y sus relaciones.

Con este modelo conseguimos representar de manera gráfica la estructura lógica de una base de datos. Los principales elementos del modelo entidad-relación son las entidades con sus atributos y las relaciones entre entidades.

Entidad.

Una entidad es un objeto en el mundo real que es distinguible de todos los demás.

Relación.

Una relación es una asociación entre diferentes entidades representativas del mundo real.

Atributo.

Los atributos describen propiedades que posee cada entidad, se clasifican en simples y compuestos, univaluados (monoevaluados) y multivaluados (multievaluados), nulos y derivados.

Diagrama Entidad-Relación.

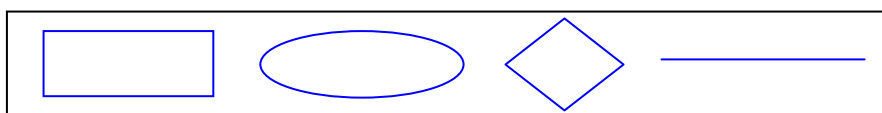
Es la representación gráfica de la estructura lógica general de una base de datos y sus componentes se describen a continuación y en la Fig. 1.3:

Rectángulos: que representan conjuntos de entidades.

Elipses: que representan relaciones entre una entidad.

Rombos: que representan relaciones entre conjuntos de entidades.

Líneas: que en los atributos son los conjuntos de entidades y los conjuntos de entidades con las relaciones



[Fig.1.3 Representación Gráfica de elementos del Modelo Entidad Relación]

1.4.7 MODELO RELACIONAL

Los modelos de datos son el instrumento principal para ofrecer abstracción de los datos. Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Contienen también un conjunto de operaciones básicas para la realización de consultas (lecturas) y actualizaciones de datos.

Los modelos de datos más modernos incluyen conceptos para especificar comportamiento, permitiendo especificar un conjunto de operaciones definidas por el usuario.

Principios básicos del modelo relacional:

1. La estructura de datos relacional y las reglas de integridad.
2. Álgebra Relacional: conjunto de operaciones para manipular la estructura de datos relacional y especificación de consultas de datos.
3. Lenguaje de Consulta SQL

El modelo relacional se basa en dos ramas de las matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden. El hecho de que el modelo relacional esté basado en la teoría de las matemáticas es lo que lo hace tan seguro y robusto. Estas ramas de las matemáticas proporcionan los elementos básicos necesarios para crear una base de datos relacional con una buena estructura, y proporcionan las líneas que se utilizan para formular buenas metodologías de diseño.

La estructura de los datos en el modelo Relacional se presenta en 3 aspectos: **Estructura de datos**, **Integridad de datos** y **Manejo de datos**.

Estructura de datos.

El modelo relacional se basa en el concepto matemático de relación. Una relación es una tabla con columnas y filas. Un SGBD sólo necesita que el usuario pueda percibir la base de datos como un conjunto de tablas.

En el modelo relacional, las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos.

Una **relación** se representa gráficamente como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas corresponden a los campos o atributos de esos registros. Los atributos pueden aparecer en la relación en cualquier orden.

Hay que tener en cuenta que un **atributo** es el nombre de una columna de una relación y que un dominio es el conjunto de valores legales de uno o varios atributos. Los dominios constituyen una poderosa característica del modelo relacional. Cada atributo de una base de datos relacional se define sobre un dominio, pudiendo haber varios atributos definidos sobre el mismo dominio.

Una **tupla** es una fila de una relación; donde los elementos de una relación son las tuplas o filas de la tabla.

El **grado de una relación** es el número de atributos que contiene. Esto quiere decir que cada fila de la tabla es una tupla con seis valores. El grado de una relación no cambia con frecuencia.

La **cardinalidad** de una relación es el número de tuplas que contiene. Ya que en las relaciones se van insertando y borrando tuplas a menudo, la cardinalidad de las mismas varía constantemente.

Una base de datos relacional es un conjunto de relaciones normalizadas.

Propiedades de las relaciones.

Cada relación tiene un nombre y éste es distinto del nombre de todas las demás.

Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor.

Se dice que las relaciones están normalizadas.

No hay dos atributos que se llamen igual.

El orden de los atributos no importa: los atributos no están ordenados.

Cada tupla es distinta de las demás: no hay tuplas duplicadas.

El orden de las tuplas no importa: las tuplas no están ordenadas.

Una **clave o llave** se distingue porque se puede identificar de modo único en una relación ya que no hay tuplas repetidas; la forma de identificarlas es mediante los valores de sus atributos.

Una **superclave** es un atributo o un conjunto de atributos que identifican de modo único las tuplas de una relación.

Una **clave candidata o llave candidata** es una superclave en la que ninguno de sus subconjuntos es una superclave de la relación, es decir se caracteriza por ser única y no se puede eliminar de algún componente sin destruir la unicidad a esto último se le conoce como minimalidad.

Una **clave ajena o llave foránea** es una superclave en la que ninguno de sus subconjuntos es una superclave de la relación.

Cuando en una tupla un atributo es desconocido, se dice que es **nulo**. Un valor nulo no representa el valor cero ni la cadena vacía, éstos son valores que tienen significado; el nulo implica ausencia de información, bien porque al insertar la tupla se desconocía el valor del atributo, o bien porque para dicha tupla el atributo no tiene sentido.

Ya que los nulos no son valores, deben tratarse de modo diferente, lo que causa problemas de implementación. De hecho, no todos los SGBD relacionales soportan los nulos.

Integridad de datos.

La **integridad de los datos** se debe cumplir todo el tiempo en todas las bases de datos relacionales y en todos sus estados o instancias para garantizar que son correctos y para ello se tienen las siguientes reglas:

Restricciones de dominios.- Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo.

Regla de integridad de entidades.- Se aplica a las claves primarias de las relaciones base y establece que ninguno de los atributos que componen la clave primaria puede ser nulo.

Por definición, una clave primaria es un identificador irreducible que se utiliza para identificar de modo único las tuplas. Irreducible significa que ningún subconjunto de la clave primaria sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas, con lo que se contradice la irreducibilidad.

Esta regla sólo se aplica a las relaciones base y a las claves primarias, no a las claves alternativas.

Regla de integridad referencial.- Se aplica a las claves ajenas (foráneas) y establece que si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia.

La regla de integridad referencial se enmarca en términos de estados de la base de datos: indica lo que es un estado ilegal, pero no dice cómo puede evitarse. La cuestión es ¿qué hacer si estando en un estado legal, llega una petición para realizar una operación que conduce a un estado ilegal? Existen dos opciones: rechazar la operación, o bien aceptar la operación y realizar operaciones adicionales compensatorias que conduzcan a un estado legal.

Manejo de datos.

Las funciones de una Base de Datos son: permitir la introducción de datos por parte de los usuarios, el almacenamiento, salida, protección y elaboración de datos, como se ve en la Fig. 1,4.



[Fig. 1.4 Manejo de Datos]

Básicamente la comunicación del usuario-programador con la base de datos se hace a través de un lenguaje de manejo de datos.

Los lenguajes de manejo o manipulación de Datos pueden ser *procedurales* o *no procedurales*.

Los *lenguajes procedurales* manipulan la base de datos registro a registro y se especifica qué operaciones se deben realizar para obtener los datos de resultado.

Los *lenguajes no procedurales* operan sobre conjuntos de registros y se especifica qué datos deben obtenerse sin decir cómo hacerlo. Uno de estos lenguajes no procedurales es el Lenguaje Estructurado de Consultas **SQL** (*Structured Query Language*) que se caracteriza por proporcionar un acceso controlado a la base de datos mediante:

-
1. Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos
 2. Un sistema de integridad que mantiene la integridad y la consistencia de los datos;
 3. Un sistema de control de concurrencia que permite el acceso compartido a la base de datos;
 4. Un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del hardware o del software;
 5. Un diccionario de datos o catálogo accesible por el usuario que contiene la descripción de los datos de la base de datos [6].

1.4.7.1 Normalización o Formas Normales.

Las formas normales no son más que tres reglas que se deben tener en cuenta dentro del Análisis conceptual, utilizando concretamente Entidad/Relación.

El proceso de aplicar las tres formas normales se llama normalización. Un diseño de base de datos que no cumpla la primera forma normal no será correcto. Cuantas más formas normales cumpla el diseño de base de datos, significará que la base de datos:

1. Evita anomalías en inserciones, modificaciones y borrados.
2. Mejora la independencia de datos.
3. No establece restricciones artificiales en la estructura de los datos.

Primera forma normal:

Una relación (tabla) está en Primera Forma Normal si los valores para cada atributo son atómicos. Es decir que cada atributo sólo puede pertenecer a un dominio y que tiene un valor único para cada fila.

Segunda forma normal:

Una relación esta en Segunda Forma Normal si esta en Primera Forma Normal y todos los atributos no clave dependen de la clave completa y no sólo de una parte de esta. Aplica para aquellas relaciones que tienen claves compuestas, es decir que están formadas por más de un atributo.

Tercera forma normal:

Una relación esta en Tercera Forma Normal si y sólo si, están en Segunda Forma Normal y además cada atributo que no está en la llave primaria no dependen transitivamente de la clave primaria [7].

1.5 MODELO CLIENTE - SERVIDOR

Un aspecto interesante de la tecnología web es el concepto CLIENTE-SERVIDOR. En Internet existen 2 tipos de equipos conectados:

Servidores.- Computadoras destinadas a ofrecer sus servicios a cualquier equipo que establezca una conexión con ellos. Se caracterizan por tener una presencia estable, por tener asignadas direcciones IPs permanentes y en ellos “viven” las páginas web.

Clientes.- Equipos que solicitan servicios a los servidores. Se caracterizan porque durante el tiempo de conexión tienen presencia física en la red, cuentan con una dirección IP variable durante su conexión ya que cambia de unas conexiones a otras, usualmente a esta característica se le conoce como IP dinámica.

Existen también otros conceptos asociados al software que manejan los ordenadores tal como:

Programa Servidor.- Implica que su ejecución permanece en el servidor con la finalidad de que éste pueda ofrecer sus servicios en la red, a esto se le conoce como aplicación o servidor web, alguno de ellos son Apache, Personal Web Server; también los hay para correo electrónico tales como Mozilla Thunderbird, Outlook Express; o de transferencia de archivos como es FTP.

Programa Cliente.- Es el software requerido del lado del cliente para tener acceso a determinado servicio, por ejemplo Internet Explorer, Opera, Mozilla, para correo electrónico Outlook, Lotus Notes y para transferencia de Ficheros esta CuteFTP o ssh.

Existen dos formas o categorías de programar la estructura Cliente-Servidor:

Programación del lado del Cliente.- Aquí los programas residen en el servidor junto a la página web y son transferidos al cliente para ser ejecutados desde su lado.

Programación del lado del Servidor.- Aquí los programas son ejecutados por el servidor y sólo se envía al cliente la respuesta o resultado de dicha ejecución.

Algunas diferencias importantes se muestran en la siguiente tabla.

<i>Programación del lado del Cliente</i>	<i>Programación del lado del Servidor</i>
Se descarga de trabajo a los servidores	El trabajo queda del lado del servidor con el riesgo de sobrecargarlo
La ejecución del programa requiere de una transmisión por la red del código a ejecutar	El cliente solo recibe el resultado de la ejecución del programa.
Los resultados son de fácil acceso para el usuario ya que no requiere de una transmisión en la red	Después del envío de algún resultado al usuario cualquier petición posterior del cliente requerirá de una nueva conexión al servidor y la ejecución de un nuevo programa

Instalación del software requerido por el cliente para la correcta ejecución del programa en el cliente	En los equipos de los clientes no se necesita ningún software especial.
El cliente tiene acceso al código y se compromete la información del mismo	El código fuente conserva su privacidad y ningún cliente tiene acceso a él
Fácil integración de los programas de las páginas en cualquier servidor web	Los servidores de alojamiento de páginas no admiten este tipo de programación.

Al igual que Internet, la programación, sus lenguajes y las Bases de Datos han tenido una evolución desde conceptos y paradigmas hasta en la forma de pensar un problema a resolver por medio de un programa; prueba de ello es el paradigma Orientado a Objetos.

Para este proyecto se utilizará una arquitectura Cliente – Servidor empleando las herramientas Microsoft Visio, Oracle10g, IBM Websphere Application Server, IBM Rational Application Developer, además de trabajar bajo el framework de Struts 2, lo que se detalla en la siguiente sección [1].

1.6 PLATAFORMA

1.6.1 MICROSOFT VISIO

Software que proporciona una amplia gama de plantillas, diagramas de flujo de procesos empresariales, diagramas de redes, diagramas de flujo de trabajo, modelos de bases de datos y diagramas de software, que pueden utilizarse para ver y racionalizar procesos empresariales, realizar el seguimiento de proyectos y recursos, crear organigramas, generar mapas de redes, confeccionar diagramas para la creación de sitios y optimizar sistemas.

En el proyecto se utilizó para generar los diagramas de caso de uso.

1.6.2 ORACLE 10g

Sistema de gestión de base de datos relacional (SGDB), considerado como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

Oracle es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones.

1.6.3 TOAD.

Aplicación de Software de desarrollo SQL y administración de Base de Datos, permite generar y ejecutar consultas, crear y modificar objetos de la base de datos y desarrollar y depurar código SQL y PL/SQL.

Esta herramienta nos permitirá interactuar con la Base de Datos generada para el sistema.

1.6.4 IBM Websphere Application Server (WAS)

Software de infraestructura para integración y aplicación o servidor de aplicaciones de software que provee la disponibilidad y seguridad de los sistemas optimizando su costo. Construye, depura y administra de manera robusta, ágil y reutilizable las aplicaciones y servicios web. Esta construido en base utilizando estándares abiertos tales como J2EE, XML y Servicios Web; a su vez funciona con varios servidores web como APACHE, HTTP SERVER, NETSCAPE por nombrar algunos.

1.6.5 IBM Rational Application Developer (RAD)

Es un entorno de desarrollo integrado para el diseño visual, construcción, pruebas y despliegue de aplicaciones, servicios y portales web y aplicaciones J2EE. Está basado en el IDE de Eclipse y soporta sus extensiones. Incluye herramientas y editores para trabajar con: conexiones a bases de datos y SQL, lenguaje de programación EGL (Enterprise Generation Language), HTML, Java, JavaServer Faces y JavaServer Pages, UML, Servicios web y XML.

1.6.6 STRUTS.

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE (Java 2, Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Struts permite reducir el tiempo de desarrollo. Su carácter de "software libre" y su compatibilidad con todas las plataformas en las que Java Enterprise esté disponible lo convierten en una herramienta altamente disponible.

Con la versión 2 del framework se introdujeron algunas mejoras sobre la primera versión, de cara a simplificar las tareas más comunes en el desarrollo de aplicaciones web, así como mejorar su integración con AJAX, etc.

Un framework es una estructura de soporte definida, mediante la cual un proyecto de software puede ser organizado y desarrollado.

El patrón MVC (Model-View-Controller) o Modelo Vista Controlador es un patrón de diseño aportado originariamente por el lenguaje SmallTalk a la Ingeniería del Software, y consiste principalmente en dividir las aplicaciones en tres partes:

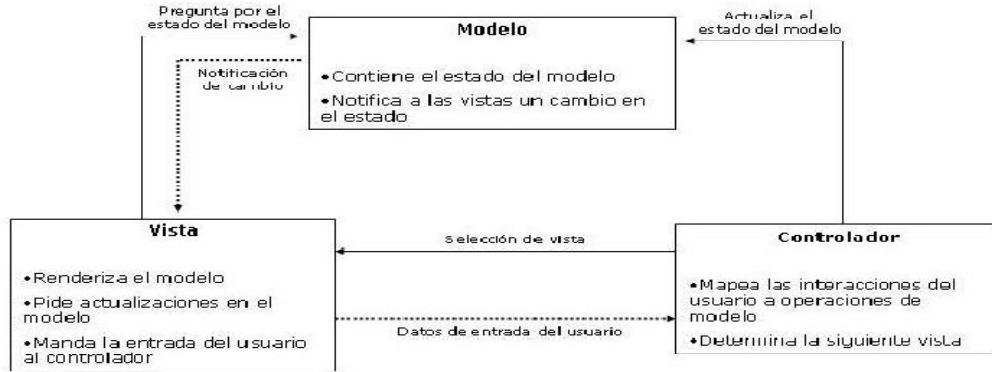
- Controlador
- Modelo
- Vistas.

El controlador es el encargado de redirigir o asignar una aplicación a cada petición y debe poseer de algún modo, un "mapa" de correspondencias entre peticiones y respuestas que se les asignan.

El modelo seria la lógica de negocio a fin de cuentas.

Una vez realizadas las operaciones necesarias el flujo vuelve al controlador y este devuelve los resultados a una vista asignada.

La siguiente figura 1.5 nos muestra la interacción entre el Modelo la Vista y el Controlador:



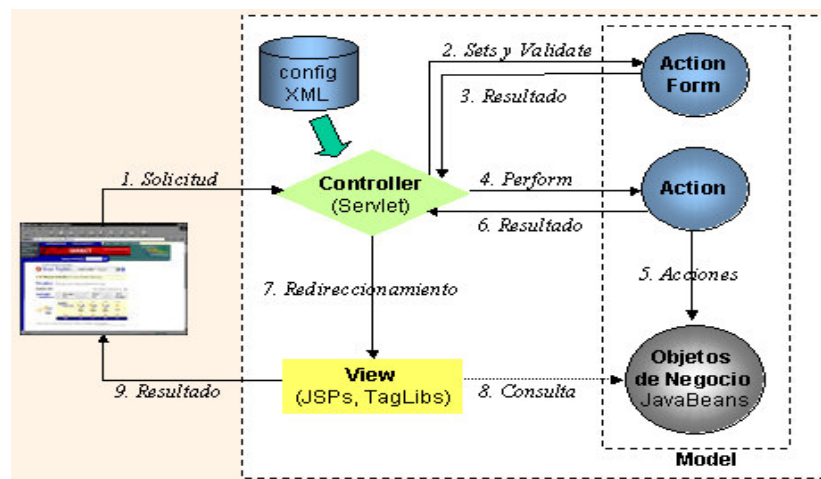
[Fig. 1.5 Patrón MVC]

Struts.

Como anteriormente se mencionó Struts implementa el patrón MVC por lo que a continuación se describe el flujo del patrón bajo Struts.

El navegador genera una solicitud que es atendida por el Controlador (un Servlet especializado). El mismo se encarga de analizar la solicitud, seguir la configuración que se le ha programado en su XML y llamar al Action correspondiente pasándole los parámetros enviados.

El Action instanciará y/o utilizará los objetos de negocio para concretar la tarea. Según el resultado que retorne el Action, el Controlador derivará la generación de interfaz a una o más JSPs, las cuales podrán consultar los objetos del Modelo para mostrar información de los mismos, lo anterior se puede apreciar en la siguiente figura 1.6 [8].

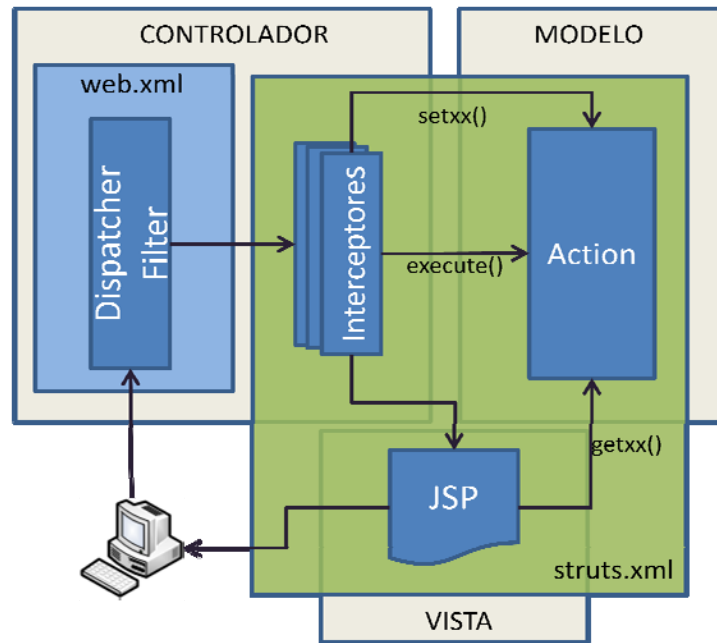


[Fig. 1.6 Framework de Struts]

Struts 2

La filosofía de trabajo en Struts2 basa en 3 conceptos “construir” (build), “desplegar” (deploy) y “mantener” (maintain).

La versión 2 de Struts incluye mejoras y cambios entre las que se encuentran: diseño mejorado, nuevos tags incorporados, mejora en la gestión de checkboxes, botones para cancelar flexibles, formularios POJO (*Plain Old Java Object*) y en lugar de los ActionFormy acciones POJO [9].



[Fig. 1.7 Framework de Struts2]

Capítulo 2.

A NÁLISIS DEL SISTEMA.

Para lograr los objetivos que se han planteado previamente es necesario comprender e indagar la lógica de lo que se necesita implementar y esta será la tarea de este capítulo.

2.1 PLANTEAMIENTO DEL PROBLEMA.

2.1.1 ANTECEDENTES.

El sistema es requerido para una empresa dedicada al rubro de la telefonía celular, cuya necesidad es realizar transferencias de equipos celulares y accesorios. Hoy en día no hay un almacén que tenga la función de mantener todos los accesorios y equipos recién adquiridos, y que, por medio de una selección se transfieran dichos artículos registrando los movimientos hacia aquellos almacenes involucrados.

2.1.2 DESCRIPCIÓN DEL PROBLEMA.

En base a lo descrito anteriormente, se requiere de un sistema vía web que controle las transacciones de:

- ✓ Generación de propuestas para adquisición de accesorios y equipos celulares por parte de la empresa fabricante, por medio de un Representante al que llamaremos *Representante del Fabricante*.
- ✓ El proceso de análisis o estudio para aceptar o rechazar dicha propuesta por parte del personal de la empresa, al que identificaremos como *Personal de Inventario*.
- ✓ La carga inicial o transferencia inicial de los artículos (accesorios y equipos celulares) involucrados en la propuesta hacia un Almacén Central.
- ✓ La distribución de accesorios y equipos celulares realizada previamente por selección entre almacenes de diferentes sucursales e incluso del Almacén Central a una de éstas.

Para concluir el flujo de la operación hay que considerar una serie de premisas y eventos o transacciones que se describen a detalle.

Los usuarios que intervienen en el sistema son:

El representante del fabricante encargado de generar la propuesta de accesorios y equipos celulares.

El personal de inventario de la empresa encargado de revisar, analizar y en su caso aceptar o rechazar las propuestas de oferta generadas por el representante de ventas del distribuidor.

El gerente de cada sucursal cuya función es la de seleccionar los accesorios y equipos que requiere como existencias para el almacén de su sucursal.

Un administrador encargado de generar y controlar los accesos para los representantes de ventas del fabricante, el personal de inventarios y el gerente de la sucursal.

Por lo tanto el representante de ventas del fabricante o distribuidor, así como el personal de inventario y el gerente de la sucursal contarán con un identificador y un password, que el sistema validará para determinar el tipo de operador y las tareas que pueden realizar dentro del sistema. En caso de que no se cuente con un identificador, el sistema le informará al interesado el procedimiento correcto para solicitar el acceso.

El flujo parte con la publicación de una propuesta u oferta, tarea correspondiente al representante del fabricante; dicha propuesta pasará por varios “estados” para indicarnos en que parte del flujo se encuentra. Cuando una propuesta es generada nace con un estado de “*En evaluación*”; posteriormente se puede encontrar en estado de “*Rechazada*”, “*Aceptada*” o “*Cancelada*”. Cabe mencionar que una propuesta tendrá una vigencia de 15 días a partir de ser generada.

Los datos que sustentan una propuesta son:

- ✓ un folio
- ✓ un estado
- ✓ una fecha de origen
- ✓ un motivo
- ✓ un registro de cuantas veces ha sido modificada una propuesta
- ✓ el nombre o identificador de quien genera la propuesta
- ✓ el nombre o el identificador de la persona asignada por el sistema encargada de analizar la propuesta por parte de la empresa
- ✓ el identificador del usuario que realizó la última modificación sobre la propuesta
- ✓ fecha de modificación
- ✓ marca del fabricante
- ✓ tipo de accesorio o equipo
- ✓ descripción
- ✓ modelo
- ✓ cantidad
- ✓ precio unitario
- ✓ el sistema se encargará de calcular el precio unitario con IVA, subtotal y total

En el momento que el representante de ventas considere que ha concluido la captura, podrá guardarla en el sistema por medio de un botón Generar, en este punto inicia la operación y el sistema enviará un correo electrónico a la persona de inventarios asignada por el sistema, para informar que hay una nueva propuesta y que necesita ser estudiada o analizada.

El sistema también dará la opción de cancelar la propuesta si el representante del fabricante encuentra algún error durante la captura.

La persona de Inventario recibirá el correo electrónico correspondiente a la propuesta indicando:

- ✓ Un folio
- ✓ El nombre del representante de ventas del fabricante
- ✓ El fabricante

-
- ✓ El estado de la propuesta, que en este punto será *"En Evaluación"*

En el momento en que el personal de inventario acceda al sistema éste le mostrará en pantalla la lista de propuestas por estado y permitirá navegar a aquellas con estado *"En Evaluación"*, de las cuales el podrá seleccionar la que le interese con el objetivo de analizarla y decidir si es *"Aceptada"* o *"Rechazada"*, además de capturar el motivo de rechazo, en su caso.

El estado *"Rechazada"* de una propuesta se asignará cuando, para el personal la propuesta no es viable, y, para este caso el sistema enviará un correo electrónico para notificar al representante de ventas del fabricante, que la propuesta ha sido rechazada, indicando:

- ✓ El folio
- ✓ El personal asignado que estudio la propuesta y,
- ✓ El estado de la propuesta como *"Rechazada"*

Una propuesta *"Rechazada"* puede ser modificada por el representante del Fabricante, y para ello tendrá que acceder al sistema, el cual le indicará el número de propuestas *"Rechazadas"*, *"Aceptadas"*, y *"Canceladas"* asociadas a él.

Además el sistema también le podrá dar la opción de capturar algún dato conocido de una propuesta para mostrar su detalle.

El sistema también permitirá al representante de ventas modificar la propuesta siempre y cuando dicha propuesta todavía se encuentre vigente, en caso contrario el sistema sólo mostrará el detalle, sin permitir modificación alguna, por lo que el representante del fabricante en caso de estar interesado tendrá que generar una nueva propuesta.

Cuando una propuesta se modifica, el sistema le permitirá al representante del Fabricante cambiar algunos puntos de la propuesta, el estado de la propuesta se modifica a *"En evaluación"* e iniciará nuevamente el flujo de aceptación o rechazo de la misma.

Una propuesta también puede situarse en estado *"Rechazada"* debido a que el personal asignado de la empresa, está interesado en la oferta pero requiere alguna modificación en el detalle de la misma, lo cual se indicará en el campo de motivo. Este caso por ejemplo se refiere a incrementar o disminuir las cantidades de accesorios o equipos ofertados en la propuesta, o eliminar un producto que no requiere la empresa.

Igual que para el punto en que la propuesta no es viable, el sistema modificará el estado de la propuesta a *"En evaluación"* y enviará un correo electrónico para notificar al representante de ventas del fabricante, que la propuesta requiere de algunos cambios, indicando:

- ✓ El folio
- ✓ El personal asignado que estudió la propuesta
- ✓ El estado de la misma

También se le permitirá al representante del fabricante modificar la propuesta siempre y cuando ésta, todavía se encuentre vigente, en caso contrario el sistema sólo mostrará el detalle, sin

permitir modificación, por lo que el representante del fabricante tendrá que generar una nueva propuesta.

Si la propuesta se encuentra vigente el sistema permite la modificación e inicia el flujo nuevamente.

En caso de que la propuesta sea *“Aceptada”*, el sistema pedirá al personal que indique la forma de pago para la misma, y solicitará una confirmación al personal, por lo que en este punto aún se puede cambiar el estado de la propuesta a *“Rechazada”*.

Al confirmar la forma de pago el sistema enviará un correo electrónico para notificar al representante de ventas del fabricante, que la propuesta ha sido aceptada, indicando:

- ✓ El folio
- ✓ El personal asignado que estudio la propuesta
- ✓ El estado correspondiente

El representante de ventas accederá al sistema y al enfocar esta solicitud para ver su detalle también se incluirá la indicación de forma de pago.

Cuando una propuesta ya ha sido *“Aceptada”* y se encuentre vigente, el representante del fabricante tendrá la opción de modificar la propuesta para reofertar, y el sistema cambiará su estado a *“En evaluación”*.

También el sistema dará sólo la opción de *“Cancelar”* en caso de que la propuesta se encuentre fuera de período de vigencia, o el representante determine cancelarla especificando un motivo.

Para el caso en que una propuesta ya fue *“Aceptada”* el sistema desplegará en pantalla el detalle de la propuesta y agregará los campos necesarios para capturar el identificador de fábrica asociado al accesorio o al equipo celular, esto se realizará para cada artículo solicitado.

Al terminar la captura el sistema pedirá al representante la confirmación de los datos y almacenará esta información; inmediatamente el sistema enviará un correo electrónico a la persona de inventarios para informar que tiene artículos por aceptar provenientes de una *transferencia o Primera carga* con un folio determinado.

Cuando el personal de inventario ingrese al sistema éste le mostrará el detalle de la transferencia y solicitará la aceptación de los artículos con la finalidad de que el sistema los registre en un Almacén Central, para que puedan ser consultados y posteriormente asignados a algún almacén de una sucursal.

Las consultas sobre el Almacén Central se podrán realizar en base al tipo de artículo (fundas, cargadores, manos libres, equipos, etc.), descripción del artículo, fabricante, identificador de fábrica o modelo del artículo sí lo tiene.

Para los gerentes de las sucursales, después de firmarse al sistema, éste les mostrará la pantalla con las opciones de consulta sobre el Almacén Central, el Almacén correspondiente a la sucursal a

la que corresponden, y las opciones de búsqueda descritas anteriormente. En la opción de consulta sobre el Almacén Central, y al mostrar los artículos, resultado de la búsqueda, el sistema le permitirá seleccionar los artículos que les interesen para transferir a su almacén, así, en el momento que él decida pueda presionar el botón ya sea de “Cancelar” o de “Transferir”.

Posteriormente el sistema pedirá confirmación sobre los artículos que se moverán hacia el almacén de la sucursal y para identificar el movimiento el sistema generará un registro histórico para el movimiento y para los artículos involucrados el cual incluye:

- ✓ Un folio
- ✓ Un almacén origen
- ✓ Un almacén destino
- ✓ Un identificador de fábrica o código de barras que asocie a los artículos con la transacción

Cuando un artículo ha sido transferido del Almacén Central a un almacén de alguna sucursal, el artículo quedará asociado a este nuevo almacén con un estado de “Existencia”.

Sí un artículo no es vendido el gerente podrá regresarlo al Almacén Central, firmándose al sistema y seleccionando la opción del Almacén de su sucursal y Traspaso de Equipo, además tendrá que seleccionar al Almacén Central como almacén destino. Para terminar la transacción presionará el botón de “Transferir”, con su correspondiente confirmación y registro.

2.2 ESPECIFICACION DE REQUERIMIENTOS

Los requerimientos obtenidos de la descripción del problema son:

R1.- El representante de ventas del fabricante o distribuidor como el personal de inventario y el gerente de la sucursal contarán con un identificador y un password, que el sistema validará para determinar el tipo de operador y las tareas que puede realizar dentro del mismo.

R2.- En caso de que no se cuente con usuario o identificador el sistema le informará al interesado el procedimiento correcto para solicitar el acceso.

R3.- Las consultas sobre el Almacén Central o almacén de la sucursal se podrán realizar en base a:

- ✓ Tipo de artículo (fundas, cargadores, manos libres, equipos, etc.)
- ✓ Descripción del artículo
- ✓ Fabricante
- ✓ Identificador del fabricante
- ✓ Modelo del artículo sí lo tiene

R4.- El sistema permitirá transferencias de los almacenes de la sucursal hacia el Almacén Central

R5.- El sistema se encargará de calcular el precio unitario con IVA, monto subtotal y monto total.

R6.- Los datos asociados a una propuesta son:

- ✓ Folio generado por el sistema

-
- ✓ Estado de la propuesta
 - ✓ Una fecha de origen
 - ✓ Un motivo
 - ✓ Un registro de cuantas veces ha sido modificada dicha propuesta
 - ✓ El identificador de quien genera la propuesta
 - ✓ El identificador de la persona de Inventario asignada por el sistema encargada de analizar la propuesta por parte de la empresa
 - ✓ El identificador del usuario que realizó la última modificación sobre la propuesta
 - ✓ Fecha de modificación
 - ✓ Marca del fabricante
 - ✓ Tipo de accesorio o equipo
 - ✓ Descripción
 - ✓ Modelo
 - ✓ Cantidad
 - ✓ Precio unitario
 - ✓ Precio unitario con IVA
 - ✓ Monto subtotal
 - ✓ Monto Total

R6.1.- El sistema almacenará la información anterior tras la selección del botón “*Generar*”, por parte del Representante de ventas del fabricante.

R6.2.- El sistema también dará la opción de “*Cancelar*” la propuesta si el representante del fabricante encuentra algún error durante la captura.

R7.- El sistema se encargará del envío de notificaciones vía correo electrónico en los casos siguientes:

R7.1.- Generación de una propuesta. Los datos a tomar en cuenta para enviar el correo serán un folio de propuesta, el nombre del representante de ventas del fabricante, el fabricante, la fecha de generación de la propuesta y un estado de “*En Evaluación*” de la propuesta.

R7.2.- Propuesta Rechazada- El correo indicará el folio, el personal asignado que estudió la propuesta, la fecha de origen, la fecha de modificación y el estado.

R7.3.- Propuesta Aceptada.- El correo indicará el folio, el personal asignado que estudió la propuesta, la fecha de origen, la fecha de modificación y el estado.

R7.4.- Notificación de Primera Carga.- El objetivo del correo será informar que el personal asignado de inventarios, tiene artículos por aceptar provenientes de una propuesta, indicando el folio de la Transacción.

R8.- El sistema le permitirá al Personal de Inventarios:

R8.1.- Ver en pantalla la lista de propuestas por estado y permitirá navegar a aquellas con estado “*En Evaluación*”, “*Rechazadas*”, “*Aceptadas*” o “*Canceladas*” de las cuales el podrá seleccionar la que le interese.

R8.2.- Visualizar el detalle de la solicitud seleccionada con las opciones de “*Aceptar*” o “*Rechazar*” para aquellas que se encuentran en estado de “*En evaluación*”.

R8.3.- Registrar el nuevo estado asignado por el personal de inventario y su motivo si se requiere.

R8.4.- Establecer la forma de pago para la propuesta, y le solicitará una confirmación cuando la propuesta haya sido calificada como "Aceptada".

R8.5.- Aceptar los artículos de la propuesta como parte de la Primera Carga de artículos con la finalidad de que el sistema los registre en un Almacén Central.

R9.- El sistema le permitirá al Representante del Fabricante:

R9.1.- Visualizar el número de propuestas "Rechazadas", "Aceptadas", y "Canceladas" asociadas a él.

R9.2.- Realizar consultas de propuesta por criterios ya sea por folio, estado, fecha de origen, etc. para mostrar su detalle.

R9.3.- Modificar la propuesta siempre y cuando ésta todavía se encuentre vigente, y se encuentre en estado de "Rechazada"

R9.4.- Actualizar el estado de la propuesta a "En evaluación" en caso de ser modificada.

R9.5.- Ver el detalle, sin permitir modificación alguna, de una propuesta si ésta excedió el período de vigencia

R9.6.- Para las propuestas "Aceptadas" al ser enfocadas, el sistema mostrará la forma de pago asignada por el personal de inventario

R9.7.- Cuando una propuesta se encuentre "Aceptada", el sistema desplegará en pantalla el detalle de la propuesta y agregará los campos necesarios para capturar el identificador de fábrica de los accesorios o equipos celulares, esto se realizará para cada artículo solicitado.

R9.8.- Al final el sistema pedirá al representante la confirmación de los datos y almacenará esta información.

R9.9.- El sistema también permitirá que una propuesta "Aceptada" pueda ser:

R9.9.1 Modificada para Reofertar, en este caso el sistema almacenará los cambios y la regresará a estado "En evaluación"

R9.9.2 Cancelada almacenando este estado (Cancelada) y el motivo de Cancelación.

R10.- El sistema le permitirá al Gerente de una sucursal:

R10.1.- Visualizar en pantalla las opciones de consulta sobre: el Almacén Central, el Almacén correspondiente a la sucursal en donde laboran, y las opciones de búsqueda en base al tipo de artículo (fundas, cargadores, manos libres, equipos, etc.), descripción del artículo, fabricante, identificador de fábrica o modelo del artículo si lo tiene.

R10.2.- En los resultados de las consultas generadas, el sistema permitirá seleccionar los artículos que se desee transferir a su almacén de su sucursal, y los botones "Cancelar" y "Transferir", donde:

R10.2.1.- El sistema quitará la selección del artículo en pantalla como parte del funcionamiento del botón "Cancelar"

R10.2.2.- El sistema pedirá confirmación sobre los artículos que se moverán hacia el almacén de la sucursal con las siguientes características:

El sistema generará un registro de datos históricos para el movimiento y para los artículos involucrados el cual incluye:

- ✓ Un folio
- ✓ Un almacén origen
- ✓ Un almacén destino

-
- ✓ Un identificador de fábrica que asocie a los artículos con la transacción
Los artículos que han sido transferidos quedarán con un estado de "*En existencia*".

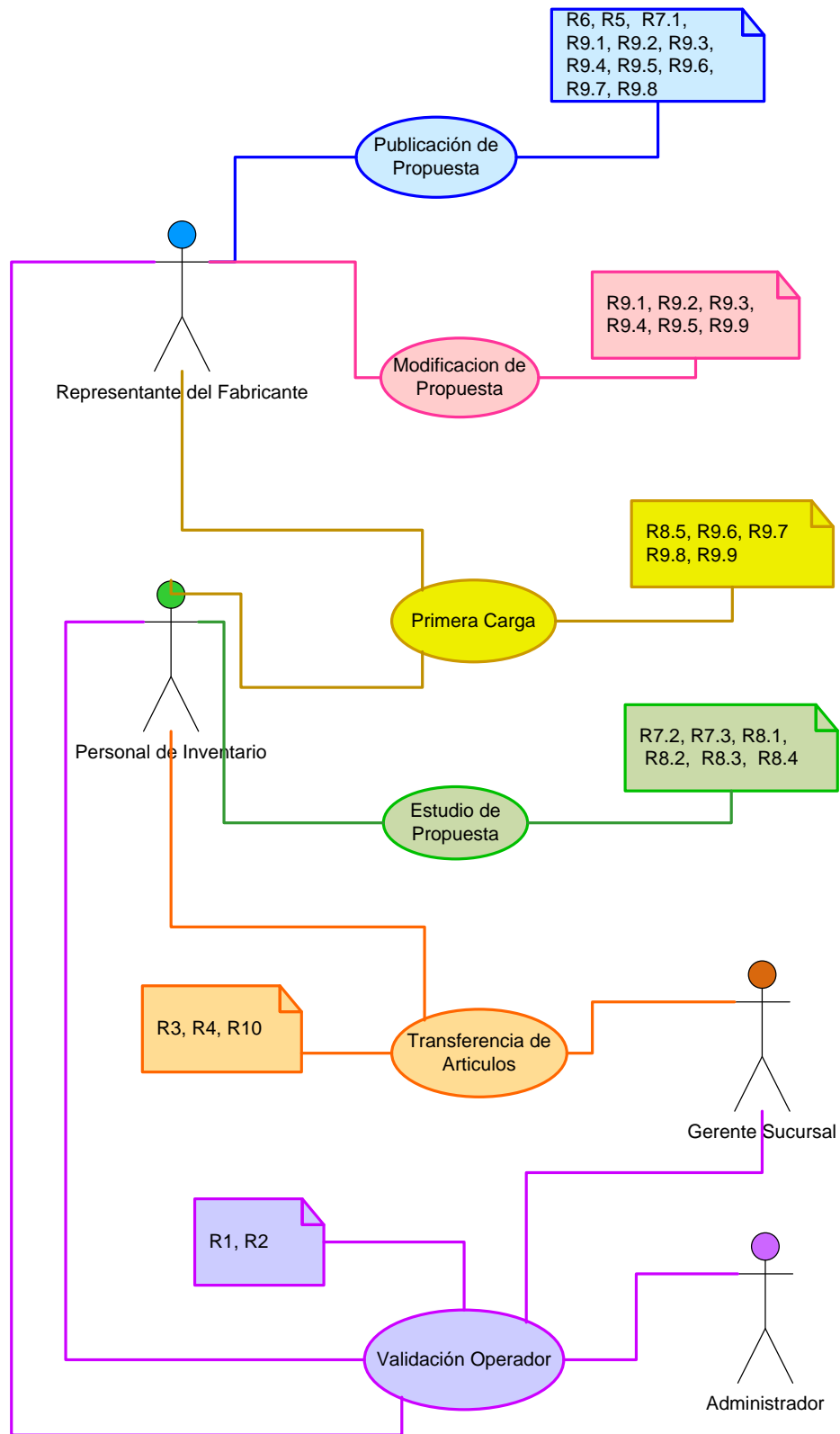
2.3 MODELO DE CASOS DE USO

Para mejorar la definición de los requisitos en la realización de programas, utilizamos la creación de casos de uso, la cual es una descripción narrativa de los procesos. Fundamentalmente el modelo de casos de uso indica quien usa o interactúa con el sistema a desarrollar y como lo hace.

El análisis comprende aquellas actividades que parten de los casos de uso y de los requerimientos funcionales para producir un modelo de análisis del sistema. Mientras que el modelo de diseño es un modelo de objetos describiendo la realización de los casos de uso y sirve como una abstracción del modelo de implementación y código fuente. El modelo de diseño es usado como entrada para las actividades a realizar dentro de la implementación y pruebas del sistema.

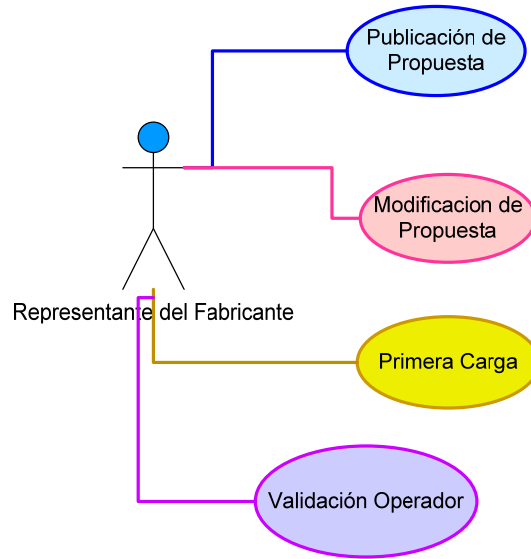
Capturando los requerimientos y usando diagramas de casos de uso en UML, se encontró que los actores que intervienen en el sistema son Representante del Fabricante, Persona de Inventarios, Gerente de Sucursal y el Administrador del sistema.

A continuación se presenta la representación de casos de uso del sistema en general en la figura 2.1, tomando los diagramas de casos de uso para realizar el análisis.

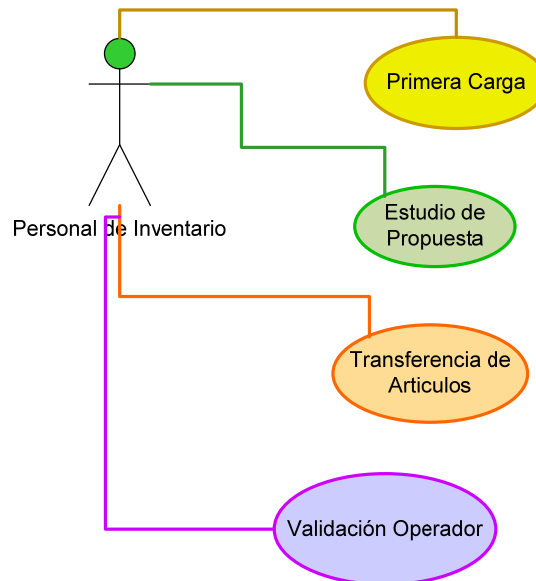


[Fig. 2.1 Casos de Uso – Sistema General]

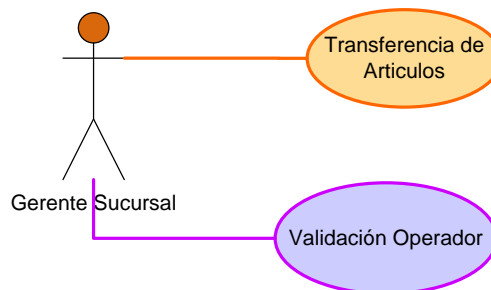
A continuación se muestran los casos de uso específicos asociados a cada actor.



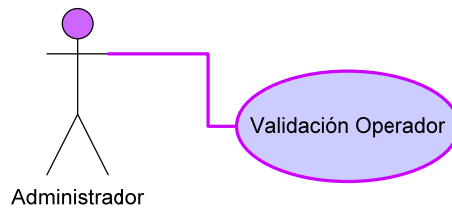
[Fig. 2.2 Casos de Uso que controla el actor Representante de Fabricante]



[Fig. 2.3 Casos de Uso que controla el actor Persona de Inventario]



[Fig. 2.4 Casos de Uso que controla el actor Gerente Sucursal]



[Fig. 2.5 Casos de Uso que controla el actor Administrador]

2.3.1 ESPECIFICACIÓN DE CASOS DE USO.

En esta sección se detalla en base al flujo de eventos cada Caso de Uso.

Caso de Uso:	Publicación de Propuesta.	
Descripción General:	Involucra el flujo que va desde formular la propuesta hasta el almacenamiento de la misma en la Base de Datos.	
Actores:		
Objetivo del Caso de Uso:	Almacenar en la BD la propuesta generada por el representante del Fabricante.	
Precondiciones:	1.- El Representante del Fabricante cuenta con una clave y password solicitados previamente. 2.- El Representante del Fabricante elige la opción "Generar Propuesta".	
Postcondiciones:	1.- Una propuesta nace con un estado de "en evaluación".	
Camino Básico		
Flujo de Eventos	ACTOR	SISTEMA
1	El Representante del Fabricante accede a la sección de generar propuesta.	
2		El sistema muestra o despliega el formulario correspondiente y los botones "Generar Propuesta" y "Cancelar".
3	El Representante del Fabricante llena los campos correspondientes a: 3.1 Seleccionar la marca del Fabricante 3.2 Tipo de accesorio (funda, cargador, manos libres, cables) o equipo celular 3.3 Captura una breve descripción del accesorio o equipo 3.4 Captura el modelo del accesorio o	

	equipo celular 3.5 Cantidad de artículos a promover. 3.6 Captura de Precio Unitario	
4		El sistema calculará 4.1 El precio unitario con IVA 4.2 Importe Subtotal 4.3 Importe Total
5	El Representante del Fabricante decide presionar el botón "Generar Propuesta"	
6		El sistema generará: 6.1 Folio Propuesta 6.2 Fecha de origen de Propuesta 6.3 Inicializar campo contador de modificaciones de la propuesta con valor 1 6.4 Asignará una persona de inventarios para el análisis de la propuesta 6.5 Un Estado Inicial "en evaluacion" para la propuesta. 6.6 Fecha de última modificación de la Propuesta.
7		El sistema almacenará la propuesta en la BD con los datos capturados por el Representante del Fabricante, los generados por el sistema y aquellos que complementaran la información como: 7.1 El idOperador del Representante del Fabricante (quien genera la propuesta). 7.2 El idOperador de quien modificó por última vez la propuesta
8		El sistema desplegará un mensaje para informar que el registro de la propuesta fue exitoso y proporcionará el Folio que se generó para la propuesta y el personal de inventarios asignado.
9		El sistema genera un correo electrónico dirigido tanto al Personal de Inventario como al Representante del Fabricante, con la siguiente información: + Folio de la propuesta + Nombre del Representante del Fabricante. +Fecha de generación de propuesta. +Estado de la propuesta.
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA

4	El Representante del Fabricante se equivoca al llenar algún campo del formulario por lo que presiona el botón "Cancelar"	
5		El sistema limpia los campos del formulario
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA
5	El Representante del Fabricante decide no generar la propuesta por lo que presiona el botón "Cancelar"	
6		El sistema limpia los campos del formulario

Caso de Uso:	Estudio de Propuesta.	
Descripción General:	Involucra el análisis que realizará el personal de inventario sobre la propuesta y el almacenamiento del resultado del estudio en la Base de Datos.	
Actores:	<pre> graph LR Actor[Personal de Inventario] --- UC((Estudio de Propuesta)) Note[R7.2, R7.3, R8.1, R8.2, R8.3, R8.4] --- UC </pre>	
Objetivo del Caso de Uso:	Aceptar o Rechazar la propuesta y almacenar en la BD las modificaciones realizadas.	
Precondiciones:	1.- El Personal de Inventarios cuenta con una clave y password solicitados previamente. 2.- El Personal de Inventarios recibe una notificación de asignación de propuesta por medio de correo electrónico. 3.- El personal de Inventarios accede al sistema e ingresa a la sección de propuesta en evaluación	
Postcondiciones:		
Camino Básico		
Flujo de Eventos	ACTOR	SISTEMA
1	El Personal de Inventarios accede a la sección de propuestas en evaluación.	
2		El sistema despliega un listado por folio de todas las propuestas por evaluar. Mostrando sólo datos Generales como: 2.1 Folio 2.2 Representante del Fabricante que

		<p>generó la propuesta</p> <p>2.3 Compañía</p> <p>2.4 Fecha de origen de la propuesta</p>
3	El Personal de Inventarios selecciona una propuesta por medio de la liga asociada al folio de la propuesta	
4		<p>El sistema despliega en una nueva pantalla el detalle de la propuesta con la información general (mostrada al inicio) además de:</p> <p>4.1 El número de veces que ha sido modificada una propuesta</p> <p>4.2 El nombre de la persona asignada por el sistema como encargada de analizar la propuesta</p> <p>4.3 El idOperador de quien realizó la última modificación sobre la propuesta</p> <p>4.4 Fecha de modificación</p> <p>4.5 Tipo de accesorio o equipo celular</p> <p>4.6 Descripción del accesorio o equipo celular</p> <p>4.7 Modelo</p> <p>4.8 Cantidad</p> <p>4.9 Precio unitario</p> <p>4.10 Precio unitario con IVA</p> <p>4.11 Importe subtotal</p> <p>4.12 Importe total</p> <p>Dentro de esta interfaz también se muestra un Campo texto que indicará el "Motivo", una lista para seleccionar la forma de pago de los accesorio o equipos celulares y dos botones "Aceptar" y "Rechazar"</p>
5	El Personal de inventarios decide Aceptar la propuesta por lo que tiene que seleccionar una forma de pago	
6	El Personal de inventarios presiona el botón "Aceptar"	
7		<p>El sistema generará o modificará:</p> <p>7.1 El estatus de la propuesta a "Aceptada"</p> <p>7.2 Incrementará el número de veces que ha sido modificada una propuesta</p> <p>7.3 El idOperador de quien realizó la última modificación sobre la propuesta</p> <p>7.4 Fecha de modificación</p>

8		El sistema almacena las modificaciones realizadas sobre la propuesta y envía mensaje de "Propuesta Aceptada"
9		El sistema genera un correo electrónico dirigido tanto al Personal de Inventario como al Representante del Fabricante, con la siguiente información: + Folio de la propuesta + Nombre del Personal de Inventario que evaluó + Fecha de modificación de propuesta. + Estado de la propuesta.
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA
5	El Personal de inventarios decide Aceptar la propuesta pero No selecciona una forma de pago	
6	El Personal de inventarios presiona el botón "Aceptar"	
7		El sistema indicará que falta seleccionar forma de pago, y mostrará la misma pantalla del detalle de la propuesta.
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA
5	El Personal de inventarios decide Rechazar la propuesta por lo que tiene que llenar el campo de "Motivo" para indicar la razón por la cual no fue aceptada la propuesta	
6	El personal de inventarios presiona el botón "Rechazar"	
7		El sistema generará o modificará: 7.1 El estatus de la propuesta a "Rechazada" 7.2 Incrementará el número de veces que ha sido modificada una propuesta 7.3 El idOperador de quien realizó la última modificación sobre la propuesta 7.4 Fecha de modificación
8		El sistema almacena las modificaciones realizadas sobre la propuesta y envía mensaje de "Propuesta Rechazada"
9		El sistema genera un correo electrónico dirigido tanto al Personal de Inventario

		como al Representante del Fabricante, con la siguiente información: + Folio de la propuesta + Nombre del Personal de Inventario que evaluó + Fecha de modificación de propuesta. + Estado de la propuesta.
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA
5	El Personal de inventarios decide Rechazar la propuesta pero No llena el campo "Motivo"	
6	El Personal de inventarios presiona el botón "Rechazar"	
7		El sistema indicará que falta llenar el campo "Motivo", y mostrará la misma pantalla del detalle de la propuesta.

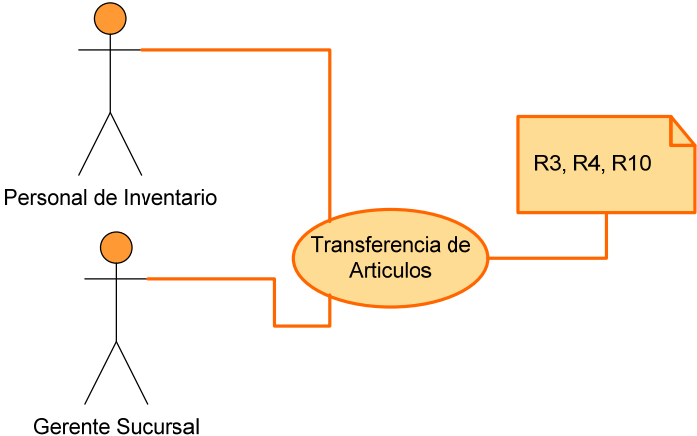
Caso de Uso:	Modificación de Propuesta.	
Descripción General:	Consiste en el flujo que se seguirá para modificar una propuesta que ha sido "Rechazada" previamente.	
Actores:	<p>Representante del Fabricante</p>	
Objetivo del Caso de Uso:	Regenerar una propuesta o en su caso cancelarla.	
Precondiciones:	El representante de ventas cuenta con una propuesta en estado "Rechazada" y un período de vigencia válido.	
Postcondiciones:	El período de vigencia inválida una propuesta "Rechazada"	
Camino Básico:		
Flujo de Eventos	ACTOR	SISTEMA
1	El Representante del Fabricante accede al detalle de la propuesta por medio del la liga localizada en el folio de la misma	
2		El sistema muestra el detalle de la propuesta con los campos de texto habilitados para modificar permitiendo dicha modificación sobre:

		+ Tipo de Accesorio o equipo celular + Descripción +Modelo + Cantidad + Precio Unitario
3		También agrega en pantalla dos botones “Modificar” y “Cancelar”
4	El Representante del Fabricante decide “Modificar” la propuesta, así que escribirá la nueva información en los campos permitidos por el sistema y presionará el botón “Modificar”	
5		El sistema asignará el estado de “En evaluación” y almacenará las modificaciones realizadas sobre la propuesta.
6		El sistema desplegará en pantalla un mensaje para informar que el registro de la propuesta fue exitoso y volverá a indicar el Folio original y el Personal de Inventario Asignado previamente.
7		El sistema genera un correo electrónico dirigido tanto al Personal de Inventario como al Representante del Fabricante, con la siguiente información: + Folio de la propuesta + Nombre del Representante del Fabricante. +Fecha de modificación de la propuesta. +Estado de la propuesta.
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA
4	El Representante del Fabricante decide “Cancelar” la propuesta, así que presionará el botón “Cancelar”	
5		El sistema asignará el estado de “Cancelado” y almacenará los datos de fecha de modificación y el usuario que realizó la última modificación.
6		El sistema desplegará en pantalla un mensaje para informar que la propuesta ha sido cancelada.

Caso de Uso:	Primera Carga.	
Descripción General:	Consiste en el flujo que se seguirá para realizar la primera Carga de los artículos al Almacén Central.	
Actores:		
Objetivo del Caso de Uso:	Almacenar en BD el detalle de cada artículo que ha sido adquirido.	
Precondiciones:	El representante de ventas cuenta con una propuesta en estado "Aceptada" y con un período de vigencia válido.	
Postcondiciones:		
Camino Básico:		
Flujo de Eventos	ACTOR	SISTEMA
1	El Representante del Fabricante accederá al detalle de una propuesta "Aceptada"	
2		El sistema mostrará el detalle de dicha propuesta y agregará un formulario para capturar la información de cada artículo propuesto. El formulario involucrará: + identificador de fábrica del equipo celular o accesorio.
3		El sistema también mostrará un botón de "Transferencia" el cual sólo se habilitará cuando se llene el número de artículos ofertados en la propuesta. En caso contrario el Representante de Ventas no podrá continuar
4	Cuando el Representante del Fabricante haya terminado de llenar el formulario presionará el botón "Transferencia"	
5		El sistema pedirá confirmación de la información a almacenar dándole opción al Representante del Fabricante de Cancelar o Aceptar.
6	El Representante del Fabricante	

	selecciona Aceptar	
7		El sistema almacena la información en BD generando un nuevo folio para la transferencia, fecha en que se origina la transferencia
8		El sistema genera un correo electrónico dirigido tanto al Personal de Inventario como al Representante del Fabricante, con la siguiente información: + Folio de la Transferencia + Nombre del Representante del Fabricante. +Fecha de Transferencia.
9	El Personal de Inventario ingresa al sistema y accede a la opción de Transferencias por Aceptar	
10		El sistema muestra el detalle de la transferencia y los artículos que involucra tales como: Folio de la transferencia, fecha en que se originó, id del Representante del Fabricante que la generó, el tipo de los accesorios involucrados, el modelo, la descripción, la cantidad, el precio Unitario, el monto Subtotal y el monto Total, además de proporcionar el detalle de cada equipo o al accesorio.
11		El sistema muestra en pantalla para aceptar Transferencia un botón con dicha funcionalidad.
12	El personal de Inventario presiona el Botón "Aceptar Transferencia"	
13		El sistema almacena en BD el cambio de estado para cada artículo y manda mensaje de "Transferencia Realizada".
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA
6	El Representante del Fabricante selecciona Cancelar	
7		El sistema sólo limpia los campos capturados y muestra la misma pantalla.

Caso de Uso:	<i>Transferencias de Accesorios y Equipos.</i>
Descripción General:	Consiste en el flujo que se seguirá para realizar Transferencias de artículos entre el Almacén Central y el Almacén de la sucursal y, viceversa.

Actores:		
Objetivo del Caso de Uso:	Almacenar en el Almacén de la sucursal cada artículo que ha sido seleccionado previamente.	
Precondiciones:	El Gerente de la sucursal cuenta con un operador y clave de acceso solicitados previamente.	
Postcondiciones:		
Camino Básico:		
Flujo de Eventos	ACTOR	SISTEMA
1	El Gerente de la sucursal accederá a la opción "Consultas"	
2		<p>El sistema en dicha sección permitirá consultas sobre:</p> <ul style="list-style-type: none"> a) Almacén Central. b) Almacén de la sucursal al que pertenece el Gerente, él no podrá consultar otros Almacenes pertenecientes a otras sucursales. <p>Los criterios de consultas son:</p> <ul style="list-style-type: none"> + Tipo de artículo + Fabricante + Modelo + Identificador de fábrica + Descripción <p>También habrá un botón "Buscar"</p>
3	El Gerente de la sucursal seleccionará el almacén sobre el cual realizará la consulta y el criterio de la misma. Posteriormente presionará el botón "Buscar"	
4		<p>El sistema mostrará el resultado de la consulta indicando:</p> <ul style="list-style-type: none"> + Tipo de artículo + Fabricante

		+ Modelo + Identificador de fábrica + Descripción Y un checkbox así como los botones "Transferir", "Cancelar"
5	El Gerente de la sucursal seleccionará los equipos que desea trasladar a su almacén por medio del checkbox y posteriormente presionará el botón "Transferir"	
6		El sistema generará los movimientos pertinentes en BD para realizar la transferencia y enviará un mensaje informativo de "Transferencia Realizada".
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA
5	El Gerente de la sucursal seleccionará los equipos que desea trasladar a su almacén por medio del checkbox, pero decide presionará el botón "Cancelar"	
6		El sistema sólo limpia los campos capturados y muestra la misma pantalla.

Caso de Uso:	Valida Operador.
Descripción General:	Determinar si el operador tiene acceso al sistema y sus tareas dentro del mismo.
Actores:	<p>The diagram shows a central use case 'Validación Operador' (represented by a purple oval) connected to four actors: 'Representante del Fabricante', 'Personal de Inventario', 'Gerente Sucursal', and 'Administrador' (represented by stick figures). A note box labeled 'R1, R2' is attached to the use case.</p>
Objetivo del Caso de Uso:	Permitir el acceso al Sistema.
Precondiciones:	El operador ya ha sido registrado previamente y ya cuentan con un

	operador o identificador y una contraseña o clave.	
Postcondiciones:		
Camino Básico:		
Flujo de Eventos	ACTOR	SISTEMA
1	Operador captura clave y password	
2		Lee Operador y clave
3		Valida la dupla operador y clave
4		El sistema determina las tareas asociada a dicho operador
5		El sistema en base a las tareas del operador muestra las opciones que le corresponden en pantalla
Camino Alternativo		
Flujo de Eventos	ACTOR	SISTEMA
1	Operador captura operador y clave erróneos	
4		Operador no encontrado
5		Notifica error con un mensaje
6		Presenta pantalla de Inicio

DCapítulo 3. DISEÑO DEL SISTEMA.

Los objetivos del diseño de la base de datos son:

Representar los datos que requieren las principales áreas de aplicación y los grupos de usuarios, y representar las relaciones entre dichos datos.

Proporcionar un modelo de datos que soporte las transacciones que se vayan a realizar sobre los datos.

Especificar un esquema que alcance las prestaciones requeridas para el sistema con suficiente detalle.

Aplicando para ello ciertas técnicas y principios.

Esta etapa consta de tres fases: diseño conceptual, diseño lógico y diseño físico de la base de datos.

3.1 DISEÑO CONCEPTUAL.

Consiste en la construcción de un esquema conceptual de la información usado en el proyecto, independientemente de cualquier consideración física, después de validar y comprobar el esquema conceptual con los requisitos o requerimientos este modelo se transformará en un esquema lógico.

Para realizar el diseño conceptual se considera:

- ✓ Identificar entidades
- ✓ Identificar relaciones
- ✓ Identificar atributos que se asocian a entidades y relaciones
- ✓ Determinar los dominios de los atributos
- ✓ Determinar los identificadores
- ✓ Determinar las jerarquías de generalización (si las hay)
- ✓ Dibujar el diagrama entidad-relación
- ✓ Revisar el esquema conceptual local con el usuario

Para el sistema se establecen las siguientes entidades:

Operador (que puede ser Representante del Fabricante, Persona de Inventario o Gerente Sucursal).

Descripción: Información general asociada al operador que accede al sistema

Atributos:

- Id_Op: Identificador asociado al Operador que puede acceder al sistema.
- Clave: Usuario asociado al Operador.
- Password: Contraseña de acceso al sistema.
- Nombre: Nombre completo del Operador o Usuario que tiene acceso al sistema.

-
- Correo_Elect: Correo Electrónico u email del Operador.
 - Tel_contacto: Teléfono para poder contactar al Operador.

Representante del Fabricante

Descripción: Información específica asociada al operador de tipo Representante del Fabricante.

Atributos:

- Id_Op: Identificador de Operador en el sistema.
- Fabricante: Descripción de la marca Fabricante del accesorio o Equipo Celular.

Persona de Inventario

Descripción: Información específica asociada al operador de tipo Persona de Inventario.

Atributos:

- Id_Op: Identificador de Operador en el sistema.
- Numero de Empleado: Identificador numérico asociado al Operador Persona de Inventario.
- Zona Geográfica: Asociado a este tipo de Operador.

Gerente Sucursal

Descripción: Información específica asociada al operador de tipo Gerente de Sucursal.

Atributos:

- Id_Op: Identificador de Operador en el sistema.
- Numero de Empleado: Identificador numérico asociado al Operador Gerente de Sucursal.

Propuesta

Descripción: Información general asociada a la propuesta generada por el Representante del Fabricante

Atributos:

- Folio: Identificador numérico asignado por el sistema para la propuesta
- Estado_Propuesta: Estatus asociado a la propuesta que indica el flujo por el cual ha pasado.
- Rec_version: Contador que indica el numero de cambios que ha sufrido el registro de la propuesta
- Fecha_origen: Fecha en que nace una propuesta.
- Fecha_mod: Fecha que se modifica de acuerdo al flujo natural de una propuesta.
- User_last_mod: Identificador de Operador que ha realizado algún último cambio sobre la propuesta.
- Precio Subtotal: Costo general subtotal, de la propuesta
- Precio Total: Costo con IVA de la propuesta.

Detalle Propuesta

Descripción: Información específica asociada a la propuesta generada por el Representante del Fabricante

Atributos:

- IdDP: Identificador numérico del detalle de una propuesta.

-
- Marca: Marca Fabricante que genera la propuesta.
 - Cantidad: Numero de accesorios o equipos celulares ofertados.
 - Tipo_ac: Indicador de accesorios o equipos celulares que se ofertan.
 - ModeloDet: Modelo correspondiente al accesorio o equipo celular que se oferta en la Propuesta.
 - Descripcion: Descripción asociada al accesorio o equipo celular.
 - P_Unit: Precio Unitario del accesorio o equipo celular.
 - P_IVA: Precio con IVA del accesorio o equipo celular.

Dictamen

Descripción: Información específica asociada al dictamen realizado por la Persona de Inventario sobre la propuesta generada por el Representante del Fabricante.

Atributos:

- Folio_Dic: Identificador numérico asociado al Dictamen.
- Fecha_mod: Fecha en que se genera el Dictamen o modifica.
- Motivo: Descripción breve que justifica el dictamen generado.

Primera Carga

Descripción: Información específica de la Transferencia de artículos detallados en la propuesta hacia el Almacén Central

Atributos:

- Folio_PC: Identificador numérico que se genera para la primera carga.

Artículo

Descripción: Información detallada del accesorio o equipo celular que se tiene en algún Almacén

Atributos:

- IdFab: Identificador natural del accesorio o equipo celular.
- Tipo: Indicador del registro para conocer si es accesorio o equipo celular.
- Marca: Marca del accesorio o equipo celular.
- Estado: Indicador para el accesorio o equipo celular que indica si se encuentra en existencia, en transferencia o fue vendido.
- Modelo: del accesorio o equipo.
- Descripción: Observación que se puede realizar sobre el accesorio o equipo celular.
- P_Unit: Precio Unitario del equipo celular o accesorio.

Almacén

Descripción: Abstracción lógica de un Almacén para asociar a los accesorios y equipos celulares.

Atributos:

- Id: Identificador asociado al Almacén.

Sucursal

Descripción: Abstracción lógica de una Sucursal para asociar a algún Operador de tipo Gerente Sucursal o algún Almacén.

Atributos:

- Id_Tienda: Identificador asociado a la Sucursal.
- Nombre_Sucursal: Nombre descriptivo de la Sucursal.
- Dirección: Lugar donde se encuentra la Sucursal.

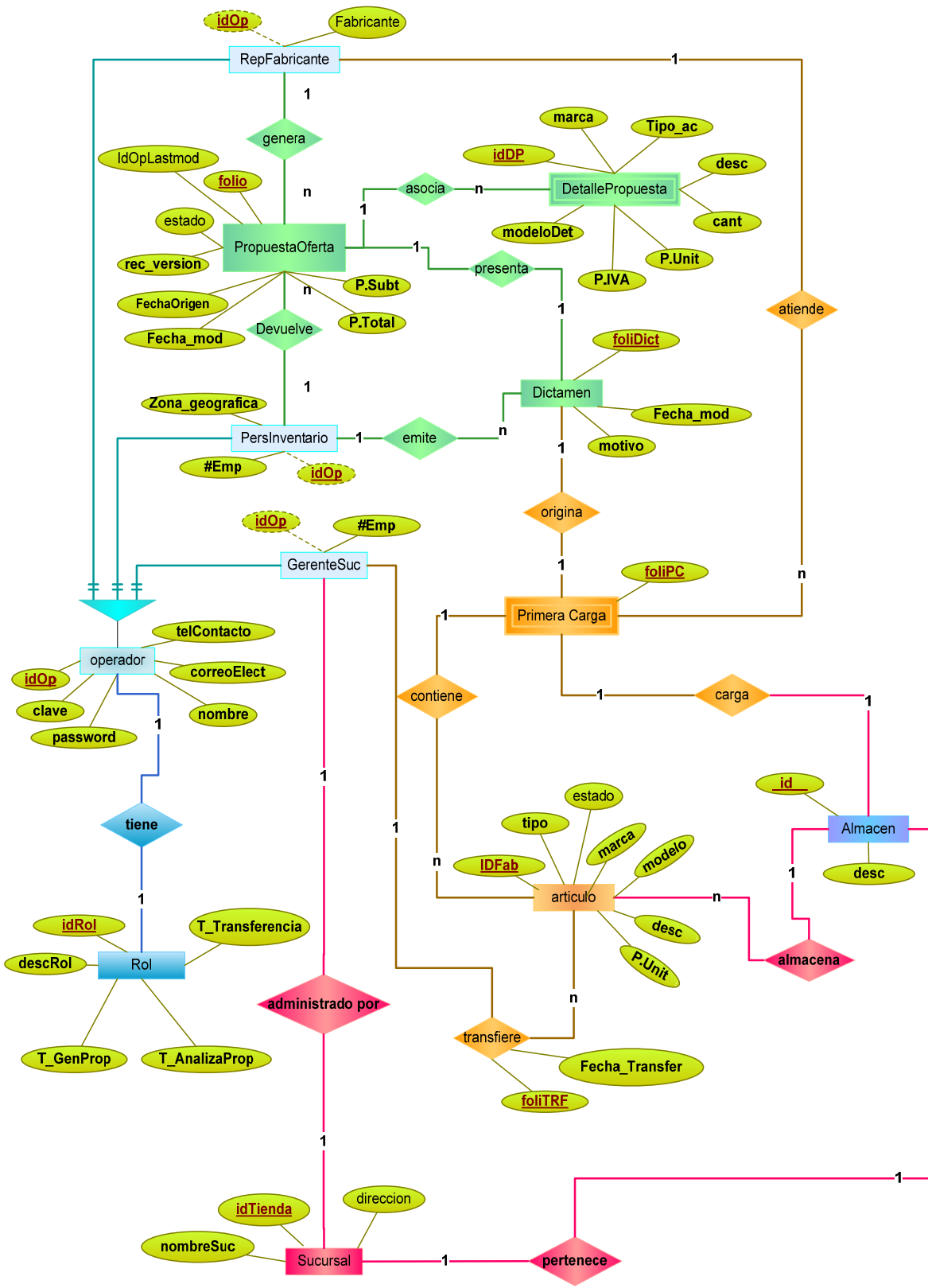
Rol

Descripción: Información asociada a las tareas que desempeñan los diferentes tipos de Operador.

Atributos:

- Id_rol: Identificador numérico para un rol.
- Desc_Rol: Nombre del rol.
- T_genProp: Tarea para generar Propuestas.
- T_AnalizaProp: Tarea para generar un Dictamen sobre alguna Propuesta.
- T_Transferencia: Tarea para generar Transferencias.

Tomando como base las entidades encontradas podemos crear el diagrama entidad-relación mostrado en la siguiente figura.



[Fig. 3.1 Diagrama Entidad Relación]

3.2 DISEÑO LÓGICO.

El diseño lógico es el proceso de construir un esquema de la información, basándose en un modelo de base de datos específico por ejemplo el modelo relacional, modelo de red, modelo jerárquico o el modelo orientado a objetos, independiente del SGBD (Sistema Gestor de Base de Datos) concreto que se vaya a utilizar y de cualquier otra consideración física.

Este esquema será una fuente fundamental para el diseño físico además puede tomar mucha importancia durante la etapa de mantenimiento del sistema.

Para llevar a cabo la transformación del modelo entidad-relación al modelo relacional hay que basarse en las siguientes reglas:

- ✓ Toda entidad convierte en una tabla
- ✓ Toda relación N:M se genera una nueva tabla que contenga cada una de las llaves primarias de las entidades involucradas en la relación
- ✓ Toda relación 1:M se incluye en la tabla M, la llave primaria de la tabla 1 como llave foránea.

Por lo tanto el diseño lógico de la base de datos o Modelo Relacional para el sistema será:

Operador						
<u>idOp</u>	clave	nombre	password	correoElect	telContacto	idRol

FK

RepFabricante	
<u>idOpFab</u>	fabricante

Personal_Inventario		
<u>idOpPI</u>	numEmpleado	ZonaGeografica

Gerente		
<u>idOpGte</u>	idSuc	numEmpleado

Rol				
<u>idRol</u>	descRol	T_genProp	T_AnalizaProp	T_Transferencia

Propuesta									
<u>folioProp</u>	estado	fecha Orig	Fec_last_ mod	idOp_last_ _mod	ver_reg	subtotal	total	idOpFab	idOpPI

FK FK

DetallePropuesta								
<u>idDP</u>	modeloDet	tipoArt	marca	desc	cant	P_unit	P_UnitIVA	folioProp

FK

Dictamen				
<u>folioDict</u>	Fecha_mod	motivo	folioProp	idOpPI
			FK	FK

Sucursal		
<u>idSuc</u>	descSuc	direccion

PrimeraCarga				
<u>folioPc</u>	idOpFab	idFab	folioDict	idAlm
			FK	FK

Almacen		
<u>idAlm</u>	desc	idSuc
		FK

Articulo									
<u>idFab</u>	modelo	desc	tipo	marca	estado	P_Unit	P_IVA	idAlmc	folioPC
								FK	FK

Transferencia			
<u>folioTrf</u>	fechaTrf	idOpGte	idFab
		FK	FK

3.2.1 Normalización de Base de Datos del Sistema.

La normalización se utiliza para mejorar el esquema lógico, de modo que satisfaga ciertas restricciones que eviten la duplicidad de datos, además de ser un proceso que permite decidir a qué entidad pertenece cada atributo y organiza los datos de acuerdo a sus relaciones lógicas.

✓ Primera Forma Normal (1FN).

La primera forma normal plantea que todos los atributos de las relaciones deben ser atómicos, es decir no hay grupos repetidos.

Como se puede observar en el esquema anterior todas las tablas poseen atributos atómicos, por lo que podemos decir que la base de datos cuenta con la 1FN.

✓ Segunda Forma Normal (2FN).

Una tabla se encuentra en segunda forma normal si cumple con la primera forma normal y todos sus atributos que no son claves o llaves dependen por completo de la llave.

En cada una de las tablas del esquema anterior se puede observar que los atributos no llave dependen de su clave principal.

✓ Tercera Forma Normal (3FN).

Una relación está en tercera forma normal si, y sólo si se encuentra en segunda forma normal y no existen dependencias transitivas entre los atributos, es decir, una dependencia transitiva existe cuando hay más de una forma de llegar a referenciar a un atributo de una relación.

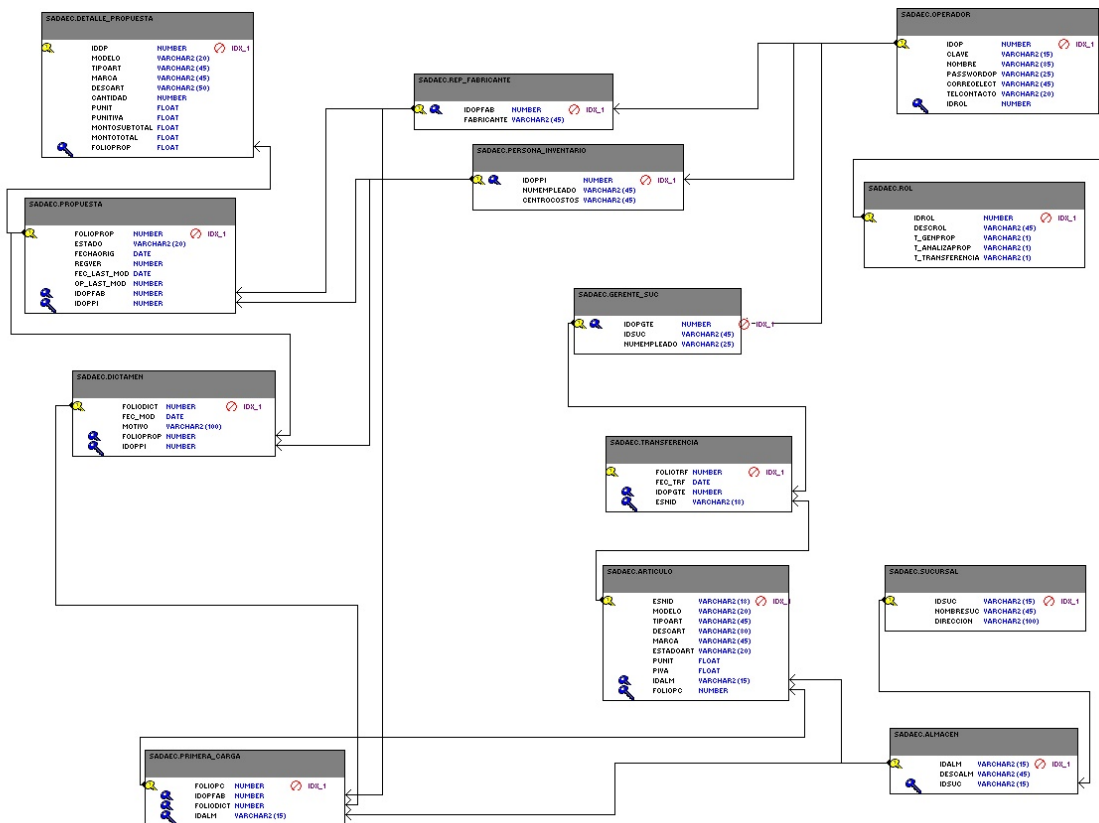
En el esquema anterior se puede apreciar que en todas las tablas de la Base de Datos se encuentran en tercera forma normal porque no existe dependencia funcional entre los atributos no llave.

3.3 DISEÑO FÍSICO.

El diseño físico es el proceso de producir la descripción de la implementación de la base de datos en estructuras de almacenamiento y métodos de acceso que garanticen un acceso eficiente a los datos.

El objetivo del diseño físico es describir cómo se va a implementar físicamente el esquema lógico de la fase anterior.

Se debe considerar que para llevar a cabo esta etapa se debe contar con un SGBD. En este caso el sistema utilizará Oracle; y para realizar la administración y desarrollo de la Base de Datos utilizamos la herramienta TOAD, el cual nos genera el siguiente diagrama cuando ya se encuentra construida la Base de Datos.



[Fig. 3.2 Diagrama generado por Toad]

3.4 DICCIONARIO DE DATOS.

El diccionario de datos se utiliza para llevar un recuento detallado de todas las tablas dentro de la base de datos que han sido creadas por el usuario, el diseñador o ambos. Así pues, el diccionario de datos contiene (por lo menos) todos los nombres y características de los atributos de cada una de las tablas en el sistema.

Tablas del sistema.

A continuación se muestra una pequeña descripción de las tablas resultantes de las relaciones que se presentan en el modelo Entidad-Relación:

Tabla	Descripción
ROL	Catálogo de los tipos de roles y sus tareas asociadas.
OPERADOR	Almacena información general acerca de los usuarios registrados en el sistema.
REP_FABRICANTE	Contiene información específica relacionada al Operador con rol Representante del Fabricante
PERSONA_INVENTARIO	Incluye información específica relacionada al Operador con rol Persona de Inventario
GERENTE_SUC	Guarda información específica relacionada al Operador con rol Gerente
PROPUESTA	Reúne datos asociados a la propuesta que genera un Representante del Fabricante
DETALLE_PROPUESTA	Contiene los datos que forman el detalle de la Propuesta generada por el Representante del Fabricante
DICTAMEN	Almacena la información relacionada al dictamen emitido por el Operador Persona_Inventario
SUCURSAL	Reúne los datos referentes a una sucursal
ALMACEN	Guarda información del almacén que pertenece a una sucursal
PRIMERA_CARGA	Acumula información relacionada a la Primera Transferencia o Carga, que genera el Representante del Fabricante al Almacén Central
ARTICULO	Contiene la información referente al detalle del artículo (accesorio o equipo celular) transferido.
TRANSFERENCIA	Reúne los datos que se generan durante la Transferencia de un artículo.

Detalle de las Tablas del sistema.

A continuación se define el detalle de cada una de las tablas especificando su descripción, tipo de dato, longitud y dato requerido:

Tabla 3.1

Nombre de la Tabla:	ROL
----------------------------	------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDROL	SI	NUMBER	SI	Identificador asociado a un rol
DESCROL		VARCHAR2 (45)		Nombre que describe un rol
T_GENPROP		VARCHAR2 (1)		Tarea asociada a la actividad de Generar Propuesta
T_ANALIZAPROP		VARCHAR2 (1)		Tarea asociada a la actividad de Analizar Propuesta
T_TRANSFERENCIA		VARCHAR2 (1)		Tarea asociada a la actividad de Transferir equipos

Tabla 3.2

Nombre de la Tabla:	OPERADOR
----------------------------	-----------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDOP	SI	NUMBER	SI	Identificador asociado a un operador
CLAVE		VARCHAR2 (15)		Identificador asociado al operador del sistema
NOMBRE		VARCHAR2 (85)		Nombre del operador del sistema
PASSWORDOP		VARCHAR2 (25)		Contraseña que permite al operador ingresar al sistema
CORREOELECT		VARCHAR2 (45)		Correo Electrónico del operador del sistema
TELCONTACTO		VARCHAR2 (20)		Teléfono de contacto del operador del sistema
IDROL		NUMBER	SI	Identificador asociado a un rol

Tabla 3.3

Nombre de la Tabla:	REP_FABRICANTE
----------------------------	-----------------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDOPFAB	SI	NUMBER	SI	Identificador asociado a un operador de tipo Representante del fabricante
FABRICANTE		VARCHAR2 (45)		Nombre de la marca fabricante del celular o accesorio

Tabla 3.4

Nombre de la Tabla:	PERSONA_INVENTARIO
----------------------------	---------------------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDOPPI	SI	NUMBER	SI	Identificador asociado a un operador de tipo Persona de Inventario
NUMEMPLEADO		VARCHAR2 (45)		Número de empleado del operador
ZONAGEOGRAFICA		VARCHAR2 (45)		Zona geográfica a la que pertenece el operador

Tabla 3.5

Nombre de la Tabla:	GERENTE_SUC
----------------------------	--------------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDOPGTE	SI	NUMBER	SI	Identificador asociado a un operador de tipo Gerente de Sucursal
IDSUC		VARCHAR2 (45)		Identificador de la Sucursal o tienda a la que pertenece el operador
NUMEMPLEADO		VARCHAR2 (25)		Número de empleado del operador

Tabla 3.6

Nombre de la Tabla:	PROPUESTA
----------------------------	------------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
FOLIOPROP	SI	NUMBER	SI	Identificador asociado a una Propuesta
ESTADO		VARCHAR2 (20)		Estado de una propuesta. Valores asociados "Aceptada", "En proceso", "Rechazada", "Cancelada"
FECHAORIG		DATE		Fecha origen de la Propuesta
REGVER		NUMBER		Contador de modificaciones sobre el registro
FEC_LAST_MOD		DATE		Fecha de última modificación del registro
OP_LAST_MOD		NUMBER		Identificador del último operador que modificó el registro
IDOPFAB		NUMBER		Identificador de operador tipo

				Representante del Fabricante que genera la propuesta
IDOPPI		NUMBER		Identificador de operador tipo Personal de Inventario que estudiará la propuesta.

Tabla 3.7

Nombre de la Tabla:	DETALLE_PROPUESTA
----------------------------	--------------------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDDP	SI	NUMBER	SI	Identificador propio del Detalle de una propuesta
MODELO		VARCHAR2 (20)		Modelo del articulo involucrado en la propuesta
TIPOART		VARCHAR2 (45)		Tipo de Artículo
MARCA		VARCHAR2 (45)		Marca del Fabricante del Artículo
DESCART		VARCHAR2 (50)		Descripción del Artículo
CANTIDAD		NUMBER		Cantidad del artículo que se oferta
PUNIT		FLOAT		Precio Unitario
PUNITIVA		FLOAT		Precio Unitario con IVA
MONTOBTOTAL		FLOAT		Subtotal de todo el Detalle
MONTOTOTAL		FLOAT		Total del Detalle
FOLIOPROP		NUMBER		Folio de Propuesta a la que pertenece este Detalle

Tabla 3.8

Nombre de la Tabla:	DICTAMEN
----------------------------	-----------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
FOLIODICT	SI	NUMBER	SI	Identificador asociado a un Dictamen(Estado de la propuesta) asignado por un Operador tipo Persona Inventario
FEC_MOD		DATE		Fecha de modificación del registro
MOTIVO		VARCHAR2 (100)		Descripción de la Justificación del Dictamen asignado
FOLIOPROP		NUMBER		Folio de la propuesta a la que se le asigna el dictamen
IDOPPI		NUMBER		Identificador del operador Persona Inventario que asigna el dictamen

Tabla 3.9

Nombre de la Tabla:	SUCURSAL
----------------------------	-----------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDSUC	SI	VARCHAR2 (15)	SI	Identificador asociado a una Sucursal
NOMBRESUC		VARCHAR2 (45)		Nombre asociado a la sucursal
DIRECCION		VARCHAR2 (100)		Dirección asociada a la sucursal

Tabla 3.10

Nombre de la Tabla:	ALMACEN
----------------------------	----------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDALM	SI	VARCHAR2 (15)	SI	Identificador asociado a un Almacén
DESCALM		VARCHAR2 (45)		Nombre asociado al almacén
IDSUC		VARCHAR2 (15)		Identificador de una Sucursal a la que pertenece el almacén

Tabla 3.11

Nombre de la Tabla:	PRIMERA CARGA
----------------------------	----------------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
FOLIOPC	SI	NUMBER	SI	Folio asociado a la Primera Carga de artículos
IDOPFAB		NUMBER		Identificador de operador Rep_Fabricante
FOLIODICT		NUMBER		Folio de Dictamen "Aceptada" que se asocia a la Primera Carga
IDALM		VARCHAR2(15)		Identificador del almacén sobre el que se realiza la Primera Carga

Tabla 3.12

Nombre de la Tabla:	ARTICULO
----------------------------	-----------------

Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
IDFAB	SI	VARCHAR2 (18)	SI	Identificador asociado a un Artículo.
MODELO		VARCHAR2 (20)		Modelo del articulo
TIPOART		VARCHAR2 (45)		Tipo "accesorio", "equipo celular"
DESCART		VARCHAR2 (80)		Descripción del artículo
MARCA		VARCHAR2 (45)		Marca del Fabricante asociada al

				artículo
ESTADOART		VARCHAR2 (20)		Estado del articulo. "Existencia", "Vendido", "Transferencia"
PUNIT		FLOAT		Precio Unitario del Artículo
PIVA		FLOAT		Precio con IVA del Artículo
IDALM		VARCHAR2 (15)		Identificador del Almacén en donde se encuentra el artículo
FOLIOPC		NUMBER		Folio de Primera Carga asociado al artículo.

Tabla 3.13

Nombre de la Tabla:	TRANSFERENCIA
----------------------------	----------------------

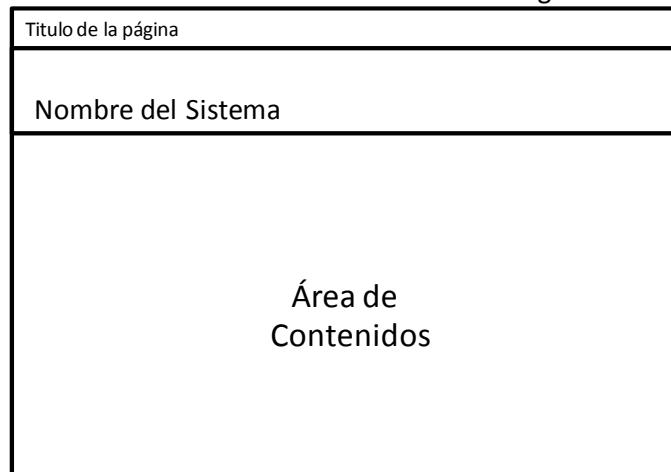
Columna	Primary Key	Tipo de Dato	Dato Requerido	Descripción
FOLIOTRF	SI	NUMBER	SI	Folio asignado a la Transferencia
FEC_TRF		DATE		Fecha de la Transferencia
IDOPGTE		NUMBER		Identificador del Operador Gerente que realiza la transferencia
ID		VARCHAR2 (18)		Identificador del Artículo que participa en la Transferencia

3.5 DISEÑO DE LA INTERFAZ DEL SISTEMA.

Una interfaz es un conjunto de elementos de la pantalla que permiten al usuario interactuar o realizar acciones sobre el sitio web.

El sistema en base a sus características requiere de los siguientes bosquejos de interfaces base:

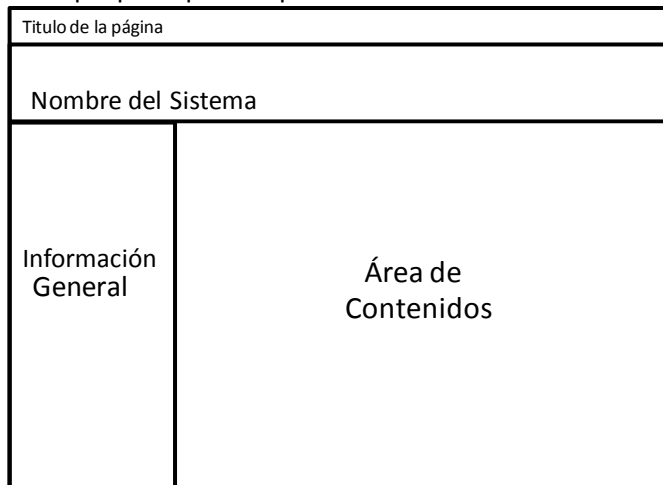
- a) Interfaz de Identificación o Autenticación se muestra en la Fig.3.3



[Fig. 3.3 Interfaz Inicio]

Como su nombre lo indica esta interfaz interactuará con el operador para validar su acceso al sistema. Esta interfaz cuenta con un área de contenidos destinada para que el operador introduzca la información requerida para su acceso.

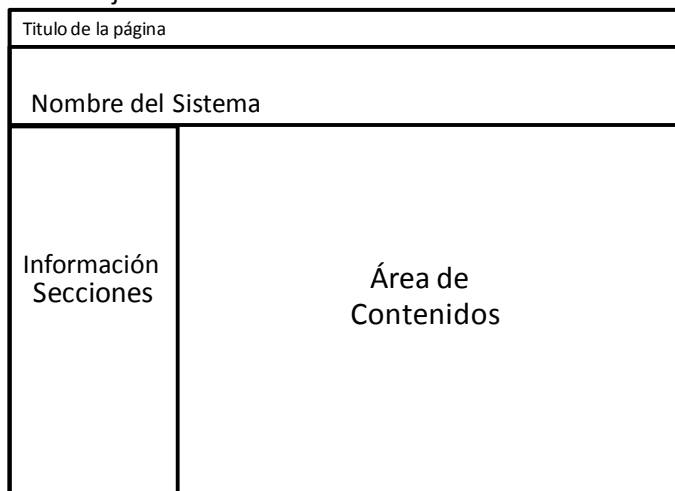
b) Interfaz Principal por Tipo de Operador o Rol se muestra en la Fig. 3.4



[Fig. 3.4 Interfaz Principal]

Interfaz orientada a desplegar información de la última transacción generada, datos del operador y las tareas que por el tipo o rol de operador le corresponden.

c) Interfaz de Trabajo.



[Fig. 3.5 Interfaz de área de Trabajo para cada Operador]

Interfaz de Trabajo asociada a las tareas y generación de transacciones, además de permitir la navegación hacia las demás tareas que le corresponden al operador.

IV Capítulo 4. IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA.

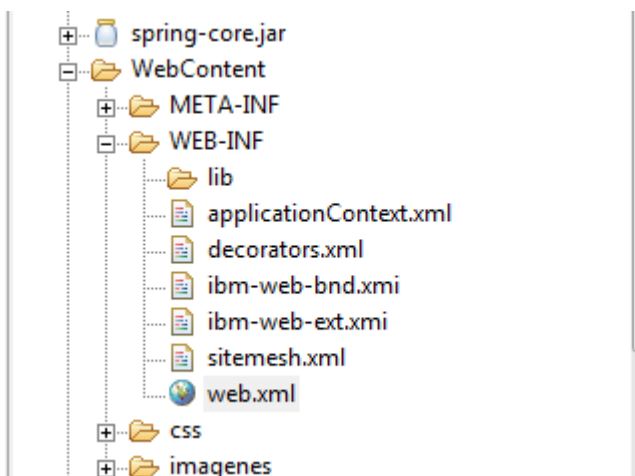
4.1 IMPLEMENTACION DEL SISTEMA.

Como se mencionó en capítulos anteriores el sistema se implementó bajo el framework de struts2 empleando para ello IBM Websphere Application Server (WAS), IBM Rational Application Developer (RAD) para la parte web; mientras que para la parte de administración y almacenamiento de datos se optó por utilizar Oracle10g.

Lo relevante aquí es el framework de Struts2 que utiliza el patrón MVC (*Modelo-Vista-Controlador*) y cuya manipulación de la información comienza desde que una petición es recibida por el *controlador*, conocido como el filtro *FilterDispatcher* ; encargado de comunicar la vista y el modelo, ya sea respondiendo a eventos generados por el usuario en la vista, invocando cambios en el modelo, o devolviendo a la vista la información del modelo, necesaria para que pueda generar la respuesta adecuada para el usuario; por medio de un mapeo al *action* apropiado.

El *FilterDispatcher* es por lo tanto el punto de entrada dentro del framework Struts2 para cualquier petición, especificado en el archivo *descriptor de despliegue* “web.xml”.

La Fig. 4.1 y 4.2 muestran la forma en que se organiza y se especifica el punto de entrada *FilterDispatcher*.



[Fig. 4.1 Organización del framework Struts2 – archivo web.xml]

```

Descriptor de despliegue Web X
<filter>
  <filter-name>action2-cleanup</filter-name>
  <filter-class>org.apache.struts2.dispatcher.ActionContextCleanUp</filter-class>
</filter>

<filter>
  <filter-name>sitemesh</filter-name>
  <filter-class>com.opensymphony.module.sitemesh.filter.PageFilter</filter-class>
</filter >

<filter>
  <filter-name>action2</filter-name>
  <filter-class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>
</filter>
<!-- Definiendo Archivo de Configuracion de Spring -->
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>classpath*:com/iusacell/proyecto/pojo/config/beans.xml</param-value>
</context-param>

  <!-- Levantando el Listener del Contexto -->
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

```

[Fig. 4.2 Parte Declarativa – archivo web.xml]

Un componente *action* corresponde al *modelo* dentro del patrón MVC encargado de referenciar los datos que maneja la aplicación y las reglas de negocio que operan sobre ellos. La Fig. 4.3 muestra la referencia al *action*.

```

Descriptor de despliegue Web index.jsp X
index.jsp
<BODY>
  <BR/>
  <BR/>
  <s:form action="ValidaOperadorAction">
    <TABLE width="479" height="276" border="1" align="center">
    <TBODY>

```

[Fig. 4.3 index.jsp - Código referenciando Action]

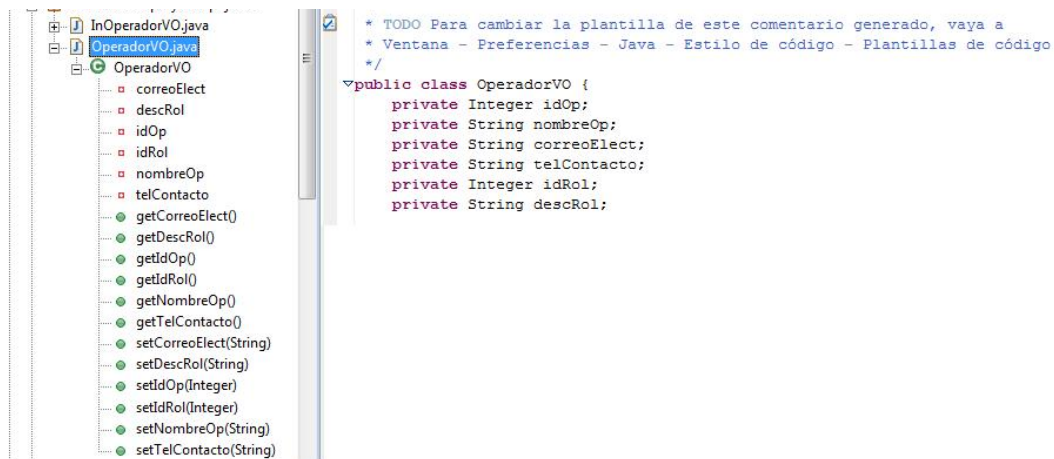
El framework busca un mapeo de *action* en el archivo struts.xml y considera que esta asignación corresponde a la clase definida en *class*, como lo muestra la Fig. 4.4.



[Fig. 4.4 struts.xml –Asignación de clase action]

El action es parte del modelo dentro del patrón MVC, y suele estar dividido en dos subcapas, siguiendo el patrón DAO (*Data Access Object*) y POJOs (Plain Old Java Objects).

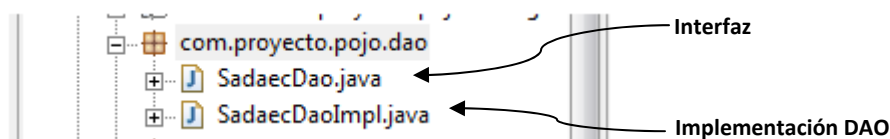
En el proyecto se utilizaron los POJOs con la finalidad de establecer y manipular estructuras orientadas a Objetos. La Fig. 4.5 muestra un ejemplo de implementación.



[Fig. 4.5 Implementación de un POJO]

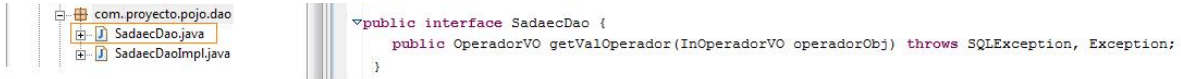
Y para establecer una interfaz común entre la aplicación y la Base de Datos se optó por manejar DAOs, lo que implica tener una interfaz e implementar por cada interfaz el DAO en sí.

La Fig. 4.6, muestra que tanto la interfaz y la implementación del DAO deben considerarse.



[Fig. 4.6 Elementos que forman un DAO]

a) Interfaz DAO. La Fig.4.7 muestra la declaración en código de la Interfaz DAO.

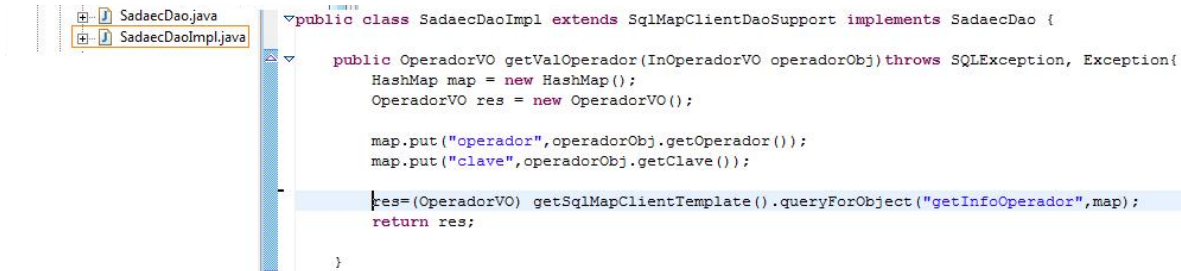


```
com.proyecto.pojo.dao
├── SadaecDao.java
└── SadaecDaoImpl.java
```

```
public interface SadaecDao {
    public OperatorVO getValOperator(InOperatorVO operadorObj) throws SQLException, Exception;
}
```

[Fig. 4.7 Código que constituye a la Interfaz DAO]

b) Implementación DAO. La Fig.7.8 muestra parte del código de implementación.



```
public class SadaecDaoImpl extends SqlMapClientDaoSupport implements SadaecDao {

    public OperatorVO getValOperator(InOperatorVO operadorObj) throws SQLException, Exception{
        HashMap map = new HashMap();
        OperatorVO res = new OperatorVO();

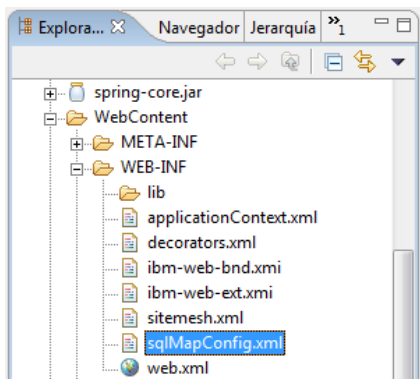
        map.put("operador", operadorObj.getOperador());
        map.put("clave", operadorObj.getClave());

        res=(OperatorVO) getSqlMapClientTemplate().queryForObject("getInfoOperador",map);
        return res;
    }
}
```

[Fig. 4.8 Implementación del componente DAO]

Como parte de las características del framework de ibatis se requiere utilizar sqlMap que permite la persistencia de objetos en base de datos relacionales; para ello utiliza archivos xml que permiten asociar DAOs con sentencias SQL.

El archivo de configuración es un archivo xml dentro del cual se especifica el archivo donde se encuentran los mapeos SQL que utiliza la aplicación, cuya declaración se muestra en la Fig. 4.9.



```
spring-core.jar
WebContent
├── META-INF
├── WEB-INF
│   ├── lib
│   ├── applicationContext.xml
│   ├── decorators.xml
│   ├── ibm-web-bnd.xmi
│   ├── ibm-web-ext.xmi
│   ├── sitemesh.xml
│   └── sqlMapConfig.xml
web.xml
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE sqlMapConfig PUBLIC "-//ibatis.apache.org//DTD SQL Map Config 2.0//EN" "http://ibatis.apache.
<sqlMapConfig>
  <sqlMap resource="com/proyecto/pojo/config/consultas.xml"/>
</sqlMapConfig>
```

[Fig. 4.9 Declaración del archivo que contiene el mapeo]

En el archivo consultas xml se proporciona el código SQL y los mapeos para cada unos de los objetos parámetro y de los objetos resultado, esto se aprecia en la Fig. 4.10.

```

<sqlMap>
  <typeAlias alias="operador" type="com.proyecto.pojo.vo.OperadorVO" />
  <typeAlias alias="hashMap" type="java.util.HashMap" />
  <typeAlias alias="string" type="java.lang.String" />

  <resultMap id="get-validaOperador-map" class="operador">
    <result property="idOp" column="idop"/>
    <result property="nombreOp" column="nombre"/>
    <result property="correoElect" column="correoElect"/>
    <result property="telContacto" column="telContacto"/>
    <result property="idRol" column="idRol"/>
    <result property="descRol" column="descRol"/>
  </resultMap>

  <select id="getInfoOperador" parameterClass="map" resultMap="get-validaOperador-map">
    <![CDATA[
      SELECT op.idop idop, op.nombre nombre, op.correoElect correoElect,
      op.telContacto telContacto, r.idRol idRol, r.descRol descRol
      FROM operador op, rol r
      WHERE op.idRol=r.idRol
      AND op.clave=#parametroClave:VARCHAR#
      AND op.password=#parametroPsw:VARCHAR#

    ]]>
  </select>
</sqlMap>

```

[Fig. 4.10 Mapeos entre objetos parámetros u objetos de entrada y salida]

Posteriormente el resultado se retorna por medio de la vista, que es la encargada de generar la interfaz con la que la aplicación interactúa con el usuario. En Struts 2 equivale a los resultados o *result*; generalmente implementada por medio de JSPs.

4.2 INTERFAZ DEL SISTEMA.

En base al mapa de navegación expuesto en el capítulo anterior se definieron las siguientes interfaces del sistema:

- Control de Acceso.
- Área de Trabajo Personalizada por Operador.
- Generar Propuesta.
- Dictaminar Propuesta.
- Generar Primera Carga.
- Aceptar Primera Carga.
- Realizar Transferencias.
- Consultas asociadas a :
 - Propuestas
 - Accesorios/Equipo Celular.
 - Transferencias.

4.2.1 Interfaz Control de Acceso

Esta interfaz está dedicada a verificar que el operador cuente con el acceso y los permisos necesarios para trabajar con el sistema, esta interfaz se encuentra representada en la Fig. 4.11.

Consulta asociada:

```
SELECT  
op.idop, op.nombre, op.correoElect, op.telContacto, r.descRol  
FROM OPERADOR OP, ROL R  
WHERE op.idRol=r.idRol  
AND op.clave=:parametroClave  
AND op.password=:parametroPsw
```



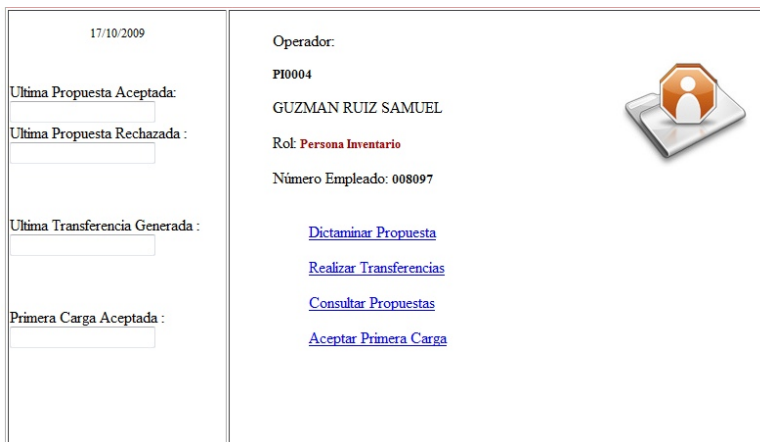
[Fig. 4.11 Interfaz de Inicio]

4.2.2. Interfaz Área de Trabajo Personalizada por Operador.

Pantalla dedicada a interactuar con el usuario proporcionándole las tareas que puede realizar en base a su rol, como se muestra en la Fig. 4.12.

Consulta asociada:

```
SELECT t_genprop, t_analizaprop, t_transferencia  
FROM ROL WHERE idrol=1
```



[Fig. 4.12 Interfaz Principal]

4.2.3 Interfaz Genera Propuesta.

Interfaz encargada de interactuar con el usuario para generar una propuesta por medio de la captura de la información asociada a una propuesta, representada en la Fig. 4.13.

Consultas Asociadas:

```
a) SELECT seq_propuesta.NEXTVAL  
INTO PFOLIOPROP  
FROM DUAL;
```

```
b) INSERT INTO PROPUESTA (folio_prop,estado, fechaorig, regver, fec_last_mod, op_last_mod,  
idopfab, idppi)  
VALUES  
(pFolioProp, "EN EVALUACION", TRUNC(SYSDATE), 1, SYSDATE, pidRF, pidRF, pidPi);
```

```
c)SELECTseq_propuesta.NEXTVAL  
INTO pldDetalle  
FROM DUAL;
```

```
d) INSERT INTO DETALLE_PROPUESTA (iddp, modelo, tipoart, marca, descart, cantidad, punit,  
punitiva, montosubtotal, montototal, folioprop)  
VALUES  
(pldDetalle, pmodelo, ptipo, pmarca, descripcion, pcantidad, punit, punitive,  
psubtotal, ptotal, pFolioProp)
```

24/09/2009

RF0001 Folio 01 Estado Propuesta En Evaluacion

Operador Espino Cisneros Esperanza

Marca Seleccione...

Cantidad	Tipo Articulo	Modelo	Descripción	Precio	PrecioIVA
	Seleccione...				
	Seleccione...				
	Seleccione...				
	Seleccione...				
	Seleccione...				

Siguiente Cuadrícula

Cancelar Generar Subtotal Total

Generar Propuesta

Consulta Propuesta

Generar Primera Carga

Principal

Cerrar Sesion

[Fig. 4.13 Interfaz Genera Propuesta]

4.2.4 Interfaz Dictaminar Propuesta.

Interfaz específica para el rol de Persona de Inventario, que le permite interactuar con el sistema para continuar el flujo de la transacción, cuya imagen se ve en la Fig. 4.14.

Consultas Asociadas:

```
a) SELECT seq_dictamen.NEXTVAL  
INTO pFolioDictamen  
FROM DUAL;
```

```
b) INSERT INTO DICTAMEN (foliodict, fec_mod, motivo, folio_prop, idoppi)  
VALUES (pFolioDictamen, SYSDATE, pmotivo, pFolioProp, pidoppi);
```

```
c) UPDATE PROPUESTA  
SET estado=pEstadoProp  
WHERE folio_prop= pFolioProp;
```

29/09/2009

PI0004 Folio 01 Estado Propuesta En Evaluacion

Operador Guzman Ruiz Samuel

Marca

Cantidad	Tipo Articulo	Modelo	Descripción	Precio	PrecioIVA
5	Equipo Celular	LG			

[Dictaminar Propuesta](#)
[Realizar Transferencia](#)
[Consulta Propuesta](#)
[Aceptar Primera Carga](#)
[Principal](#)

Motivo Rechazo:

[Rechazar](#) [Aceptar](#) Subtotal Total

[Fig. 4.14 Interfaz Dictamina Propuesta]

4.2.5 Interfaz Primera Carga.

Vía encargada de actuar con el operador Representante de Fabricante para generar la primera carga hacia el almacén central. Fig. 4.15.

Consultas Asociadas:

a) *SELECT seq_primeraCarga.NEXTVAL
INTO pldPc
FROM DUAL;*

b) *INSERT INTO PRIMERA CARGA(foliopc, idopfab, folioDict)
VALUES(pldPc, pidRF pFolioDictamen);*

c) *INSERT INTO ARTICULO (idFab, modelo, tipoart, descart, marca, estadoArt, punit, pIVA, idalm,
foliopc)
VALUES(pifab, pmodelo, ptipo, descripción, pmarca, 'En existencia', punit, punitive, pldPc);*

24/09/2009

Genera Primera Carga

Operador RF0001 Espino Cisneros Esperanza

Folio Propuesta: 1 Folio Transferencia: 1 Propuesta compuesta por 3 Elementos

Elementos de Propuesta	Cantidad	Tipo Articulo	Modelo	Descripción
1/3	5	Equipo Celular	LG	

Sección de Captura de IDs.

01119303				

Faltan Elementos a Mostrar 2/3

Siguiete Elemento Generar Principal

[Cerrar Sesion](#)

[Fig. 4.15 Interfaz Genera Primera Carga]

4.2.6 Interfaz Aceptar Primera Carga.

Interfaz que le permite al operador con rol de Persona de Inventario interactuar con el sistema para aceptar los accesorios o Equipos Celulares involucrados en la propuesta aceptada con anterioridad. Fig. 4.16.

Consultas Asociadas:

```
UPDATE PRIMERA_CARGA
```

```
SET idalm=palmCentral
```

```
where foliopc=pldPc
```

Elementos de Propuesta	Cantidad	Tipo Articulo	Modelo	Descripción
1/3	5	Equipo Celular	LG	

Sección de Captura de IDs.				
01119303	01117308	0111937	011394	0111429

[Fig. 4.16 Interfaz Acepta Primera Carga]

4.2.7 Interfaz para Realizar Transferencias.

Destinada al operador con rol de Gerente de Sucursal para interactuar con el sistema por medio de la operación de Transferencias de equipos, generadas desde el almacén central hacia el almacén asociado a la sucursal. Fig. 4.17.

Consultas Asociadas:

```
a) SELECT seq_Transferencias.NEXTVAL  
INTO pldTr  
FROM DUAL;
```

```
b) SELECT id, tipoArt, modelo, descArt  
FROM ARTICULO  
WHERE TIPOART=:pTipoArt  
AND MARCA=:pMarca  
AND ESTADOART='EX';
```

```
c) INSERT INTO TRANSFERENCIA  
(foliotrf,fec_trf,idopgte,id)  
values(pldTr, SYSDEATE,:pOperador,:pld);
```

```
d) UPDATE ARTICULO  
SET ESTADOART='TR', IDALM=:pAlmacenDest  
WHERE TIPOART=:pTipoArt  
AND MARCA=:pMarca;
```

sucursal S0005

GS0003

Operador Huerta Aguilar
Gilberto

[Realizar Transferencia](#)

[Consultar Transferencia](#)

[Consultar
Accesorio/Equipo Celular](#)

[Principal](#)

[Cerrar Sesión](#)

Transferencias

Folio Transferencia: Tipo a Transferir: Marca: Modelo:

Número de Elementos Encontrados: Almacen Origen: Almacen Destino:

Seleccione	ID	Tipo Articulo	Modelo	Descripción
<input type="checkbox"/>	0111930305	Equipo Celular	LG	
<input checked="" type="checkbox"/>	01119303	Equipo Celular	LG	
<input type="checkbox"/>	01119059	Equipo Celular	LG	
<input type="checkbox"/>	0111930311	Equipo Celular	LG	
<input type="checkbox"/>	011193012	Equipo Celular	LG	
<input checked="" type="checkbox"/>	01119113	Equipo Celular	LG	
<input type="checkbox"/>	01128303	Equipo Celular	LG	
<input type="checkbox"/>	011129	Equipo Celular	LG	
<input type="checkbox"/>	0111930200	Equipo Celular	LG	
<input type="checkbox"/>	0111976	Equipo Celular	LG	

[Resultados Siguientes](#)

IDs Seleccionados				
<input type="text" value="01119303"/>	<input type="text" value="01119113"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Elementos a Transferir: 2

[Fig. 4.17 Interfaz de Transferencias]

4.2.8 Interfaz de Consultas.

Las consultas dentro del sistema están orientadas a buscar Propuestas y Accesorios o Equipos Celulares, dependiendo del operador y las tareas asociadas a él.

a) Consulta de Propuestas.

Interfaz manipulada por los roles de Representante de Fabricante y Persona de Inventario que les permite buscar propuestas específicas ya sea por el folio, el estado de la Propuesta o la marca. Fig. 4.18.

Consultas Asociadas:

```
SELECT dp.cantidad, dp.tipoArt, dp.modelo, dp.descArt, dp.pUnit, dp.pUnitIVA, p.montoSubtotal,
p.montoTotal
FROM PROPUESTA p, detalle_propuesta dp
WHERE p.folioProp=dp.folioProp
and p.folioProp=:pFolioProp;
```

```
SELECT dp.cantidad, dp.tipoArt, dp.modelo, dp.descArt, dp.pUnit, dp.pUnitIVA, p.montoSubtotal,
p.montoTotal
FROM PROPUESTA p, detalle_propuesta dp
WHERE p.folioProp=dp.folioProp
and p.estado=:pestadoProp;
```

```

SELECT dp.cantidad, dp.tipoArt, dp.modelo, dp.descArt, dp.pUnit, dp.pUnitIVA, p.montoSubtotal,
p.montoTotal
FROM PROPUESTA p, detalle_propuesta dp
WHERE p.folioProp=dp.folioProp
and p.marca=:pMarca;

```

24/09/2009

RF0001

Operador Espino Cisneros Esperanza

Folio: Estado Propuesta: Seleccione... Marca: Seleccione...

Consultar

Cantidad	Tipo Articulo	Modelo	Descripción	Precio	PrecioIVA

Siguiente Cuadrícula

Subtotal Total

[Generar Propuesta](#)
[Consulta Propuesta](#)
[Generar Primera Carga](#)
[Principal](#)
[Cerrar Sesión](#)

[Fig. 4.18 Interfaz Consulta Propuesta]

b) Consulta de Accesorios o Equipos Celulares.

Destinada a los roles de Persona de Inventario y Gerente Sucursal, esta interfaz les permite consultar Accesorios o Equipos Celulares, donde los criterios de búsqueda son por , marca, modelo y estatus del equipo o accesorio. Fig. 4.19.

30/09/2009

sucursal S0005

Operador Huerta Aguilar Gilberto

ID a Consultar: Tipo a Consultar: Equipo Celular Marca: LG Modelo:

Status: Seleccione... Almacén: Seleccione...

Buscar

ESNID	Status	Tipo Articulo	Modelo	Almacén
0111930305		Equipo Celular	LG	Central
01119303		Equipo Celular	LG	S0005
01119059		Equipo Celular	LG	Central
0111930311		Equipo Celular	LG	Central
011193012		Equipo Celular	LG	Central
01119113		Equipo Celular	LG	S0005
01128303		Equipo Celular	LG	Central
011129		Equipo Celular	LG	Central
0111930200		Equipo Celular	LG	Central
0111976		Equipo Celular	LG	

Resultados Sigüientes...

[Realizar Transferencia](#)
[Consultar Transferencia](#)
[Consultar](#)
[Accesorio/Equipo Celular](#)
[Principal](#)
[Cerrar Sesión](#)

[Fig. 4.19 Consulta de Accesorio y Equipos Celulares]

c) **Consulta de Transferencias.**

Su objetivo es interactuar con rol de Persona de Inventario y Gerente Sucursal para conocer el detalle de una Transferencia. Fig. 4.20.

17/11/2009

sucursal S0005

GS0003

Operador Huerta Aguilar
Gilberto

[Realizar Transferencia](#)

[Consultar Transferencia](#)

[Consultar Accesorio/Equipo Celular](#)

[Principal](#)

[Cerrar Sesion](#)

Consultar Tranferencias

Folio a Consultar

Almacen Origen: Seleccione...
Almacen Destino: Seleccione...

Buscar

Folio Transferencia	Fecha Origen	Fecha Movimiento
2	16/11/2009	17/11/2009

[Resultados Sigientes...](#)

Detalle de Transferencia

0111930305	011193113			

Elementos Transferidos: 2

[Fig. 4.20 Interfaz para Consultar Transferencias]



CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES.

Con respecto a nuestros objetivos se puede concluir que se generó un sistema eficiente, confiable y consistente desde el lanzamiento y trazado de la transacción hasta el almacenamiento de los datos.

El análisis, diseño e implementación cumplió su objetivo con la culminación del sistema producido bajo la metodología del Modelo de Cascada.

Con respecto al framework de Struts2 bajo el cual se desarrolló el sistema se puede decir que se logró emplear el estándar y con ello lograr una aplicación eficiente para la empresa.

También se logró conocer más a fondo la parte administrativa de Oracle 10g, además de la instalación del gestor de la Base de Datos y el cliente correspondiente que nos permite interactuar con la base radicada en el servidor.

El sistema se encuentra en la fase de pruebas y validaciones por parte del personal de la empresa.

RECOMENDACIONES.

Para una etapa posterior del sistema se propone trabajar con la forma de pago de aquellas propuestas aceptadas.

Permitir que las propuestas hagan referencia hacia una imagen del equipo o del accesorio, con el objetivo de hacer gráfica la aplicación y apoyar al operador.

Diseñar un formulario más llamativo para el operador con la ayuda de "tooltip".

Informar a la aplicación actual que genera las salidas de equipos y accesorios por medio de la venta que se generó en la nueva BD, la salida del equipo para empezar a generar un histórico de los accesorios.

Monitoreo del sistema para detectar posibles incongruencias dentro del flujo e implementar las nuevas reglas de negocio que surjan.

Bibliografía.

- [1] Pressman Roger: *Ingeniería De Software. Un Enfoque Práctico*, McGraw Hill, 2002.
- [2] Grady Booch, James Rumbaugh e Ivar Jacobson: *El Lenguaje Unificado de Modelado, Guía del Usuario 2/E*, Addison Wesley, 2006.
- [3] Alfredo Weitzenfiel: *Ingeniería De Software Orientada A Objetos Con Uml, Java E Internet*, Thompson Editores, 2004.
- [4] Ramez Elmasri, Shamkant B. Navathe: *Fundamentos de Sistemas de Bases de Datos*, Pearson/Addison Wesley, 2002
- [5] Silberschatz Abraham, F. Korth Henry, Sudarshan S.: *Fundamentos de Bases de Datos*, McGraw Hill, 2002.
- [6] Somodevilla María Josefa, *Sistemas de Bases de Datos*. Diplomado en Tecnologías de la Información, Facultad de Ciencias de la Computación, BUAP 2009.
- [7] Cesar Pérez: *ORACLE 10g Administración y Análisis de Bases de Datos*, ALFA OMEGA RA-MA, 2008.
- [8] Donald Brown, Chad Michael Davis, Scott Stanlick: *Struts In Action*, Manning, 2003
- [9] Ian Roughley, *Practical Apache Struts 2 Web2.0 Projects*, Apress, 2007.