



**BENEMÉRITA UNIVERSIDAD  
AUTÓNOMA DE PUEBLA**  
*Facultad de Ciencias de la Computación*

**SMovSuper: Sistema Móvil de Toma de  
Decisiones para la Compra en  
Supermercados**

Tesis Profesional

Para obtener el Título de

**Licenciada en Ciencias de la Computación**



Presenta:

**Landero Heredia Perla Alejandra**

Asesor:

**M.C. Meliza Contreras González**

**Puebla, Puebla, Septiembre 2012**

---

# Dedicatoria

## **A Dios...**

*Esto te lo dedico a ti, porque eres algo muy especial en mi vida y corazón, porque siempre me has dado fe para seguir adelante.*

## **A ti Mama...**

*Te dedico esta tesis a ti porque siempre has estado apoyándome en todo momento, brindándome la fortaleza necesaria para seguir mirando hacia adelante a través de los obstáculos. Gracias a ti por todo lo que has hecho en todos estos años de mi vida, por creer en mi y por tus grandes consejos.*

*¡Gracias mama!*

## **A ti Papa....**

*Por tu inmenso apoyo, por tus sabios consejos, por la fuerza y actitud que me has enseñado para no rendirme si me caigo y la manera de levantarme sin mirar de frente, por cultivar e inculcar ese sabio don de la responsabilidad. Por todo eso y mucho más.*

*¡Gracias Papa!*

## **A mis Hermanas....**

*A ustedes hermanas por soportar mi carácter, mi forma de ser, por aceptarme tal y como soy, porque son mis mejores, verdaderas y grandes amigas en todo lo bueno y malo, por todo ese ánimo que me dieron para llegar hasta esta etapa de mi vida, por todo su apoyo incondicional.*

*¡Gracias!*

*Si Gracias a ustedes mama, papa, hermanas, por la gran paciencia y comprensión que me tuvieron en el camino de mi carrera profesional, es por ello que estoy aquí para decirles que..... ¡Lo hemos logrado!*

## *Agradecimientos*

*Agradezco de manera muy especial a mi M.C Meliza Contreras González y Asesora de mi tesis profesional.*

*Agradezco infinitamente todo su apoyo, por compartir desinteresadamente sus conocimientos, experiencia, consejos y por su amistad.*

*Gracias por su ejemplo, paciencia y por haberme enseñado tanto.*

*A la Vicerrectoría de Investigación y Estudios de Posgrado de la Benemérita Universidad Autónoma de Puebla pues gracias al proyecto VIEP "Computación Humana para la detección de hábitos de consumo en clientes de portales web" se facilitó el proceso de terminación de la tesis puesto con la bibliografía financiada a través de él se concluyó satisfactoriamente.*

# ÍNDICE GENERAL

<b>INTRODUCCIÓN .....</b>	<b>6</b>
<b>CAPITULO 1 – MARCO TEORICO .....</b>	<b>8</b>
1.1 Dispositivos Móviles .....	8
1.1.1 Conceptos de dispositivos móviles .....	8
1.1.2 Tipos de dispositivos móviles .....	9
1.1.3 Tipos de aplicaciones .....	13
1.1.4 Herramientas para desarrollo sobre dispositivos móviles .....	16
1.1.5 Sistemas Operativos para dispositivos móviles .....	17
1.1.6 Micronavegadores .....	20
1.1.7 Lenguajes de los navegadores para dispositivos móviles .....	22
1.1.8 Emuladores .....	25
1.1.9 Manejadores de Bases de Datos para móvil .....	26
1.1.10 Ingeniería de Software .....	28
1.1.11 Conceptos de Diagramas de Casos de Uso .....	30
1.1.12 Servlets .....	32
<b>CAPITULO 2 – ANALISIS DEL SISTEMA .....</b>	<b>34</b>
2.1 Análisis del Sistema .....	34
2.2 Requerimientos del Sistema .....	34
2.3 Diagramas de Casos de Uso .....	35
2.4 Casos de Uso .....	38
<b>CAPITULO 3 – DISEÑO DEL SISTEMA .....</b>	<b>52</b>
3.1 Diagrama de Clases .....	52
3.2 Diagrama Entidad Relación .....	54
3.3 Modelo Relacional .....	56
3.4 Normalización .....	57
<b>CAPITULO 4 – IMPLEMENTACION Y PRUEBAS .....</b>	<b>63</b>
4.1 Diseño con Interfaz de Android .....	63
4.2 Implementación de la Base de Datos .....	70

4.3 Implementación del Sistema .....	71
4.3.1 Interfaz del Cliente .....	71
4.3.1.1 Pagina Principal .....	72
4.3.1.2 Registrarse .....	72
4.3.1.3 Ingresar .....	73
4.3.1.4 Menú de Opciones Principal.....	73
4.3.1.5 Formulario Crear Lista .....	73
4.3.1.6 Mis Listas .....	74
4.3.1.7 Menú de Opciones Listas.....	74
4.3.1.8 Formulario de Cotización.....	74
4.3.1.9 Menú de Opciones Productos .....	75
4.3.2 Interfaz del Administrador.....	75
4.3.2.1 Pagina Principal .....	76
4.3.2.2 Menú Principal.....	76
4.3.2.3 Menú Principal Altas.....	76
4.3.2.4 Formulario de Altas Marcas .....	77
4.3.2.5 Formulario de Altas Categorías .....	77
4.3.2.6 Formulario de Altas Supermercado .....	78
4.3.2.7 Formulario de Altas Sucursal .....	78
4.3.2.8 Formulario de Altas Producto Generico .....	79
4.3.2.9 Formulario de Altas Producto Comercial .....	79
4.3.2.10 Menú Principal Bajas .....	80
4.3.2.11 Formulario de Eliminar .....	80
4.3.2.12 Menú Principal Modificaciones .....	80
4.3.2.13 Formulario de Editar .....	81
4.3.2.14 Menú Principal Consultar .....	81
4.3.2.15 Formulario de Consultar .....	82
4.4 Pruebas del Sistema .....	83
4.4.1 Pruebas Servidor .....	83
4.4.1.1 Validación .....	83
4.4.1.2 Menú Principal Altas .....	84
4.4.1.3 Formulario Altas .....	84
4.4.1.4 Formulario Bajas .....	86
4.4.1.5 Formulario Editar .....	86
4.4.1.6 Formulario Consultar .....	88
4.4.2 Pruebas Cliente .....	89
4.4.2.1 Menú Principal .....	89
4.4.2.2 Opción Registrarse .....	89
4.4.2.3 Opción Ingresar .....	90
4.4.2.4 Crear Lista .....	92
4.4.2.5 Ver Mis Listas .....	92
4.4.2.6 Ver productos de lista .....	92
4.4.2.7 Cotizar .....	93
4.4.2.8 Opción Visualizar Productos .....	93
4.4.2.9 Opción Detalles .....	94
4.4.2.10 Opción Eliminar Lista .....	94
4.4.2.11 Opción Vaciar Lista .....	95
<b>CONCLUSION Y TRABAJO FUTURO .....</b>	<b>97</b>
<b>BIBLIOGRAFIA .....</b>	<b>98</b>

# Introducción

Los dispositivos móviles representan una nueva oportunidad de acceso para los sistemas de información. La rapidez y flujo de los negocios actuales requiere de tecnologías que brinden a las empresas de actualizaciones de datos en tiempo real y les permitan la mayor movilidad posible. La creación y desarrollo de aplicaciones y sistemas para dispositivos móviles responde de manera efectiva a estos requerimientos como una plataforma ideal que aporta grandes ventajas y utilidades específicas dentro de los procesos de una empresa como la comunicación con clientes y proveedores, enviar y recibir información, administrar su negocio, tener una oficina móvil, o la realización de trámites y/o operaciones propias del comercio electrónico sin importar donde se encuentre.

Actualmente, un dispositivo móvil representa un medio de comunicación ligero y portátil. Un dispositivo móvil le evita los inconvenientes de llevar una laptop a todos lados sin dejar de gozar de sus beneficios como agenda electrónica, programas, y todo aquello que requiere para continuar con su trabajo. Se podría decir que un dispositivo móvil permite realizar todas las funciones básicas de los equipos fijos brindándole, adicionalmente, la libertad de movimiento.

La tecnología actualmente, permite que algunos lenguajes de alto nivel se puedan utilizar para la programación de estos dispositivos móviles; tomando mucha fuerza en el ámbito laboral. Así el desarrollo para dispositivos móviles abarca uno de los aspectos más importantes y con más evolución dentro de la informática de los últimos años.

En esta tesis se presenta la realización de un cliente-servidor utilizando dispositivos móviles.

Dicho sistema consiste en generar cotizaciones de los productos de menor precio, es decir, el usuario accederá a los datos almacenados de una página web a través de un dispositivo móvil, mediante el uso de dicho dispositivo el usuario visualizará los productos deseados a través de la cotización generada, por el cual el sistema mostrará una lista de estos productos con el costo correspondiente, lugar y ubicación de dicho producto, así el usuario denotará cual es el producto de menor costo con el que su toma de decisión se facilitará de una manera rápida y eficaz.

Esta tesis se encuentra dividida de la siguiente manera:

### **Capítulo 1.**

Marco Teórico.

En este Capítulo se define los dispositivos móviles (Conceptos de dispositivos móviles, Tipos de dispositivos móviles, Tipos de aplicaciones, Herramientas para desarrollo sobre dispositivos móviles, Sistemas operativos para dispositivos móviles, Micronavegadores, Lenguajes de los navegadores para dispositivos móviles, Emuladores, Manejadores de Bases de datos para móvil, Ingeniería de Software, Conceptos de Diagramas de Casos de Uso).

### **Capítulo 2.**

Análisis del Sistema.

En este capítulo se define el análisis que es llevado a cabo en el sistema (Planteamiento del Problema, Objetivo General, Objetivos Específicos, Análisis de Requerimientos, Factores que influyen en la toma de decisión de la compra de supermercado, Casos de uso).

### **Capítulo 3.**

Diseño del Sistema.

En este capítulo se realiza el diseño del Sistema (Diagrama de Clases, Diagrama Entidad Relación, Modelo Relacional, Normalización).

### **Capítulo 4.**

Implementación y Pruebas del Sistema.

En este capítulo se realiza la implementación y se realizan las pruebas necesarias para la obtención de resultados satisfactorios (Diseño con Interfaz de Android, Implementación de la Base de Datos, Implementación del Sistema, Pruebas del Sistema).

### **Conclusión y Trabajo Futuro.**

En esta sección se describe la importancia de utilizar un alista de compras a través de un dispositivo móvil, así como las mejoras de este sistema.

### **Bibliografía.**

En esta sección se mencionan los libros utilizados durante el desarrollo de la Tesis.

# Capítulo 1

## Marco Teórico

### 1.1 Dispositivos Móviles

#### 1.1.1 Conceptos de dispositivos móviles

Los dispositivos móviles son aquellos que son suficientemente pequeños para ser transportados y empleados durante su transporte. Normalmente se sincronizan con un sistema de sobremesa para actualizar aplicaciones y datos. En la figura 1.1 se muestra los diferentes tipos de dispositivos móviles.

En inglés existe una amplia gama de términos para referirse a los dispositivos móviles: "information device", "information appliance", "consumer electronic", "embedded device" o "small device", por ejemplo. En definitiva:

- Son aparatos pequeños.
- Con funcionalidad limitada.
- Más económico.
- Con algunas capacidades de procesamiento (como son las funciones que tiene cada uno de los dispositivos móviles).
- Con conexión permanente o intermitente a una red.
- Con memoria limitada.
- Diseñados específicamente para una función, pero que pueden llevar a cabo otras más generales.
- Normalmente se asocian al uso individual de una persona, tanto en posesión como en operación, el cual puede adaptarlos a su gusto.
- La mayoría de estos aparatos pueden ser transportados en el bolsillo del propietario [R1].

## 1.1.2 Tipos de dispositivos móviles

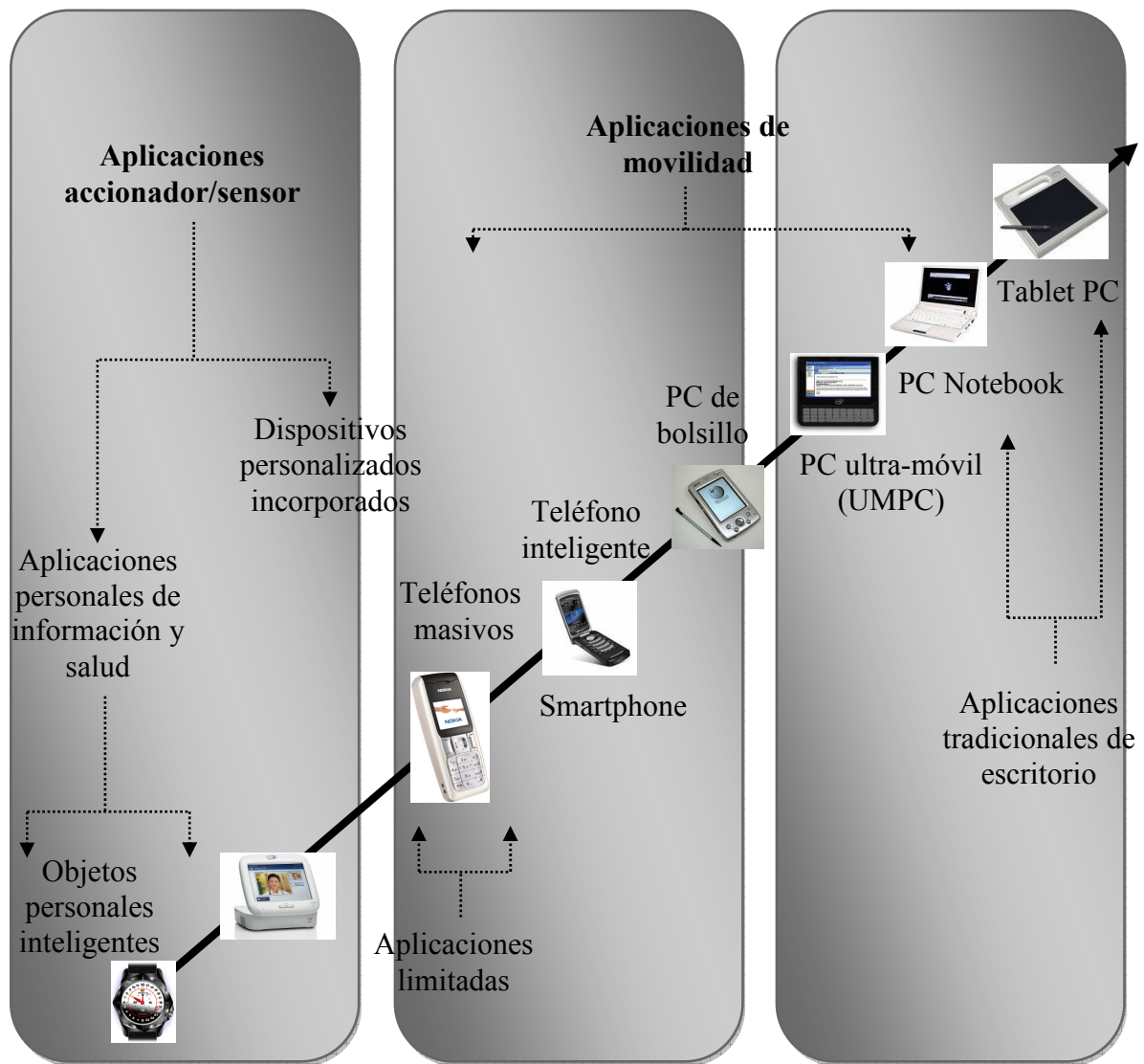


Figura 1.1 - Secuencia de los dispositivos móviles

A continuación se describe la figura 1.1:

Existen diversos tipos de dispositivos móviles como son: teléfonos masivos los cuales tienen aplicaciones limitadas, teléfonos inteligentes, PC de bolsillo, PC ultra-móvil, PC Notebook, Tablet PC.

**Dispositivo Móvil de Datos Limitados:**

(Limited Data Mobile Device): dispositivos que tienen una pantalla pequeña, principalmente basada en pantalla de tipo texto con servicios de datos generalmente limitados a SMS y acceso WAP. Un ejemplo de estos dispositivos móviles son los teléfonos móviles.

**Notebooks:**

Se utiliza el término inglés notebook para llamar a las computadoras o computadoras portátiles. También se utiliza la palabra laptop. Realiza las mismas tareas y tiene las mismas capacidades que una computadora de escritorio. Dentro de sus características se tiene:

- Capacidad de prestación: alta.
- Difusión: baja.
- Practicidad (uso, autonomía y conectividad): baja.

**Teléfono inteligente:**

Un Smartphone (Teléfono Inteligente en español) es un dispositivo que funciona como un teléfono celular con características similares a las de una computadora personal. Con un teléfono inteligente puedes recibir llamadas, revisar tu agenda mientras se visualizan videos o mientras se sincronizan el dispositivo con otros y todo esto sin necesidad de interrumpir alguna de las tareas. Dentro de las características de un teléfono inteligente se tiene:

- Permite la instalación de programas para incrementar el procesamiento de datos y la conectividad.
- Soporta correo electrónico.
- Cuenta con GPS.
- Utiliza cualquier interfaz para el ingreso de datos, como por ejemplo teclado QWERTY, pantalla táctil.
- Permite la conexión a Internet.
- Poseen agenda digital, administración de contactos.
- Permiten leer documentos en distintos formatos, entre ellos los PDFs y archivos de Microsoft Office.
- Cuenta con un Sistema Operativo.

### **PDA:**

(Personal Digital Assistant o Ayudante digital personal) es un dispositivo de pequeño tamaño que combina una computadora, teléfono/fax, Internet y conexiones de red. Dentro de las características de un PDA se tiene:

- Un PDA típico tiene una pantalla táctil para ingresar información, un slot de memoria para almacenarla.
- Algunos PDAs también contienen soporte para navegar por la red a través de Wi-Fi y revisar el correo electrónico.
- Se pueden trabajar documentos de Office.
- La mayoría de las PDAs permite el uso de tarjetas SD (Secure Digital SD es un formato de tarjeta de memoria flash).
- Una de las funciones más importantes de los PDAs es la sincronización con las computadoras personales. Esto permite la actualización del directorio, haciendo que la información de la computadora y del PDA sea la misma.

La sincronización también evita la pérdida de la información almacenada en caso de que el accesorio se pierda. La sincronización se realiza mediante un programa el HotSync Manager (Palm OS) y el Windows Mobile Device (Windows Mobile).

### **Tablet PC:**

Un Tablet PC es una computadora portátil con la capacidad de aceptar entrada de texto e imagen desde dispositivos de escritura manual: un lápiz sensible. De esta forma el usuario puede introducir datos manuscritos en la computadora simplemente escribiéndolos sobre la pantalla como si se tratase de una hoja de papel, de forma similar a los PDAs o asistentes personales. Estos dispositivos, incorporan un pequeño teclado gráfico sobre el que se puede escribir directamente. El lápiz digital proporciona una interactividad con el equipo informático más natural que otros dispositivos de comunicación con la computadora utilizados convencionalmente. Dentro de las características de la Tablet PC se tiene:

- Pantalla TFT
  - Puede ser activa (electromagnética) o pasiva (resistiva).
  - Mayor calidad y ángulo de visión las activas.
- Procesadores hasta 1 GHz.
- Memoria hasta 512 MB SDRAM.
- Sistema Operativo: Windows XP professional con funciones específicas para Tablet PCs.
- Tecnología wireless Wi-Fi, puerto USB, ethernet (RJ-45).

**PC Ultra Móvil:**

El Ultra Mobile PC (UMPC) o PC Ultra Móvil en español, es un Tablet PC más pequeño. Con pantalla táctil que se puede usar con lápiz o con la mano. Dentro de las características de la PC Ultra Móvil se tiene:

- Tamaño de pantalla máximo de 20 cm.
- Pantalla sensible al tacto (touch screen)
- Capaces de ejecutar cualquier variante de Linux, Windows XP y Vista.
- Debido a que los UMPC ejecutan el sistema operativo Windows XP, serán capaces de trabajar con cualquier software que se haya escrito para esta plataforma, aunque el reducido tamaño de la pantalla hará necesario que se ajusten algunos elementos en la interfaz.
- Para el manejo fácil del sistema operativo se incluye un paquete software conocido como el Touch Pack.
- Wi-Fi, Bluetooth o Infrarrojos para el intercambio de archivos con otros dispositivos y conexión a Internet sin cables.
- Contienen suficiente capacidad de proceso para soportar la edición de texto.
- Disco duro entre 20 y 100 GB.
- Procesadores AMD Geode (LX900, LX800) de 550, 600 MHz, Intel Celeron-M Core Dúo (U2400 de 1060MHz).
- 512 a 1GB de RAM.

### 1.1.3 Tipos de Aplicaciones

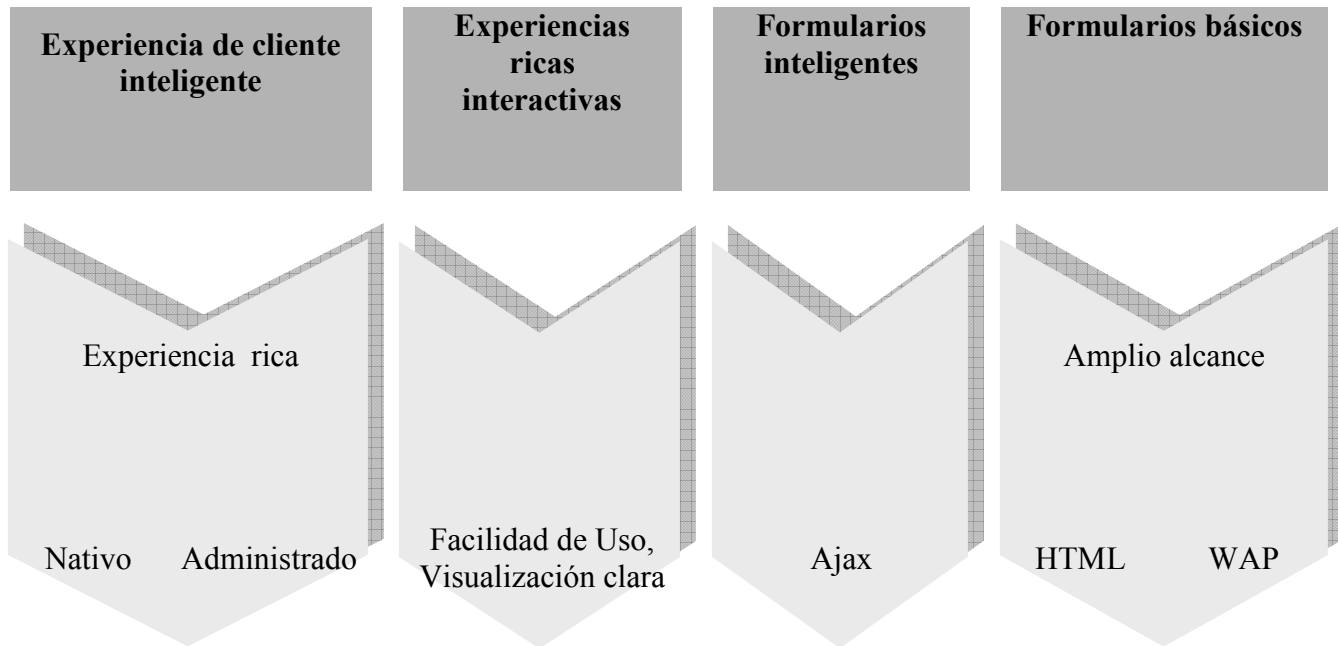


Figura 1.2 - Aplicaciones

A continuación se describe la figura 1.2:

Los tipos de aplicaciones se basan en experiencia de cliente inteligente el cual obtiene una experiencia rica de acuerdo al lenguaje nativo y al lenguaje administrado, de igual manera es basado en experiencias ricas interactivas como facilidad de uso del sistema, visualización clara de dicho sistema es decir, que el sistema sea entendible al usuario, existen formularios inteligentes como es el ajax, formularios básicos el cual contienen un amplio alcance como son: HTML y WAP. La experiencia de cliente inteligente se basa en una experiencia rica la cual contiene dos tipos de lenguajes: lenguaje nativo y lenguaje intermedio.

- **El lenguaje nativo:**

	IDE
Symbian OS	Eclipse + Plugin + SDK Visual Studio + SDK + Plugin
Windows Mobile	Visual Studio + SDK
Iphone OS	Xcode + SDK
Palm	Codewarrior + SDK

Figura 1.3 – Lenguaje Nativo

En la tabla de la figura 1.3 se menciona los sistemas operativos con los respectivos IDE's necesarios en cada uno de ellos, es decir, en los Sistemas Operativos es necesario que los lenguajes de programación contengan herramientas útiles para el programador como son los SDK's y Plugins. Algunos lenguajes de programación que contienen IDE's para los diversos Sistemas Operativos Móviles, son descritos a continuación:

- ✓ Eclipse contiene herramientas útiles como son los Plugins y SDK's, utilizado en el Sistema Operativo Symbian y Android.
- ✓ Visual Studio contiene herramientas útiles como son los SDK's, utilizado en el Sistema Operativo Windows Mobile.
- ✓ Xcode contiene herramientas útiles como son los SDK's, utilizado en el Sistema Operativo Iphone.
- ✓ Codewarrior contiene herramientas útiles como son los SDK's, utilizado en el Sistema Operativo Palm.

▪ **El lenguaje intermedio:**

El lenguaje intermedio es el lenguaje de una maquina abstracta diseñada para ayudar en el análisis de los programas. Es decir actúa como un compilador ya que primero traduce el código fuente del programa, en una forma mas apropiada para las transformaciones de mejora del código como un paso intermedio antes de generar el archivo objeto o el código maquina para una maquina específica.

	<b>Eclipse (Java ME)</b>	<b>Netbeans (Java ME)</b>	<b>Visual Studio (CF)</b>
Nokia	Plugin (SDK)	Plugin (SDK)	
Sony Ericsson	Plugin (SDK)	SDK	Plugin (SDK)
Samsung	SDK	Plugin (SDK)	SDK
Motorola	SDK	Plugin (SDK)	Plugin (SDK)
Blackberry	Plugin(SDK)	SDK	
Android	Plugin (SDK)	SDK	
Palm	SDK	SDK	SDK
Windows Based			SDK

**Figura 1.4 – Lenguaje Intermedio**

En la tabla 1.4 se menciona los diversos dispositivos móviles y lenguajes de programación para dichos dispositivos móviles y herramientas necesarias. Ejemplo:

En el dispositivo móvil Nokia el lenguaje de programación Eclipse contiene herramientas necesarias como son los Plugins.

### Las experiencias ricas interactivas:

Es la interacción que va adquiriendo el programador mediante las aplicaciones del lenguaje nativo y lenguaje intermedio sobre los dispositivos móviles.

### Formularios Inteligentes:

Un ejemplo de dichos formularios es la validación de cliente servidor. Como se muestra en la figura 1.5.

Nombre	<input type="text"/>
Apellidos	<input type="text"/>
	<small>Su nombre y dos apellidos se enviarán con todos los mensajes de correo electrónico salientes.</small>
Idioma	<input type="text" value="Español"/>
País/región	<input type="text" value="España"/> <small><a href="#">Privacidad para residentes de la Unión Europea</a></small>
Provincia	<input type="text" value="([Elija una])"/>
Zona horaria	<input type="text" value="Horario universal - GMT"/>
Sexo	<input type="radio"/> Hombre <input type="radio"/> Mujer
Fecha de nacimiento	<input type="text" value="Día"/> <input type="text" value="Mes"/> <input type="text"/> (p. ej., 1999)
Ocupación	<input type="text" value="([Seleccione una ocupación])"/>

Figura 1.5 – Validación cliente - servidor

Al presionar el botón enviar una función JS se encarga de obtener los valores de los campos y validarlos en cliente utilizando diferentes criterios. Si algún campo es erróneo se cambia su className y se muestra un mensaje de error si no existen errores se llama a la función JS la cual abrirá la conexión XMLHttpRequest con el servidor enviándole los datos que el usuario ha completado. El cliente recibirá la cadena de respuesta del servidor.

### Formularios básicos:

- **WAP:**

Son las siglas de Wireless Application Protocol, un estándar seguro que permite que los usuarios accedan a información de forma instantánea a través de dispositivos inalámbricos como PDAs, teléfonos móviles, teléfonos inteligentes (Smartphones.).

- **HTML:**

Siglas de HyperText Markup Language, es el lenguaje de marcado predominante para la construcción de paginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

## 1.1.4 Herramientas para desarrollo sobre dispositivos móviles

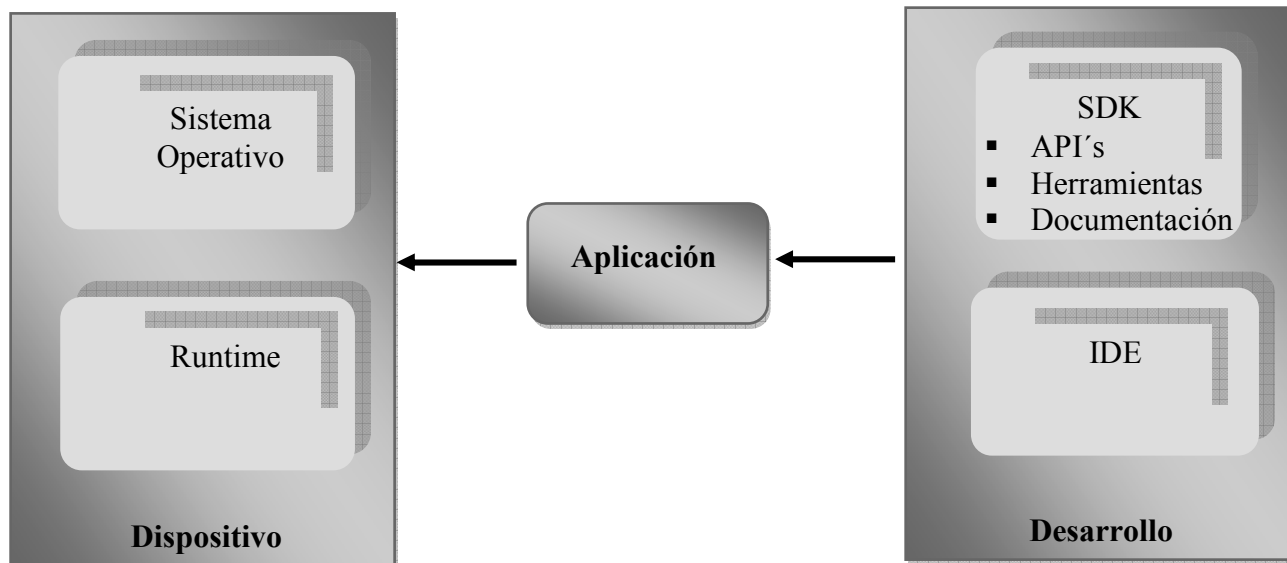


Figura 1.6 - Herramientas para Desarrollo sobre Dispositivos Móviles

A continuación se describe la figura 1.6:

### **Sistema Operativo:**

Existen diversos sistemas operativos para los dispositivos móviles: Symbian, BlackBerry OS, Windows Mobile, iPhone OS, Android, Linux.

### **SDK:**

Un software development kit (SDK) o kit de desarrollo de software es generalmente un conjunto de herramientas de desarrollo que le permite al programador crear aplicaciones para un sistema concreto, como son paquetes de software, frameworks. Estos SDK contienen archivos de cabecera, bibliotecas, muestras, documentación y herramientas que utilizan las API necesarias para desarrollar aplicaciones.

### **API:**

Application Programming Interface. Es un grupo de rutinas que provee un sistema operativo, una aplicación o una biblioteca, que definen como invocar desde un programa un servicio que estos presentan. Una API representa un interfaz de comunicación entre componentes software.

### **IDE:**

Integrated Development Environment. Aplicación compuesta por un conjunto de herramientas útiles para un programador.

Un entorno IDE puede ser exclusivo para un lenguaje de programación o bien, puede utilizarse para varios. Suele consistir de un editor de código, un compilador, un debugger y un constructor de interfaz grafica GUI. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

## 1.1.5 Sistemas Operativos para dispositivos móviles

Un sistema operativo móvil o SO móvil es un sistema operativo que controla un dispositivo móvil al igual que los PCs utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos [R2].

A continuación se listan los sistemas operativos existentes para dispositivos móviles:

### **Symbian OS:**

Symbian OS es un sistema operativo propietario, es una colección compacta de código ejecutable y varios archivos, la mayoría de ellos bibliotecas DLL. Está diseñado para dispositivos móviles, se ejecuta exclusivamente en procesadores ARM.

El sistema operativo Symbian se encuentra cargado en la memoria flash del teléfono móvil. La ventaja de tener un sistema operativo en los dispositivos móviles es porque tiene la capacidad de instalar aplicaciones existentes y obtener funcionalidad que se desee, como leer documentos, agendas privadas. La mayor ventaja de Symbian OS es que fue construido para dispositivos portátiles, con recursos limitados.

Pone fuerte énfasis en la conservación de la memoria, usando idiomas de programación específicos de Symbian tales como descriptores y compiladores de limpieza.

Junto con otras técnicas esto mantiene el uso de la memoria bajo. Además la programación de el OS Symbian esta basada en eventos. Todo esto hace del C++ del Symbian OS muy especializado, sin embargo muchos dispositivos pueden también ser programados en OPL, Pitón, Visual Basic, Simkin, Perl, junto con Java ME y modalidades de Personal Java y Java.

### **Windows Mobile:**

Es un sistema operativo compacto, con una suite de aplicaciones básicas para dispositivos móviles basados en la API Win32 de Microsoft. Los dispositivos móviles que llevan Windows Mobile son Pocket PC, Smartphones y Media Center portátil.

Tanto el Windows Mobile para Pocket PC, como Windows Mobile para Smartphone, poseen bastantes aspectos parecidos, pero en cuanto a las aplicaciones desarrolladas, una aplicación no servirá para ambas versiones Windows Mobile PPC (Pocket PC) y Windows Mobile SP (Smartphone).

Versiones de Windows Mobile:

- Originalmente apareció con el nombre de Pocket PC 2000 para los dispositivos del mismo nombre. Este sistema operativo estaba basado en Windows CE 3.0.
- Luego se llamo Pocket PC 2002.
- Se lanza Windows Mobile 2003, fue el primero en llevar el nombre de Windows Mobile.
- Windows Mobile 2003 SE.
- Windows Mobile 5.0.
- Windows Mobile 6.

### **iPhone OS:**

El iPhone OS es el sistema operativo que utiliza el iPod touch y el iPhone. Está basado en una variante del Mac kernel que se encuentra en Mac OS X. El iPhone OS incluye el componente de software Animación Core de MAC OS X v10.5 que, junto con el Power VR MBX el hardware de 3D, es responsable de las animaciones usadas en la interfaz de usuario. iPhone OS tiene 4 capas de abstracción: la capa del núcleo del sistema operativo, la capa de servicios principales, la capa de medios de comunicación y la capa de Cocoa Touch. El sistema operativo ocupa bastante menos de medio GB del total del dispositivo, 8 GB o el almacenaje de 16 GB. Este sistema operativo no tenía un nombre oficial hasta que salió la primera versión beta del iPhone SDK. Antes Apple había declarado que el iPhone controla un sistema operativo OS X.

### **Palm OS:**

Es un sistema operativo para PDAs, PPC para pseudo PCs de bolsillo. Las aplicaciones Palm se ejecutan en un entorno emulado denominado el Entorno de Compatibilidad de Aplicaciones Palm disminuyendo velocidad pero permitiendo gran compatibilidad con programas antiguos. Hay muchas aplicaciones interesantes para el sistema operativo Palm OS que se pueden añadir. Una de las características de las máquinas con Palm OS es que contiene infrarrojo es decir que se pueden comunicar con otros dispositivos que incluyan un puerto infrarrojo como por ejemplo algunos modelos de impresora, computadoras portátiles y modelos de teléfono móviles así como también a las redes de computadora con las que pueda establecer comunicación por medio de Wi-Fi.

### **Android:**

Es una plataforma de software y un sistema operativo para dispositivos móviles basado en el núcleo de Linux. Android permite a los desarrolladores escribir código gestionado en el lenguaje de programación Java y controlar los dispositivos por medio de bibliotecas desarrolladas. Dentro de las características de Android tiene:

- Framework de aplicaciones: permite reuso y reemplazo de componentes.
- Máquina virtual Dalvik: optimizada para dispositivos móviles.
- Navegador integrado: basado en el motor open source WebKit.
- Gráficos optimizados, con una librería de gráficos 2D; gráficos 3D basado en la especificación OpenGL ES 1.0 (aceleración de hardware opcional).
- SQLite para almacenamiento de datos estructurados.
- Telefonía GSM (dependiente del hardware).
- Bluetooth, EDGE, 3G, y Wi-Fi (dependiente del hardware).
- Ambiente rico de desarrollo incluyendo un emulador de dispositivo, herramientas para depurar, perfiles de memoria y rendimiento, y un plugin para el IDE Eclipse.
- Pantalla táctil.

- Android Market permite que los desarrolladores pongan sus aplicaciones, gratuitas o de pago, en el mercado a través de esta aplicación accesible desde todos los teléfonos con Android.

**BlackBerry OS:**

BlackBerry OS 4.7 es un sistema completo con todas las funcionalidades necesarias en un smartphone, mientras que el OS X 2.1 es un sistema en desarrollo. Dentro de las características de Blackberry Os tiene:

- Sistema operativo de RIM (Research In Motion Limited es una compañía canadiense de dispositivos inalámbricos mas conocido como el promotor del dispositivo de comunicación de mano BlackBerry).
- Disponibles versiones.
- Paquetes de utilidades como es Office y PDF.
- SDK gratuito.
- JME (Java Mobile Edition).
- Familia BlackBerry.

## 1.1.6 Micronavegadores

También llamado microbrowser, mobile browser o mini-browser. Es un tipo de navegador diseñado para ser usado en pantallas pequeñas de diferentes tipos de dispositivos móviles. Generalmente es usado en teléfonos basados en WAP, que requieren un Gateway, para traducir páginas web, feeds y otros contenidos de Internet [R3]. Dentro de los micronavegadores que existen para los dispositivos móviles se encuentran:

### **OperaMini:**

Es un navegador web para dispositivos móviles desarrollados con Java ME. Funciona en la mayoría de los teléfonos que soportan midlets Java. OperaMini obtiene todo el contenido a través de un proxy que corre el motor de maquetación del navegador Opera. Este motor en el servidor proxy reformatea las páginas web con la anchura adecuada para pantallas pequeñas.

### **Mini Mozilla:**

Es un navegador web para pequeños dispositivos como PDAs y teléfonos móviles. El navegador usa tecnología de renderizado de pantalla pequeña para reformatear páginas web para pantallas de PDA.

### **Pocket Internet Explorer:**

(PIE), Pocket Internet Explorer es un navegador de Internet para Pocket PC y Handheld PC que viene instalado por defecto en Windows Mobile y Windows CE para Handheld PC. Hace operaciones parecidas que el popular navegador Internet Explorer, Microsoft desarrollo Pocket Internet Explorer e Internet Explorer para Windows CE, pero el Pocket Internet Explorer consume menos recursos y ofrece menos funciones.

### **Skyfire:**

Skyfire es un navegador web para móviles basado en Windows Mobile, y que ofrece poderosas características como soporte para Flash, Silverlight y AJAX. Skyfire ofrece una barra de direcciones, denominada Super Bar y que cuenta con características de búsqueda similar a la que podemos ver en Google Chrome. Skyfire es compatible con móviles basados en Windows Mobile, aunque se está trabajando en el desarrollo de una versión para móviles basados en Symbian.

### **NetFront:**

Es un microbrowser para dispositivos embebidos. Principalmente usado en teléfonos celulares, NetFront está disponible para muchas plataformas y es usado también en PDAs, web phones (teléfonos web), consolas de juegos, terminales de correo electrónica y otros tipos de dispositivos. Ofrece fácil personalización y una amplia gama de plug-ins. NetFront fue diseñado para funcionar como un navegador embebido en dispositivos móviles. Entre sus características se encuentran:

- Permite la navegación por pestañas.
- PagePilot (ampliación y reducción de imágenes).
- Virtual Pointer (imita a un puntero).
- Manejo inteligente de marcos (selecciona y amplía marcos por separado).
- Zoom ajustable continuo.
- Navegación sin conexión y grabación de páginas e imágenes.
- Bloqueador de ventanas emergentes inteligente.
- Búsqueda en Internet integrada.
- Rellenado automático de formularios.
- Administrador de contraseñas.
- Nombres de dominio internacionales.

### **Safari:**

Es un navegador web desarrollado por Apple Inc. Es el navegador incluido en Mac OS X y recientemente se han desarrollado versiones ejecutables para Microsoft Windows. Safari está escrito sobre el framework WebKit, que incluye a WebCore, el motor de renderizado, y JavaScriptCore, el intérprete de JavaScript. El motor de renderizado utilizado para renderizar páginas web llamado WebKit está basado en el motor KHTML.

Como resultado de esto, el motor interno de Safari es software libre. Las mejoras al código de KHTML por parte de Apple son incorporadas al código de KDE rápidamente. Dentro de sus funcionalidades se encuentran:

- Interfaz de navegación por pestañas, la cual permite arrastrar las pestañas para reordenarlas, moverlas entre ventanas, o crear nuevas ventanas a partir de pestañas.
- Una barra de búsqueda web, con la posibilidad de modificar su longitud. Se usa el buscador Google para Mac OS X y Google o Yahoo para búsquedas bajo Microsoft Windows.
- Bloqueo de ventanas emergentes.
- Corrección ortográfica al rellenar campos de texto.
- Cuadros para introducir texto con tamaño expandible.
- Manejo de contraseñas integrado bajo el sistema Keychain.
- Funcionalidad de suscripción a feeds de contenido nuevo en sitios web.
- Alto nivel en el cumplimiento de estándares web.

### **Nokia N60 browser:**

Es un navegador, open source, creado por Nokia a partir del WebKit de Apple y el KHTML de Konqueror, para su serie de móviles 60, incluidos: Nokia E60, Nokia E61, Nokia E70, Nokia 3250, Nokia N80, Nokia N71, Nokia N91 and Nokia N92. Está licenciado bajo LGPL.

## 1.1.7 Lenguajes de los navegadores para dispositivos móviles

### Compact HTML (cHTML):

Subconjunto de HTML para dispositivos pequeños de información como teléfonos inteligentes y PDAs.

CHTML es una versión del HTML regular más reducida. Estas restricciones se deben a que los pequeños dispositivos suelen tener pequeñas capacidades de memoria, poco poder de procesamiento, limitada o ninguna capacidad de almacenamiento, pequeñas pantallas monocromáticas, entre otros. En la figura 1.7 se muestra el navegador compacto para dispositivos móviles.

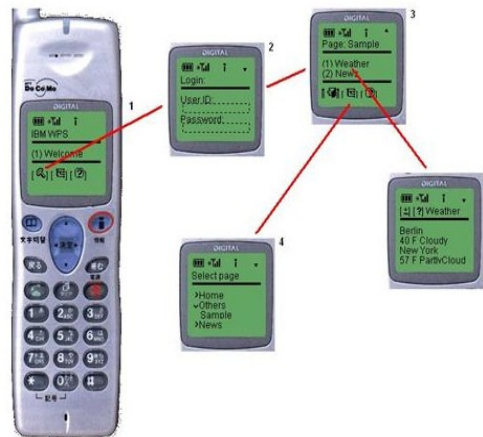


Figura 1.7 – Compact HTML

### HTML:

Es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML o en forma descendida directamente de SGML.

### WML:

Es un lenguaje cuyo origen es el XML. Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales (PDA) dotados de tecnología WAP. Es una versión reducida del lenguaje HTML que facilita la conexión a Internet de dichos dispositivos y que además permite la visualización de páginas web en dispositivos inalámbricos que incluyan la tecnología WAP. WML es un metalenguaje, lo que implica que además de usar etiquetas predefinidas se pueden crear componentes

propios y tiene ciertas similitudes con otro lenguaje de etiquetas bastante conocido, el HTML, utilizado para la creación de páginas web convencionales. El soporte de WML tiene:

- Entrada del Usuario. WML soporta diversas listas de selecciones, lista de selección multi-nivel, captura de texto, y control de tareas.
- Estado y Administración del contexto. Para maximizar los recursos de la red, WAP permite variables para el traslado entre archivos WML. Esto en lugar de estar enviando una cadena completa, las variables, pueden ser enviadas y sustituidas en el tiempo de ejecución.

### **VoiceXML:**

VoiceXML, es un estándar de comunicación basado en voz, el cual es considerado como un lenguaje XML, el cual juega el rol de lenguaje en una aplicación de comunicación de voz, similar al rol jugado por HTML en aplicaciones WEB. VoiceXML, es relativamente una nueva especificación en el campo de la tecnología inalámbrica. Este fue diseñado por Motorola.

Las ventajas del uso de VoiceXML, como un lenguaje de audio son que:

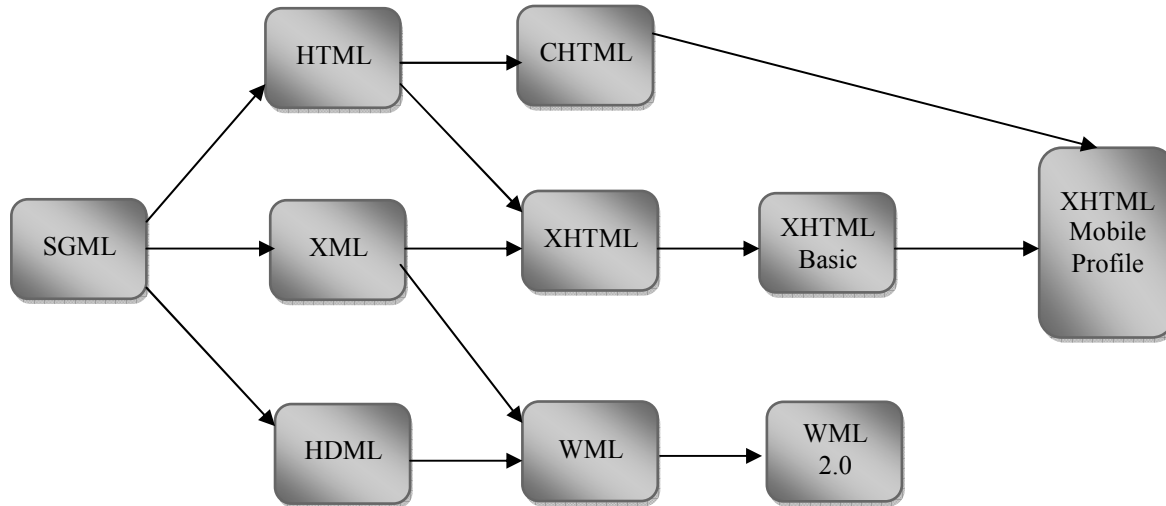
- Minimiza las interacciones entre cliente/servidor, mediante múltiples interacciones específicas en los documentos.
- Protege las aplicaciones de los desarrolladores, desde un nivel bajo, así como los detalles de plataformas específicas.
- Separa el código de interacción del usuario, lo que le da a VoiceXML, un servicio más lógico.
- Promueve servicios de portabilidad, a lo largo de la implementación de plataformas VoiceXML, ya que es un lenguaje común para proveedores de contenido, proveedores de herramientas y proveedores de plataformas.

### **XHTML:**

Es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. En su versión 1.0, XHTML es solamente la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. Como XHTML tiene un código más limpio, elimina gran cantidad de errores que antes ocurrían con HTML, además de la incompatibilidad entre dispositivos. No importa si un sitio utiliza tecnología PHP, ASP, JSP o cualquier otra, porque XHTML se integra a ellas y las complementa de manera exitosa.

**XHTML-MP:**

El tipo de documento XHTML Mobile Profile o XHTML-MP es un superconjunto de XHTML Basic creado por la Open Mobile Alliance, con algunas extensiones añadidas. Es una alternativa de facto a XHTML Basic y forma parte de la especificación WAP 2.0 (aunque se sigue manteniendo también el soporte de WML 1.x).



**Figura 1.8** – Diagrama de aplicaciones XHTML Profile

El diagrama de la figura 1.8 se define de la siguiente manera:

SGML son las siglas de Standard Generalized Markup Language o Lenguaje de Mercado Generalizado. Consiste en un sistema para la organización y etiquetado de documentos. La Organización Internacional de Estándares (ISO) normalizó este lenguaje en 1986. El lenguaje SGML sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial. El lenguaje HTML está definido en términos del SGML. XML es un estándar de creación posterior, que incorpora un subconjunto de la funcionalidad del SGML. Es una evolución de los lenguajes que existen para dispositivos móviles.

## 1.1.8 Emuladores

Un emulador es un software que permite ejecutar programas de computadora en una plataforma (arquitectura hardware o sistema operativo) diferente de aquella para la cual fueron escritos originalmente.

Cada tipo de emulador tiene su propio formato de Roms, variando también dependiendo de la consola. Dentro de los emuladores se tienen:

<http://www.winwap.com/downloads/downloads>: En esta liga se proporcionan emuladores para dispositivos móviles como: Winwap para Windows Mobile Professional (touch - screen), Winwap para Windows Mobile standard (non touch - screen), Winwap Smartphone Browser Emulator. Esta página nos muestra un emulador con el nombre de WinWAP Smartphone Browser Emulator, dicho emulador muestra un ejemplo que contiene un menú y en dicho menú las opciones de pagina principal, búsqueda, colocar una dirección URL.

<http://emulator.mtld.mobi/emulator.php>: En esta liga se proporcionan emuladores para dispositivos móviles como: .mobi emulador móvil. En esta pagina muestra dos emuladores para dispositivos móviles, Nokia N70 y Sony K750. En dichos emuladores muestra un ejemplo sencillo de una página Web en la cual contiene opciones básicas como: pagina principal, limpiar la pagina principal, limpiar cache.

<http://www.netbeans.org/>: En esta liga se proporcionan emuladores para dispositivos móviles (Java ME), dicho software contiene algunas aplicaciones para dispositivos móviles que pueden ser visualizadas de dispositivos como Motorola, Nokia, Sony Ericsson o en una nueva configuración. Contiene una paleta la cual muestra formas básicas como son: Botón, CheckBox, Textfield, entre otras, de las cuales nos permite crear diversas interfaces en los dispositivos móviles. Permite editar imágenes que pueden ser exportadas como archivos PNG y JPG.

<http://developer.android.com/sdk/index.html>: En esta liga se proporciona el SDK de Android, el cual los emuladores proporcionados trabajan bajo la plataforma de Eclipse. Este tipo de emuladores se visualizan en dispositivos móviles actuales como Sony Ericsson, Nokia, Samsung o en una nueva configuración.

## 1.1.9 Manejadores de Bases de Datos para móvil

En los últimos años se han producido grandes avances en las tecnologías de comunicación inalámbricas. Gracias a estos avances, los usuarios pueden acceder a una base de datos remota en cualquier momento y en cualquier lugar. Las bases de datos móviles son portables, físicamente independiente del servidor desde sitios remotos para compartir datos corporativos. Entre las bases de datos basadas en dispositivos móviles se encuentran:

- SQL Server Compact Edition (Windows Mobile)

<http://www.microsoft.com/SqlServer/2005/en/us/compact.aspx>: En esta liga se proporciona información sobre las herramientas para bases de datos para dispositivos móviles. Dichas herramientas permiten crear aplicaciones móviles que funcionan en plataformas como Windows, Windows XP, Windows Vista, Pocket PC y Smartphone. El funcionamiento de dicho software esta basado en .NET Compact Framework. SQL Server Compact Edition, permite crear base de datos de SQL Server CE en dispositivos móviles y establecer conexiones con las bases de datos de SQL Server que se encuentran en el dispositivo o en un servidor remoto.

- DB4O (Java and .NET)

<http://developer.db4o.com/>: En esta liga se proporciona información sobre las herramientas de bases de datos en dispositivos móviles. DB4O es el motor de base de objetos que es utilizado en Java y .NET

- HandBase

<http://www.ddhsoftware.com/developers.html>: En esta liga se proporciona información sobre las herramientas de bases de datos en dispositivos móviles. HandBase es una base de datos relacional que trabaja bajo las plataformas PDA. Dicho software incluye registros con campos de texto, numéricos, flotantes, fechas, cálculos entre campos, entre otras posibilidades. HandBase contiene una opción en la que se puede agregar el emulador del dispositivo móvil para ello es necesario registrar el dispositivo móvil con un costo, así de esta manera será visualizado dicho dispositivo dentro del software. HandBase Forms Designer es un software que es complemento de HandBase, dicho software tiene la opción de realizar formularios los cuales son manejados dentro de la base de datos creada desde HandBase, y así sean visualizadas en el dispositivo móvil. Un ejemplo de ello se muestra en la figura 1.22.

En la figura se muestra una base de datos elaborada bajo el software HandBase, en la opción de new record (nuevo registro) muestra un formulario que se ha elaborado bajo el software HandBase Forms Designer, en dicho formulario se agregaran los registros a la base de datos.

- DB2 Everyplace

<http://www-01.ibm.com/software/data/db2/everyplace/edition-db.html>: En esta liga se proporciona información sobre las herramientas de bases de datos en dispositivos móviles. DB2 Everyplace es una base de datos relacional y un servidor de sincronización que permite extender las aplicaciones y los datos empresariales a dispositivos móviles, esta base de datos puede integrarse en dispositivos como PDAs y teléfonos móviles, permite diseñar formularios, asignar botones y campos a los mismos y definir otras acciones.

- Oracle Lite 10g

<http://www.oracle.com/technology/products/lite/index.html>: En esta liga se proporciona información sobre las herramientas de bases de datos en dispositivos móviles. Es conocido como el Oracle Móvil, esta basado en los sistemas operativos Windows CE y Palm OS.

- SQL Anywhere

<http://www.sybase.com/products/databasemanagement/sqlanywhere>: En esta liga se proporciona información sobre las herramientas de bases de datos en dispositivos móviles. SQL Anywhere proporciona base de datos de tipo empresarial que pueden usarse en servidores y acceder desde dispositivos móviles.

- SQLite

SQLite es un sistema multiplataforma y pequeño gestor de base de datos como MySQL, con la diferencia que es menor en tamaño y peso, la base de datos se guarda en archivos que se almacenan en el dispositivo. En esta liga se explica detalladamente los requisitos de uso e instalación: <http://www.sqlite.org/>

## 1.1.10 Ingeniería de Software

La ingeniería de Software es una disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después de que se utiliza. La ingeniería de software no solo comprende los procesos técnicos del desarrollo del software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas. Las herramientas de ingeniería de software son los métodos necesarios para desarrollar el sistema, como son las herramientas CASE y el uso de lenguajes de diseño como el UML [1].

### ▪ Herramientas CASE:

Las herramientas CASE son un conjunto de métodos, utilidades y técnicas (diagrama de flujo de datos, modelo entidad - relación) que facilitan la automatización del ciclo de vida de desarrollo de sistemas de información, completamente o en alguna de sus fases.

El empleo de herramientas CASE permiten integrar:

- Generación de interfaces entre el análisis y el diseño.
- Generación del código a partir del diseño.

Las herramientas CASE pueden clasificarse atendiendo a:

- Las plataformas que soportan.
- Las fases del ciclo de vida del desarrollo de sistemas que cubren.
- Su funcionalidad.

### ▪ UML:

UML es un lenguaje de modelado de sistemas más utilizado. UML es un lenguaje gráfico que permite visualizar, especificar, construir y documentar un sistema de software. Es importante decir que UML es un lenguaje que especifica y no describe métodos o procesos. En UML existen diferentes tipos de diagramas, como son los Diagramas de casos de uso, el cual se describirá en la sección 1.1.10.

Entre las ventajas que ofrece UML se encuentran:

1. Diseño y documentación.
2. Código reutilizable.
3. Descubrimiento de fallas.
4. Las modificaciones son más fáciles.

En el diagrama de la figura 1.9 se describe la funcionalidad de un Sistema Restaurante. Los casos de uso están representados por elipses y los actores están representados por las figuras humanas. El actor Crítico de comidas puede probar la comida, pagar la comida, o beber vino. Sólo el actor Chef puede preparar la comida. El marco define los límites del sistema Restaurante, es decir, los casos de uso se muestran como parte del sistema que está siendo modelado, los actores no.

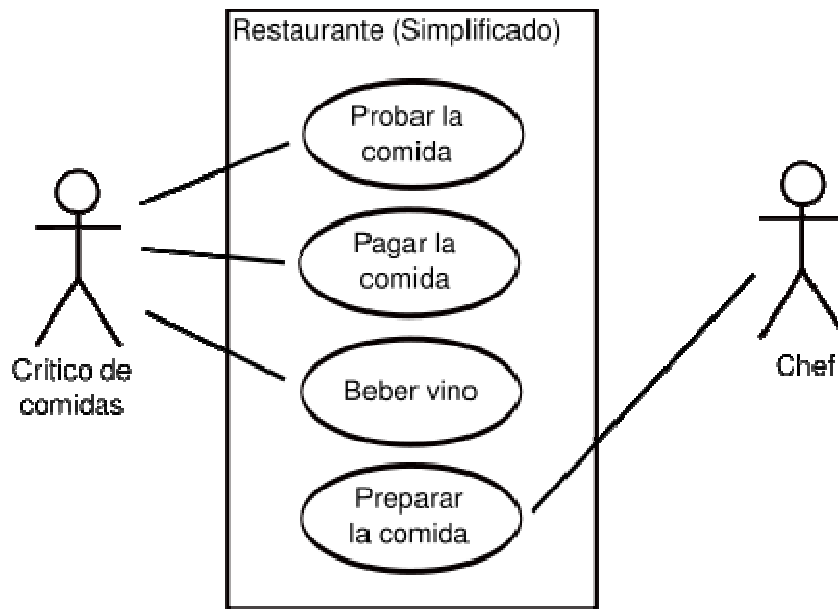


Figura 1.9 – Sistema Restaurante

## 1.1.11 Conceptos de Diagramas de Casos de Uso

Un diagrama de Casos de Uso representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Los elementos que pueden aparecer en un diagrama de casos de uso son:

- **Actores:**

Se le llama actor a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos así como a entidades abstractas como el tiempo. Se representa mediante una figura humana con líneas y círculos.

- **Casos de Uso:**

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema que lleva a una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el diagrama de Casos de Uso mediante una elipse con el nombre de caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

- **Relaciones entre casos de uso:**

Un caso de uso, en principio, debería describir una tarea que tiene un sentido completo para el usuario. Sin embargo, hay ocasiones en las que es útil describir una interacción con un alcance menor como caso de uso. La razón para utilizar estos casos de uso no completos en algunos casos, es mejorar la comunicación en el equipo de desarrollo, el manejo de la documentación de casos de uso. Para el caso de que queramos utilizar estos casos de uso más pequeños, las relaciones entre estos y los casos de uso ordinarios pueden ser de los siguientes tres tipos:

- *<<extends>>*: Se puede incluir una relación entre dos casos de uso de tipo incluye, si se desea especificar diferentes variantes del mismo caso de uso. Es decir, esta relación implica que el comportamiento de un caso de uso, es diferente dependiendo de ciertas circunstancias. En principio esas variaciones pueden también mostrarse como diferentes descripciones de escenarios asociadas al mismo caso de uso.
- *Generalización*: En un diagrama de casos de uso también pueden mostrarse generalizaciones (relaciones de herencia) así mostrar que elementos diferentes están relacionados como tipos de otros. Son aplicables a actores o casos de uso, pero en estos últimos la semántica es muy similar a las relaciones extend.

- <<*include*>>: Se puede incluir una relación entre dos casos de uso tipo include si se desea especificar comportamiento común en dos o más casos de uso. Ejemplo, existen tres casos de uso X, Y, Z, entonces X, Y hacen algo en común y ambos necesitan de Z.

## 1.1.12 Servlets

Un servlet es un programa que se ejecuta en un servidor Web. Los clientes pueden invocarlo utilizando el protocolo HTTP. Comparativamente, lo mismo que un applet es cargado y ejecutado por un explorador, un servlet es cargado y ejecutado por un servidor Web.

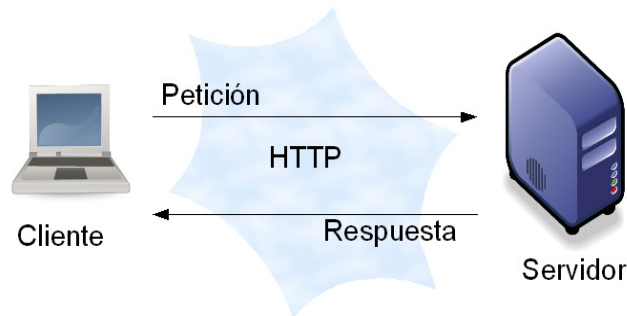


Figura 1.10 – Estructura de un servlet

La figura 1.10 indica que un servlet acepta peticiones de un cliente, procesa la información y le devuelve resultados que podrán ser mostrados mediante applets, páginas HTML, etc. Sin embargo esta no es la única función de un servlet; puede realizar también otras tareas como comunicarse con otro servlet para ayudarlo en su trabajo, o bien facilitar el acceso a bases de datos.

En definitiva las características principales de los Servlets son:

- Al estar escritos en java, son independientes de la plataforma.
- Consumen menos recursos porque solo son cargados la primera vez que solicitan sus servicios. Las siguientes peticiones crearan hilos de ejecución.
- Son más rápidos que los programas CGI y que los scripts porque, por un lado se precompilan y por otro, no se tiene que generar un nuevo proceso cada vez que se invocan.
- Son seguros y portables debido a que se ejecutan bajo la máquina virtual de java, al mecanismo de excepciones de java, y al uso del administrador de seguridad de java.
- No requieren soporte para java en el explorador del cliente, ya que operan en el dominio del servidor y envían los resultados en HTML. No obstante, se pueden utilizar otros interfaces de cliente como aplicaciones java o applets.

### Estructura de un servlet

A continuación se describe la figura 1.11:

#### Método **init()**

- Se ejecuta una sola vez al inicializar el Servlet
- Inicializar variables y operaciones costosas en tiempo de ejecución

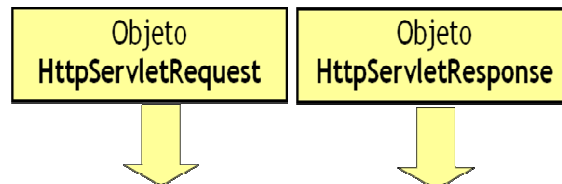
#### Métodos **doGet()** o **doPost()**

- Recoger peticiones del usuario y ejecutar operaciones
- Mandar respuesta al usuario (en forma de HTML)

Otros métodos de usuario

#### Método **destroy()**

- Lo llama el servidor al “apagarse”
- Cerrar procesos en curso, liberar memoria, cerrar ficheros



**Figura 1.11** – Estructura de un servlet

El API Servlet es claro y simple. Un servlet es una clase Java que implementa la interfaz Servlet, que define cinco métodos:

- ✓ **service()**: Es el corazón de los servlets. El servidor invoca al método `service()` para ejecutar respuestas. El método `service()` acepta como parámetros objetos `ServletRequest`, que encapsula la petición del cliente y `ServletResponse`, que dispone de métodos para devolver información al cliente.
- ✓ **init()**: Es el lugar donde se inicializa el servlet. Es un método que acepta como parámetros un objeto de tipo `ServletConfig`, que contiene la configuración del servidor, y se garantiza que solamente se llamará una vez durante la vida del servlet
- ✓ **getServletConfig()**: Debe devolver el objeto `ServletConfig` que se pasa como parámetro al método `init()`.
- ✓ **destroy()**: Libera el servlet. Se llama cada vez que el servlet debe ser descargado. Todos los recursos del sistema bloqueados por `init()` son liberados al invocar este método y se garantiza que solo se le llamará una vez durante el la vida del servlet.
- ✓ **getServletInfo()**: devuelve una cadena con la información de Copyright.

Para asegurar un óptimo rendimiento, el servidor solamente carga una vez el servlet. Una vez cargado, permanece en memoria, estando disponible en cualquier instante para procesar cualquier petición. Por lo tanto, varias tareas pueden llamar simultáneamente al método `service()`, por lo que la sincronización dentro de `service` debe ser una premisa a no olvidar jamás.

# Capítulo 2

## Análisis del Sistema

### 2.1 Análisis del Sistema

La manera en que se pensó en desarrollar el sistema fue dada a las necesidades del usuario por lo cual cuando acude a un supermercado realizan una lista de compras mediante una hoja de papel, por lo que dicho usuario acude a diversos supermercados para comparar precios y tomar la decisión de compra. Por lo tanto es una pérdida de tiempo y gastos adicionales.

Para desarrollar el sistema propuesto se considera que los usuarios a través de un dispositivo móvil accedan al sistema.

### 2.2 Requerimientos Funcionales

Para construir algo primero debe entenderse lo que debe ser ese “algo”. Al proceso de entender y documentar este algo se denomina “análisis de requerimientos”. En general, los requerimientos expresan qué se supone que debe hacer una aplicación [2].

Los Requerimientos Funcionales del Sistema son los siguientes:

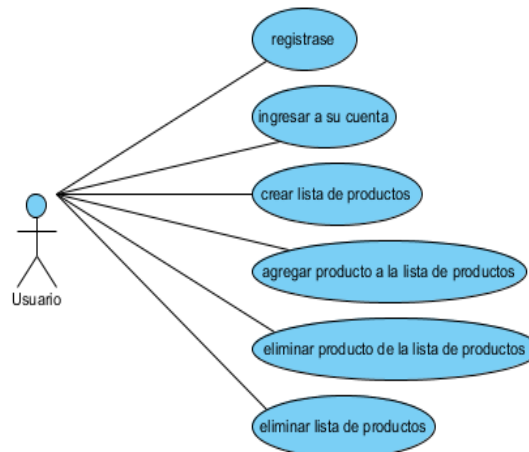
- El sistema debe requerir que el usuario se identifique mediante un nombre de usuario y una contraseña
- El sistema debe permitir registrar los datos de nuevos usuarios
- El sistema debe permitir al usuario crear una lista de compras
- El sistema debe permitir al usuario agregar productos a la lista de compras
- El sistema debe permitir al usuario eliminar la lista de compras
- El sistema debe generar la cotización de la lista de productos existentes
- El sistema debe permitir al usuario vaciar la lista de productos
- El sistema debe mostrar la cotización resultante en un orden de menos a mayor dentro de un rango de 5 productos
- El sistema debe permitir al Administrador las operaciones de altas, bajas, consultas, modificaciones de los productos genericos y comerciales.

## 2.3 Diagramas de Casos de Uso

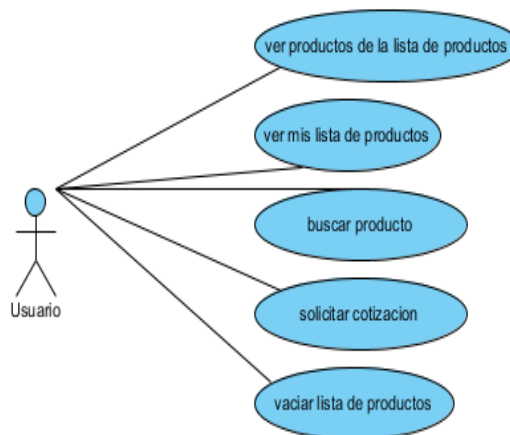
En general, un caso de uso es, simplemente, un texto escrito que describe el papel de un actor que interactúa con el acontecer del sistema [3]. A continuación en la figura 2.1 y 2.2 se muestra dividido en dos bloques los casos de uso del actor usuario. En las figuras 2.3, 2.4, 2.5 2.6 y 2.7 se muestran dividido en cinco bloques los casos de uso del actor administrador.

Rol Involucrado

- Usuario



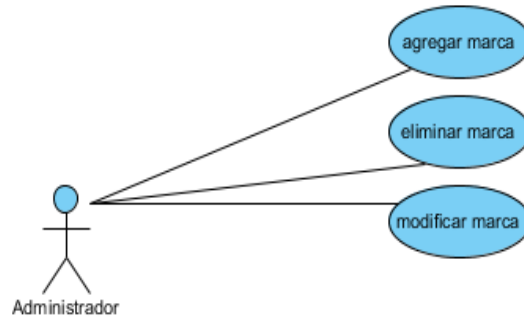
**Figura 2.1** Primer bloque de casos de uso del usuario



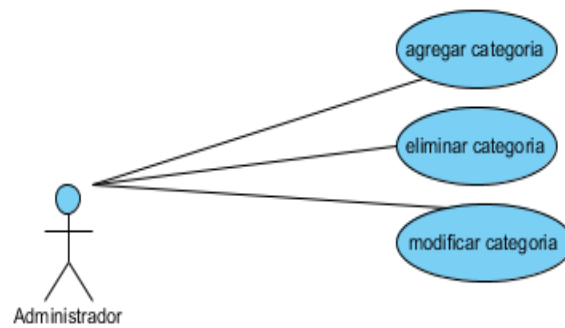
**Figura 2.2** Segundo bloque de casos de uso del usuario

Rol Involucrado

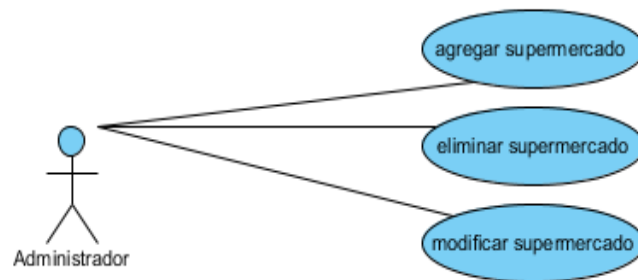
- Administrador



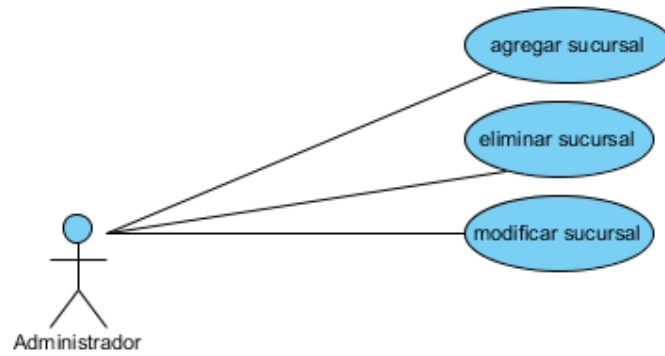
**Figura 2.3** Primer bloque de casos de uso del administrador



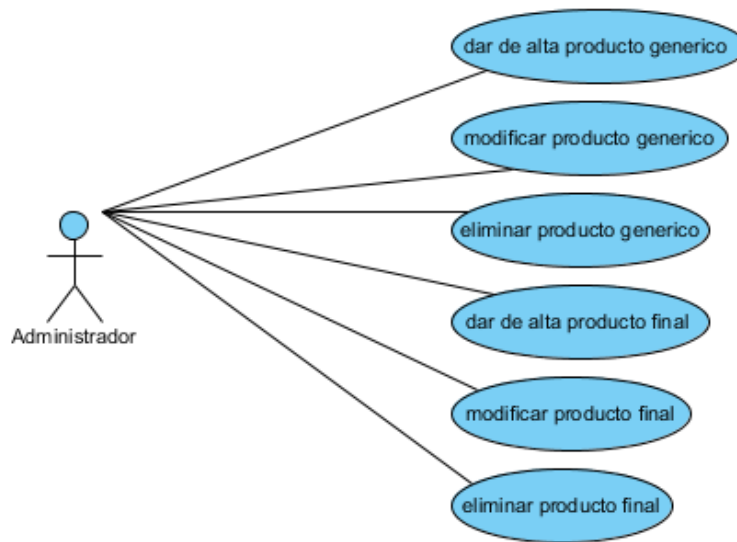
**Figura 2.4** Segundo bloque de casos de uso del administrador



**Figura 2.5** Tercer bloque de casos de uso del administrador



**Figura 2.6** Cuarto bloque de casos de uso del usuario



**Figura 2.7** Quinto bloque de casos de uso del usuario

## 2.4 Casos de Uso

<b>Casos de Uso:</b>	Registrarse
<b>Descripción:</b>	El usuario se registra en el sistema
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario no debe existir en la base de datos
<b>Flujo Principal:</b>	<ol style="list-style-type: none"> <li>1.- El usuario ingresa a la página principal</li> <li>2.- El usuario selecciona la opción registrarse</li> <li>3.- El sistema muestra al usuario el formulario con los campos nombre de usuario y contraseña donde proporciona sus datos para registrarse</li> <li>4.- El usuario llena el formulario con la información</li> <li>5.- El usuario da click en el botón crear cuenta</li> <li>6.- El sistema almacena los datos</li> </ol>
<b>Poscondición:</b>	El usuario existe en la base de datos
<b>Flujo Alternativo:</b>	El sistema envía un mensaje de error si el usuario existe

<b>Casos de Uso:</b>	Ingresar a su cuenta
<b>Descripción:</b>	El usuario ingresa a su cuenta
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario existe en la base de datos
<b>Flujo Principal:</b>	<ol style="list-style-type: none"> <li>1.- El sistema pregunta nombre de usuario y contraseña</li> <li>2.- El usuario ingresa su nombre de usuario, contraseña y selecciona ingresar</li> <li>3.- El sistema valida que el nombre de usuario y contraseña existan.</li> </ol>
<b>Poscondición:</b>	El usuario ha ingresado al sistema
<b>Flujo Alternativo:</b>	El sistema envía un mensaje de error si el usuario y contraseña no existen. En caso contrario el usuario ingresa a su cuenta

<b>Casos de Uso:</b>	Crear lista de productos
<b>Descripción:</b>	El usuario genera su lista de productos del supermercado
<b>Actores:</b>	Usuario

<b>Precondición:</b>	El usuario debe haber ingresado a su cuenta
<b>Flujo Principal:</b>	<ol style="list-style-type: none"> <li>1.- El usuario pulsa el botón crear lista de productos</li> <li>2.- El sistema muestra los campos nombre de lista y fecha</li> <li>3.- El usuario llena los campos con los datos requeridos</li> <li>4.- El usuario da click en el botón ok</li> <li>4.- El sistema almacena los datos</li> </ol>
<b>Poscondición:</b>	El sistema almacena la nueva lista de productos
<b>Flujo Alternativo:</b>	El sistema envía un mensaje de error si se deja en blanco alguno de los campos

<b>Casos de Uso:</b>	Agregar producto a la lista de productos
<b>Descripción:</b>	Se agregan los productos ingresados por el usuario a la lista de productos
<b>Actores:</b>	Usuario
<b>Precondición:</b>	Existir una lista de productos
<b>Flujo Principal:</b>	<ol style="list-style-type: none"> <li>1.- El usuario selecciona la opción añadir producto</li> <li>2.- El sistema muestra un formulario con los campos nombre del producto y fecha</li> <li>3.- El usuario llena los campos con los datos requeridos</li> <li>4.- El usuario da click en el botón añadir</li> <li>5.- El sistema agrega el nuevo producto a la lista de productos</li> </ol>
<b>Poscondición:</b>	El nuevo producto ha sido almacenado a la lista de productos
<b>Flujo Alternativo:</b>	El sistema envía un mensaje de error si se deja algún campo en blanco

<b>Casos de Uso:</b>	Eliminar producto de la lista de productos
<b>Descripción:</b>	Elimina un producto seleccionado por el usuario de la lista de productos
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El producto debe encontrarse en la lista
<b>Flujo Principal:</b>	<ol style="list-style-type: none"> <li>1.- Se ejecuta el caso de uso buscar producto</li> <li>2.- El usuario da click en el botón eliminar</li> <li>3.- El sistema borra de la lista el producto</li> </ol>

<b>Poscondición:</b>	El usuario guarda cambios de la lista presionando el botón guardar
<b>Flujo Alternativo:</b>	El usuario inicia la acción de eliminar producto y el sistema le muestra un mensaje de éxito.

<b>Casos de Uso:</b>	Eliminar lista de productos
<b>Descripción:</b>	Se elimina la lista de productos seleccionada
<b>Actores:</b>	Usuario
<b>Precondición:</b>	Debe existir una lista de productos
<b>Flujo Principal:</b>	1.- El usuario visualiza las listas de productos existente 2.- El usuario selecciona la opción eliminar lista 3.- El sistema borra la lista de productos seleccionada por el usuario
<b>Poscondición:</b>	El usuario regresa al menú principal
<b>Flujo Alternativo:</b>	El sistema le muestra un mensaje de éxito

<b>Casos de Uso:</b>	Ver productos de la lista de productos
<b>Descripción:</b>	Ver el contenido de una lista de productos
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario debe haber ingresado a su cuenta
<b>Flujo Principal:</b>	1.- Se ejecuta el caso de uso ver mis listas de productos 3.- El usuario da click en el botón ver contenido de la lista de productos 4.- El sistema muestra los productos contenidos
<b>Poscondición:</b>	El usuario visualiza los productos contenidos en la lista de productos seleccionada
<b>Flujo Alternativo:</b>	Si la lista de productos se encuentra vacía se muestra un mensaje de decisión, si desea agregar productos a la lista

<b>Casos de Uso:</b>	Ver mi lista de productos
<b>Descripción:</b>	El sistema mostrara al usuario las listas de productos existentes
<b>Actores:</b>	Usuario

<b>Precondición:</b>	El usuario debe haber ingresado a su cuenta
<b>Flujo Principal:</b>	1.- El usuario da click en la opción mis listas de productos 2.- El sistema muestra las listas de productos existentes
<b>Poscondición:</b>	El usuario visualiza las listas de productos existentes
<b>Flujo Alternativo:</b>	Si no existen listas de productos, se envía un mensaje de decisión, si desea crear una nueva lista de productos.

<b>Casos de Uso:</b>	Buscar producto
<b>Descripción:</b>	permite realizar la búsqueda del producto en la lista de productos
<b>Actores:</b>	Usuario
<b>Precondición:</b>	Deben existir productos en la lista de productos
<b>Flujo Principal:</b>	1.- El usuario coloca el nombre del producto que desea buscar en el campo búsqueda 2.- El usuario da click en el botón buscar 3.- El sistema muestra el producto que el usuario solicito en la búsqueda
<b>Poscondición:</b>	El sistema muestra el producto solicitado por el usuario
<b>Flujo Alternativo:</b>	El sistema envía un mensaje de error si no existen productos

<b>Casos de Uso:</b>	Solicitar cotización
<b>Descripción:</b>	El usuario solicita la cotización de los productos a partir de una lista existente
<b>Actores:</b>	Usuario
<b>Precondición:</b>	El usuario debe haber seleccionado una lista de productos
<b>Flujo Principal:</b>	1.- El usuario visualiza las listas existentes 2.- El usuario selecciona la opción generar cotización 3.- El sistema genera la cotización seleccionada por el usuario 4.- El sistema muestra el resultado de la cotización desde el producto mas económico hasta el producto de mayor precio en un rango de 5 productos

<b>Poscondición:</b>	El sistema muestra los productos resultantes de la cotización solicitada por el usuario
<b>Flujo Alternativo:</b>	El sistema envía un mensaje de error si no existen productos en la lista

Casos de Uso:	Vaciar lista de productos
Descripción:	Se eliminan todos los productos existentes de la lista de productos
Actores:	Usuario
Precondición:	El usuario debe haber seleccionado una lista de productos a vaciar
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El usuario da click en el botón vaciar lista de productos</li> <li>2.- El sistema elimina todos los productos existentes de esa lista dejando la lista de productos vacía</li> <li>3.- El sistema muestra la lista de productos vacía</li> </ol>
Poscondición:	Los productos han sido eliminados de la lista de productos
Flujo Alternativo:	Se envía un mensaje de decisión si desea vaciar la lista de productos.

Casos de Uso:	Agregar marca
Descripción:	Se agrega la nueva marca
Actores:	Administrador
Precondición:	El administrador debe haber seleccionado el formulario a llenar del menú principal
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador da click en añadir marca</li> <li>2.- El administrador ingresa los datos de la nueva marca, introduciendo el nombre en el campo nombre de marca</li> <li>3.- El administrador pulsa el botón guardar</li> <li>4.- El sistema comprueba la validez de los datos y los almacena</li> </ol>
Poscondición:	La nueva marca ha sido almacenadas en el sistema
Flujo Alternativo:	El sistema envía un mensaje de error si el campo se encuentra en blanco

Casos de Uso:	Agregar categoria
Descripción:	Se agrega la nueva categoria
Actores:	Administrador
Precondición:	El administrador debe haber seleccionado el formulario a llenar del menú principal
Flujo Principal:	<p>1.- El administrador da click en añadir categoria</p> <p>1.- El administrador ingresa los datos de la nueva categoria, introduciendo el nombre en el campo nombre de categoria</p> <p>2.- El administrador pulsa el botón guardar</p> <p>3.- El sistema comprueba la validez de los datos y los almacena</p>
Poscondición:	La nueva categoria ha sido almacenada en el sistema
Flujo Alternativo:	El sistema envía un mensaje de error si el campo se encuentra en blanco o introduce un carácter distinto a una letra

Casos de Uso:	Agregar supermercado
Descripción:	Se agrega el nuevo supermercado
Actores:	Administrador
Precondición:	El administrador debe haber seleccionado el formulario a llenar del menú principal
Flujo Principal:	<p>1.- El administrador da click en añadir supermercado</p> <p>1.- El administrador ingresa los datos del nuevo supermercado, introduciendo el nombre en el campo nombre de supermercado</p> <p>2.- El administrador pulsa el botón guardar</p> <p>3.- El sistema comprueba la validez de los datos y los almacena</p>
Poscondición:	El nuevo supermercado ha sido almacenado en el sistema
Flujo Alternativo:	El sistema envía un mensaje de error si el campo se encuentra en blanco

Casos de Uso:	Agregar sucursal
Descripción:	Se agregan la nueva sucursal dependiendo del supermercado al que pertenecen
Actores:	Administrador
Precondición:	El administrador debe haber seleccionado el formulario supermercado
Flujo Principal:	1.- El administrador ingresa una nueva sucursal dependiendo del supermercado , introduciendo como campo el nombre de la sucursal y establecimiento en el campo ubicación 2.- El administrador pulsa el botón guardar 3.- El sistema almacena los datos
Poscondición:	La nueva sucursal ha sido almacenada en el sistema
Flujo Alternativo:	El sistema envía un mensaje de error si el campo se encuentra en blanco

Casos de Uso:	Eliminar marca
Descripción:	Elimina la marca que ha sido seleccionada
Actores:	Administrador
Precondición:	El administrador visualizara las marcas existentes del formulario
Flujo Principal:	1.- El administrador busca la marca que desea eliminar 2.- El administrador selecciona dicha marca 3.- El administrador pulsa el botón eliminar 4.- El sistema muestra las marcas existentes
Poscondición:	La marca seleccionada ha sido eliminada del sistema
Flujo Alternativo:	El administrador inicia la acción de eliminar una marca y el sistema le muestra un mensaje de decisión

Casos de Uso:	Eliminar categoria
Descripción:	Elimina la categoria que ha sido seleccionada
Actores:	Administrador
Precondición:	El administrador visualizara las categorias existentes del formulario

Flujo Principal:	1.- El administrador busca la categoría que desea eliminar 2.- El administrador selecciona dicha categoría 3.- El administrador pulsa el botón eliminar 3.- El sistema muestra las categorías existentes
Poscondición:	La categoría seleccionada ha sido eliminada del sistema
Flujo Alternativo:	El administrador inicia la acción de eliminar una categoría y el sistema le muestra un mensaje de decisión

Casos de Uso:	Eliminar supermercado
Descripción:	Elimina el supermercado que ha sido seleccionado
Actores:	Administrador
Precondición:	El administrador debe haber ingresado a la opción eliminar supermercado del menú supermercado
Flujo Principal:	1.- El administrador busca el supermercado que desea eliminar 2.- El administrador selecciona dicho supermercado 3.- El administrador pulsa el botón eliminar 4.- El sistema muestra los supermercados existentes
Poscondición:	El supermercado seleccionado ha sido eliminado del sistema
Flujo Alternativo:	El administrador inicia la acción de eliminar un supermercado y el sistema le muestra un mensaje de decisión

Casos de Uso:	Eliminar sucursal
Descripción:	Elimina una sucursal seleccionada por el administrador dependiendo del supermercado
Actores:	Administrador
Precondición:	El administrador visualiza el formulario de supermercados existentes con las sucursales correspondientes de dicho supermercado
Flujo Principal:	1.- El administrador busca la sucursal que desea eliminar 2.- El administrador selecciona la sucursal 3.- El administrador pulsa el botón eliminar 4.- El sistema muestra las sucursales existentes de

	dicho supermercado
Poscondición:	La sucursal seleccionada ha sido eliminada del sistema
Flujo Alternativo:	El administrador inicia la acción de eliminar una sucursal y el sistema le muestra un mensaje de decisión

Casos de Uso:	Modificar marca
Descripción:	Modifica el nombre de la marca reemplazando con un nuevo nombre
Actores:	Administrador
Precondición:	El administrador visualizara las marcas existentes en el formulario
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador busca la marca que desea modificar</li> <li>2.- El administrador selecciona dicha marca</li> <li>3.- El administrador selecciona la opción modificar</li> <li>4.- El sistema muestra el campo nombre de marca</li> <li>5.- El administrador modifica la marca que selecciono</li> <li>6.- El administrador da click en el botón guardar cambios</li> <li>7.- El sistema almacena los datos modificados</li> </ol>
Poscondición:	El administrador regresa a la página principal
Flujo Alternativo:	El sistema envía un mensaje de error si el campo se ha dejado en blanco

Casos de Uso:	Modificar categoria
Descripción:	Modifica el nombre de la categoria reemplazando con un nuevo nombre
Actores:	Administrador
Precondición:	El administrador visualizara las categorias existentes en el formulario
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador busca la categoria que desea modificar</li> <li>2.- El administrador selecciona dicha categoria</li> <li>3.- El administrador selecciona la opción modificar</li> <li>4.- El sistema muestra el campo nombre de categoria</li> </ol>

	<p>5.- El administrador modifica la categoría que selecciono</p> <p>6.- El administrador da click en el botón guardar cambios</p> <p>7.- El sistema almacena los datos modificados</p>
Poscondición:	El sistema ha almacenado los nuevos datos reemplazados , regresa a la página principal del menú categoría
Flujo Alternativo:	El sistema envía un mensaje de error si el campo se ha dejado en blanco

Casos de Uso:	Modificar supermercado
Descripción:	Modifica el nombre del supermercado reemplazando con un nuevo nombre
Actores:	Administrador
Precondición:	El administrador visualizara los supermercados existentes en el formulario
Flujo Principal:	<p>1.- El administrador busca el supermercado que desea modificar</p> <p>2.- El administrador selecciona dicho supermercado</p> <p>3.- El administrador selecciona la opción modificar</p> <p>4.- El sistema muestra el campo nombre de supermercado</p> <p>5.- El administrador modifica el nombre de supermercado que selecciono</p> <p>6.- El administrador da click en el botón guardar cambios</p> <p>7.- El sistema almacena los datos modificados</p>
Poscondición:	El sistema ha almacenado los nuevos datos reemplazados , regresa a la página principal del menú supermercado
Flujo Alternativo:	El sistema envía un mensaje de error si el campo se ha dejado en blanco o si se ha introducido un espacio

Casos de Uso:	Modificar sucursal
Descripción:	Modifica el nombre de la sucursal reemplazando con un nuevo nombre
Actores:	Administrador
Precondición:	El administrador visualizara las sucursales

	existentes en el formulario
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador busca la sucursal que desea modificar</li> <li>2.- El administrador selecciona dicha sucursal</li> <li>3.- El administrador selecciona la opción modificar</li> <li>4.- El sistema muestra el campo nombre de sucursal</li> <li>5.- El administrador modifica el nombre de la sucursal que selecciono</li> <li>6.- El administrador da click en el botón guardar cambios</li> <li>7.- El sistema almacena los datos modificados</li> </ol>
Poscondición:	El sistema ha almacenado los nuevos datos reemplazados , regresa a la página principal del menú supermercado
Flujo Alternativo:	El sistema envía un mensaje de error si el campo se ha dejado en blanco o si se ha introducido un espacio

Casos de Uso:	Dar de alta producto genérico
Descripción:	El administrador captura información del nuevo producto genérico
Actores:	Administrador
Precondición:	El producto no ha de existir en la base de datos
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador selecciona el menú producto genérico</li> <li>2.- El administrador da click en la opción alta producto</li> <li>3.- El sistema muestra un formulario del producto, con sus respectivos datos a llenar como: nombre de producto, costo proveedor, descripción.</li> <li>4.- El administrador introduce los datos del producto</li> <li>5.- El sistema incorpora un nuevo producto en la base de datos</li> </ol>
Poscondición:	El sistema ha incorporado un nuevo producto a la base de datos
Flujo Alternativo:	Si el sistema detecta que falta por rellenar algún campo mostrara un mensaje de error

Casos de Uso:	Modificar producto genérico
Descripción:	El administrador modifica la información del producto deseado
Actores:	Administrador
Precondición:	El administrador visualiza el formulario con la lista de productos registrados en la base de datos
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador elige el producto a modificar</li> <li>2.- El administrador presiona el botón modificar</li> <li>3.- El sistema muestra el formulario modificar producto con los datos del producto</li> <li>4.- El administrador modifica los datos necesarios</li> <li>5.- El administrador presiona el botón guardar cambios</li> <li>6.- El sistema valida las entradas del administrador en los campos de texto</li> <li>7.- El sistema inserta los datos modificados en la base de datos</li> </ol>
Poscondición:	El caso de uso termina cuando el administrador presiona el botón guardar cambios o salir en dicho formulario
Flujo Alternativo:	El sistema muestra un mensaje de éxito o error de la operación de modificación

Casos de Uso:	Eliminar producto genérico
Descripción:	Elimina el producto que ha sido seleccionado
Actores:	Administrador
Precondición:	El administrador visualiza el formulario con la lista de productos registrados en la base de datos
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador elige el producto a eliminar</li> <li>2.- El administrador presiona el botón eliminar</li> <li>3.- El sistema borra dicho producto de la base de datos</li> </ol>
Poscondición:	El caso de uso termina cuando el usuario decide salir del formulario
Flujo Alternativo:	El administrador inicia la acción de eliminar un producto genérico y el sistema le muestra un mensaje de decisión

Casos de Uso:	Dar de alta producto final
Descripción:	El administrador captura la información del nuevo producto con precio al público
Actores:	Administrador
Precondición:	El producto no ha de existir en la base de datos
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador selecciona el menú producto final</li> <li>2.- El administrador da click en la opción alta producto final</li> <li>3.- El sistema muestra un formulario del producto, con sus respectivos datos a llenar como: supermercado en el que se encuentra, sucursal, marca, categoría, costo por sucursal .</li> <li>4.- El administrador introduce los datos del producto</li> <li>5.- El administrador da click en el botón dar de alta</li> <li>6.- El sistema incorpora un producto en la base de datos</li> </ol>
Poscondición:	El sistema ha incorporado un producto a la base de datos
Flujo Alternativo:	Si el sistema detecta que falta por rellenar algún campo mostrara un mensaje de error

Casos de Uso:	Modificar producto final
Descripción:	El administrador modifica la información del producto deseado reemplazándola con nueva información
Actores:	Administrador
Precondición:	El administrador visualiza el formulario con la lista de productos registrados en la base de datos
Flujo Principal:	<ol style="list-style-type: none"> <li>1.- El administrador elige el producto a modificar</li> <li>2.- El administrador presiona el botón modificar</li> <li>3.- El sistema muestra el formulario modificar producto con los datos del producto</li> <li>4.- El administrador modifica los datos necesarios</li> <li>5.- El administrador presiona el botón guardar cambios</li> <li>6.- El sistema valida las entradas del administrador introducidas en los campos</li> <li>7.- El sistema inserta los datos modificados en la base de datos</li> </ol>

Poscondición:	El caso de uso termina cuando el administrador presiona el botón guardar cambios o salir en dicho formulario
Flujo Alternativo:	El sistema muestra un mensaje de éxito o error de la operación de modificación

Casos de Uso:	Eliminar producto final
Descripción:	Elimina el producto que ha sido seleccionado
Actores:	Administrador
Precondición:	El administrador visualiza el formulario con la lista de productos registrados en la base de datos
Flujo Principal:	1.- El administrador elige el producto a eliminar 2.- El administrador presiona el botón eliminar 3.- El sistema borra dicho producto de la base de datos
Poscondición:	El caso de uso termina cuando el usuario decide salir del formulario
Flujo Alternativo:	El administrador inicia la acción de eliminar un producto final y el sistema le muestra un mensaje de decisión

# Capítulo 3

## Diseño del Sistema

Diseño es el proceso creativo de transformación del problema en una solución, la descripción de una solución también se denomina diseño [4].

### 3.1 Diagrama de Clases

Los Diagramas de Clase describen los componentes esenciales de la arquitectura de un Sistema [5].

Dado el Diagrama de Casos de Uso se puede modelar el Diagrama de Clases ver figura 3.1 utilizando el lenguaje UML, el cual nos proporciona las herramientas necesarias para la representación gráfica de dicho diagrama.

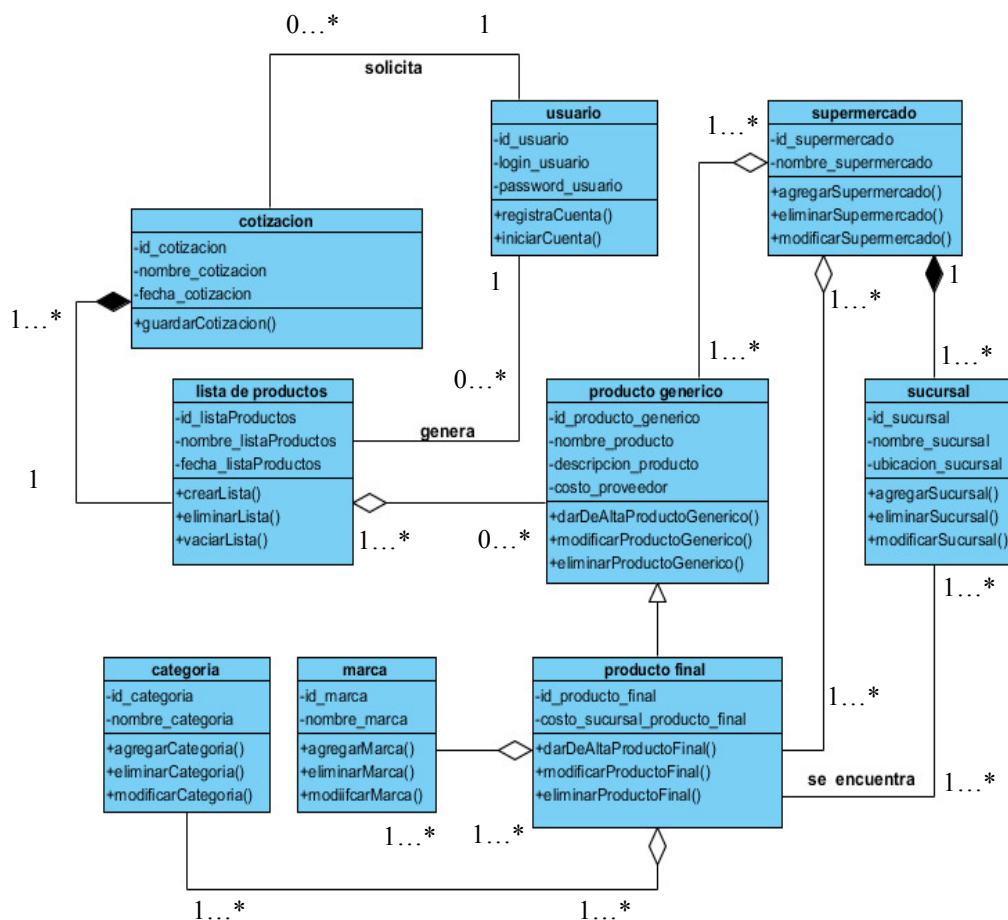


Figura 3.1 Diagrama de Clases para el Sistema SMovSuper

## Descripción de las Clases

**Producto Genérico:** Clase que contiene los datos sobre los productos de costo genérico. Esta clase se especializa en otra clase que es producto final identificada por el *costo\_sucursal\_producto\_final*. Uno o más productos genéricos forman parte de un supermercado. Uno o más productos genéricos forman parte de una lista de productos.

**Categoría:** Clase que contiene las categorías en las que se encuentran divididos los productos. Una o más categorías forman parte de un producto final.

**Marca:** Clase que contiene las diversas marcas que caracterizan un producto. Una o más marcas forman parte de un producto final.

**Producto Final:** Clase que contiene los datos sobre los productos con precio al público. Uno o más productos finales forman parte de un supermercado.

**Sucursal:** Clase que contiene los datos de las sucursales de acuerdo al supermercado correspondiente. Una sucursal es parte de un supermercado. Se relaciona con la clase producto final a través de la asociación *se encuentra*, pudiéndose encontrar varios productos finales.

**Usuario:** Clase que contiene la información de los usuarios identificados únicamente por *login\_usuario* y *password\_usuario*. Se relaciona con la clase Lista de Productos a través de la asociación *genera*, pudiendo generar varias lista de productos. Se relaciona con la clase cotización a través de la asociación *solicita*, pudiendo solicitar varias cotizaciones.

### 3.2 Diagrama Entidad Relación

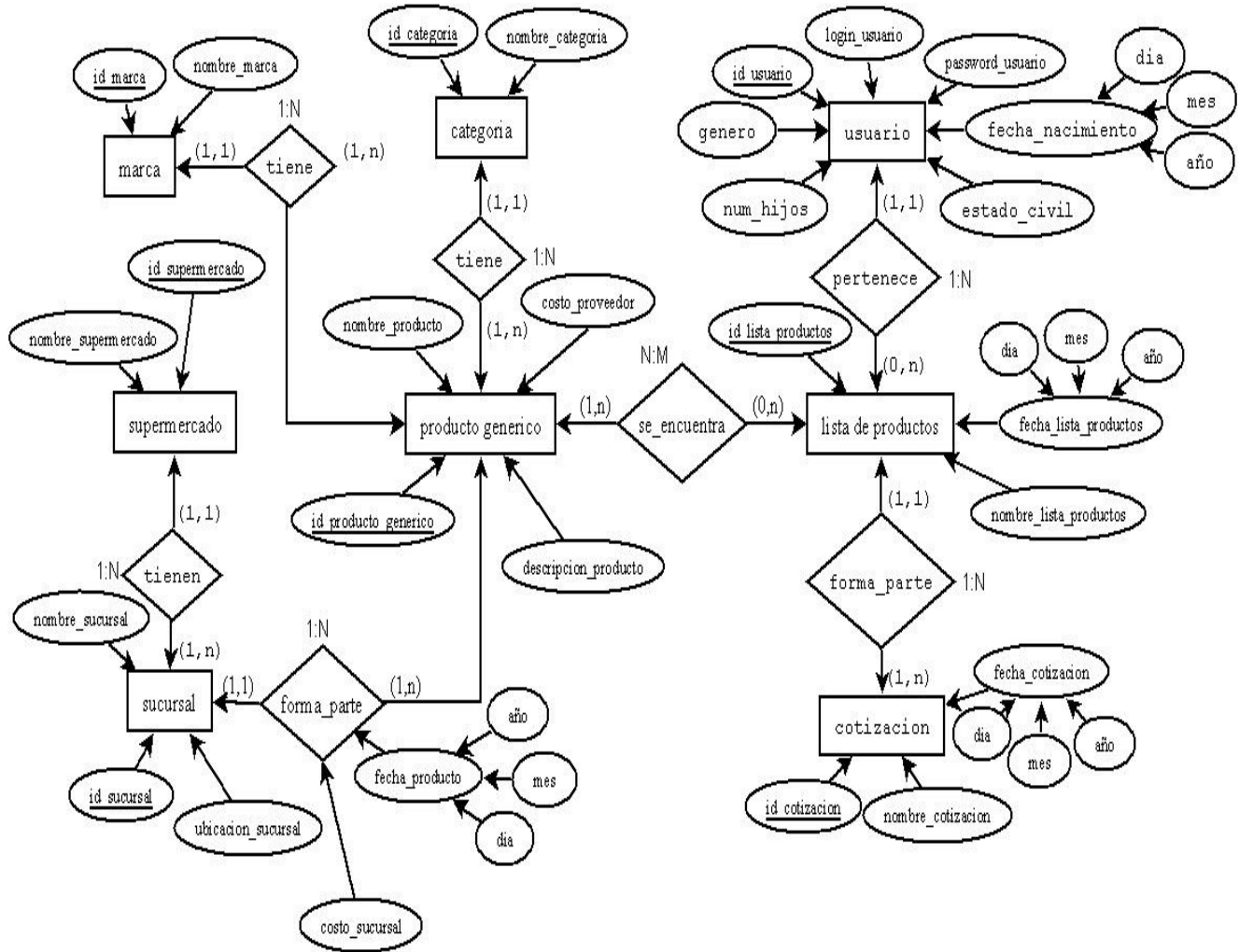


Figura 3.2 Diagrama Entidad Relación para el sistema SuperMov

En la figura 3.2 se tiene el Diagrama Entidad Relación en la que esta basada la aplicación, en el cual se desprenden 8 entidades (que corresponden con 8 tablas de la base de datos). Dichas tablas serian las siguientes:

- Marca
- Categoria

- Usuario
- Supermercado
- Producto generico
- Lista de productos
- Sucursal
- Cotización

Sin embargo, en la base de datos existen otras 7 tablas más, las cuales proceden de las relaciones que tienen atributos. Dichas relaciones son: producto generico-tiene-marca, producto generico-tiene-categoria, usuario-pertenece-lista de productos, supermercado-tiene-sucursal, producto generico-forma\_parte-sucursal, producto generico-se\_encuentra-lista de productos, lista de productos-forma\_parte-cotizacion

### 3.3 Modelo Relacional

El modelo relacional representa la base de datos como una colección de relaciones. En términos informales, cada relación semeja una tabla o hasta cierto punto un archivo simple [6]. Ver figura 3.3:

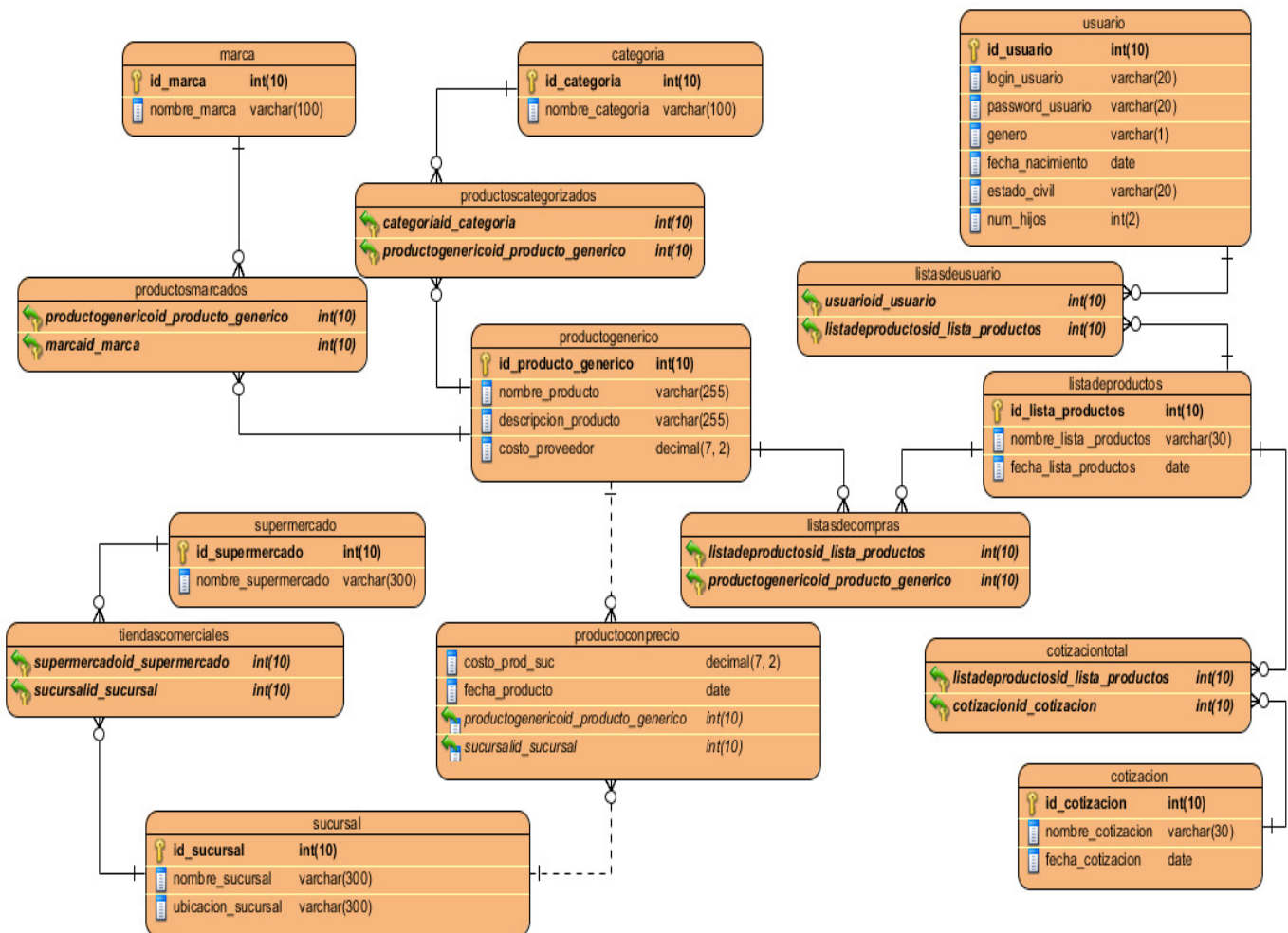


Figura 3.3 Esquema del Modelo Relacional para el sistema SuperMov

### 3.4 Normalización

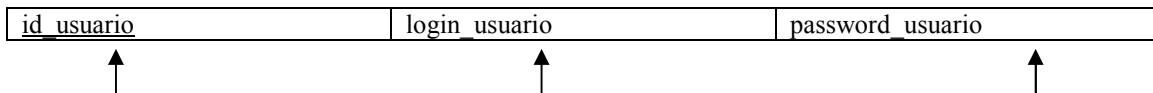
La normalización de los datos puede considerarse como un proceso durante el cual los esquemas de relación insatisfactorios se descomponen repartiendo sus atributos entre esquemas de relación mas pequeños que poseen propiedades deseables [6].

#### Tabla usuario

1FN: La siguiente tabla esta en 1FN ya que los atributos son atómicos

<u>id_usuario</u>	login_usuario	password_usuario
-------------------	---------------	------------------

2FN: La tabla usuario contiene atributos que dependen totalmente de la llave primaria por tanto se encuentra en 2FN



3FN: Se observa que los atributos de la tabla no poseen dependencias transitivas lo cual la tabla se encuentra en 3FN

#### Tabla categoria

1FN: La tabla categoria está en 1FN ya que los atributos son atómicos

<u>id_categoria</u>	nombre_categoria
---------------------	------------------

2FN: Se observa que los atributos de la tabla no poseen dependencias funcionales lo cual la tabla se encuentra en 2FN

3FN: La tabla categoria no posee dependencias transitivas por lo tanto se encuentra en 3FN

### Tabla marca

1FN: La siguiente tabla contiene atributos atómicos por lo que se encuentra en 1FN

<u>id_marca</u>	nombre_marca
-----------------	--------------

2FN: Los atributos no poseen dependencias funcionales por lo tanto la tabla se encuentra en 2FN

3FN: Los atributos de la tabla no poseen dependencias transitivas por lo tanto se encuentra en 3FN

### Tabla supermercado

1FN: La siguiente tabla esta en 1FN ya que los atributos son atómicos

<u>id_supermercado</u>	nombre_supermercado
------------------------	---------------------

2FN: Los atributos de la tabla supermercado no contienen dependencias funcionales por lo que la tabla se encuentra en 2FN

3FN: Se observa que los atributos de la tabla no poseen dependencias transitivas lo cual la tabla se encuentra en 3FN

### Tabla sucursal

1FN: La tabla cumple con la primera forma normal ya que los datos son atómicos.

<u>id_sucursal</u>	nombre_sucursal	ubicacion_sucursal
--------------------	-----------------	--------------------

The diagram shows a horizontal line below the table. From the left end of this line, an arrow points up to the 'id\_sucursal' attribute. From the middle of the line, an arrow points up to the 'nombre\_sucursal' attribute. From the right end of the line, an arrow points up to the 'ubicacion\_sucursal' attribute. This indicates that both 'nombre\_sucursal' and 'ubicacion\_sucursal' functionally determine 'id\_sucursal'.

2FN: Los atributos no clave dependen funcionalmente de la clave primaria por lo tanto la tabla se encuentra en 2FN.

3FN: Los atributos de la tabla no poseen dependencias transitivas por lo tanto se encuentra en 3FN.

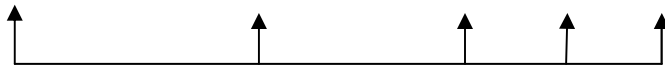
### Tabla cotizacion

<u>id_cotizacion</u>	nombre_cotizacion	fecha_cotizacion
----------------------	-------------------	------------------

1FN: La primera forma normal nos dice que los campos deben ser atómicos, pues fecha cotizacion es un campo que puede descomponerse en día, mes y año. Tal como se muestra en la siguiente tabla:

1FN

<u>id_cotizacion</u>	nombre_cotizacion	dia	mes	año
----------------------	-------------------	-----	-----	-----



2FN: La tabla cotizacion se encuentra en 2FN ya que los atributos no clave dependen funcionalmente de la clave primaria.

3FN: La tabla se encuentra en 3FN ya que los atributos no poseen dependencias transitivas.

### Tabla lista de productos

<u>id_lista_productos</u>	nombre_lista_productos	fecha_creacion_lista
---------------------------	------------------------	----------------------

1FN: La primera forma normal nos dice que los campos deben ser atómicos, pues fecha creación es un campo que puede descomponerse en día, mes y año. Tal como se muestra en la siguiente tabla:

1FN

<u>id_lista_productos</u>	nombre_lista_productos	dia	mes	año
---------------------------	------------------------	-----	-----	-----




2FN: La tabla lista de productos contiene atributos que dependen totalmente de la llave primaria por tanto se encuentra en 2FN.

3FN: Se observa que los atributos de la tabla no poseen dependencias transitivas lo cual la tabla se encuentra en 3FN

### Tabla producto genérico

1FN: La siguiente tabla contiene atributos atómicos por lo que se encuentra en 1FN.

<u>id_producto_generico</u>	nombre_producto	descripcion_producto	costo_proveedor
-----------------------------	-----------------	----------------------	-----------------



2FN: La tabla producto generico contiene atributos que dependen totalmente de la llave primaria por tanto se encuentra en 2FN

3FN: Los atributos de la tabla no poseen dependencias transitivas por lo tanto se encuentra en 3FN.


### Tabla producto con precio

<u>id_producto_generico</u>	<u>id_sucursal</u>	fecha_producto	costo_prod_suc
-----------------------------	--------------------	----------------	----------------

1FN: La primera forma normal nos dice que los campos deben ser atómicos, pues fecha producto es un campo que puede descomponerse en día, mes y año. Tal como se muestra en la siguiente tabla:

1FN

<u>id_producto_generico</u>	<u>id_sucursal</u>	día	mes	año	costo_prod_suc
-----------------------------	--------------------	-----	-----	-----	----------------



2FN: La tabla cotizacion se encuentra en 2FN ya que los atributos no clave dependen funcionalmente de la clave primaria.

3FN: La tabla se encuentra en 3FN ya que los atributos no poseen dependencias transitivas.

### Tabla productos marcados

1FN: Los atributos de la tabla son atómicos por lo tanto se encuentra en 1FN.

<u>id_marca</u>	<u>id_producto_generico</u>
-----------------	-----------------------------

2FN: Se observa que los atributos de la tabla no poseen dependencias funcionales lo cual la tabla se encuentra en 2FN

3FN: Se observa que los atributos de la tabla no poseen dependencias transitivas lo cual la tabla se encuentra en 3FN

### **Tabla productos categorizados**

1FN: La siguiente tabla esta en 1FN ya que los atributos son atómicos

<u>id_categoria</u>	<u>id_producto_generico</u>
---------------------	-----------------------------

2FN: Los atributos no poseen dependencias funcionales por lo tanto la tabla se encuentra en 2FN

3FN: Los atributos de la tabla no poseen dependencias transitivas por lo tanto se encuentra en 3FN

### **Tabla listas de usuario**

1FN: La siguiente tabla contiene atributos atómicos por lo que se encuentra en 1FN

<u>id_usuario</u>	<u>id_lista_productos</u>
-------------------	---------------------------

2FN: Se observa que los atributos de la tabla no poseen dependencias funcionales lo cual la tabla se encuentra en 2FN

3FN: La tabla categoria no posee dependencias transitivas por lo tanto se encuentra en 3FN

### **Tabla lista de compras**

1FN: Se observa que los atributos de la tabla lista de compras son atómicos, por lo cual se encuentra en 1FN.

<u>id_producto_generico</u>	<u>id_lista_productos</u>
-----------------------------	---------------------------

2FN: La tabla lista de compras no contiene atributos con dependencia funcional por lo tanto se encuentra en 2FN.

3FN: No existen dependencias transitivas en la tabla, entonces se encuentra en 3FN.

### **Tabla cotizacion total**

1FN: La tabla se encuentra en 1FN, ya que los atributos son atómicos.

<u>id_cotizacion</u>	<u>id_lista_productos</u>
----------------------	---------------------------

2FN: La tabla cotizacion total esta en 2FN, ya que no posee dependencias funcionales.

3FN: Se observa que la tabla no posee dependencias transitivas por lo tanto la tabla esta en 3FN.

### **Tabla tiendas comerciales**

1FN: La tabla se encuentra en 1FN, ya que los atributos son atómicos.

<u>id_supermercado</u>	<u>id_sucursal</u>
------------------------	--------------------

2FN: La tabla tiendas comerciales esta en 2FN, ya que no posee dependencias funcionales.

3FN: Se observa que la tabla no posee dependencias transitivas por lo tanto la tabla esta en 3FN.

# Capítulo 4

## Implementación y Pruebas

En este capítulo se presenta la descripción paso a paso de la implementación y la realización de pruebas del Sistema.

### 4.1 Diseño con Interfaz de Android

Para el desarrollo de una aplicación de Android se realiza lo siguiente [7]:

Paso 1: Instalar Java 6.0 SDK o el mas actual

El archivo se encuentra en la siguiente liga:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Paso 2: Descargar e Instalar IDE Eclipse (Indigo)

Se encuentra en la siguiente liga: <http://www.eclipse.org/downloads/> , y se descarga Eclipse IDE for Java EE Developers, ya que es el que contiene las herramientas para aplicaciones web.

Paso 3: Descargar, Instalar y Configurar Tomcat

Se encuentra en la siguiente liga: <http://tomcat.apache.org/> , se descarga la versión de Tomcat dependiendo de la versión jdk de java instalada en el paso 1, es decir si la versión de java corresponde a la 6 entonces la versión de Tomcat será la 6.

Una vez instalado Tomcat, se configura la opción de arranque, ya que si se deja de manera automática causara problemas en la ejecución de una aplicación web dentro de Eclipse.

Para deshabilitar el arranque automático del servicio se realiza lo siguiente, ver figura 4.1:

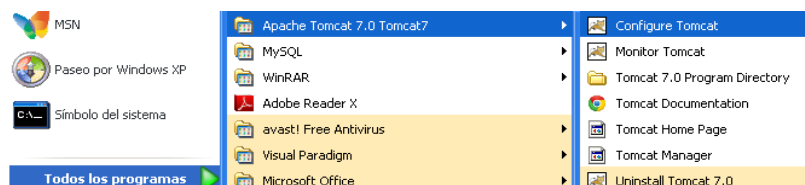


Figura 4.1 Configuración de Tomcat

En la ventana de propiedades ver figura 4.2, se fija el tipo de inicio como manual, se detiene el servicio (Stop) y se marca Aplicar y Aceptar.

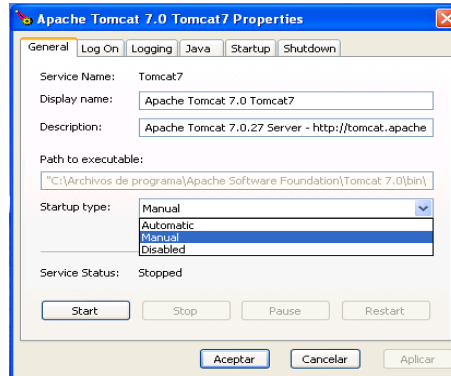


Figura 4.2 Ventana de propiedades de Tomcat

#### Paso 4: Descargar el Android SDK

El SDK de Android se encuentra en la siguiente liga:

<http://developer.android.com/sdk/index.html> , se descarga installer\_r18-windows.exe.

Una vez instalado, se selecciona las API's (Android 2.1(API 7), Android 2.2 (API 8)) que se van a instalar para utilizar en Eclipse. Ver figura 4.3.

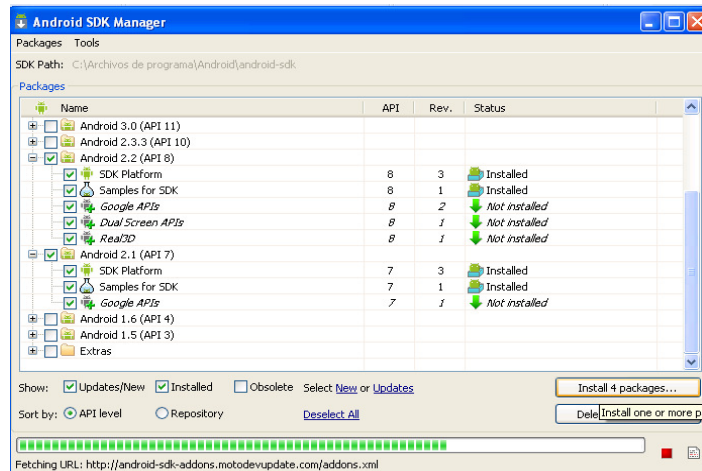


Figura 4.3 Android SDK

#### Paso 5: Integrar Tomcat con Eclipse

Se selecciona “Window” a continuación “Preferences”. Como se muestra en la figura 4.4:

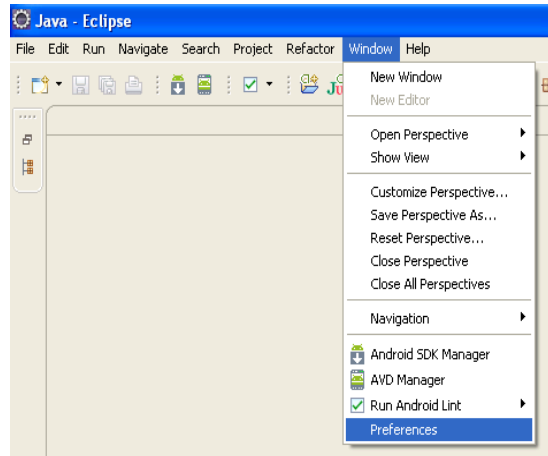


Figura 4.4 Eclipse

Se selecciona “Server” a continuación “Runtime Environments” y se da click en “Add”. Como se muestra en la figura 4.5.

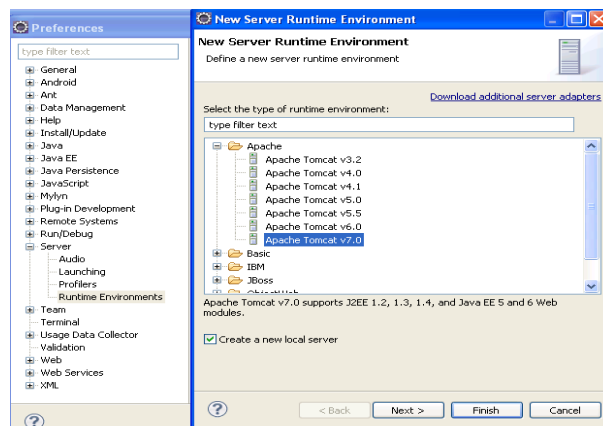


Figura 4.5 Server

Se escoge una de las opciones de Apache Tomcat dependiendo de la versión jdk de java instalada en el paso 1. Se da click en “next” en el botón “Browse” se selecciona la carpeta en donde ha sido instalado Tomcat, ver figura 4.6.

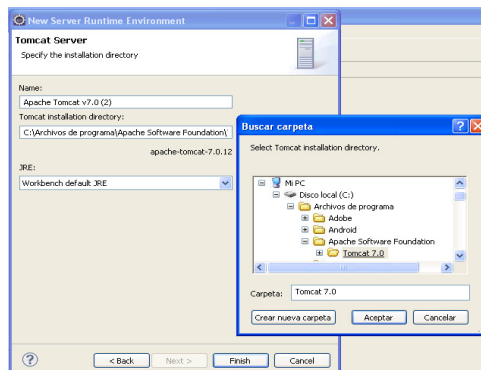


Figura 4.6 Browse Tomcat

En “JRE” se selecciona el jdk de java instalado. Ver figura 4.7 a continuación se selecciona “Finish”, se pulsa “OK”

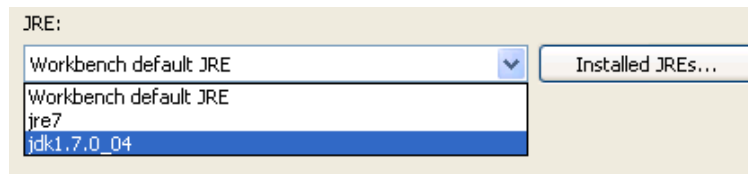


Figura 4.7 JRE Tomcat

Una vez agregado el servidor Tomcat en Eclipse, se puede visualizar en la pestaña “Servers”, a continuación se da doble click sobre el nuevo servidor y se muestra la pagina de configuración, se selecciona “Server Location” a continuación se habilita la opción “Use Tomcat installation (takes control of Tomcat installation)”, se cierra la pagina de configuración se da click en “yes”. Ver figura 4.8.

Si esta configuración no es habilitada ocasiona problemas al correr una aplicación web.

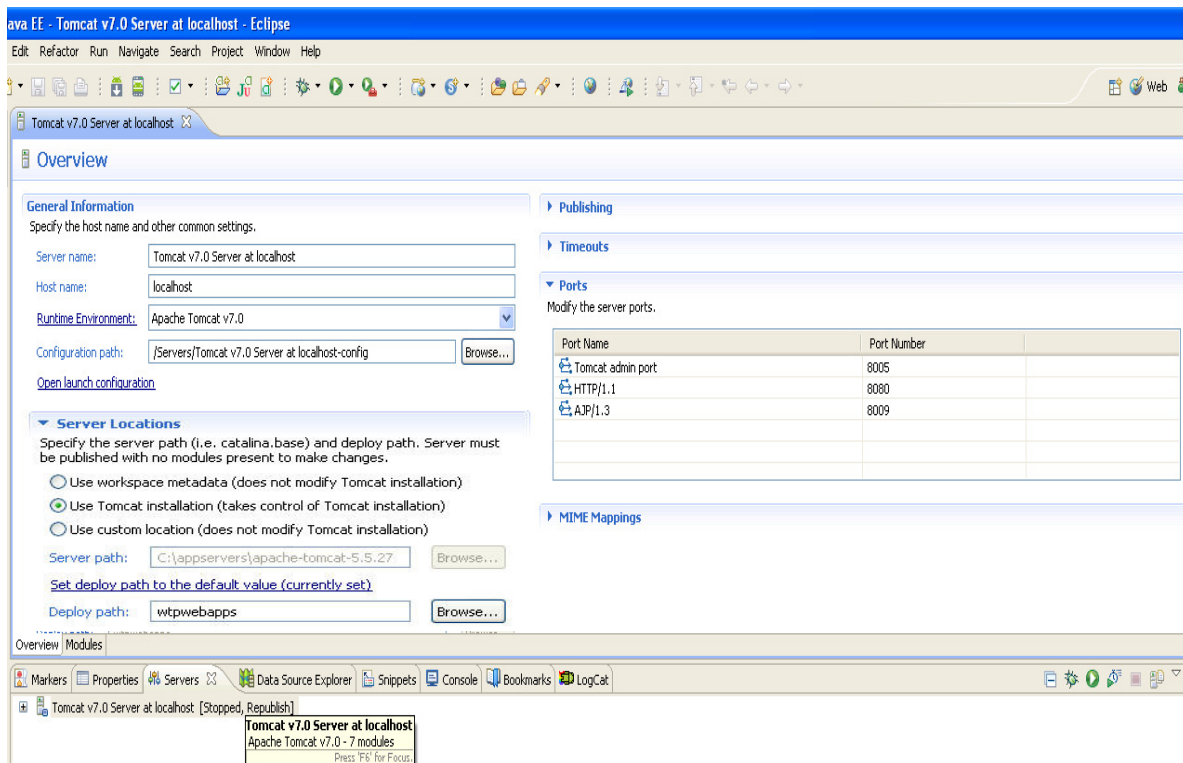
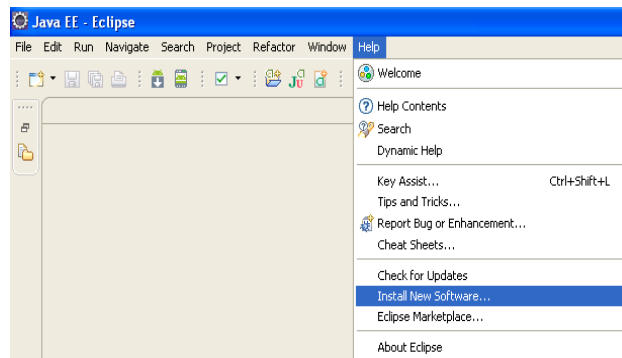


Figura 4.8 Página de Configuración Tomcat

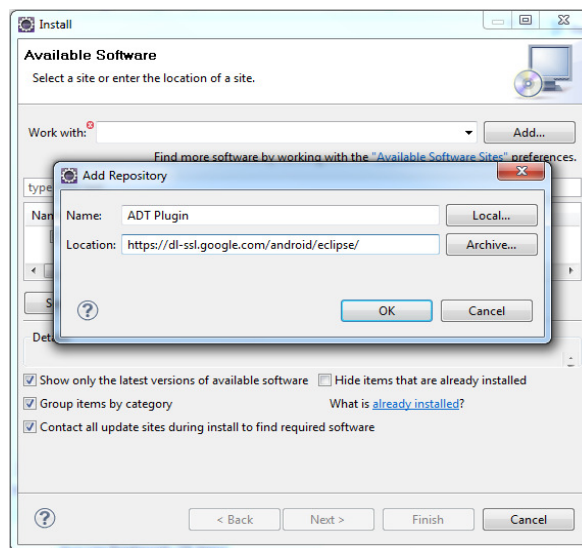
### Paso 6: Instalar Plugin ADT de Android en Eclipse

Se selecciona “Help” > “Install New Software”. Como se muestra en la figura 4.9.



**Figura 4.9** Install New Software

Click “Add” dentro del cuadro de dialogo “Name” se coloca ADT Plugin, en “Location” se coloca la siguiente dirección: <https://dl-ssl.google.com/android/eclipse/>, se da click en “OK”, ver figura 4.10.



**Figura 4.10** Plugin de Android

Se marca la casilla “Developer Tools”, se da click en “next”, como se muestra la figura 4.11. En la siguiente ventana se visualiza la lista de herramientas descargadas, se da click en “next”

y a continuación click en “finish”, cuando la instalación ha sido completada, se reinicia Eclipse.

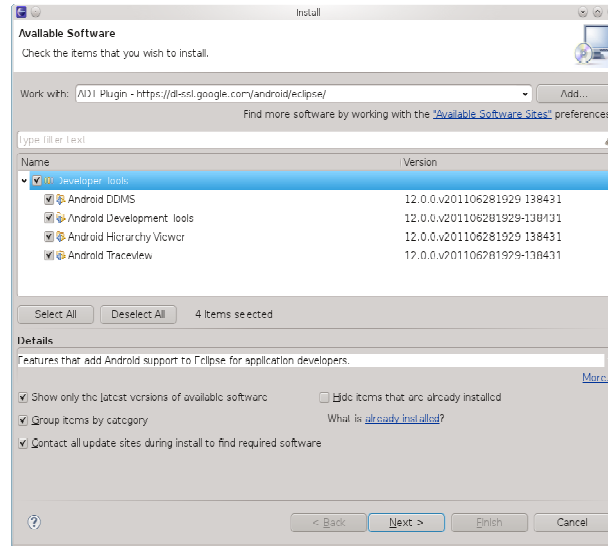


Figura 4.11 Developers Tools de Android

### Paso 7: Configurar el Plugin ADT

Una vez que se ha reiniciado Eclipse, se selecciona “Window” > “Preferences”. Ver figura 4.12.

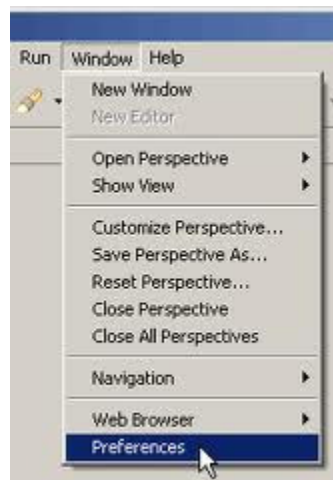


Figura 4.12 Configuración Plugin ADT

Se selecciona “Android” del panel izquierdo de Preferencias, se da click en el botón “Browse” se selecciona la ruta donde ha instalado el SDK de Android. Dicha ruta se visualiza en el SDK Android Manager en la parte SDK Path muestra la ubicación. Una vez agregada la dirección en el “SDK Location” se da click en “Apply” a continuación click en “OK”. Como se muestra en la figura 4.13.

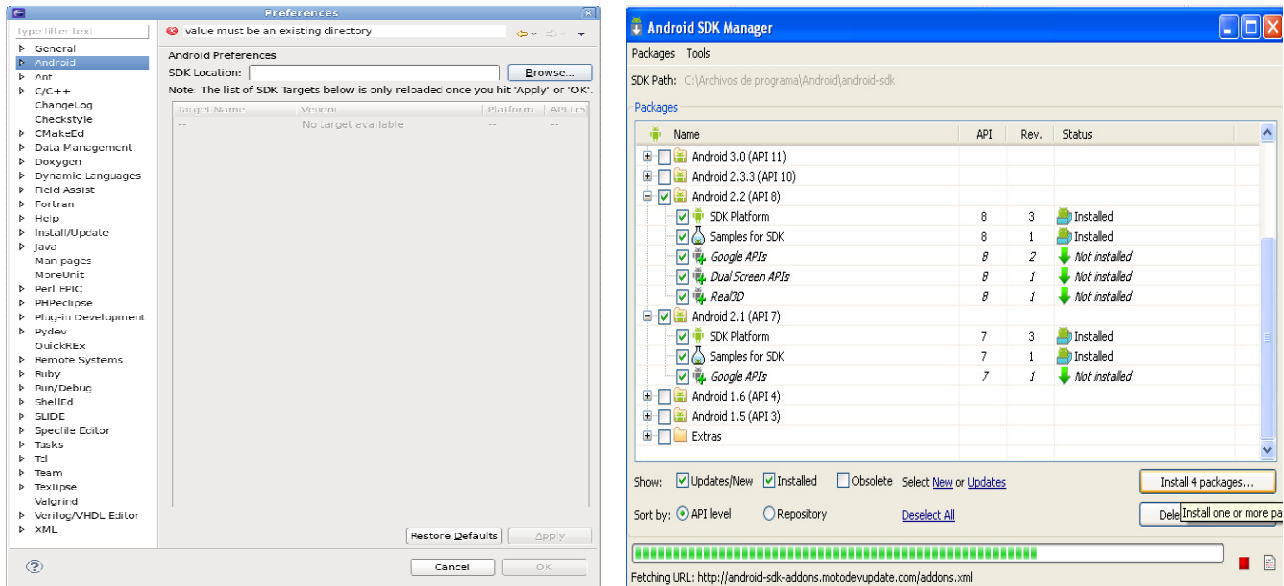


Figura 4.13 Panel de Configuración

### Paso 7: Crear un Dispositivo Virtual de Android (AVD)

Un dispositivo virtual de Android es un emulador que permite visualizar un modelo actual del dispositivo. En la figura 4.14 se muestra como añadir un AVD. Para agregar un nuevo dispositivo virtual de android en eclipse se selecciona “Window” seguido de “AVD Manager”.

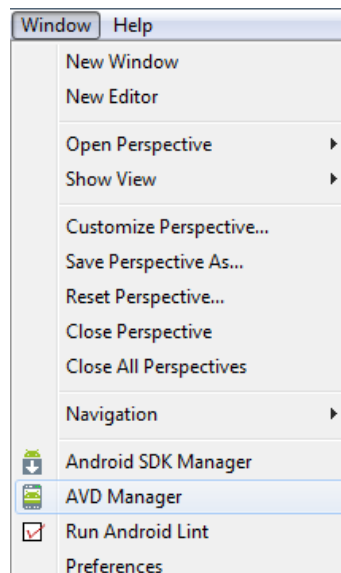


Figura 4.14 AVD Manager

Click en el botón “new” > en “name” se escribe el nombre del dispositivo virtual, por ejemplo genérico, después se selecciona el “target”, el target es la versión de Android con la que se desea trabajar, por ejemplo Android 4.0- API Level 14, en “Size” se escribe el tamaño del dispositivo virtual, la opción “Snapshot” puede estar o no seleccionada, esta característica permite arrancar los AVDS de manera mas rápida. En “Built-in” nos permite seleccionar rápidamente un tamaño de pantalla predefinido, como el WVGA800 defecto, o seleccionar una resolución específica. Una vez que se ha llenado la información se da click en el botón “Create AVD”. Ver figura 4.15.

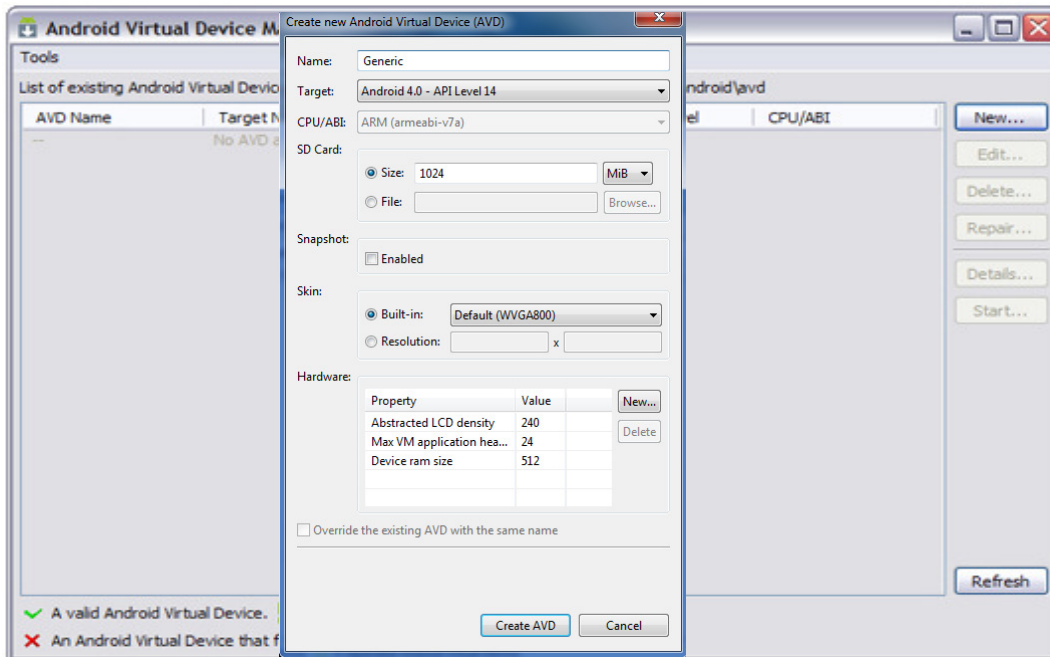


Figura 4.15 Android Virtual Device Manager

## 4.2 Implementación de la Base de Datos

El propósito de la implementación es satisfacer los requerimientos de la manera que es específica el diseño detallado [2].

En la Figura 4.16 se presenta la base de datos del sistema SMovSuper: Sistema Móvil de Toma de Decisiones para la Compra en Supermercados implementada con MySQL.

En ella se distinguen las siguientes tablas:

**Categoría:** Permite almacenar las categorías en las que se clasifican los productos.

**Cotización:** Permite almacenar los nombres de las cotizaciones generadas por el usuario incluyendo fecha en la que fue generada la cotización.

**Lista de Productos:** Permite almacenar los datos de las listas creadas por el usuario.

**Marca:** Permite almacenar los diversas marcar existentes.

**Producto Con Precio:** Permite almacenar los datos de los productos comerciales para las diferentes sucursales.

**Producto Generico:** Permite almacenar los datos de los productos genéricos existentes en los supermercados.

**Sucursal:** Permite almacenar los datos de las diferentes sucursales.

**Supermercado:** Permite almacenar los datos de los diferentes supermercados.

**Usuario:** Permite almacenar los datos de registro de cada usuario.

```
Tables_in_smovsuper
+-----+
categoria
cotizacion
cotizaciontotal
listadeproductos
listasdecompras
listasdeusuario
marca
productoconprecio
productogenerico
productoscategorizados
productosmarcados
sucursal
supermercado
tiendascomerciales
usuario
+-----+
```

Figura. 4.16 Base de datos implementada con MySQL.

### 4.3 Implementación del Sistema

Como ya se mencionó anteriormente, para la creación de la base de datos se utilizo la herramienta MySQL en donde se crearon las tablas y campos. Dado que MySQL es una base de datos open source [8]. Para el manejo de la base de datos dentro de un servidor se utilizo Servlets. Un servlet es un programa que se ejecute en un servidor Web, los clientes pueden invocarlo utilizando el protocolo HTTP, y es cargado y ejecutado por un servidor Web [8].

En esta sección se analizan cada uno de los componentes del SMovSuper: Sistema Móvil de Toma de Decisiones para la Compra en Supermercados.

#### 4.3.1 Interfaz del Cliente

Esta interfaz se realizo bajo la plataforma de Eclipse utilizando, servlet, SDK de Android y Servidor Apache Tomcat. A continuación se presenta una lista con los archivos de la aplicación que corresponden la interfaz del cliente.

- ✓ BotonesListasProductos.java
- ✓ Categorias.java
- ✓ CrearListaProductos.java
- ✓ Cuentaingresar.java
- ✓ Mislistas.java
- ✓ ProductosGenerico.java
- ✓ Registrarse.java
- ✓ SmovSupermerMovilActivity.java
- ✓ ConexionBDConsultas.java
- ✓ ServletMetodosSmovSuper.java

### 4.3.1.1 Pagina Principal

Esta página solicita al usuario la selección de las opciones de Ingresar o Registrarse, para posteriormente ser enviada a otro activity dependiendo de la opción seleccionada por el usuario, Ver figura 4.17.



Figura. 4.17 Pagina Principal

### 4.3.1.2 Registrarse

Esta página permite al usuario realizar el registro de sus datos llenando los siguientes campos usuario, password, fecha de nacimiento, género, estado civil y número de hijos. Con el menú de opciones permite regresar a la página principal y limpiar los campos. Ver figura 4.18.

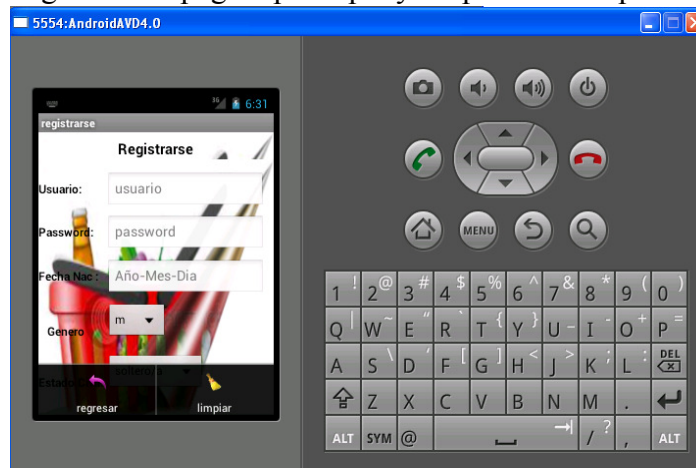


Figura. 4.18 Pagina de Registro

### 4.3.1.3 Ingresar

Esta página de la pagina 4.19, permite al usuario ingresar a su cuenta llenando los siguientes campos usuario y password. Con el menú de opciones permite al usuario regresar a la página principal y limpiar los campos.

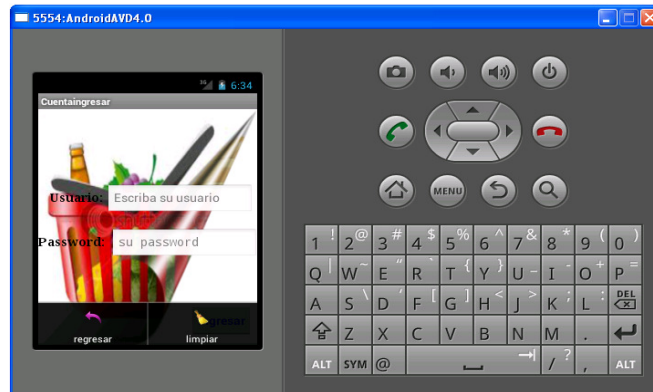


Figura. 4.19 Pagina de Ingreso

### 4.3.1.4 Menú de Opciones Principal

Esta página muestra al usuario las opciones de crear lista de productos, ver mis listas y visualizar productos. Con el menú de opciones permite al usuario ir a la página principal. Ver figura 4.20.



Figura. 4.20 Menú de Opciones

### 4.3.1.5 Formulario crear lista

En la figura 4.21 se muestra el formulario crear lista. Esta página permite al usuario crear una lista llenando los campos nombre de lista y fecha.

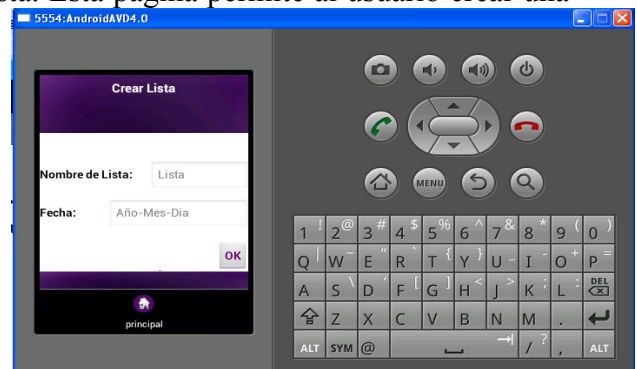


Figura. 4.21 Formulario crear lista

### 4.3.1.6 Mis Listas

Esta página permite al usuario ver las listas existentes. En la figura 4.22 se muestra dicha página.

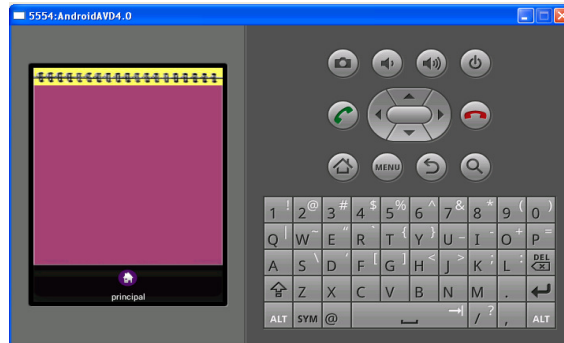


Figura. 4.22 Mis Listas

### 4.3.1.7 Menú de Opciones Listas

Esta página muestra al usuario las opciones de cotizar, ver productos de lista, eliminar lista, vaciar lista. Ver figura 4.23.

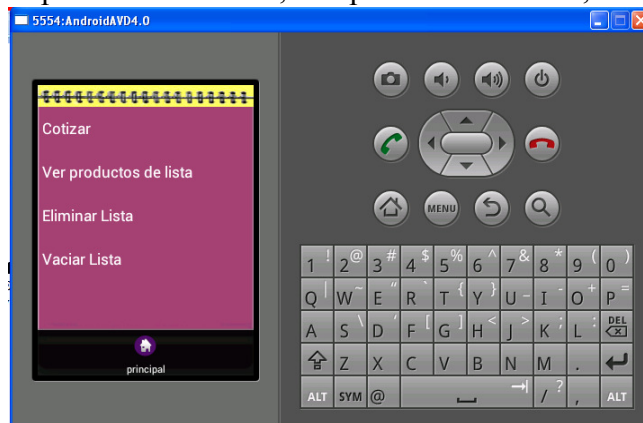


Figura. 4.23 Menú de opciones

### 4.3.1.8 Formulario de Cotización

Esta página muestra la cotización total correspondiente a la lista solicitada por el usuario. Como se muestra en la figura 4.24.

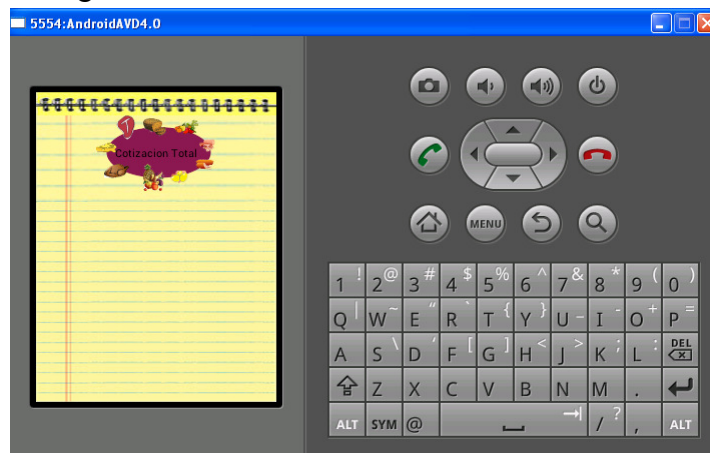


Figura. 4.24 formulario de Cotización

### 4.3.1.9 Menú de Opciones Productos

Esta página ver figura 4.25, muestra al usuario las opciones de agregar y detalles al seleccionar un producto.

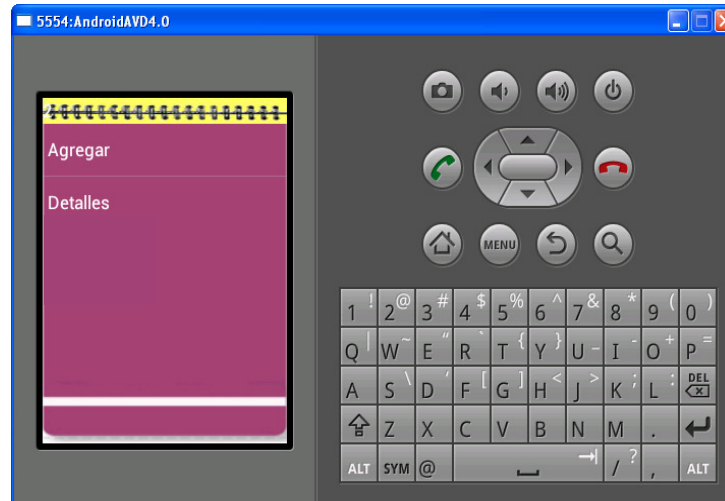


Figura. 4.25 Menú de Opciones Producto

### 4.3.2 Interfaz del Administrador

Esta interfaz se realizó bajo la plataforma de Eclipse utilizando Servlets, HTML, JS, CSS, servidor Apache Tomcat. A continuación se presenta una lista con los archivos de la aplicación que corresponden a la interfaz del cliente.

- ✓ actualizar.html
- ✓ altas.html
- ✓ bajas.html
- ✓ altas.css
- ✓ categoria.html
- ✓ consultar.html
- ✓ estilos\_web.css
- ✓ form\_modificar.css
- ✓ formulario\_alta.css
- ✓ index.html
- ✓ marcas.html
- ✓ menus.css
- ✓ modificar\_marca.html
- ✓ principal.html
- ✓ supermercado.html
- ✓ visualizar.css
- ✓ ConexionBDConsultas.java
- ✓ ServletAdministrador.java

### 4.3.2.1 Pagina Principal

Esta pagina solicita al administrador su usuario y password, como se muestra en la figura 4.26, al ser enviada esta información el sistema revisa que los datos correspondan al administrador de lo contrario se negara el acceso a este.

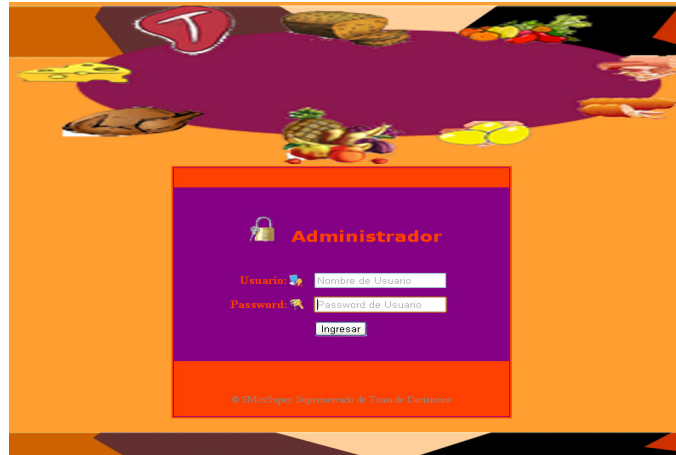


Figura. 4.26 Pagina de Registro

### 4.3.2.2 Menú Principal

El menú se encuentra en la parte central de la página en el cual cada una de las opciones direcciona a la tarea seleccionada. Este menú permite seleccionar la tarea que el administrador desee realizar como altas, bajas, modificaciones, consultas.

Este menú contiene el botón salir en el cual permite al administrador salirse del sistema cuando lo desee. Ver figura 4.27.

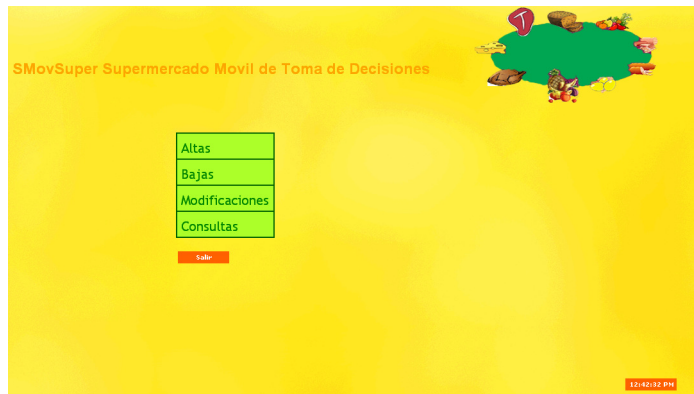


Figura. 4.27 Interfaz de Menú Principal

### 4.3.2.3 Menú Principal Altas

En la figura 4.28, se muestra el menú principal Altas. Este menú permite seleccionar la tarea que el administrador desee realizar como alta marca, alta categoria, alta supermercado, alta sucursal, alta producto generico, alta producto comercial, y permite al administrador salirse del sistema o regresar a la página anterior.



Figura. 4.28 Menú Principal Altas

#### 4.3.2.4 Formulario de Alta Marcas

Este formulario permite dar de alta a la nueva marca y permite al administrador regresar a la página anterior. Ver figura 4.29.

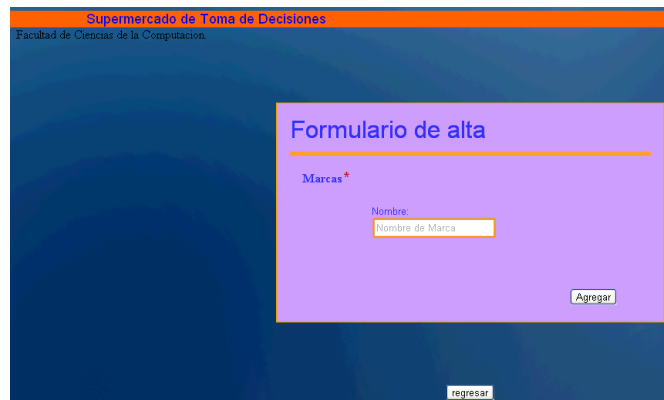


Figura. 4.29 Formulario de Alta Marcas

#### 4.3.2.5 Formulario de Alta Categorías

En la figura 4.30 se muestra el formulario de alta categorías. Este formulario permite dar de alta a la nueva categoría y permite al administrador regresar a la página anterior.



Figura. 4.30 Formulario de Alta Categorías

### 4.3.2.6 Formulario de Alta Supermercado

En la figura 4.31 se muestra el formulario de alta supermercado. Este formulario permite al administrador dar de alta al nuevo supermercado como también le permite regresar a la página anterior.

Figura. 4.31 Formulario de Alta Supermercado

### 4.3.2.7 Formulario de Alta Sucursal

Este formulario permite dar de alta a la nueva sucursal dependiendo del supermercado al que pertenece y permite al administrador regresar a la página anterior. Ver figura 4.32.

Figura. 4.32 Formulario de Alta Sucursal

### 4.3.2.8 Formulario de Alta Producto Generico

Este formulario permite dar de alta al nuevo producto generico y permite al administrador regresar a la página anterior. Ver figura 4.33.

The screenshot shows a web interface for adding a generic product. At the top, there is a header with 'Supermercado de Toma de Decisiones' and 'Facultad de Ciencias de la Computacion'. The main content area is titled 'Formulario de alta' and contains a section for 'Productos Genericos \*'. The form includes the following fields: 'Nombre Producto:' (text input), 'Descripcion:' (text input), 'Costo Proveedor \$' (text input), 'Seleccione una Categoria:' (dropdown menu with 'Abarrotes' selected), and 'Seleccione una Marca:' (dropdown menu with 'Acroz' selected). An 'Agregar' button is located at the bottom right of the form area.

Figura. 4.33 Formulario de Alta Producto Generico

### 4.3.2.9 Formulario de Alta Producto Comercial

Este formulario permite dar de alta al nuevo producto comercial y permite al administrador regresar a la página anterior. Ver figura 4.34.

The screenshot shows a web interface for adding a commercial product. At the top, there is a header with 'Supermercado de Toma de Decisiones' and 'Facultad de Ciencias de la Computacion'. The main content area is titled 'Formulario de alta' and contains a section for 'Productos Comerciales \*'. The form includes the following fields: 'Fecha:' (text input), 'Costo Comercial \$' (text input), 'Costo Proveedor' (text input), 'Seleccione un Producto:' (dropdown menu with 'CerealMul' selected), and 'Selecciona una Sucursal:' (dropdown menu with 'Walmart Reforma' selected). An 'Agregar' button is located at the bottom right of the form area. A 'regresar' button is located at the bottom center of the page.

Figura. 4.34 Formulario de Alta Producto Comercial

### 4.3.2.10 Menú Principal Bajas

En la figura 4.35 se muestra el menú principal bajas. Este menú permite seleccionar la tarea que el administrador desee realizar como eliminar marca, eliminar categoría, eliminar supermercado, eliminar sucursal, eliminar producto generico y eliminar producto comercial, así como también le permite salirse del sistema o regresar a la página anterior.



Figura. 4.35 Menú Principal Bajas

### 4.3.2.11 Formulario de Eliminar

En la figura 4.36 se muestra el formulario de eliminar. Este formulario permite eliminar las marcas seleccionada por el administrador. Dicho formulario permite regresar a la página anterior. El procedimiento de eliminar categoría, eliminar supermercado, eliminar sucursal, eliminar producto generico y eliminar producto comercial es similar al formulario eliminar marcas.

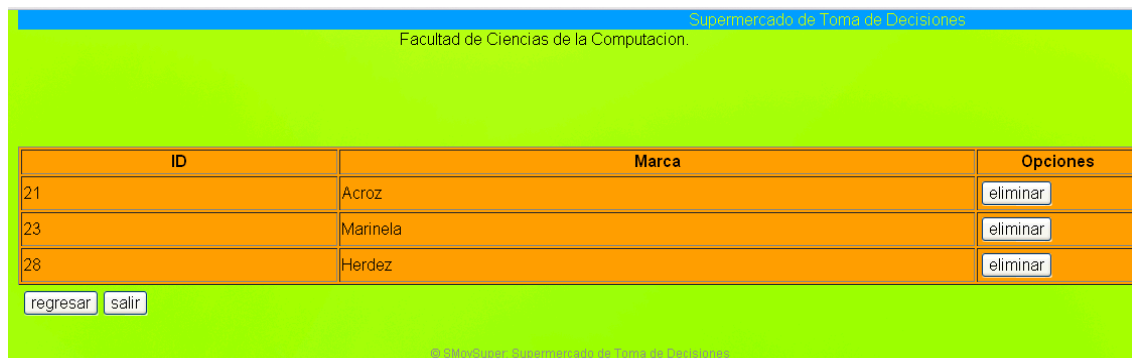


Figura. 4.36 Formulario de eliminar

### 4.3.2.12 Menú Principal Modificaciones

De la figura 4.37 se muestra el formulario principal modificaciones el cual permite al administrador seleccionar la tarea que desee realizar como editar marca, editar categoría, editar supermercado, editar sucursal, editar producto generico y editar producto comercial, así como también permite al administrador salirse del sistema o regresar a la página anterior.



Figura. 4.37 Menú Principal Editar

#### 4.3.2.13 Formulario de Editar

Este formulario permite modificar la marca seleccionada por el administrador. Dicho formulario permite regresar a la página anterior. El procedimiento de editar categoría, editar supermercado, editar sucursal, editar producto generico y editar producto comercial es similar al formulario editar marca. Ver figura 4.38.

Figura. 4.38 Formulario de Editar Marca

#### 4.3.2.14 Menú Principal Consultar

Este menú permite seleccionar la tarea que el administrador desee realizar como consultar marca, consultar categoría, consultar supermercado, consultar sucursal, consultar producto generico y consultar producto comercial, también permite al administrador salirse del sistema o regresar a la página anterior cuando lo desee. En la figura 4.39 se muestra el menú principal consultar.



Figura. 4.39 Menú Principal Consultar

#### 4.3.2.15 Formulario de Consultar

Este formulario permite consultar las marcas existentes en la Base de Datos. Dicho formulario permite regresar a la página anterior. El procedimiento de consultar categoría, consultar supermercado, consultar sucursal, consultar producto genérico y consultar producto comercial es similar a consultar marca. Ver figura 4.40.

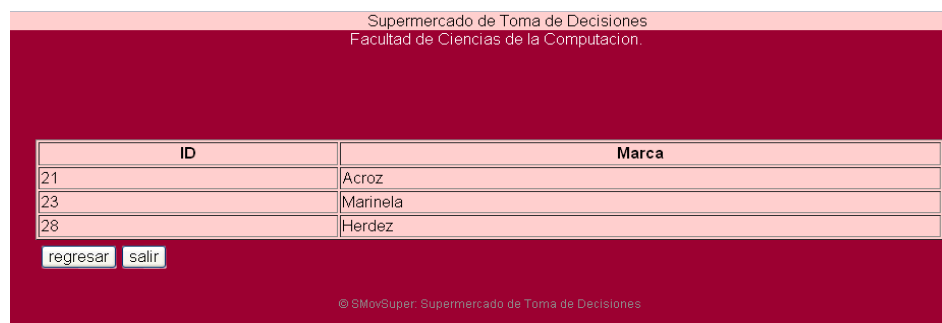


Figura. 4.40 Formulario de consultar

## 4.4 Pruebas del Sistema

Los requisitos del software se comprueban utilizando técnicas de diseño de casos de prueba de caja negra. Las pruebas funcionales o de caja negra son un enfoque para llevar a cabo pruebas donde estas se derivan de la especificación del programa o componente [10].

A continuación se presentan las pruebas realizadas al Sistema a través de un emulador y una computadora portátil mostrando los resultados obtenidos.

### 4.4.1 Pruebas Servidor

Se mostrará un recorrido por el sistema accedendo a cada ventana y menú, así como también la información resultante de las diferentes consultas solicitadas por los usuarios, además de mensajes de error y éxito proporcionados por el sistema.

#### 4.4.1.1 Validación

Se muestra la página principal, en ella se pide nombre de usuario y password los campos se dejan en blanco por lo que el sistema muestra un mensaje de error hasta que se introduzca texto en cada uno de los campos. Tal y como se muestra en la figura 4.41.

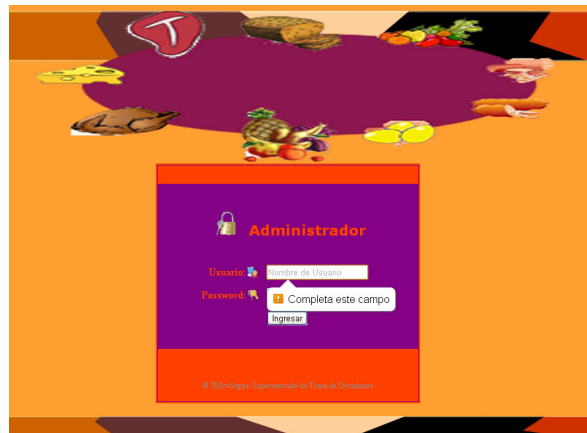


Figura 4.41 Mensaje de error

El administrador introduce de manera errónea su nombre de usuario entonces se muestra un mensaje de error, realiza la prueba introduciendo de manera errónea su password nuevamente se muestra el mensaje de error, tal y como muestra la figura 4.42 a continuación:



Figura 4.42 Mensaje de error

En caso contrario, el administrador introduce de manera correcta, los datos por lo que el sistema valida de manera correcta los datos introducidos y muestra al Administrador el menú principal con las opciones de altas, bajas, modificaciones, consultas y el botón de salir permitiendo la salida del sistema e ingresar nuevamente como se muestra en la figura 4.43.

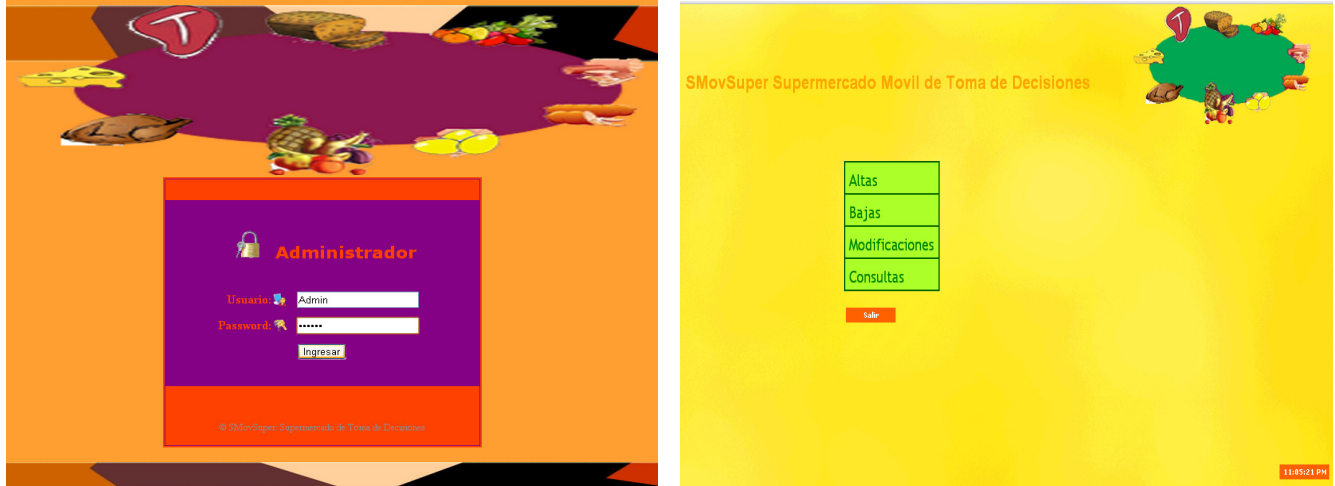


Figura 4.43 Menú principal

#### 4.4.1.2 Menú Principal Altas

El administrador selecciona la opción altas del menú principal, por lo que el sistema despliega el menú principal de altas, tal y como se muestra en la figura 4.44.

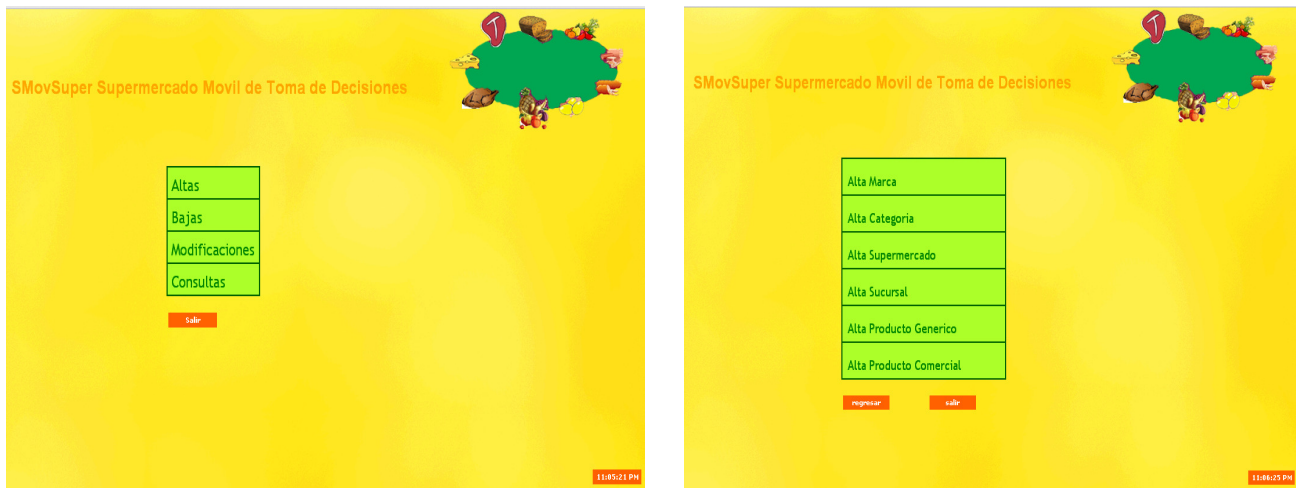


Figura 4.44 Menú principal Altas

#### 4.4.1.3 Formulario Altas

Dentro del menú principal altas, el administrador selecciona la opción alta marca, presiona el botón agregar entonces el sistema muestra un mensaje de error ya que el campo se encuentra en blanco. Ver figura 4.45.

Los formularios de las opciones de alta categoria, alta supermercado, alta sucursal, alta producto generico y alta producto comercial es similar a la opción de alta marca.

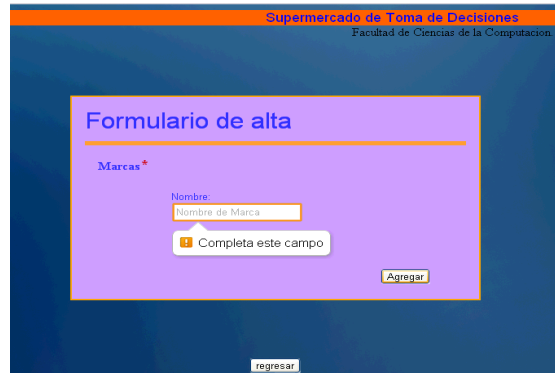


Figura 4.45 Mensaje de error

En caso contrario, el administrador introduce datos en el campo nombre marca, seguidamente el sistema envía un mensaje de error ya que los datos introducidos existen en la base de datos, como en la figura 4.46. Seguidamente el administrador presiona el botón regresar, para introducir nuevos datos.

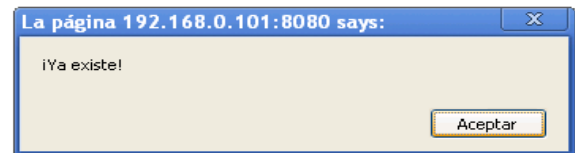
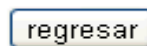
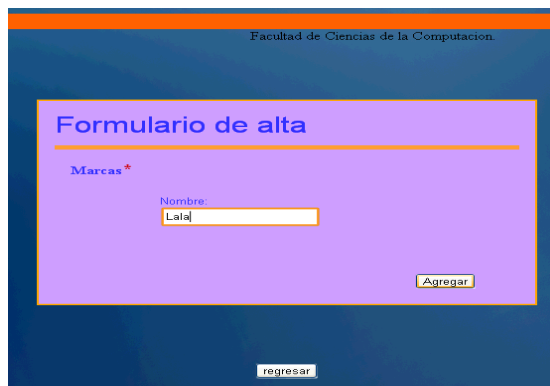


Figura 4.46 Formulario altas y mensaje de error

El administrador nuevamente introduce datos en el campo nombre de marca, da click en el botón agregar, seguidamente el sistema muestra un mensaje de éxito. Como se muestra en la figura 4.47.

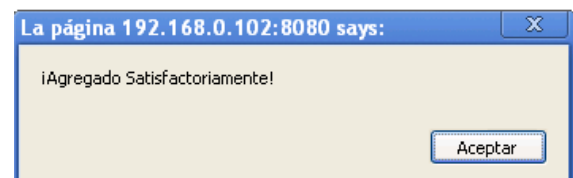
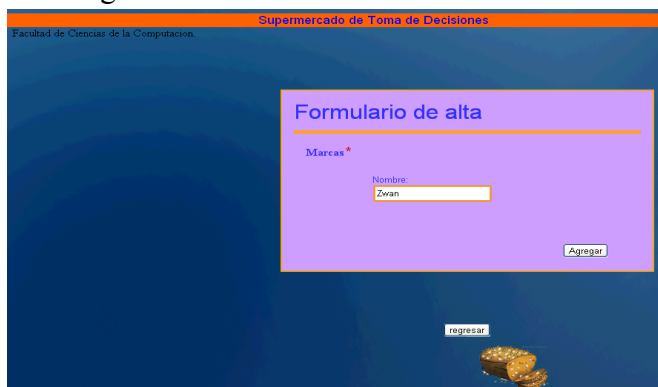


Figura 4.47 Formulario de altas y mensaje de éxito

#### 4.4.1.4 Formulario Bajas

El administrador selecciona la opción eliminar marca dentro del menú principal bajas, el sistema muestra el formulario correspondiente. Ver figura 4.48. Los formularios de eliminar categoría, eliminar supermercado, eliminar sucursal, eliminar producto generico y eliminar producto comercial es similar al formulario eliminar marca.

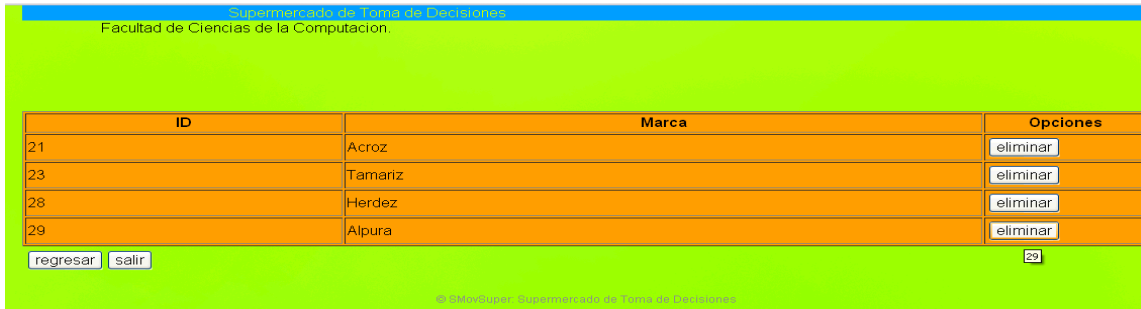


Figura 4.48 Formulario Bajas

El administrador da click en la opción eliminar y a continuación se muestra un mensaje de éxito como en la figura 4.49.

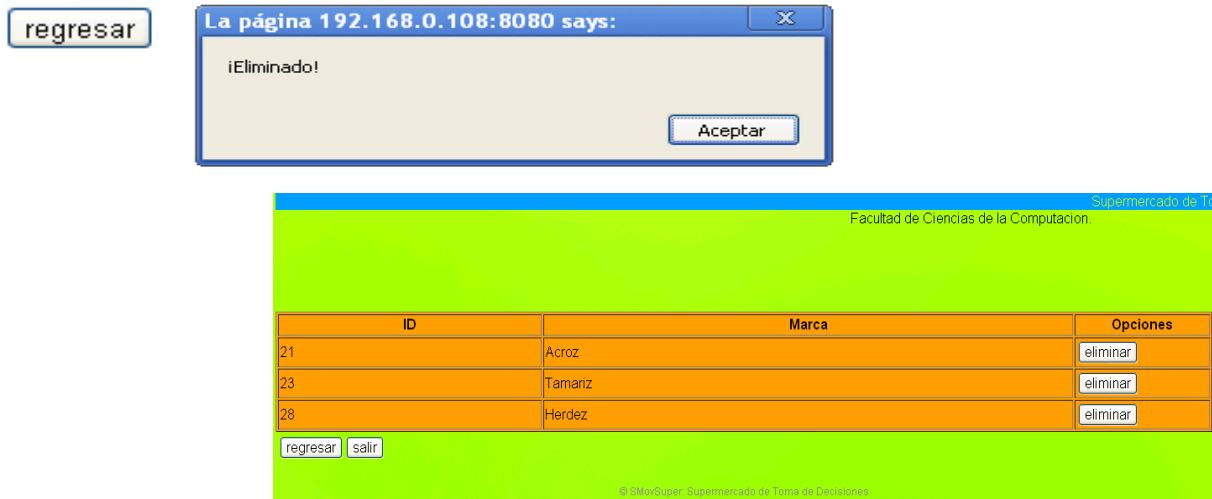
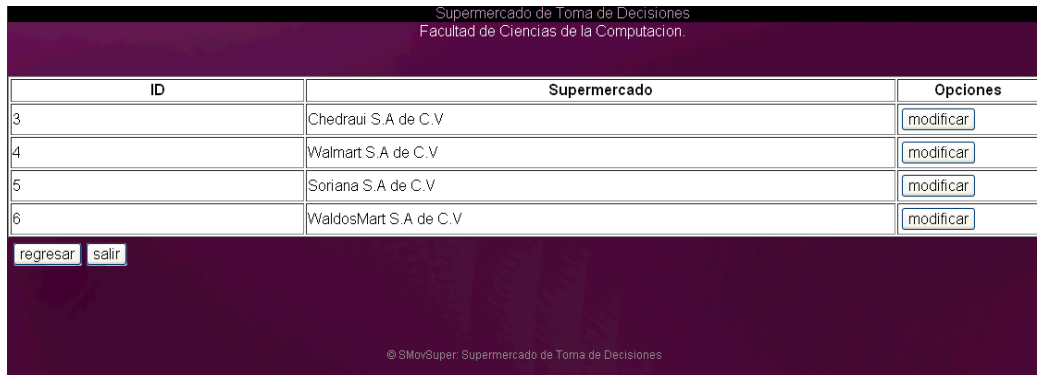


Figura 4.49 Mensaje de éxito

#### 4.4.1.5 Formulario Editar

El administrador selecciona la opción editar marca dentro del menú principal modificaciones, seguidamente el sistema muestra el formulario correspondiente como en la figura 4.50. Las opciones de editar marca, editar categoría, editar supermercado, editar sucursal, editar producto generico y editar producto comercial es similar a la opción editar marca.



ID	Supermercado	Opciones
3	Chedraui S.A de C.V	<input type="button" value="modificar"/>
4	Walmart S.A de C.V	<input type="button" value="modificar"/>
5	Soriana S.A de C.V	<input type="button" value="modificar"/>
6	WaldosMart S.A de C.V	<input type="button" value="modificar"/>

© SMovSuper: Supermercado de Toma de Decisiones

**Figura 4.50** Formulario Editar

El administrador da click en el botón modificar, el sistema seguidamente muestra el formulario actualizar con los campos correspondientes, el administrador deja los campos en blanco, da click en “Guardar Cambios” y aparece un mensaje de error, como se muestra en la figura 4.51.



Formulario de Actualizar

ID:

Supermercado:

© SMovSuper: Supermercado de Toma de Decisiones

**Figura 4.51** Formulario Actualizar

El administrador introduce datos en los campos seguidamente el sistema envía un mensaje de error ya que los datos introducidos existen en la base de datos. Ver figura 4.52.



Formulario de Actualizar

ID:

Supermercado:

© SMovSuper: Supermercado de Toma de Decisiones

La página 192.168.0.101:8080 says:  
¡Ya existe!

**Figura 4.52** Mensaje de Error

En caso contrario el administrador introduce nuevos datos, el sistema valida los datos introducidos y como resultado envía un mensaje de éxito, seguidamente el administrador da click en el botón regresar y el sistema muestra el formulario de editar como el de la figura 4.53.

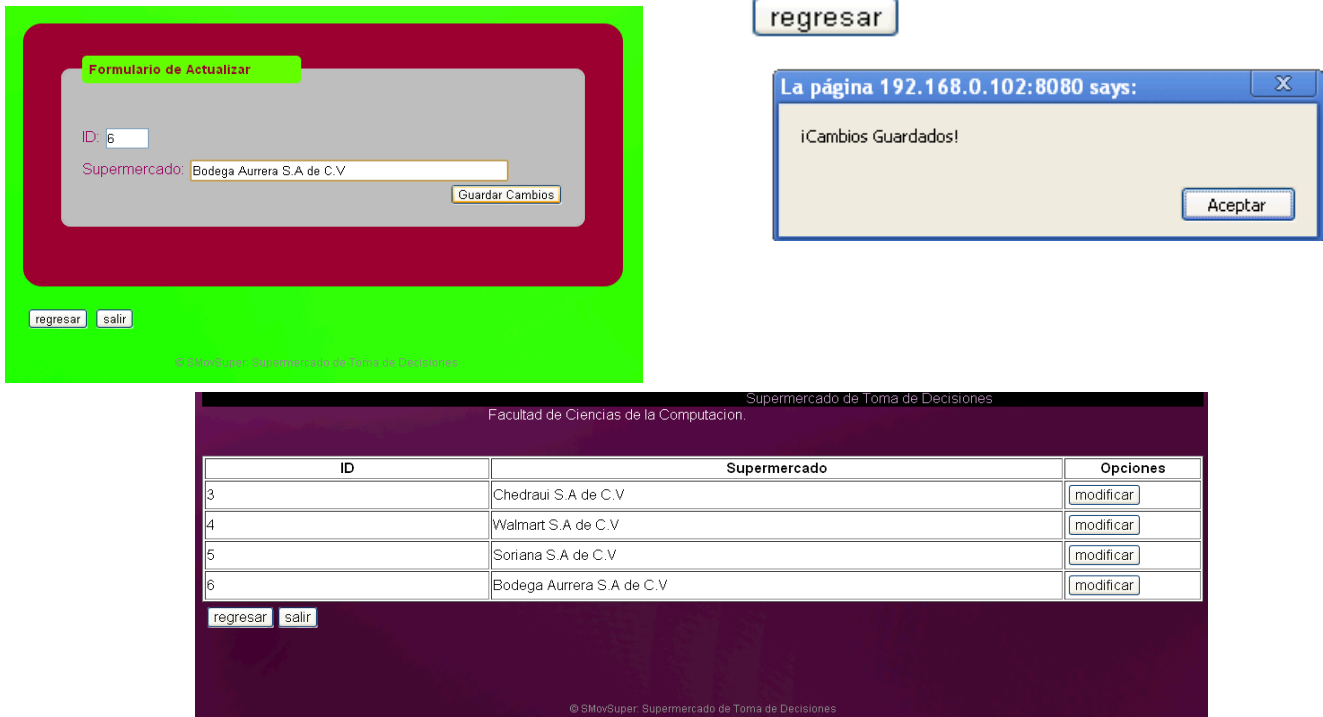


Figura 4.53 Mensaje de éxito y formulario editar

#### 4.4.1.6 Formulario consultar

El administrador selecciona la opción consultar producto generico dentro del menú principal consultar, el sistema muestra el formulario correspondiente como en la figura 4.54. Los formularios de las opciones consultar marca, consultar categoria, consultar supermercado, consultar sucursal y consultar producto comercial es similar al formulario de consultar marca.

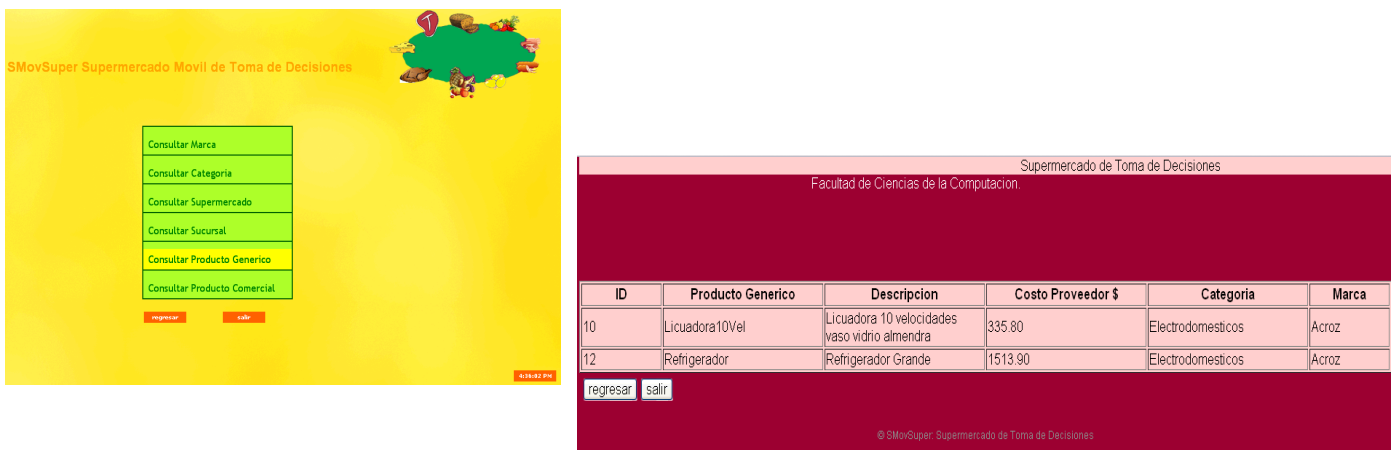


Figura 4.54 Formulario consultar

## 4.4.2 Pruebas Cliente

Se mostrara un recorrido por las diferentes ventanas y menús del sistema. Por lo que debe de arrojar la información respectivamente consultada o si no errores.

### 4.4.2.1 Menú Principal

Dentro del menú principal el usuario visualiza las opciones de “ingresar” y “registrarse”, selecciona la opción ingresar el sistema muestra el formulario con sus respectivos campos, selecciona la opción registrarse el sistema muestra el formulario con sus campos correspondientes. En la figura 4.55 se muestran las opciones mencionadas anteriormente.

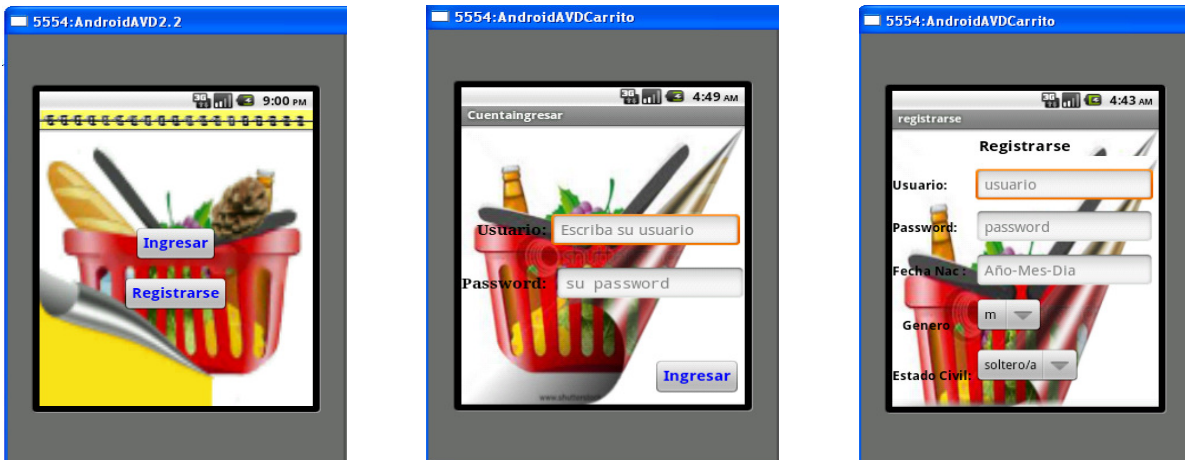


Figura 4.55 Opciones Menú Principal

### 4.4.2.2 Opción Registrarse

El usuario selecciona la opción registrarse seguidamente el sistema muestra el formulario de registro con los campos correspondientes a llenar, el usuario ha dejado en blanco uno de los campos seguidamente el sistema muestra un mensaje de error. Ver figura 4.56.

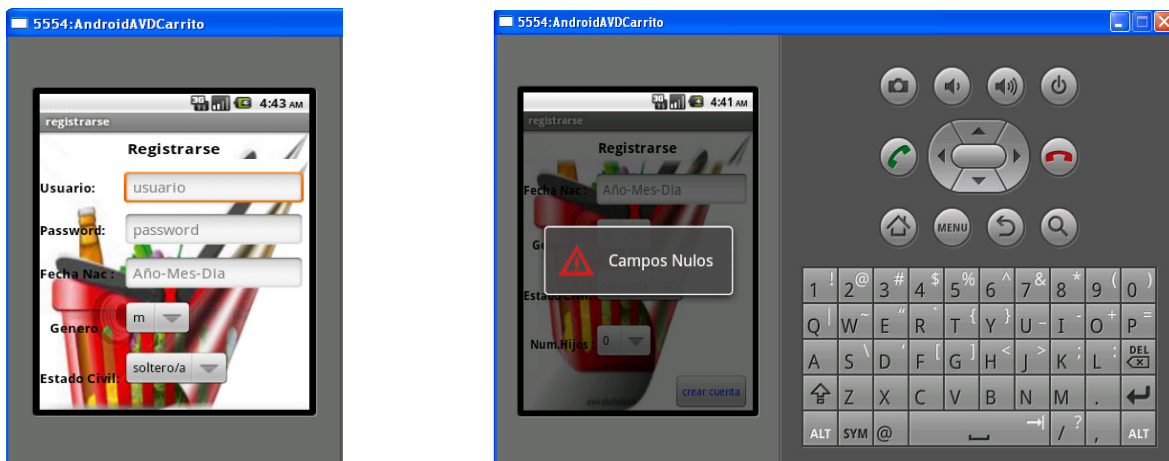


Figura 4.56 Formulario de Registro

El usuario selecciona la opción limpiar del formulario de registro por lo que los campos son nuevamente dejados en blanco. En la figura 4.57 se muestra un ejemplo de la opción limpiar.

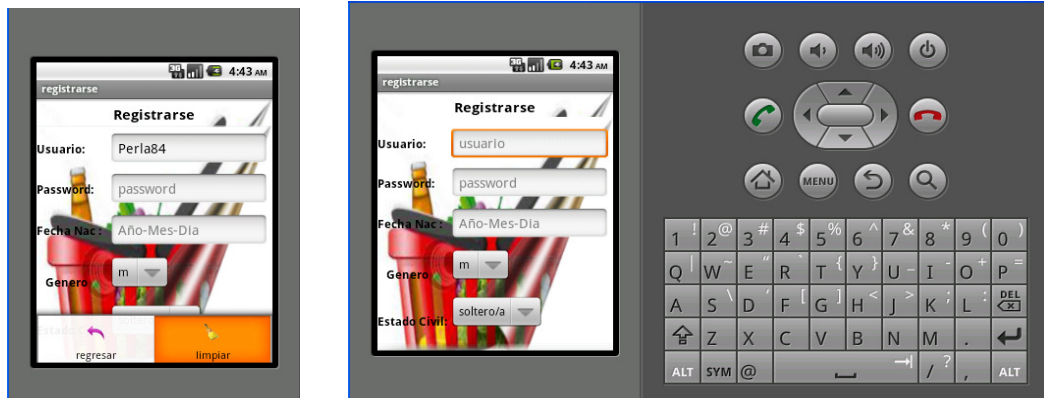


Figura 4.57 Opción Limpiar

El usuario llena los campos requeridos en el formulario de registro y da click en el botón crear cuenta, el sistema valida los campos llenados por el usuario, seguidamente se muestra un mensaje de éxito regresando al menú principal. Ver figura 4.58.

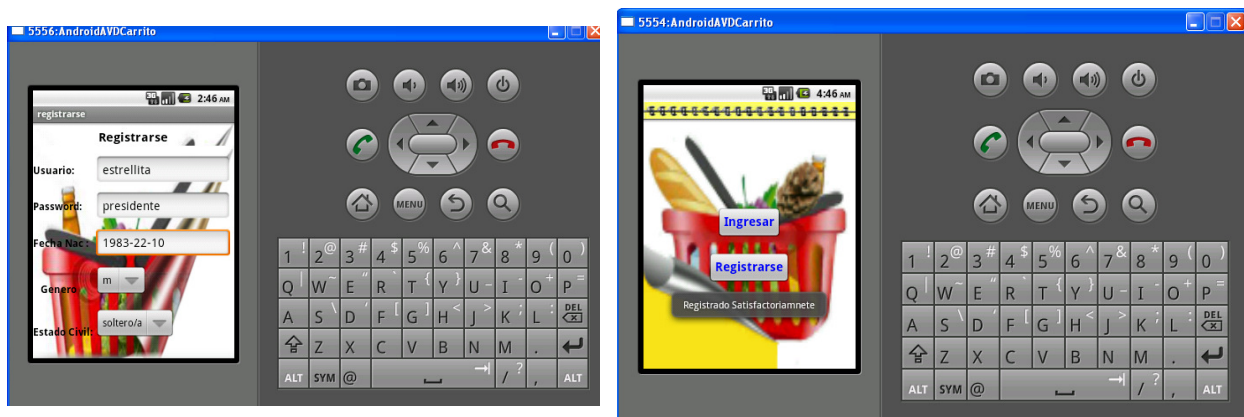


Figura 4.58 Registro

### 4.4.2.3 Opción Ingresar

El usuario selecciona la opción ingresar a continuación se muestra el formulario con los campos usuario y password, el usuario deja los campos en blanco, seguidamente se muestra un mensaje de error, como en la figura 4.59.

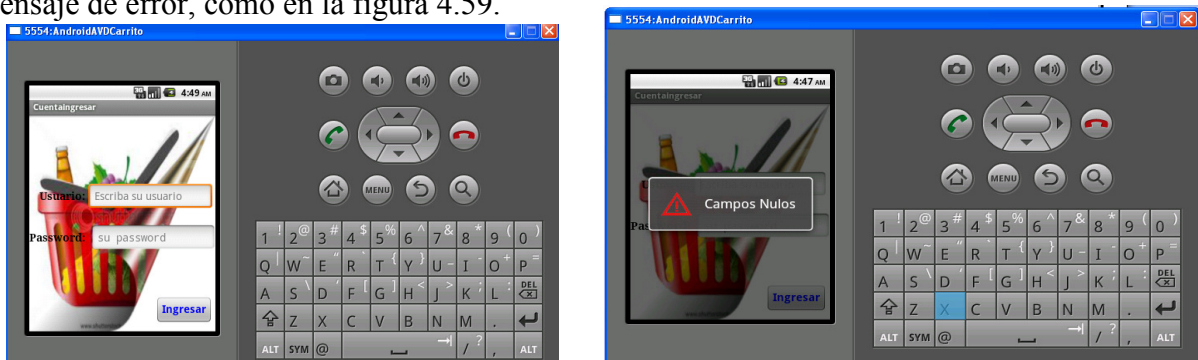


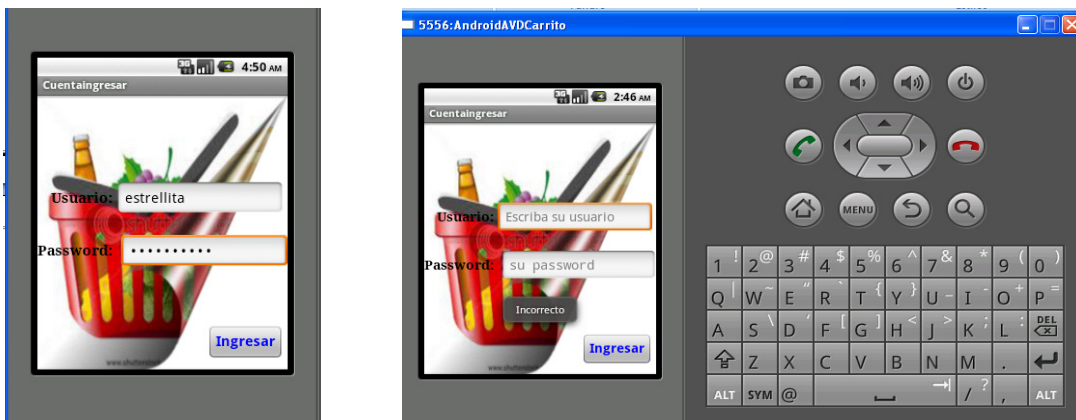
Figura 4.59 Formulario de Ingreso

El usuario selecciona la opción limpiar del formulario ingresar por lo que los campos son nuevamente dejados en blanco. En la figura 4.60 se muestra el menú del formulario ingresar con las opciones regresar y limpiar.



Figura 4.60 Opción Limpiar

El usuario llena los campos requeridos en el formulario ingresar da click en el botón ingresar, el sistema valida los datos mostrando un mensaje de error ya que el usuario no existe en la base de datos.



El usuario realiza su registro e ingresa nuevamente al sistema, los datos son validados por el sistema, se muestra la página principal. Ver figura 4.61.



Figura 4.61 Ingresar

#### 4.4.2.4 Crear Lista

El usuario crea una lista nueva llenando los campos requeridos del formulario crear lista. El usuario deja los campos en blanco el sistema muestra un mensaje de error hasta que se introduzca texto en los campos, en caso contrario el usuario introduce texto en los campos por lo que el sistema muestra un mensaje de éxito indicando que la lista ha sido creada. En la figura 4.62 se muestran los diferentes tipos de mensajes en cada uno de los casos mencionados anteriormente.

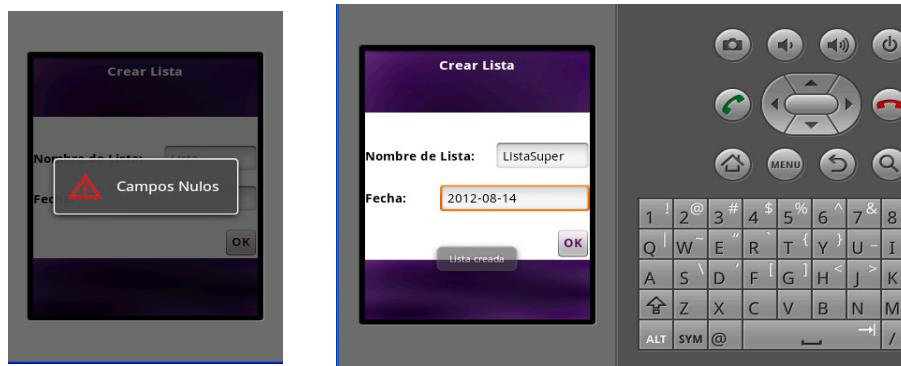


Figura 4.62 Formulario crear lista

#### 4.4.2.5 Ver Mis Listas

El usuario selecciona la opción ver mis listas del menú principal, a continuación se despliegan las listas existentes del usuario. En la figura 4.63 se muestra la pagina resultante de la opción ver mis listas solicitada por el usuario.



Figura 4.63 Opción ver mis listas

#### 4.4.2.6 Ver productos de lista

El usuario selecciona una lista, seguidamente se muestran las opciones de cotizar, ver productos de lista, eliminar lista y vaciar lista, el usuario selecciona la opción ver productos de lista por lo que el sistema muestra al usuario el formulario con los productos existentes de dicha lista. En caso contrario aparece un mensaje de lista vacía. Ver figura 4.64.

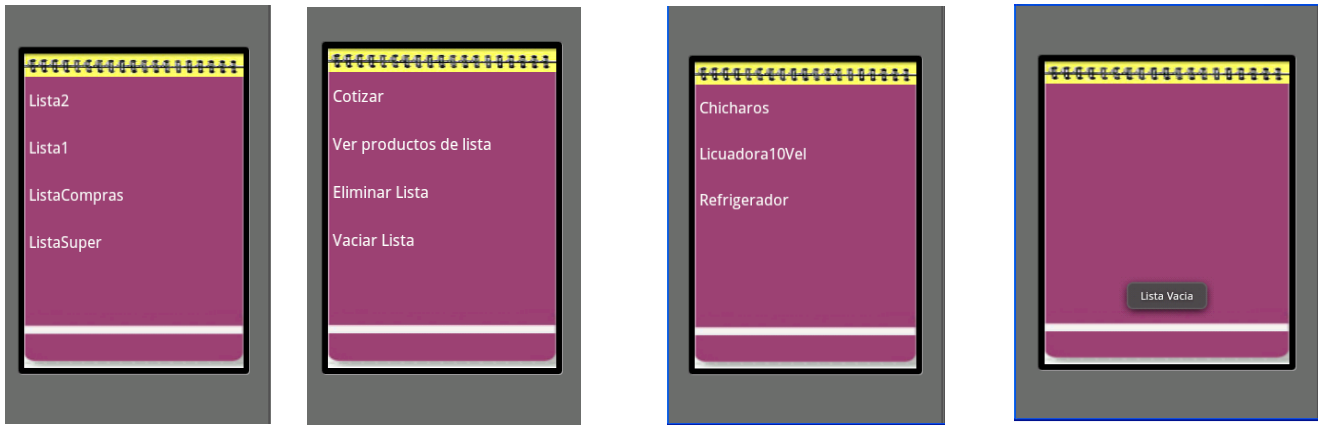


Figura 4.64 Opción ver productos de lista

#### 4.4.2.7 Cotizar

El usuario selecciona una lista seguidamente se muestran las opciones de cotizar, ver productos de lista, eliminar lista y vaciar lista, el usuario selecciona la opción cotizar por lo que se muestra la lista de productos cotizados en un rango de 5 productos mas económicos. En la figura 4.65 muestra un ejemplo detallado sobre la opción cotizar.

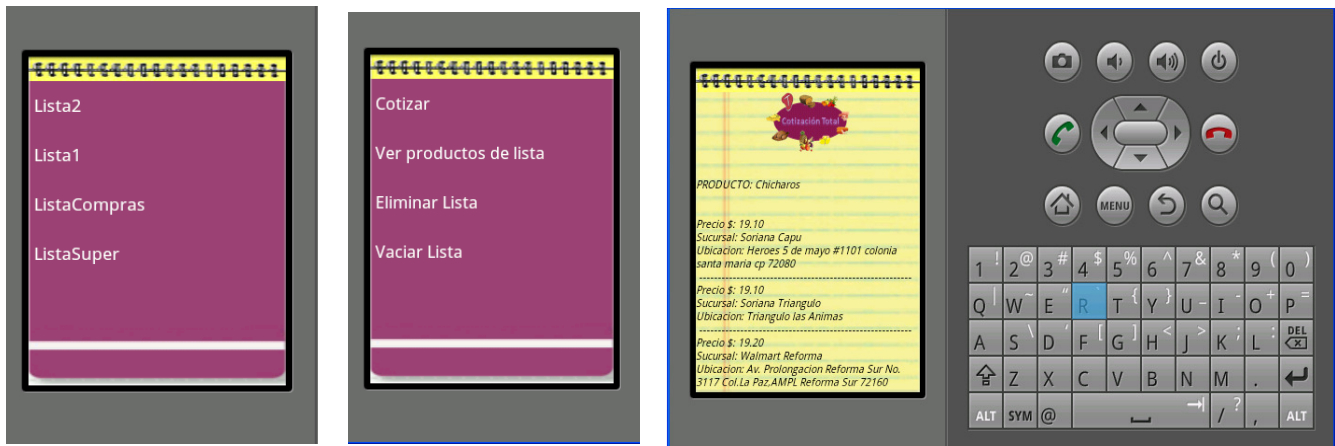


Figura 4.65 Opción cotizar

#### 4.4.2.8 Opción visualizar productos

El usuario selecciona la opción visualizar productos del menú principal por lo que se despliega el formulario de las categorías en la que los productos se encuentran divididos, seguidamente el usuario da click en una de las categorías el sistema despliega el formulario con los productos correspondientes. En la figura 4.66 se detalla un ejemplo de la opción visualizar productos.



Figura 4.66 Opción visualizar productos

#### 4.4.2.9 Opción detalles

El usuario selecciona un producto de las categorías existentes automáticamente se despliegan las opciones de agregar y detalles, por lo que el usuario selecciona la opción detalles en la que se visualiza la descripción del producto. Ver figura 4.67.

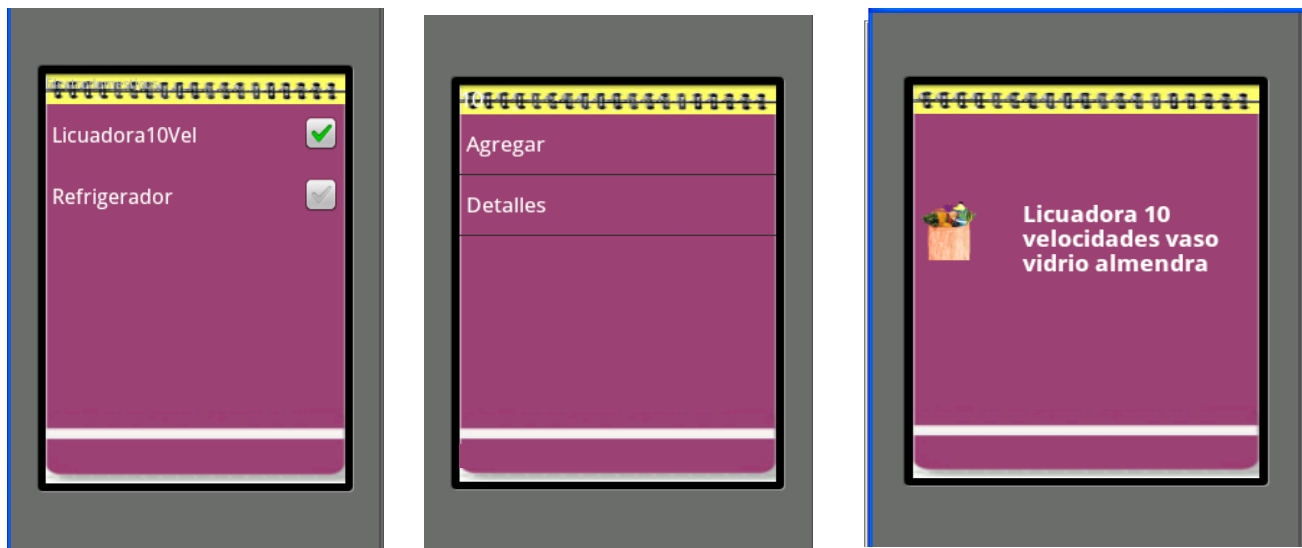


Figura 4.67 Opción detalles

#### 4.4.2.10 Opción Eliminar Lista

El usuario selecciona una lista de productos, automáticamente se despliegan las opciones de cotizar, ver productos de lista, eliminar lista y vaciar lista, por lo que el usuario selecciona la opción eliminar lista, el sistema elimina la lista seleccionada por el usuario. Ver figura 4.68.

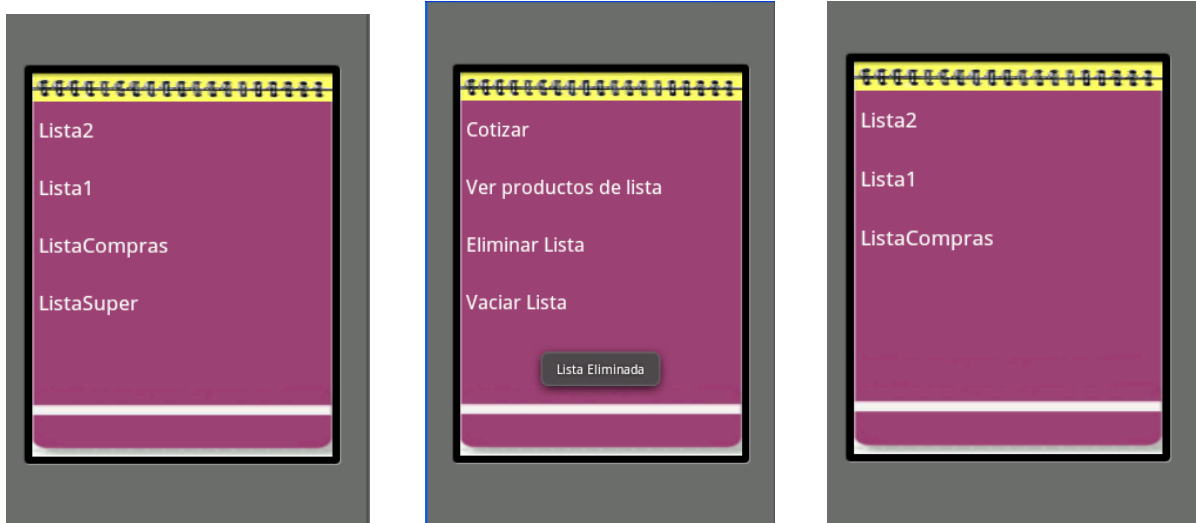


Figura 4.68 Opción eliminar lista

#### 4.4.2.11 Opción Vaciar Lista

El usuario selecciona una lista de productos, seguidamente se despliegan las opciones de cotizar, ver productos de lista, eliminar lista y vaciar lista, por lo que el usuario selecciona la opción vaciar lista en la el sistema elimina los productos contenidos de esa lista. Como se muestra en la figura 4.69.

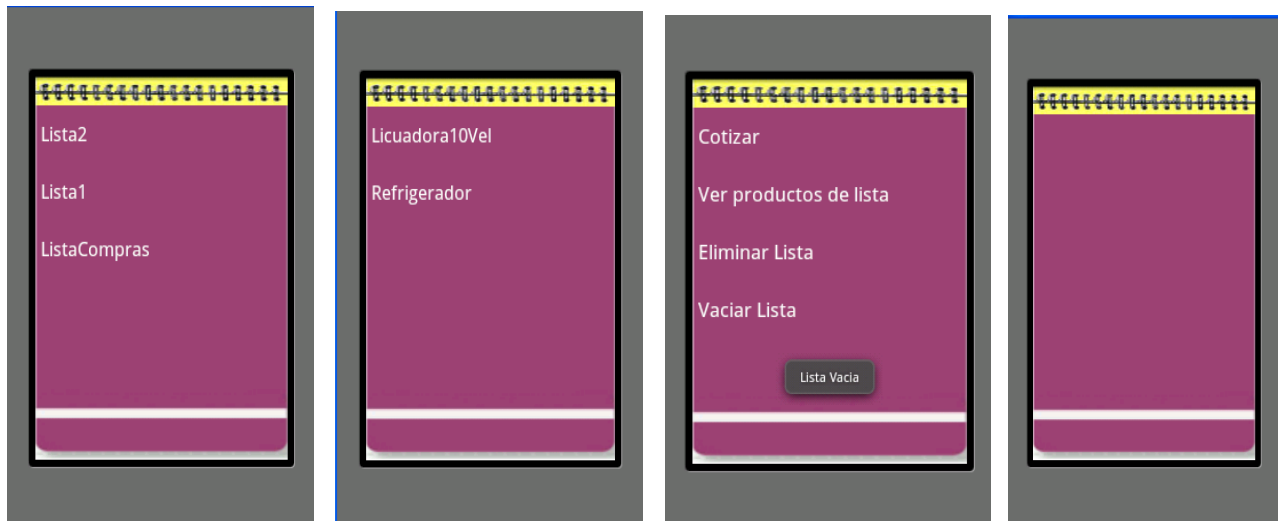


Figura 4.69 Opción vaciar lista

Los pruebas realizadas en este capítulo fueron llevadas a cabo con personas externas a la Facultad. Se fueron realizando paso a paso y verificando con los requerimientos planteados en el sistema así como también los diferentes métodos de pruebas que agrupa la prueba de caja negra. Es por ellos que los resultados fueron los siguientes:

- La prueba de la opción registrarse es satisfactoria ya que en caso de no existir el usuario en la base de datos permite registrarse y así ingresar al sistema haciendo uso del mismo, es así como cubre uno de los requerimientos del sistema.
- La prueba de la opción ingresar es satisfactoria ya que permite al usuario ingresar al sistema y hacer uso del mismo, cubriendo otro de los requerimientos del sistema.
- La prueba de la opción crear lista es satisfactoria ya que permite al usuario crear el numero de listas necesarias para el usuario siendo así cumple con otro de los requerimiento planteado en el sistema.
- La prueba de la opción eliminar es satisfactoria ya que permite al usuario eliminar la lista de productos cuando ya no sea de su utilidad, con esta prueba se cumple otro de los requerimientos del sistema.
- La prueba de cotizar la cual es la prueba principal y objetivo de este sistema, se cumple de manera satisfactoria ya que el sistema genera las cotizaciones necesarias para el usuario permitiéndole así la toma de decisiones en la compra de supermercados. Esta prueba se llevo a cabo de manera más detallada con los usuarios, para obtener una mejor conclusión.
- Cada uno de las pruebas mencionadas en este capitulo cumplieron satisfactoriamente con los requerimientos del sistema.

## Conclusión y Trabajo Futuro

La importancia del uso de tecnologías como son los dispositivos móviles cada vez es mayor, es por ello que la tradicional lista de la compra realizada con lápiz y papel ha llegado hasta las pantallas táctiles. Uno de los grandes beneficios de los dispositivos móviles hacia las listas de compras es la portabilidad.

Las pruebas presentadas en el capítulo 4 confirman que la aplicación generada es capaz de almacenar datos, conexión a red, facilidad de uso en sus interfaces, portabilidad al realizar las listas de la compra, principales objetivos de este sistema.

Por otro lado las ventajas de la plataforma Android es que es una de las más actuales plataformas para móviles y de tipo open source, con un mercado más amplio, donde existe mayor probabilidad de que la gente tenga dicha plataforma.

Actualmente el desarrollo de aplicaciones hacia los dispositivos móviles es extenso, es por ello que una mejora al sistema consistiría en la visualización de la ubicación de la sucursal a través de mapas esto facilitaría al cliente su ubicación.

Dicho Sistema de Toma de Decisiones aun no esta automatizado para generar cotizaciones automáticas, en este caso se podría vincular un agente inteligente que genere la cotización dada las preferencias del usuario. Tal caso seria que el sistema localice el lugar donde se encuentre el usuario y únicamente tome en cuenta las sucursales cercanas para generar las cotizaciones.

Otras de las mejoras en el sistema consistiría en mencionar las ofertas existentes de acuerdo a la cotización que solicite el usuario.

# BIBLIOGRAFIA

## LIBROS

- [1] Ian Sommerville  
Ingeniería de Software  
Pearson Educacion; Madrid 2005  
ISBN 84-7829-074-5
  
- [2] Braude, Eric J.  
Ingeniería de Software: una perspectiva orientada a objetos  
Alfaomega México; Febrero 2003  
ISBN 970-15-0851-3
  
- [3] Roger S. Pressman.  
Ingeniería de Software: un enfoque práctico  
McGraw-Hill; 2002  
ISBN 844-8132-14-9
  
- [4] Shari Lawrence Pflieger  
Ingeniería de Software Teoría y Práctica  
Prentice Hall; Enero 2002  
SBN 987-9460-71-5
  
- [5] Weitzenfeld Alfredo  
Ingeniería de Software Orientado a Objetos con UML, Java e Internet  
Mexico Thompson; 2005  
ISBN 9706861904
  
- [6] B. Navathe Shamkant  
Sistema de Bases de Datos: conceptos fundamentales  
Addison Wesley Longman de México, S.A de C.V  
ISBN: 968-444-399-4
  
- [7] Amaro Soriano José Enrique  
Android Programación de dispositivos móviles a través de ejemplos  
Alfaomega Grupo Editor, S.A de C.V., México  
ISBN: 978-607-707-370-3
  
- [8] Larry Ullman  
Guía de aprendizaje mysql  
Pearson Educación S.A, Madrid 2003  
ISBN: 8420538434

- [9 ] Fco. Javier Ceballos Sierra  
Java 2 Curso de programación  
Alfaomega México, 2003  
ISBN: 9701508491
- [10] Roger S. Pressman  
Ingeniería del Software un enfoque practico  
McGraw-Hill  
ISBN: 0072853183
- [11] Froufe Quintas Agustín  
Java 2 Manual de usuario y tutorial 4ª edición  
Alfaomega Grupo editor México, Octubre 2005  
ISBN: 970-15-1156-5

#### REFERENCIAS EN INTERNET

- [R1] [http://leo.ugr.es/J2ME/INTRO/intro\\_4.htm](http://leo.ugr.es/J2ME/INTRO/intro_4.htm)
- [R2] <http://www.slideshare.net/oskr crazy/sistemas-operativos-para-dispositivos-moviles>
- [R3] <http://www.alegsa.com.ar/Dic/micronavegador.php>