



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**FACULTAD DE INGENIERÍA**



**Módulo de autoservicio para selección de equipos  
y planes tarifarios para telefonía móvil**

**T E S I S**

QUE PARA OBTENER EL TÍTULO  
DE  
INGENIERO EN COMPUTACIÓN

PRESENTAN:

**Lara Ruiz Lorenzo**  
**Lima Ángeles Sergio Javier**  
**Vázquez Ortega Irene**

DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA

CIUDAD UNIVERSITARIA

México D.F. 2015



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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



## **Dedicatorias.**

A mi Madre, Norma Ángeles; tu amor, tu apoyo incondicional y tu ejemplo han hecho posible este y todos los logros en mi vida... te amo con el corazón.

A mis Hijos, Mateo y Mariana; siempre y pase lo que pase, son y serán mi mejor obra y mi mayor orgullo; los veo y me pregunto ¿Quién le dio vida a Quien?, siempre serán mi mayor motivación para seguir avanzando y aprendiendo. Espero que la conclusión de este proyecto que queda plasmada en esta tesis, les sirva de ejemplo de que nunca es tarde para cumplir con sus propósitos y de que con esfuerzo y dedicación, toda meta es alcanzable y todo sueño es posible.... Los amo hijos.

A mis Abuelitos, Luchita y Fili, por su tiempo, sus cuidados y su amor, gracias por inculcarme la Fe que me ha sacado adelante aún en mis peores momentos.

A mi Padre, Sergio Lima, gracias por todo tu apoyo.

A Irene Vázquez.

A Lorenzo Lara, gracias por demostrarme el valor de una buena amistad, sincera e incondicional.

Por enseñarme con tu ejemplo de vida que cuando se quiere se puede y que no hay obstáculo o adversidad que el trabajo duro y el amor no puedan vencer, por ser amiga, cómplice, compañera, por el apoyo incondicional en las metas y sueños que hemos de concretar; por ayudarme a recuperar la Fe en mí y por animarme, obligarme, a dar el primer paso de este proyecto que concluirá al obtener mi título.... Gracias Totales Lily.

*Sergio Javier Lima Angeles*

## **Dedicatorias.**

A Mateo porque gracias a su gran corazón, día a día me enseña que la vida es solo una y que es muy importante vivir con gran intensidad, amar y tener pasión por las cosas. Hijo gracias por ser tan amoroso y enseñarme que el tiempo con la familia es el mejor regalo que podemos tener, por ser parte importante de mi vida. Te amo hijo.

A Luz Mariana que a pesar de ser una pequeña, me ha demostrado que lo que siembras cosechas, que hay que ver la vida tan simple y disfrutarla. Que podemos equivocarnos y ser mejores cada día. Gracias hija por demostrar esa fuerza y lucha y enseñarme que no hay que olvidar al niño que llevamos dentro.

Delfina, gracias por darme la vida y demostrarme día a día que una mujer es capaz de alcanzar muchas metas, sin poner pretextos. Hacer las cosas con pasión y amor. Que se nos está permitido caer y levantarnos para ser mejores personas. A 11 años de tu partida, no me queda más que darte las gracias por ser mi madre y decirte que al fin logre conquistar una meta más. Gracias mamita por tu amor.

Cliserio, gracias por todo el esfuerzo, dedicación, coraje, valor e inteligencia que día a día me has demostrado. Siempre agradeceré tu apoyo incondicional, en las buenas y en las malas. Gracias por toda la educación que me diste. Te amo papito.

Abue Pini, gracias por haberme cuidado y amado tanto, tus comidas, conversaciones y rezos siempre me hicieron la vida más tranquila, lo logré, al fin tu nieta logró ser profesionalista!

A Lorenzo y Sergio por ser parte de esto.

Arturo, Bety y Laura, gracias por ser y estar.

A ti que formas parte importante en mi vida, que me ha dado la fuerza para quitarme esas telarañas de mi cabeza y saber que una mujer es capaz de hacer tanto. Ser mamá, hija, hermana, amiga y sobre todo ser una gran profesionalista. Que todo llega gracias a nuestro esfuerzo diario, a nuestra constancia y dedicación. Que saber manejar los tiempos nos da mejores resultados. Que la vida es tan simple como un suspiro!

*Irene Vázquez Ortega*

## **Dedicatorias.**

A mis padres:

Por su apoyo, amor, trabajo, sacrificios, por su confianza, su consejo siempre oportuno y sus jalones de oreja, durante todos estos años. Por todo ello he podido convertirme en el hombre que ahora soy y gracias a ello he llegado a este importante momento.

A Sergio:

Gracias por tú amistad, por haber sido parte de este proyecto que ahora nos pone a un paso de ser ingenieros. Agradezco, por ser los oídos para mis problemas y la voz para tú consejo.

A Irene:

Gracias por tú apoyo y aportaciones a este importante proyecto, por tú amistad, profesionalismo y humildad durante todo el proceso.

A Juan Carlos Roa:

Por su paciencia, dirección y apoyo para terminar y cerrar un ciclo como estudiante, para convertirme en un ingeniero digno de mi Facultad, de mi Universidad y de mi Patria.

*Lorenzo Lara Ruíz*

# INDICE

---

## Capítulo 1 Entorno del problema

1.1	Introducción .....	1
1.2	Descripción básica de los parámetros importantes para la elección de un equipo móvil .....	4
1.3	Normatividad que deben satisfacer los equipos móviles .....	10

## Capítulo 2 Marco teórico

2.1	Características, ventajas y desventajas de la metodología MVC y bases de datos objeto-relacionales .....	14
2.2	Características, ventajas y desventajas de JBoss, JQuery .....	20
2.3	Características, ventajas y desventajas de la base de datos Oracle .....	26
2.4	Características, ventajas y desventajas Java Script, HTML, CSS y JSP .....	28
2.5	Características, ventajas y desventajas de la arquitectura cliente-servidor .....	34

## Capítulo 3 Desglose del sistema

3.1	Análisis del problema .....	40
3.2	Requerimientos particulares y generales del usuario .....	42
3.3	Posicionamiento del software sugerido contra productos similares en el mercado .....	45
3.4	Separación por módulos utilizando la metodología MVC .....	62
3.5	Diseño del front-end a nivel de bloques sugerido al usuario .....	70

## Capítulo 4 Diseño y construcción de la aplicación

4.1	Modelo esencial ambiental del sistema	
4.1.1	Diagrama de contexto .....	49
4.1.2	Diagrama Entidad-Relación .....	52
4.1.3	Diagrama de casos de uso, procesos y clases .....	54
4.1.4	Diccionario de datos .....	68
4.1.5	Normalización de la base de datos .....	71
4.2	Construcción de la base de datos .....	81
4.3	Construcción del front-end y sitio Web .....	91
4.4	Aplicación de pruebas y mantenimiento .....	100
4.5	Generación de reportes .....	109
	<b>Conclusiones</b> .....	112
	<b>Bibliografía</b> .....	114
	<b>Apéndices</b> .....	117

# **CAPÍTULO**

# **I**

## **Capítulo 1 Entorno del problema**

### **1.1 Introducción**

La telefonía móvil está siendo cada vez más necesaria en nuestra vida diaria. Las llamadas, los mensajes, los correos, las fotos y una gran diversidad de aplicaciones hacen que la telefonía móvil nos brinde un servicio eficaz y rápido.

En la vida personal como en la empresarial, el celular, el Smartphone, el iPad, etc. nos facilitan muchas actividades, llegando a hacer cosas que eran inimaginables o que solo las podíamos ver en alguna computadora personal o portátil. Actualmente la tecnología móvil nos pone en nuestras manos todo lo que una computadora puede realizar; crear o modificar documentos de office (Word, Excel, PowerPoint), capturar video y fotos, navegar en Internet, imprimir documentos, grabar audio, GPS, WhatsApp, aplicaciones que monitorean el ejercicio, descargar música, radio, SMS, Facebook, Instagram, MSN, entre otras aplicaciones más.

Hoy en día existen en el mercado una gran variedad de marcas de telefonía celular, cada teléfono cuenta con diversos sistemas operativos: Android, IOS, Windows Phone, BlackBerry entre otros, varias opciones para adquirir equipos móviles seleccionando el servicio de pago que más nos convenza, para que el usuario escoja muy bien que teléfono se adopta a sus necesidades y dependiendo del costo conozca los beneficios que adquirirá al comprar un teléfono.

Es importante que el usuario sea consciente de que es lo que en verdad necesita hacer con su dispositivo móvil, para que no le resulte costoso el plan adquirido o el móvil comprado si solo necesita recibir llamadas o sólo necesita escribir puros mensajes de texto.

El usuario se ve en la necesidad de adquirir un equipo móvil, para estar comunicado e informado del día a día. Es importante señalar, que en la actualidad con un teléfono móvil se envían documentos por correo, se modifican se imprimen y se entregan al usuario de forma rápida y exitosa.

Es por esa razón que nosotros quisimos desarrollar un sistema que le muestre al usuario en una interfaz gráfica y amigable los teléfonos celulares que se ofrecen en el mercado, sus principales características así como las distintas opciones de planes tarifarios, para que el usuario decida qué opción se ajusta a sus necesidades.

Sabemos que antes, el usuario tenía que ir a las instalaciones donde se adquirían los equipos móviles, ver todos los teléfonos que se tenían en ese momento y leer la ficha técnica de cada móvil de su interés, preguntar al vendedor si había en otro color, esperar la respuesta y volver a preguntar si mejor había otro equipo con mejores características, como sabrán era mucho tiempo perdido y el usuario confundía las especificaciones de cada teléfono visto.

La finalidad de este sistema es brindarle al usuario de forma clara y rápida las características de todos los equipos que hay en este momento, así como llevarlo de forma dinámica a través del software al seleccionar un plan tarifario e ir manipulando los diferentes modelos de teléfonos.

El usuario no escribirá, todo será con filtros para que eso le permita mover cada selección que se presente. En cuanto al diseño de nuestro aplicativo, será desarrollado en Oracle 11g como motor de base de datos y emplearemos el software libre como Jboss para el servidor de aplicaciones, para el front-end se utilizará HTML( **HyperText Markup Language**), CSS ( **Cascading Style Sheets** ) y JSP ( **Java Servlet Pages** ), Java script y JQuery para que la aplicación sea mucho más dinámica.

A continuación se mencionan los objetivos esperados al concluir nuestra aplicación:

- Proporcionar una interfaz gráfica atractiva y amigable.
- Permitir la navegación sencilla entre las diferentes pantallas.
- Presentar de forma clara y con imágenes atractivas los diferentes modelos de teléfonos celulares con los que cuenta el mercado, permitiéndole el acceso sencillo a información detallada de dichos teléfonos.

- Presentar de forma clara y en una vista amigable las características más importantes de los planes tarifarios con los que se cuentan a nivel general.
- Relacionar de forma sencilla el teléfono que el usuario elija con el o los planes tarifarios para los cuales está disponible.
- Recopilar la información necesaria para levantar el pedido del teléfono y plan tarifario y a su vez pasarlo al sistema de contratación-facturación.
- Posicionar nuestra aplicación ante los clientes como una excelente opción para la compra de teléfonos y contratación de planes tarifarios.
- Vender una excelente imagen del desarrollo ante los clientes, promocionando de forma clara, sencilla y vistosa los productos y servicios que se ofrecen.
- Evitar en lo posible la excesiva captura de datos por parte del usuario.
- Filtrar la información de acuerdo a las necesidades del usuario, ya sea por marca, precio, nombre y plan.

El Capítulo 1 Entorno del problema, se explica de forma clara el por qué nuestro sistema será de gran ayuda al usuario que piensa adquirir un teléfono móvil. Daremos a conocer los parámetros más importantes que el usuario debe considerar antes de adquirir un teléfono celular, así como algunas Normas que deben de satisfacer los equipos móviles. Mostraremos como es un organigrama genérico que se tiene en una empresa de telefonía móvil, explicando sus responsabilidades. Se tomaron en cuenta algunos conceptos mercadológicos que influyen en el usuario al adquirir un teléfono móvil.

El Capítulo 2 encontrará el Marco Teórico. En donde explicamos en que nos basamos para desarrollar nuestro sistema con base a los conocimientos adquiridos durante la carrera. Nuestro sistema será desarrollado con la metodología MVC y las bases de datos objeto-relacionales, explicaremos las características, ventajas y desventajas. Para el servidor de aplicaciones usamos JBoss, mencionaremos sus características, ventajas y desventajas.

Para el Front-end usamos HTML, CSS, JSP, JavaScript y JQuery mencionando sus características, ventajas y desventajas de cada uno. Para la base de datos trabajaremos con Oracle, explicando sus características, ventajas y desventajas. Nuestro sistema está diseñado con la arquitectura cliente-servidor, explicaremos las características, ventajas y desventajas.

El Capítulo 3 Desglose del Sistema. Se analiza el problema mediante gráficos para el diseño de nuestro sistema, mencionaremos los requerimientos particulares y generales del usuario; así como el posicionamiento del software que emplearemos para nuestro sistema con respecto a productos similares en el mercado. Separaremos cada módulo empleando la metodología MVC y el diseño a nivel de bloques de cómo queremos el front-end sugerido al usuario.

El Capítulo 4 Diseño y construcción de la aplicación. Se anexaran todos los diagramas: c de contexto, de entidad-relación, de casos de uso, proceso y clases que empleamos en nuestro sistema. Así como el diccionario de datos y la normalización. Cómo construimos la base de datos, el front-end y el sitio Web para nuestro sistema. Mostraremos las pruebas y el mantenimiento de nuestro sistema, así como la generación de reportes.

## **1.2 Descripción básica de los parámetros importantes para la elección de un equipo móvil**

Cuando el usuario quiere adquirir un teléfono celular, siempre es muy importante darle a conocer los parámetros más importantes de cada equipo móvil.

Todas las compañías ofrecen una ficha que permite conocer todas las especificaciones técnicas del equipo; de estas especificaciones, fuera de las características básicas con las que todo teléfono debe contar (redes Wi-Fi, conexión Bluetooth, reproductor de música, reproductor de video, agenda, etc.) es importante que el usuario vea las características de cada equipo antes de poder adquirir algún equipo móvil.



- (4) Tecla de inicio y sensor de reconocimiento de huellas dactilares, fingerprint (sensor de huella digital) y si maneja contraseña; puede ser PIN o un patrón. Así como el acceso al móvil mediante el reconocimiento de gestos.
- (5) Tecla de aplicaciones recientes.
- (6) Toma de audio.
- (7) Clavija multifunción. Se puede cargar la batería con el cable usb al tomacorriente.
- (8) Micrófono.
- (9) Altavoz.
- (10) Tecla Atrás.
- (11) Entrada de tarjeta SIM.
- (12) Tecla de encendido.
- (13) Cámara frontal
  - Resolución
  - Resolución de grabación de vídeo
- (14) Auriculares.
- (15) Puerto Infrarojo.
- (16) Micrófono.
- (17) Cámara trasera
  - Resolución
  - con flash y autofocus.
- (18) Flash.
- (19) Antena GPS, sistema de localización GPS.
- (20) Tecla de volumen.
- (21) Antena NFC trasera de carga inalámbrica. NFC (Near Field Communication)
- (22) Antena principal.

El usuario deberá tomar en cuenta las siguientes características antes de que adquiera un equipo móvil:

- Sistema Operativo Android, el cual permite abrir varias aplicaciones (multitarea) y actualmente se cuentan con más de 650.000 aplicaciones. Así como iOS, Windows Phone, Blackberry, Symbian y Ubuntu Touch.
- Número de Procesadores y velocidad.
- Memoria Interna de 32 GB, 64 GB y 128 GB.
- Memoria externa de 1 - 32 GB.
- Batería. Capacidad en mili amperes. La Autonomía de cada móvil varía respecto a su memoria procesador. Tiempo de uso de Internet. Tiempo de reproducción de video, tiempo de reproducción de Audio, tiempo en conversación.
- Algunos móviles cuentan con sensores como: acelerómetro (sensor de movimiento), Barómetro (sensor de la presión), Giroscopio (cuando el móvil detecta un giro en la pantalla), Geomagnético (capaz de detectar cambios en el campo electromagnético, optimizando así el uso del GPS en el móvil), Hall (sensor que sabe si el móvil lleva funda o no), HR (detecta la frecuencia cardiaca al ejercitarte).
- Servicio de redes. Tarjeta Nano-SIM o SIM.
- Wi-Fi
- Bluetooth
- PC sync.

Accesorios:

- Cartera, ver la figura 1.2.2



**Figura 1.2.2 Cartera**

- Si cuenta con cargador Wireless Ver figura 1.2.3



**Figura 1.2.3 Cargador wireless**

- Si posee cubierta. Ver figura 1.2.4



**Figura 1.2.4 Cubierta**

- Si existe una funda transparente. Ver figura 1.2.5.



**Figura 1.2.5 Funda transparente**

- Si posee un protector de pantalla con pegatina para eliminar el polvo

- Si tiene un cargador para el coche, ver la figura 1.2.6



**Figura 1.2.6 Cargador para el coche**

- Si existe aditamento para selfie retráctil Monopod, ver la figura 1.2.7



**Figura 1.2.7 Bastón**

- Si posee un dispositivo para manos libres, ver figura 1.2.8



**Figura 1.2.8 Manos libres**

### **1.3 Normatividad que deben satisfacer los equipos móviles**

Conforme a lo que se establece en el Reglamento de Telecomunicaciones, todo equipo de telecomunicaciones que se comercialice, use u opere en el territorio nacional debe contar con un certificado de homologación. Los fabricantes, importadores, comercializadores, distribuidores o arrendadores de productos de telecomunicaciones, pueden obtener el Certificado de conformidad -con propósitos de homologación- de las siguientes Normas Oficiales Mexicanas (NOM)

**NOM-008-SCFI** - Esta Norma Oficial Mexicana tiene como propósito establecer un lenguaje común que responda a las exigencias actuales de las actividades científicas, tecnológicas, educativas, industriales y comerciales, al alcance de todos los sectores del país. Establece las definiciones, símbolos y reglas de escritura de las unidades del Sistema Internacional de Unidades (SI) y otras unidades fuera de este Sistema que acepte la CGPM, que en conjunto, constituyen el Sistema General de Unidades de Medida, utilizado en los diferentes campos de la ciencia, la tecnología, la industria, la educación y el comercio.

**NOM-084-SCT1** - Especificaciones técnicas de los equipos transmisores destinados al servicio móvil de radiocomunicación especializada de flotillas. Esta Norma Oficial Mexicana establece las especificaciones y métodos de prueba a que deben sujetarse los equipos para el servicio móvil de radiocomunicación especializada de flotillas.

**NOM-088/1-SCT1** - En muchos lugares de nuestro país, principalmente en aquellas regiones en vías de desarrollo (poblaciones rurales), el costo de un servicio telefónico con distribución por conductores de cobre en postes o cables enterrados es demasiado elevado y difícil de instalar y mantener, debido a las condiciones geográficas y topográficas de dichas regiones, es decir, debido a zonas densamente arboladas, grandes extensiones de agua, tramos despoblados entre poblaciones aisladas, incendios de bosques, inundaciones, etc.

Además, al difícil acceso o falta de caminos hacia esos lugares, lo que retarda la reparación de interrupciones en el servicio.

Con este motivo, la Secretaría de Comunicaciones y Transportes atribuyó la banda de frecuencias de 2 300 MHz-2 450 MHz, para la operación de sistemas de transmisión por microondas para suministrar servicios de telefonía y datos, a abonados aislados; a este tipo de servicio se le denomina TELEFONIA RURAL. Este sistema proporciona una conexión similar a la de dos hilos entre una central telefónica y un teléfono convencional remoto. Con el fin de regular su operación se ha formulado esta Norma Oficial Mexicana, ya que es un servicio de gran importancia para el desarrollo de nuestro país.

Aplica para equipos de microondas para sistemas del servicio fijo multicanal punto a punto y punto a multipunto - Parte I: Radio Acceso Múltiple. Esta Norma Oficial Mexicana establece las especificaciones y sus correspondientes métodos de prueba que deben cumplir los equipos de radiocomunicación a utilizar en un sistema de radio acceso múltiple por medio de enlaces de microondas punto a punto y/o punto a multipunto en la banda de 2300 MHz-2 450 MHz.

**NOM-088/2-SCT1** - Aplica para equipos de microondas para sistemas del servicio fijo multicanal punto a punto y punto a multipunto - Parte II: Transporte esta Norma Oficial Mexicana establece las especificaciones y sus correspondientes métodos de prueba que deben cumplir los equipos de radiocomunicación de microondas a utilizar para sistemas fijo multicanal punto a punto y punto multipunto que operan en las bandas de 7 GHz, 10.5 GHz, 15GHz, 23GHz y 38GHz.

**NOM-121-SCT1** - Los sistemas de radiocomunicación por espectro disperso además de coexistir entre ellos, coexistirán con todos los servicios de radiocomunicación mencionados anteriormente y los equipos de radiocomunicación por espectro disperso podrán operar siempre y cuando protejan de sus posibles

Interferencias perjudiciales a los equipos, redes y servicios autorizados, por lo que las especificaciones que se establecen en esta Norma Oficial Mexicana prevén esas protecciones exigidas. La Norma aplica para equipos de radiocomunicación por salto de frecuencia y por modulación digital a operar en las bandas 902-928 MHz, 2400-2483.5 MHz y 5725-5850 MHz – Especificaciones, límites y métodos de prueba.

La presente Norma Oficial Mexicana (NOM) tiene por objeto establecer las especificaciones mínimas y límites para los equipos de radiocomunicación por espectro disperso (Salto de Frecuencia, Modulación Digital e Híbrido) en las bandas de frecuencias a que se circunscribe esta NOM, previéndose que cuando operen no causen interferencias perjudiciales a otros equipos de operación autorizada, ni a las redes y servicios de telecomunicaciones de servicios autorizados; y asimismo previéndose que al operar los equipos no inhiban la existencia y coexistencia del mayor número posible de sistemas de radiocomunicación por espectro disperso y de modulación digital. También tiene por objeto establecer los métodos de prueba para la comprobación del cumplimiento de las especificaciones y límites que aquí se establecen; y asimismo contribuir al cuidado del interés de los consumidores de estos equipos.

**NOM-184-SCFI-2012** - Aplica para prácticas comerciales-Elementos normativos para la comercialización y/o prestación de los servicios de telecomunicaciones cuando utilicen una red pública de telecomunicaciones. La presente norma oficial mexicana establece los elementos normativos y de información comercial que los proveedores deben cumplir en la comercialización y/o prestación de los servicios de telecomunicaciones cuando utilicen la red pública de telecomunicaciones, así como los requisitos mínimos que deben contener los contratos de adhesión que los proveedores utilicen con los consumidores en sus relaciones comerciales para la prestación de los servicios de telecomunicaciones, a fin de que los consumidores cuenten de manera previa a la contratación con la información suficiente para tomar la decisión que más convenga a sus intereses.

La presente (NOM) es de observancia general y obligatoria para todas las personas físicas o morales que se dediquen directa o indirectamente a comercializar, prestar y/o proporcionar servicios de telecomunicaciones a consumidores cuando utilicen la red pública de telecomunicaciones en la República Mexicana.

# **CAPÍTULO**

## **II**

## Capítulo 2 Marco Teórico

### 2.1 Características, ventajas y desventajas de la metodología MVC y bases de datos objeto-relacionales

#### 2.1.1 Características y principales ventajas y desventajas de la metodología MVC

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que divide una aplicación en tres componentes distintos; los datos (Modelo), la interfaz del usuario (Vista) y la lógica de control (Controlador).

El controlador es el encargado de redirigir o asignar una aplicación (un modelo) a cada petición; el controlador debe tener un mapa de correspondencias entre peticiones y respuestas (aplicaciones o modelo) que se les asignan.

El modelo sería la aplicación que responde a una petición, es la lógica de negocio a fin de cuentas. Una vez realizadas las operaciones necesarias el flujo vuelve al controlador y este devuelve los resultados a una vista asignada.

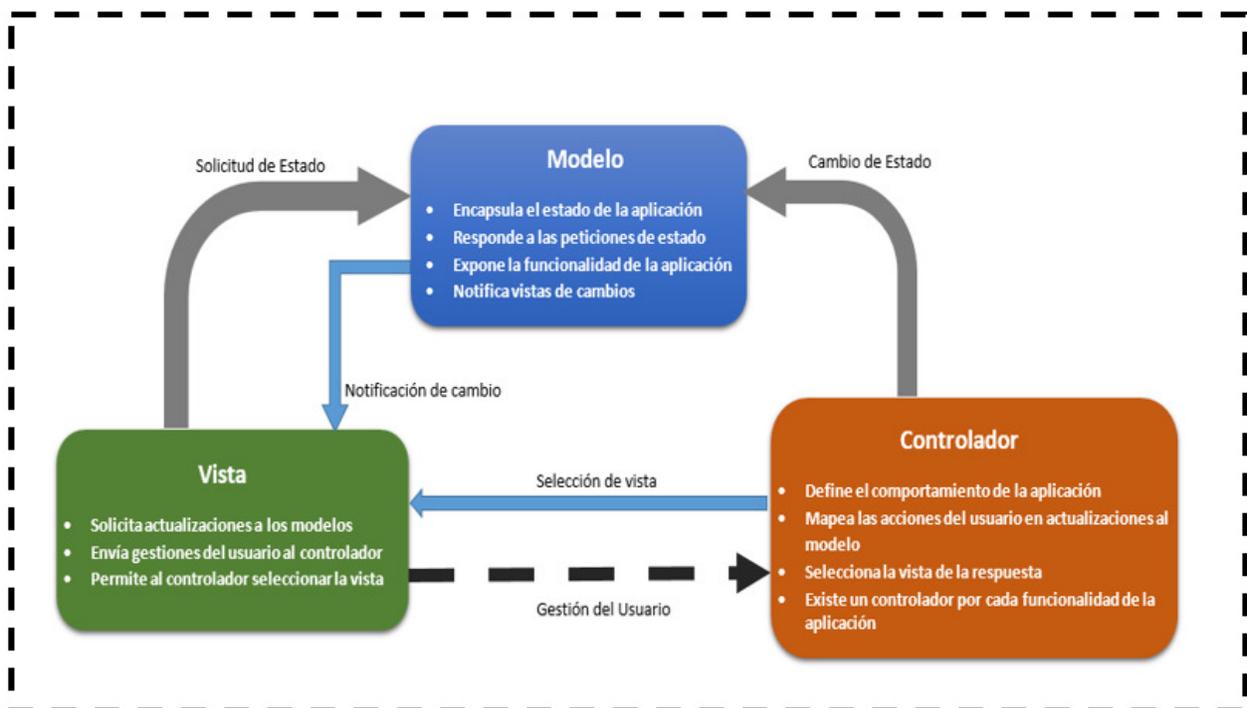


Figura 2.1.1.1 Esquema del modelo Vista Controlador

La figura 2.1.1.1 muestra cómo se lleva a cabo el procesamiento de una aplicación diseñada utilizando la metodología MVC. El proceso se divide entre sus tres componentes; primero, el controlador recibe una orden y decide cuál de las rutinas que componen al modelo es la responsable de llevar a cabo la tarea solicitada, una vez que el modelo (la lógica de negocio) concluye sus operaciones y tiene listo el resultado, devuelve el flujo al controlador y este le entrega el resultado generado a la capa de presentación (vista). A continuación, se explican de forma más amplia cada uno de los tres componentes de la metodología.

### **Modelo**

El Modelo es la representación de los datos que usa la aplicación para operar. La lógica de datos asegura la integridad de éstos y permite generar nuevos datos; dicho de otro modo, el Modelo opera con los datos y las reglas de negocio asociadas a la aplicación (incluidos el procesamiento de los datos de entrada y su análisis sintáctico) y obtiene nuevos datos como resultado.

En la metodología MVC, el Modelo es responsable de:

- Tener acceso a la capa de almacenamiento de datos; lo ideal es que el Modelo sea independiente del sistema de almacenamiento
- Definir las reglas de negocio, la funcionalidad de la aplicación como tal
- Llevar un registro de las vistas y controladores que constituyen la aplicación

### **Vista**

Este es el componente que presenta al modelo, es la interfaz con el usuario. La Vista es la capa de la aplicación con la cual el usuario interactúa por lo cual debe mostrarse en un formato adecuado para que se realice dicha interacción (interfaz gráfica).

En la metodología MVC, la Vista es responsable de:

- Recibir los datos de entrada por parte del usuario y en su caso, presentar los datos obtenidos como resultado
- Tener un registro de su Controlador asociado

## **Controlador**

Como su nombre lo indica, controla todo aquello que puede realizar la aplicación. Responde a eventos, generalmente peticiones del usuario hechas a través de la Vista, e invoca cambios en el Modelo. Está compuesto por acciones representadas con funciones en una clase.

En la metodología MVC, el Controlador es responsable de:

- Recibir los eventos de entrada (un clic, captura de texto en un campo, selección de una opción en una lista, etc.)
- Administrar las reglas que gestionan los eventos
- Secuencialmente, estos son los pasos mediante los cuales entra en acción el Controlador dentro de la metodología MVC
  - El usuario introduce o dispara un evento a través de la Vista
  - El Controlador recibe el evento y lo traduce en una petición al Modelo
  - El modelo actualiza a la Vista, si es necesario
  - Para cumplir con la actualización anterior, la Vista puede solicitar datos al Modelo
  - El controlador recibe el resultado para pasarlo a la Vista o bien a otro proceso

## **Ventajas y desventajas de la implementación de la metodología MVC**

### **Ventajas**

La popularidad de esta metodología se debe principalmente a las facilidades que aporta para diseñar y construir aplicaciones muy grandes y complejas. En general, la metodología MVC ha demostrado ser un patrón de diseño robusto y bien elaborado, las principales ventajas de utilizar la metodología MVC son

- Aporta una clara separación entre interfaz, lógica de negocio y presentación
- Sencillez para crear distintas representaciones de los mismos datos, lo cual permite una gran adaptación a los requerimientos hechos por el usuario
- Facilidad para probar los diferentes módulos o componentes de la aplicación por separado

- Permite la reutilización de los componentes, ya sea como base para componentes más complejos dentro de la misma aplicación o bien como componentes de otras aplicaciones totalmente diferentes e independientes
- Hace muy sencillo el mantenimiento de las aplicaciones, permitiendo corregir errores o desempeño no esperado al ponerlas en producción o bien adaptarlas a nuevos requerimientos por parte del usuario. Hace que las aplicaciones sean más escalables
- Las vistas siempre muestran información actualizada; los desarrolladores no deben ocuparse en actualizar las vistas ya que este proceso lo realiza el modelo de la aplicación como parte de sus funciones por definición

## **Desventajas**

Las desventajas de la metodología MVC son

- Se requiere una mayor dedicación en las etapas iniciales del desarrollo. Normalmente, esta metodología exige al programador desarrollar un mayor número de clases que, en otros entornos de desarrollo, no son necesarias. Esta desventaja es muy relativa ya que en etapas posteriores, es precisamente esto lo que permite que las aplicaciones sean mucho más fáciles de mantener, de modificar y de escalar
- La metodología MVC requiere que exista una arquitectura base inicial sobre la cual construir clases e interfaces para modificar y comunicar los módulos de una aplicación. Esta arquitectura inicial debe incluir, por lo menos, un mecanismo de eventos para poder proporcionar las notificaciones que genera el Modelo de la aplicación; una clase Modelo, una clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, las notificaciones y las actualizaciones que serán luego transparentes para el desarrollo de la aplicación
- MVC es una metodología orientada a objetos por lo que su implementación se hace muy costosa y complicada en lenguajes que no son orientados a objetos

## **2.1.2 Características y principales ventajas y desventajas de las Bases de Datos Objeto-Relacionales**

Existen aplicaciones para las cuales las bases de datos relacionales no son adecuadas ya que manejan tipos de datos complejos. [1] Por otro lado, el estándar de desarrollo más popular en la actualidad es la programación orientada a objetos; lo anterior ha generado la necesidad de incorporar los objetos en los modelos de las bases de datos. El término Base de Datos Objeto-Relacional se usa para describir a una base de datos que ha evolucionado desde el modelo relacional hacia una Base de Datos híbrida, que contiene ambas tecnologías, la relacional y la orientada a objetos.

Las Bases de Datos Objeto-Relacionales son una extensión de las bases de datos relacionales tradicionales, a las cuales se les proporcionan características de la programación orientada a objetos. Con las Bases de Datos Objeto-Relacionales, se pueden crear nuevos tipos de datos, que permiten gestionar aplicaciones más complejas con una mayor cantidad de dominios; dichos tipos de datos pueden ser compuestos, lo que implica que al menos se deben definir dos métodos para transformar, uno para pasar el nuevo tipo de datos a ASCII y otro para pasar de ASCII al nuevo tipo de datos.

Las Bases de Datos Objeto-Relacionales soportan tipos de datos complejos como referencias, listas, conjuntos, pilas, registros y arreglos; también pueden crearse funciones en algún lenguaje de programación (SQL, C, JAVA, etc.). Permiten una mayor capacidad para expresar asociaciones y conceptos.

Las Bases de Datos Objeto-Relacionales son compatibles en sentido ascendente con las bases de datos relacionales, por lo que se pueden utilizar con aplicaciones desarrolladas anteriormente sin tener que reescribirlas y posteriormente adaptarlas para que utilicen funciones orientadas a objetos

## **Ventajas y desventajas de la implementación de Bases de Datos Objeto-Relacionales**

### **Ventajas**

Al ser una evolución, un híbrido como ya se dijo, de los modelos de bases de datos relacionales y orientados a objetos, las Bases de Datos Objeto-Relacionales ofrecen las siguientes ventajas: [40]

- Resuelven muchos de los problemas que presentan las bases de datos relacionales, tales como
  - Pobre representación de las entidades del mundo real, obligando a descomponer estas entidades en varias tablas
  - La sobrecarga semántica de las tablas, ya que se utilizaban para almacenar tanto entidades como relaciones, sin que se pudiera distinguir automáticamente qué representaba cada tabla
  - Soporte limitado para expresar y mantener las reglas de integridad y las reglas de negocio
  - La dificultad de manejar consultas recursivas, es decir, consultas sobre relaciones que una tabla tiene consigo misma
  - El conjunto fijo y limitado de operaciones que se pueden realizar y que resultaba muy restrictivo para modelar el comportamiento de muchos objetos del mundo real
- Permiten y hacen mucho más sencillo la reutilización y la compartición de las bases de datos
- Mejoran significativamente la productividad, al permitir la adaptación a las necesidades del negocio
- En su implementación, permiten el uso de conocimientos y experiencia alcanzados con el uso de bases de datos relacionales, por lo que no se necesita un ciclo muy extenso de capacitación para los administradores

## **Desventajas**

Las desventajas que se presentan al implementar Bases de Datos Objeto-Relacionales son

- Se incrementa la complejidad de los desarrollos y de la administración. Al proveer mayores facilidades y herramientas para modelar objetos del mundo real, también se eleva la complejidad de las implementaciones y su posterior administración
- Mayores costos de implementación y de operación
- Solo aplicaciones grandes y complejas pueden usar todo el potencial de las Bases de Datos Objeto-Relacionales

## **2.2 Características ventajas y desventajas de Jboss y JQuery**

### **Principales características, ventajas y desventajas de JBoss**

#### **Introducción**

En la actualidad, las aplicaciones Web han cobrado cada vez mayor importancia, sobre todo con el boom del comercio electrónico, la mayoría de las grandes corporaciones que ofrecen productos y servicios dependen de sus sitios Web para interactuar con sus consumidores y usuarios ya sea para levantar pedidos, dar seguimiento a los mismos, contratar servicios, mostrar productos, etc.

Un elemento clave para estas aplicaciones Web a gran escala, son sin duda los servidores de aplicaciones, una pieza muy importante en el desarrollo de servicios Web y en general una de las piezas más estratégicas de software para las empresas que delegan en sus sitios de Internet gran parte de su estrategia de negocio. [33]

Un servidor de aplicaciones es una pieza de software que proporciona servicios de aplicación a las computadoras o dispositivos cliente; su función principal es llevar a cabo la gestión de las funciones de lógica de negocio y el acceso a los datos de aplicación.

Los principales beneficios que aporta la implementación de un servidor de aplicaciones son la centralización y la disminución de la complejidad en el desarrollo de dichas aplicaciones. Un servidor de aplicaciones permite mejorar tres aspectos fundamentales en una aplicación, la alta disponibilidad, la escalabilidad y el mantenimiento.

## **Características**

- Jboss es un servidor de aplicaciones desarrollado en código abierto e implementado en Java puro
- Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el cual esté disponible la máquina virtual de Java
- El proyecto JBoss se desarrolla y se mantiene por una red de programadores a nivel mundial y forma parte del portafolio de productos de la empresa Red Hat Inc.
- JBoss provee la plataforma para desarrollar y publicar aplicaciones Java empresariales, aplicaciones Web, así como servicios y portales.
- Permite el uso de componentes modulares estandarizados y da las herramientas necesarias para manejar muchos aspectos de la programación automáticamente.

## **Ventajas**

Las principales ventajas de implementar JBoss como servidor de aplicaciones son

- En una plataforma de código abierto y accesible que permite implementar arquitecturas orientadas a servicios
- Al estar desarrollada en código abierto, el soporte técnico y la documentación son fáciles de conseguir
- Se cuenta con acceso al código fuente y binarios
- Aumenta la productividad de las empresas
- Disminuye drásticamente los costos totales de propiedad
- Al estar soportado por una comunidad mundial de programadores, se cuenta regularmente con mejoras, parches, nuevas características, etc.

- Las revisiones y el acceso a las últimas correcciones de fallas y errores de seguridad se dan de forma más rápida y sencilla

## **Desventajas**

A continuación se enlistan las desventajas de implementar JBoss como servidor de aplicaciones

- No provee seguridad basada en roles para la parte de administración
- Presenta problemas al implementar aplicaciones que requieren garantizar la integridad de los datos al acceder simultáneamente a dos o más bases de datos
- En comparativos de desempeño contra servidores de aplicaciones comerciales como Websphere de IBM o GlassFish de Oracle, queda por debajo de estos
- Soporte Limitado en cuanto a los sistemas operativos sobre los que se puede instalar, de hecho, oficialmente solo está soportado sobre Red Hat Enterprise Linux
- La herramienta de administración de JBoss utiliza la base de datos para guardar la configuración del servidor de aplicaciones, haciéndolo más difícil de mantener y actualizar
- Como se utiliza la base de datos para guardar la configuración, se provoca carga adicional al hardware al momento de operar y se requieren configuraciones de seguridad adicionales
- La herramienta gráfica de administración está limitada y ciertas configuraciones más específicas aún tienen que hacerse editando los archivos de configuración

## **Principales características, ventajas y desventajas de JQuery**

### **Introducción**

Como se comentó en el punto anterior, los sitios Web corporativos son una herramienta que ha demostrado ser estratégica y de gran importancia para las empresas por la forma en que ha evolucionado la forma de interactuar con sus consumidores.

Parte del éxito y la popularidad que han logrado los sitios Web se debe a la vistosidad, el nivel de interacción y la facilidad de uso de los que se les ha podido dotar gracias a lenguajes de programación como Javascript, el cual está basado en acciones, el objetivo principal de Javascript es proporcionar funciones que respondan a movimientos del mouse, aperturas, uso de teclas, cargas de páginas, etc.

Javascript nació de la necesidad de los desarrolladores de crear páginas Web más interactivas, el lenguaje HTML solo permitía crear páginas estáticas donde se podían mostrar textos con estilos, pero que no permitían la interacción con los usuarios.

Javascript no requiere compilación ya que se ejecuta del lado del cliente, el navegador de Internet donde se carga la página programada en este lenguaje es el encargado de interpretar su código, es por esto que es muy fácil incorporarlo en cualquier página Web sin necesidad de instalar ningún otro programa para poder visualizar el contenido que despliega e interactuar con la página mediante las funciones que implementa. [34]

Una vez situados en el contexto de que Javascript ha venido a ser una de las herramientas más utilizada por los desarrolladores de sitios Web para hacerlos mucho más dinámicos, permitir modificaciones de forma automática desde el navegador del usuario y la ejecución de tareas muy importantes como la validación de formularios, podemos hablar de una de las librerías más importantes que conforman el lenguaje Javascript: JQuery.

### **Características**

- JQuery es una librería gratuita de Javascript creada con el objetivo de hacer más simple el desarrollo de páginas Web interactivas, funciona en todos los

navegadores de Internet actuales; permite manejar eventos, desarrollar animaciones y agregar interacción en dichas páginas con la técnica AJAX (Asynchronous Javascript + XML). Actualmente es la librería de Javascript más utilizada.

- JQuery es software libre y de código abierto y al igual que otras librerías, ofrece una serie de funcionalidades basadas en Javascript que de otra forma requerirían de mucho más código, dicho de otra forma, con las funciones propias de esta librería se logran resultados excelentes en menos tiempo y con menos programación. Esta librería consiste en un único archivo Javascript que contiene las funcionalidades comunes de DOM (Document Object Model), eventos, efectos y AJAX; su característica principal es que hace posible cambiar el contenido de una página Web sin necesidad de volverla a cargar en el navegador, esto lo hace manipulando el árbol DOM y peticiones AJAX.

## **Ventajas**

Las ventajas del uso de la librería JQuery de Javascript para el desarrollo de un sitio Web son

- Es un producto estable y bien documentado
- Requiere de mucho menos líneas de código para lograr resultados vistosos y funcionales en la páginas Web
- Permite implementar muchas más funciones en comparación con otras librerías de Javascript.
- El archivo que contiene la librería JQuery ocupa alrededor de 60 Kb, bastante compacto, por lo que no retrasará la carga de la página Web (si el servidor envía los datos comprimidos, el tamaño de JQuery se reduce a unos 20 Kb). Este archivo se carga una sola vez, para siguientes páginas, el navegador ya lo tiene disponible en caché; lo cual agiliza enormemente la navegación en el sitio Web desarrollado utilizando esta librería.

- Al ser un producto de software libre, existe un gran equipo de colaboradores a cargo de la mejora y actualización de la librería.
- Además del equipo de desarrolladores de la librería, existen otros colaboradores que crean plugins o componentes, por lo que se facilita bastante encontrar soluciones ya creadas en JQuery listas para implementar en las páginas Web, soluciones tales como interfaces de usuario, encuestas, votaciones, efectos, etc.
- Su excelente integración con AJAX, logrando interfaces más elegantes, donde las acciones se ejecutan por partes sin necesidad de cargar nuevamente toda la página, lo cual agiliza enormemente la navegación

## **Desventajas**

Las desventajas de utilizar la librería JQuery de Javascript para el desarrollo de un sitio Web son

- La funcionalidad puede ser limitada. JQuery cuenta con una gran cantidad de funciones que puede implementar, pero resultan difíciles de personalizar; su aspecto visual es estandarizado y no se integra con el de la plataforma. Por lo anterior, hay casos en los que se debe recurrir a JavaScript simple para adaptar la vista y la funcionalidad a las necesidades específicas del usuario
- El manejo que JQuery hace de las CSS (Cascading Style Sheets) suele resultar innecesariamente complejo, es complicado para los desarrolladores determinar que clases utilizar
- Existen múltiples versiones de JQuery, algunas de estas versiones trabajan bien entre ellas y algunas otras no. Por ejemplo, la compatibilidad de los navegadores con las animaciones implementadas con JQuery ha sido un problema recurrente, en este caso específico, es necesario actualizar a la versión más reciente de JQuery en el servidor
- Lentitud en las animaciones. JQuery puede hacer lenta la página Web al implementar animaciones debido a las múltiples manipulaciones que hace de DOM.

## **2.3 Características, ventajas y desventajas del manejador de bases de datos Oracle**

### **Introducción**

Hasta ahora se ha mostrado la importancia que tienen los sitios Web corporativos para las empresas, para su estrategia de negocio. Ningún sitio Web corporativo sería funcional si no contara con el respaldo de un repositorio de datos confiable. [6]

Las bases de datos son fundamentales para el desarrollo de un sitio Web corporativo, ya que proporcionan la infraestructura necesaria para hacer de dicho sitio una herramienta interactiva, donde pueda tener acceso a información acerca de productos, precios, disponibilidad, levantamiento y seguimiento de pedidos, configuración o personalización de productos, etc.

Ninguna de las opciones anteriores podría ser implementada en el sitio Web sin un repositorio confiable de datos y por supuesto un manejador de dicha base de datos.

### **Características**

- Oracle es un sistema manejador de bases de datos Objeto relacionales (ORDBMS, Object Relational Data Base Management System), desarrollado por Oracle Corporation.
- Se considera a Oracle como uno de los manejadores de bases de datos más robusto y completo por su excelente rendimiento en transacciones, su estabilidad, su escalabilidad y su soporte para ser instalado en múltiples plataformas. Ha sido diseñado para poder controlar y gestionar grandes volúmenes de contenidos no estructurados en un único repositorio con el objetivo de reducir los costos y los riesgos asociados a la pérdida de información. [7]
- Además de lo anterior, aporta reducciones en los costos de inactividad al implementar una arquitectura de máxima disponibilidad, de igual forma, permite

un cambio y adaptación de las aplicaciones mediante su característica Real Application Testing.

- Actualmente Oracle es el líder a nivel mundial con alrededor del 47% del mercado de las bases de datos; ostenta el record mundial de transacciones en pruebas de desempeño realizadas en entornos operativos Windows, UNIX y Linux, sobre este último sistema operativo, presenta el costo por transacción más bajo de la industria.

## **Ventajas**

Las principales ventajas de utilizar a Oracle como manejador de bases de datos son

- Excelente desempeño en el entorno cliente-servidor
- Excelente capacidad para la gestión de enormes bases de datos
- Excelente rendimiento en transacciones
- Manejo de un gran número de usuarios concurrentes
- Su arquitectura de alta disponibilidad
- Disponibilidad controlada de los datos por parte de las aplicaciones
- Adaptación a estándares de la industria
- Buena gestión de la seguridad
- Autogestión de la integridad de los datos
- Su portabilidad
- Su compatibilidad
- Sus características de compresión y particionamiento de tablas permiten administrar más datos a un menor costo
- Protege los datos de forma segura y facilita el cumplimiento de normas con la recuperación total de datos
- Permite integrar contenidos de medios e incluso objetos 3D spatial en los procesos de negocio

## **Desventajas**

Las principales desventajas de utilizar a Oracle como manejador de bases de datos son

- Su elevado costo de implementación. Si bien se mencionó que sus costos de operación son bajos comparativamente con otros productos, el costo del licenciamiento, servicios de instalación, configuración y su posterior mantenimiento, son bastante elevados
- Al ser un producto propietario, los parches para problemas o vulnerabilidades detectados no se liberan con la rapidez que en ocasiones se requiere
- Especialistas en seguridad han criticado a la empresa por su política de suministro de parches de vulnerabilidades, hay algunas de estas que duraron alrededor de dos años sin ser atendidas
- Si no está correctamente configurado, puede tener un pobre desempeño y ser excesivamente lento
- El costo de la capacitación para llevar a cabo la administración de Oracle, es elevado también

## **2.4 Características, ventajas y desventajas JavaScript, HTML, CSS y JSP.**

**HTML** (HyperText Markup Language)

### **Introducción**

Es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Primero surgió como SGML (Standard Generalized Markup Language) que era un sistema que permitía ordenar y etiquetar diversos documentos de una lista, de ahí se vio en la necesidad de desarrollar una descripción sobre los contenidos que aparecen como texto, título, animaciones y fotografías; denominado lenguaje de etiquetas, un listado de elementos que usualmente se utilizan en pares y pueden ser anidados. [3]

Los estándares oficiales HTML son el HTML 2.0, el HTML 3.2, el HTML 4.0 y el HTML 4.01 y la versión más actual la HTML 5.

Es importante mencionar que HTML permite códigos llamados “scripts”, que brindan instrucciones específicas a los navegadores, los más conocidos son JavaScript y PHP.

### **Características**

- Todo lenguaje de marcado de texto tiene una gramática que nos indica qué etiquetas se pueden utilizar, qué significan, cómo se relacionan entre ellas, etc. En HTML la gramática marca que un párrafo debe comenzar con la etiqueta <p> y finalizar con el cierre de la misma etiqueta (</p>).
- Divide la información en dos bloques: la cabecera (head) y el cuerpo (body).

### **Ventajas**

- Fácil de usar y es software independiente.
- El código es más simple lo que permite hacer páginas más ligeras que se cargan más rápidamente en el navegador.
- Ofrece una compatibilidad mayor con los navegadores de dispositivos móviles.
- Posibilita la inserción de vídeos y audio de forma directa.
- Tiene la capacidad de ejecutar páginas sin estar conectado.
- Incorpora nuevas capacidades Javascript que aumentan la capacidad de almacenamiento. Frente a las cookies que dejaban almacenar algunos kilobytes, ahora se puede conseguir el almacenamiento de entre 5 y 10 megas, dependiendo de la plataforma. Además, se permiten múltiples Javascripts corriendo en paralelo en una misma página.
- Permite realizar diseños adaptables a distintos dispositivos (web, tablets, móviles, etc)

### **Desventajas**

- No ofrece mucha variedad de opciones, es muy básico.
- El contenido no puede ser reconocido ni procesado por programas.
- Solo a partir de la versión 5 permite la integración y manejo de bases de datos

## JavaScript

### Introducción

Es un lenguaje de programación orientado a objetos, como el lenguaje C y C++ donde las páginas de internet inician a desarrollarse con movimiento, siendo páginas dinámicas. Es un lenguaje pequeño y liviano. [2]

Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control sobre ellos. El navegador web en ser el primero en emplear JavaScript fue Netscape.

JavaScript es un lenguaje que utiliza Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, por ejemplo; escribe funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. [4]

Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript.

Un programa Java corre en cualquier computadora que tenga instalado **JRE** (Java Runtime Environment), el cual funciona en cualquier plataforma con sistema operativo: Windows, Macintosh, Unix o Linux y algunos celulares.

La plataforma Java no se basa en compiladores, lo que emplea es el concepto de "máquina virtual" o **JVM** (Java Virtual Machine). JRE es un programa que divide el disco duro como una computadora llamada JVM, que puede ejecutar programas Java.

El compilador Java traduce el programa java al lenguaje de bajo nivel dentro de la JVM, llamado bytecode, después JRE ejecuta el bytecode dentro de JVM. El lenguaje Java maneja librerías de clases con todas las funciones llamada Java API.

### Características

- Es muy simple para entender su lenguaje de programación.
- Maneja objetos dentro de una página Web y sobre ese objeto podemos definir diferentes eventos.
- Es dinámico ya que responde a eventos en tiempo real.

- Las secuencias de comandos de JavaScript sólo pueden ejecutarse con un intérprete, que bien puede estar en un servidor Web o en un explorador de Web.
- En Java no declaramos tipos de datos, si vamos a usar una cadena e insertamos un número, java lo convierte a texto. [14]

### **Ventajas**

- Es una plataforma independiente.
- JavaScript es una excelente solución para poner en práctica la validación de datos de un formulario en el lado del cliente.
- Se pueden crear efectos dinámicos tales como imágenes dinámicas y presentaciones de diapositivas, utilizando poca memoria y manteniendo un tiempo de descarga rápido para la página web.

### **Desventajas**

- Seguridad. Cuando el usuario ejecuta alguna aplicación con java desde alguna página web, se descarga la aplicación en el navegador del cliente permitiendo así que cierto código malicioso pueda ser ejecutado en la máquina del cliente.
- Otra desventaja de JavaScript es que este tiende a introducir una cantidad enorme de fragmentos de código en nuestros sitios web. [11]
- Los usuarios pueden desactivar JavaScript en su navegador.

## **CSS (Cascading Style Sheets – Hoja de Estilo en Cascada)**

### **Introducción**

Es un lenguaje utilizado en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente “una página web”. Es aquí donde el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. [15]

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos.

CSS funciona para cada navegador, Google Chrome, Internet Explorer, Firefox, Safari y Opera.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne. El selector funciona como enlace entre el documento y el estilo, especificando los elementos que van a ver afectados por esa declaración. La declaración es la parte de la regla que establece cuál será el efecto. [16]

### **Características**

- Es muy flexible
- Un archivo CSS puede contener un número ilimitado de reglas CSS, cada regla se puede aplicar a varios selectores diferentes y cada declaración puede incluir tantos pares propiedad/valor como se desee.
- Una de las características más importantes de las hojas de estilos CSS es que permiten definir diferentes estilos para diferentes medios o dispositivos: pantallas, impresoras, móviles, proyectores.
- CSS permite incluir comentarios entre sus reglas y estilos. Los comentarios son contenidos de texto que el diseñador incluye en el archivo CSS para su propia información y utilidad.

### **Ventajas**

- lograr una apariencia uniforme de todo el sitio al activar una sola definición de estilo en cada página.
- cambiar un aspecto en todo el sitio Web con tan sólo editar unas pocas líneas.
- hacer que los códigos HTML sean más fáciles de leer ya que los estilos se definen por separado.

- permitir que las páginas se carguen más rápido ya que hay menos cantidad de HTML en cada página.
- Pueden usarse con otros lenguajes de programación (como JavaScript) para conseguir efectos dinámicos en las páginas.

### **Desventajas**

- Que Ciertas propiedades que funcionan en un navegador no funcionan en otro, o existen diferencias en un mismo navegador según sea para Windows o Mac. También existen diferencias entre distintas versiones de un mismo browser.
- Si nuestra intención es que el usuario vea una página dinámica y bien estructurada, si el navegador no es compatible, el usuario ya verá de manera diferente la página.

## **JSP (Java Server Pages – Páginas de Servidor Java)**

### **Introducción**

Es una tecnología orientada a crear páginas web con programación en Java. Se crean aplicaciones Web para ser ejecutadas en varios servidores web de múltiples plataformas.

Las paginas JSP están compuestas de código HTML/**XML** (eXtensible Markup Language - lenguaje de marcas extensible) mezclado con etiquetas para programar scripts java. [5]

Para desplegar y correr JavaServer Pages, se requiere un servidor web compatible con contenedores servlet como Apache Tomcat o Jetty.

### **Características**

- Conjunta el poder de Java en el servidor y la flexibilidad de HTML en el browser.
- Es más fácil y rápido crear aplicaciones web.

- JSP permite apreciar mejor la distinción entre el contenido de la información y su presentación.

### **Ventajas**

- Es código abierto.
- Hereda la portabilidad Java. Los JSP's y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él.
- Una de las principales ventajas de JSP sobre JavaScript es la capacidad de comunicación dentro de un servidor; por lo cual se pueden mezclar para validar formularios que han de ser enviados al servidor.
- Independencia del servidor Web.

### **Desventajas**

- La aplicación no está disponible si ocurre algún problema con el servidor o con la conexión de red.
- La carga de la interfaz es más lenta que la de una aplicación de escritorio.
- La mayor parte de la lógica de la aplicación se ejecuta en el servidor, por lo que se corre el riesgo de sobrecargar de trabajo al mismo.

### **2.5 Características, ventajas y desventajas de la arquitectura cliente-servidor.**

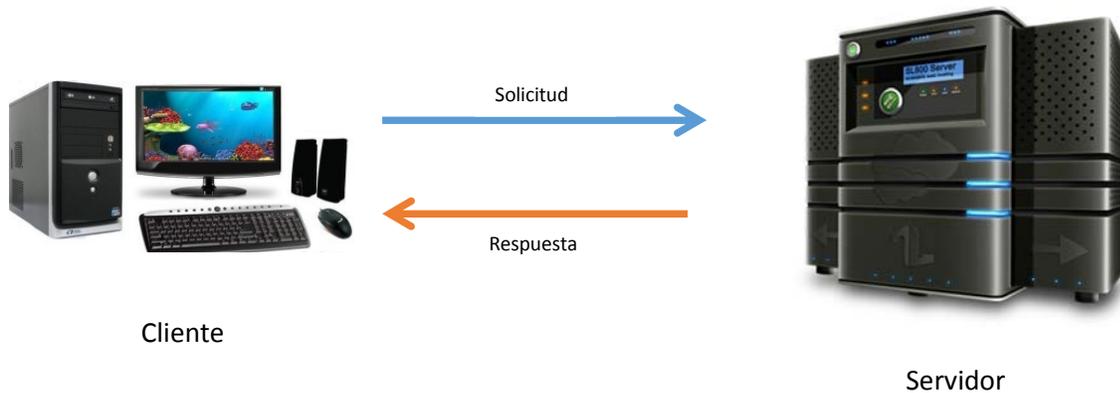
Es un modelo de aplicación en donde las tareas solicitadas son repartidas entre los servidores (que cuentan con un servicio) y los clientes (que solicitan un servicio). [18]

La arquitectura de una capa es aquella que tanto la aplicación como los datos que maneja se encuentran en la misma máquina y son administrados por la misma herramienta. Vea figura 2.5.1



**Figura 2.5.1 Arquitectura de una capa**

Algunos ejemplos del modelo cliente-servidor de dos capas requiere de dos equipos separados físicamente. Esta arquitectura es conocida como arquitectura de 2 niveles o 2 capas. Vea Figura 2.5.2



**Figura 2.5.2 Arquitectura de dos capas**

### **Características**

- El cliente tiene un papel activo en la comunicación.
- Espera y recibe las respuestas del servidor.
- Comúnmente se puede conectar a varios servidores a la vez.
- Interactúa directamente con los usuarios finales mediante una interfaz gráfica.
- El servidor espera a que lleguen las solicitudes, es el pasivo.
- El servidor cuando recibe la petición de un cliente, la procesa y le envía la respuesta al cliente.

- Cada plataforma puede ser escalable independiente sin que el usuario lo note.[19]

### **Ventajas**

- Facilita la integración entre sistemas diferentes y comparte información.
- Utiliza interfaces gráficas para el usuario.
- Facilita la integración de nuevas tecnologías, permitiendo que sea escalable.
- Seguridad mejorada.
- Administración al nivel del servidor.
- Incluye múltiples plataformas, bases de datos, redes y sistemas operativos.

### **Desventajas**

- Es necesario contar con estrategias para el manejo de errores.
- El desempeño del servidor puede degradarse debido al tráfico en la red.
- Costo de servidor elevado.

La arquitectura en 3 niveles o 3 capas es cuando el cliente solicita los recursos al servidor de aplicaciones, cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo, denominado servidor de datos, el cual proporciona al servidor de aplicaciones los datos que requiere. [20]

Una arquitectura en 3 niveles, permite un mayor grado de flexibilidad y de mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel. Mejora el rendimiento, ya que las tareas se comparten entre servidores. [21] Vea Figura 2.5.3



**Figura 2.5.3 Arquitectura de tres capas**

La arquitectura de n-capas separa hacia fuera el proceso, eso ocurre para mejorar el balance de carga en los diversos servidores; es más escalable. Aunque eso pone más carga a la red, debido a una mayor cantidad de tráfico de la red, resultando más difícil programar y probar el software que en la arquitectura de tres niveles porque tienen que comunicarse más dispositivos para terminar la transacción de un usuario. Ver Figura 2.5.4



**Figura 2.5.4 Arquitectura de n capas**

Los tipos de clientes que existen para esta arquitectura cliente servidor son.

### **Servidor Flaco - Cliente Flaco**

El cliente liviano es un PC con menos capacidades. Al diseñar un sistema informático, hay decisiones que tomar acerca del procesamiento, el almacenamiento, el software y la interfaz de usuario.

Un cliente ligero (thin client o slim client) es una computadora cliente o un software de cliente en una arquitectura de red cliente-servidor que depende primariamente del servidor central para las tareas de procesamiento, y principalmente se enfoca en transportar la entrada y la salida entre el usuario y el servidor remoto. [22]

Muchos dispositivos de cliente liviano ejecutan solamente navegadores web o programas de escritorio remoto, lo que significaba que todo el procesamiento significativo ocurría en el servidor. Algunas de las ventajas que tenemos son:

- El servidor siempre estará saturado.
- Por la red siempre habrá un gran flujo de datos.
- Menos hardware desperdiciado. Los clientes livianos pueden permanecer en servicio por más tiempo y producen menos hardware excedente que una instalación equivalente de cliente pesado.
- Menores costos de administrativos de IT. Los clientes ligeros son manejados casi enteramente en el servidor. El hardware tiene menos lugares donde puede fallar, el entorno local es altamente restringido, y el cliente es más simple y a menudo carece de almacenamiento permanente, proporcionando protección contra el malware.
- Más fácil de asegurar. Los clientes livianos pueden ser diseñados de modo que ni siquiera los datos de aplicación residan en el cliente (apenas son exhibidos en la pantalla), centralizando la protección contra el malware y reduciendo los riesgos de hurto de los datos físicos.
- Seguridad de datos mejorada. Si un dispositivo del cliente ligero sufre una seria desgracia o accidente de trabajo, no se perderá ningún dato, puesto que residen en el servidor de terminales y no en el dispositivo de punto de operación.

## **Servidor Gordo - Cliente Gordo**

Un servidor de cliente pesado no requiere tan alto nivel de desempeño como un servidor de cliente liviano (puesto que los clientes pesados por sí mismos hacen mucho del procesamiento de la aplicación). Esto puede resultar en servidores más baratos, aunque en la práctica, muchos servidores de cliente liviano son en realidad equivalentes a los servidores de archivos en cuanto a las especificaciones pero con memoria adicional. Los clientes pesados tienen ventajas en aplicaciones ricas en multimedia que serían intensivas en ancho de banda si estuvieran completamente residentes en los servidores. [23]

- Casi todo el trabajo radica en el cliente.
- Aumenta la circulación de datos que no se requieren a través de la red.

El desarrollo del módulo de autoservicio para selección de equipos y planes tarifarios para telefonía móvil, está implementado usando la arquitectura cliente – servidor de 3 capas.[24]

# **CAPÍTULO**

## **III**



## Capítulo 3 Desglose del sistema

### 3.1 Análisis del problema

Actualmente para poder comprar un equipo de telefonía celular solo es posible asistiendo a la sucursal con las implicaciones de tiempo y dinero asociadas, como el traslado ya sea en transporte público o auto particular, así mismo es necesario tener en cuenta el tiempo que va a tomar el estar revisando los equipos de interés. Para ello es necesario esperar turno para solicitar la ayuda del vendedor, esperando no esté pasando por un mal día para que atienda amablemente las preguntas que se le hagan respecto a las características de los teléfonos seleccionados y no desespere por estar mostrándolos, dado que la decisión de elegir un equipo no es tarea sencilla, es importante poner atención entre todas las características que mencione el vendedor de acuerdo a cada celular seleccionado. Posteriormente se hace necesario el elegir un plan en el que se venda el equipo y que sea acorde a las necesidades y posibilidades del cliente, por lo general son muy variados en cuanto a rentas y servicios asociados, en la figura 3.1.1 se esquematiza lo anterior.

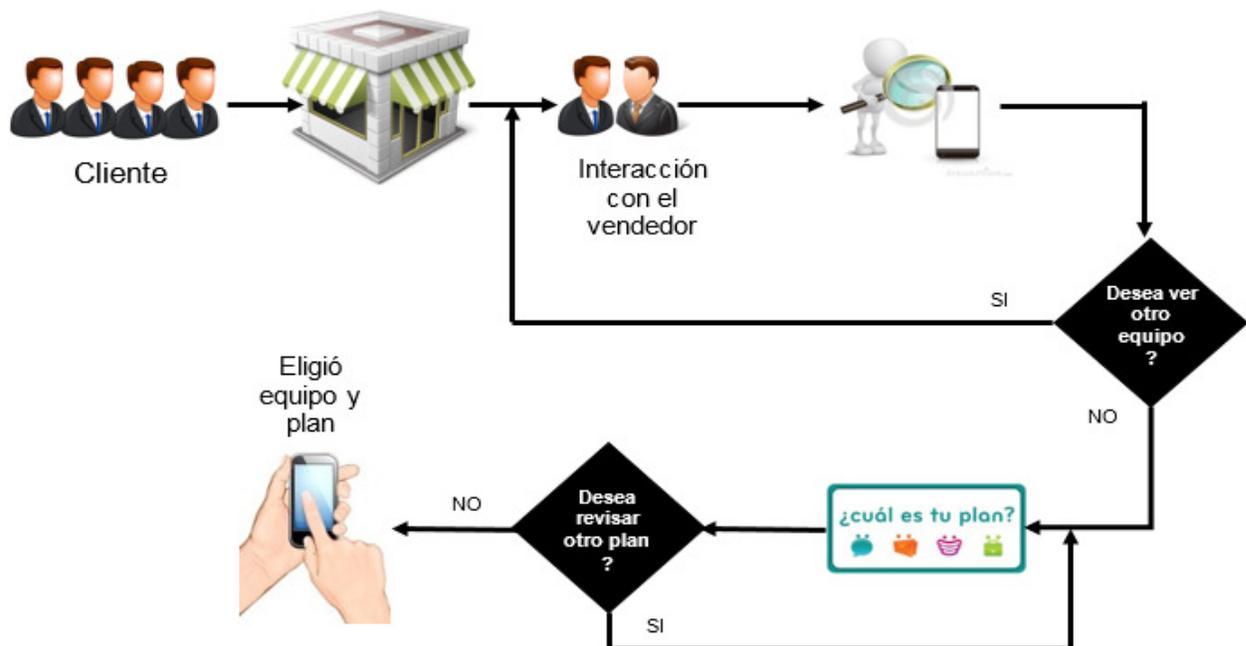


Figura 3.1.1 Esquema para elección de equipos

Dado que el uso de equipos de telefonía celular ha crecido exponencialmente, es necesario considerar la automatización de los procesos de elección, compra y distribución de equipos de telefonía celular, para de esta manera tener acceso a una mayor cantidad de personas, por lo que es importante que el proceso sea más ágil y sencillo. Buscando que el cliente no tenga que salir de su casa o trabajo, sino que pueda ingresar a internet, evitando la interacción con el vendedor y de manera intuitiva, amigable y ágil un cliente potencial pueda realizar la elección de un celular y un plan acorde a sus necesidades y presupuesto, por medio de herramientas como filtros por marca o modelo, contando con consultas que permiten visualizar las características de los equipos en forma ordenada, práctica y lo más importante que la interacción será expedita.

Adicionalmente será posible mostrar los planes en los que los teléfonos están disponibles, sin necesidad de perder tiempo en esperar a ser atendidos por un vendedor ni tener que soportar algún maltrato de su parte y crear un autoservicio de selección de teléfonos y planes con tan sólo hacer clicks en los que sean de interés.

En resumen un sitio web permitirá observar las imágenes de los productos (Equipos de telefonía celular y planes tarifarios) y leer sus especificaciones. Este servicio le proporcionará al cliente rapidez en la selección de un celular, con la posibilidad de hacerlo desde cualquier lugar y a cualquier hora. En la figura 3.1.2 se esquematiza este proceso.



**Figura 3.1.2 Esquema de solución propuesta**

### 3.2 Requerimientos particulares y generales del usuario

Se requiere que el sistema funcione bajo el sistema operativo windows xp y en los navegadores Internet explorer 10 y superiores, Google Chrome y Firefox.

Se requiere que el “Home” principal de las tiendas en línea, muestre las siguientes secciones:

- Consulta de:
  - Equipos disponibles para su venta, listado de equipos (Panel derecho) de acuerdo a los filtros de búsqueda. Al inicio se debe mostrar todos los equipos disponibles en la tienda. Cada opción que se tenga en el home debe tener una conexión al detalle del producto seleccionado, es decir:

- Características e imagen del equipo.
  - Especificaciones del equipo, imagen de especificaciones.
- Consulta detalle de equipo, esta opción debe tener liga de acceso a la consulta de las características del equipo.
  - Planes disponibles para su venta. Cada opción debe mostrar los equipos disponibles que puede comprar el cliente.
  - Accesorios disponibles para su venta. Cada opción debe mostrar los equipos disponibles que puede comprar el cliente.
  - Búsqueda: En esta sección el cliente puede realizar la búsqueda de equipos, planes, accesorios, precios, características, marcas y todo lo que se encuentre disponible en la tienda. Para ello el cliente debe ingresar palabras clave para que el motor de búsqueda localice los productos relacionados al dato de búsqueda proporcionado por el usuario.
  - Una opción de atención al cliente. Esta opción debe servir al cliente como ayuda para resolver dudas en línea, atención personalizada vía telefónica y todo lo referente a equipos, planes y accesorios.
  - Se debe contar con una sección donde se muestren los equipos y es necesario contar con filtros para acotar los resultados por marca, sistema operativo y precio.
  - Se debe contar con una herramienta que permita realizar un zoom para mostrar los equipos en tres distintos tamaños, lo cual permitirá visualizar por un lado una menor o mayor cantidad y por otro con menor o mayor detalle de imagen.
  - Se requiere una herramienta que permita visualizar un resumen de información del equipo, mostrando sus características más importantes, planes que sean más vendidos incluyendo nombre y precio del plan, marca, modelo y debe contar con un link para ver el detalle del equipo.

- Cuerpo
  - Un banner. Este debe mostrar las promociones, equipos nuevos.
  - Sección de los equipos más vendidos, donde se muestran los equipos más vendidos. Por medio de la cuál puede pasar al detalle del equipo al hacer click sobre este.
- Para contratar un plan, el sistema debe mostrar lo siguiente:
  - Breve descripción de las promesas del plan, precio minutos incluidos mensajes SMS, etc.
  - Opciones para consultar más información, preguntas frecuentes, comentarios de otros usuarios en relación al plan.
  - Opción para configurar el plan, en el cual se debe mostrar un listado de equipos que están disponibles para la venta tanto de equipo como de plan.
  - Cotización de plan, el sistema debe proporcionar al cliente una cotización de su compra, tanto del equipo como del plan que desea contratar, una vez que el cliente valide la mejor opción se debe permitir realizar la compra del plan.
  - La página muestra la información más completa del plan, debe contener:
    - Políticas de uso justo
    - Rentas del plan
    - Información de los beneficios de cada renta
    - Opciones para elegir servicios adicionales
    - Opción para comprar el plan
  - Debe contar con un carrito de compras. Esta sección debe indicar la cantidad de artículos que el cliente ha seleccionado para comprar. Y debe considerar los flujos siguientes:
  - Sucede cuando el cliente selecciona desde la pantalla para realizar el pago de un producto la opción para seguir comprando.
  - La página muestra la pantalla principal de los equipos.
  - El cliente elije un producto y selecciona la opción para comprar.

- La página actualiza el número de producto almacenados en el carrito de compras. Termina el flujo alterno
  - Este flujo ocurre cuando el cliente selecciona desde el carrito de compras la opción para borrar un producto.
  - página muestra un mensaje al cliente: “¿Esta seguro eliminar el producto del carrito?”.
  - El cliente confirma que desea borrar el producto del carrito.
  - La página muestra la pantalla principal del carrito de compras. Termina el flujo alterno
- Pie de página
    - Legales, como el aviso de privacidad de uso de datos y políticas de la empresa.
    - Teléfonos de contacto para atención a clientes.
    - Redes sociales
      - Facebook, se debe tener una conexión a la red social de la empresa.
      - Twitter, se debe tener una conexión a la red social de la empresa.

### **3.3 Posicionamiento del software sugerido contra productos similares.**

Para que funcione nuestro sistema emplearemos un servidor de aplicaciones, siendo el software Apache el que nos brindará este servicio. Nos referiremos a un servidor de aplicaciones como un sistema que incluye aplicaciones Web, conectores de servidores Web, lenguajes de programas, librerías, conectores de bases de datos y el código de administración necesarios para implementar, configurar , administrar y conectar estos componentes en un proveedor de alojamiento web. El servidor Web es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y unidireccionales.

Un servidor de aplicaciones se ejecuta en un servidor web (por ejemplo Apache, Cherokee o Nginx) y la mayoría de las veces con una base de datos SQL (por ejemplo, Oracle, Sybase e Informix).

A continuación se explica una breve introducción del software Apache, y otros dos software de nombre Cherokee y Nginx.

- Apache es capaz de suministrar un servidor seguro, eficiente y extensible que brinde servicios de HTTP en sincronía con los estándares HTTP actuales. Funciona para plataformas: Unix (BSD, GNU/Linux, etc.), Microsoft Windows y Macintosh y la noción de sitio virtual. Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la WWW (World Wide Web). Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. Apache es software libre. [25] Colocándolo en una de las posiciones más populares. Ver figura 3.3.1 [26]



**Figura 3.3.1 Logo Apache**

- El siguiente servidor de aplicaciones es Cherokee, es un servidor web multiplataforma, de alto rendimiento, su objetivo es ser rápido, flexible, y completamente funcional, sin dejar de ser liviano comparado con otros servidores web. [27] Está escrito completamente en C. Puede usarse como un sistema embebido y soporta complementos para aumentar sus funcionalidades. Es software libre, disponible bajo la Licencia Pública General de GNU. Ofrece

soporte para las tecnologías difundidas hoy en día : FastCGI , SCGI , PHP , CGI, SSI, TLS y conexiones SSL cifrada , hosts virtuales, autenticación en la codificación de marcha, equilibrio de carga, los archivos de registro compatibles Apache , Data Base Balancer , tiempo de inactividad gratuita, actualizaciones y mejoras , Reverse Proxy HTTP , y mucho más. [28] Ver figura 3.3.2



**Figura 3.3.2 Logo Cherokee**

- Como tercer rival de apache, mencionaremos al servidor de aplicaciones Nginx, este servidor que fue creado originalmente para poder soportar hasta 500 millones de peticiones simultáneas, actualmente, entre los sites que se han decantado por este servidor web destaca Wordpress, SourceForge, Hulu, Ultimate Guitar, Scribd o Badoo. [29] Originalmente, nginx fue desarrollado para satisfacer las necesidades de varios sitios web de Rambler que recibían unas 500 millones de peticiones al día en septiembre de 2008. De acuerdo con el estudio de Netcraft, Netcraft's Jul 2014 Web Server Survey, nginx es el segundo servidor web más usado en dominios activos (14,35%) superando a Microsoft Information Server. Ver figura 3.3.3



**Figura 3.3.3 Logo Nginx**

A continuación se muestra la tabla comparativa 3.3.4

	APACHE	CHEROKEE	NGINX
Características	<ul style="list-style-type: none"> <li>• Es multiplataforma</li> <li>• Es un servidor Web conforme al protocolo HTTP/1.1</li> <li>• Puede ser adaptado a diferentes entornos y necesidades.</li> <li>• Es software libre</li> </ul>	<ul style="list-style-type: none"> <li>• Es multiplataforma.</li> <li>• Está desarrollado en C.</li> <li>• Es software libre</li> <li>• Soporta complementos para aumentar sus funcionalidades.</li> </ul>	<ul style="list-style-type: none"> <li>• Es software libre</li> <li>• Es multiplataforma</li> <li>• Soporta HTTP</li> <li>• Servidores virtuales</li> </ul>
Ventajas	<ul style="list-style-type: none"> <li>• Trabaja con gran cantidad de lenguajes de desarrollo como Perl, PHP y otros lenguajes de script, Java y páginas JSP, teniendo todo el soporte que se necesita para tener páginas dinámicas.</li> <li>• Por su diseño modular es muy sencillo ampliar las capacidades del servidor web Apache</li> <li>• Se puede personalizar la respuesta ante los posibles errores que se den en el servidor.</li> <li>• Permite la creación de ficheros de log a medida del administrador, teniendo así el control máximo sobre lo que pasa en el servidor.</li> </ul>	<ul style="list-style-type: none"> <li>• Permite desplegar aplicaciones web de manera óptima.</li> <li>• Es posible desarrollar interfaces sencillas.</li> <li>• En su segundo enfoque proporciona un framework con el que dota de funcionalidades web a las aplicaciones de una forma rápida y sencilla, de forma que cualquier aplicación pueda incorporar servicios Web.</li> </ul>	<ul style="list-style-type: none"> <li>• Delega la parte de PHP haciéndolo más liviano y rápido en si mismo</li> <li>• Solo tiene funciones básicas de rewrite y ssl por lo que toda la configuración incluso la de las máquinas virtuales se puede hacer en poquito más de 50 líneas</li> </ul>

Desventajas	<ul style="list-style-type: none"> <li>• Falta de integración.</li> <li>• Posee formatos de configuración no estándar.</li> <li>• No posee buena administración.</li> </ul>	<ul style="list-style-type: none"> <li>• Como aún sigue en desarrollo tiene funcionalidades no terminadas.</li> <li>• Necesita módulos para servir contenidos dinámicos: PHP y Python.</li> </ul>	<ul style="list-style-type: none"> <li>• Es lento en el cuello de botella al mandar por red los datos php y esperar la respuesta</li> <li>• el costo de seguridad es alto</li> <li>• La velocidad ganada se pierde en el cuello de botella de la sesión de red</li> <li>• Nginx no cuenta prácticamente con ningún modulo.</li> </ul>
-------------	---	---	---

**Tabla 3.3.4 Comparativo servidores de aplicaciones**

Como se observa en las tablas, se determinó emplear APACHE porque se integra perfectamente con todas las aplicaciones que serán requeridas en nuestro sistema. Es seguro y es software libre.

Analizaremos el software para nuestra base de datos.

El sistema de administración de base de datos es el más crítico de los componentes, porque pocas de las aplicaciones pueden funcionar sin el acceso a las bases de datos implicadas. Cuando un servidor falla, las aplicaciones se caen, y los usuarios finales se quedan ociosos al perder el acceso a sus aplicaciones, lo cual puede costar miles e incluso millones de dólares! en pérdidas de negocios cada hora. Algunos de los costos asociados al tiempo muerto u ocioso de una base de datos, son: el costo de la caída de la imagen de la marca, el valor de las acciones y la baja en la eficiencia interna. Para evitar la caída de una base de datos, es necesario tener disponibilidad continua, en donde una base de datos nunca falla.

Pero la mayoría de las compañías no necesitan un nivel de disponibilidad tan estricto; para ellas es suficiente con tener una alta disponibilidad, que les permita un periodo corto de caídas ya planeadas dentro de sus procesos de operación, tiempo que normalmente es utilizado para dar mantenimiento a sus sistemas. De cualquier forma cuando un desastre hace que un sistema sea irreparable, las estrategias de disponibilidad de bases de datos deben incluir un sistema de cómputo secundario en un sitio remoto que entre en acción cuando el sistema primario se daña debido a un evento catastrófico, como por ejemplo una inundación, un huracán, un tornado, un terremoto o fuego.

- Oracle, es un sistema de gestión de base de datos objeto-relacional (ORDBMS Object-Relational Data Base Management System), desarrollado por Oracle Corporation. Es una herramienta cliente/servidor, utilizada por empresas de muy gran tamaño, debido al costo. Para el desarrollo de páginas Web ocurre una situación similar, ya que no está al alcance de los usuarios como otras bases de datos. Una base de datos de Oracle es una colección de datos tratada como una unidad, su principal propósito es almacenar y recuperar información almacenada. Una instancia Oracle consta de una estructura de memoria, llamada Area Global del sistema (SGA) y de unos procesos background utilizados por el servidor Oracle para manejar una base de datos. Ver figura 3.3.5



**Figura 3.3.5 Logo Oracle**

- Sybase, La vida de Sybase (servidor empresarial adaptable), comenzó como Sybase SQL Server, el primer sistema en manejo de bases de datos relacionales (relational database management system RDBMS), creado y vendido por Sybase. Sybase se ocupó en maximizar el desarrollo del producto y

confiabilidad para correr en HIGH END; mientras que Microsoft puso más énfasis en hacerlo “fácil de usar y windows ambiental”. [30]

Se convierte en la primera con bases de datos móviles, más de un 90% de satisfacción al cliente, la más larga librería de almacenamiento de datos, una de las aplicaciones más grandes en el mundo (AVANTGO) con más de 10 millones de usuarios registrados. Ver figura 3.3.6

The image shows the Sybase logo, which consists of the word "SYBASE" in a large, bold, serif font. A registered trademark symbol (®) is located at the top right of the letter "E".

**Figura 3.3.6 Logo Sybase**

- Informix, es el único software de base de datos OLTP disponible que presenta facilidad de integración, reduce los costos y aumenta la productividad proveyendo de una administración casi nula, tanto para los entornos pequeños como para los altamente distribuidos sin importar cuál sea la plataforma. “Instálelo y olvídense” es una frase popular de los clientes de Informix, quienes dan testimonio de la facilidad de uso y la confiabilidad legendaria de Informix. Informix ha sido diseñado desde el principio para una administración fácil y escalable, con muchas características que le permiten “desaparecer” en una aplicación. [31] Además de las características de monitoreo y desinfección automáticas integradas en Informix, éste también puede personalizarse para rastrear de manera proactiva los umbrales empresariales, reaccionar, y resolverlos antes de que se tornen un problema. Informix ofrece las opciones de duplicación y redundancia flexibles que permiten una continuidad empresarial integrada.

En caso de una falla del sistema, se pasa automáticamente a otro servidor sin que esto produzca repercusiones en su empresa. Ver figura 3.3.7



Figura 3.3.7 Logo Informix

La Tabla 3.3.8 muestra el comparativo entre los diferentes manejadores de base de datos

	ORACLE	SYBASE	INFORMIX
Características	<ul style="list-style-type: none"> <li>• motor de base de datos objeto-relacional</li> <li>• Es multiplataforma</li> <li>• Lenguaje de programación PL/SQL, PHP, Java, .NET, XM</li> <li>• Es software libre</li> <li>• Soporta cliente-servidor</li> <li>• Soporta bases de datos de todos los tamaños.</li> </ul>	<ul style="list-style-type: none"> <li>• Satisface las demandas extremas de rendimiento, eficiencia y disponibilidad de los sistemas de procesamiento de transacciones de próxima generación.</li> <li>• Simplificar la migración de datos de sistemas históricos a nuevas plataformas incluyendo el Web.</li> </ul>	<ul style="list-style-type: none"> <li>• Es multiplataforma</li> <li>• Lenguaje de programación C++</li> <li>• Es software libre</li> </ul>

Ventajas	<ul style="list-style-type: none"> <li>• Es un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.</li> <li>• Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.</li> <li>• Es considerado un gestor de base de datos muy completo.</li> </ul>	<ul style="list-style-type: none"> <li>• Rendimiento superior</li> <li>• Confiabilidad sólida</li> <li>• Menor costo total de la propiedad</li> <li>• Una base a largo plazo para la agilidad estratégica</li> <li>• Una ruta de innovación continua para los entornos esenciales</li> <li>• Tiene una de las aplicaciones móviles más grandes del mundo</li> <li>• Es número uno en bases de datos de carácter inalámbrico (Unwired).</li> </ul>	<ul style="list-style-type: none"> <li>• Proporciona funciones de desarrollo rápido y depuración interactiva.</li> <li>• Reduce los costos de la administración</li> <li>• Soporta requisitos de procesamiento de transacción online, complejos y rigurosos.</li> <li>• Optimiza capacidades de inteligencia del negocio competitivas.</li> </ul>
Desventajas	<ul style="list-style-type: none"> <li>• remodelación del sistema de almacenamiento por causa de la introducción de extensiones orientadas a objetos.</li> </ul>	<ul style="list-style-type: none"> <li>• No se especifica los puntos en las cuales sea no factible o bien sea débil la base de datos</li> <li>• El costo.</li> </ul>	<ul style="list-style-type: none"> <li>• el diferenciador más obvio es el costo de la licencia.</li> <li>• No tiene soporte para BLOBs (Objetos grandes binarios).</li> </ul>

	<ul style="list-style-type: none"> <li>• Su costo</li> <li>• Un Oracle mal configurado puede ser muy lento.</li> <li>• No hay mucha información sobre Oracle y es necesario pagar para configurarlo.</li> </ul>		<ul style="list-style-type: none"> <li>• No tiene soporte para el tipo de datos VARCHAR.</li> <li>• Son datos con una longitud fija de máximo 2000 caracteres.</li> <li>• No es recomendable utilizarlo con aplicaciones que exigen un gran rendimiento desde el punto de vista de la rapidez.</li> </ul>
--	---	--	---

**Tabla 3.3.8 Comparativo manejadores de base de datos**

Como se explica, Oracle es uno de los motores de base de datos relacional más utilizado a nivel mundial, se ejecuta en cualquier plataforma y como soporta todas las funciones, implementa diseños activos ofreciendo un ambiente de seguridad. Se decidió emplear Oracle por adaptarse a las necesidades del programa y porque se cuentan con las licencias.

Para el Front-End tenemos JSP, los cuales serán comparados con PHP y Rubi.

- JSP, es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor. JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat. Ver figura 3.3.9



## JSP Development

**Figura 3.3.9 Logo Java Server Pages**

- PHP, es un acrónimo recursivo que significa Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools) PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. PHP se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor. Ver figura 3.3.10 [32]



**Figura 3.3.10 Logo Php**

- Rubi, El lenguaje Ruby está diseñado para la productividad y la diversión del desarrollador, siguiendo los principios de una buena interfaz de usuario. Sostiene que el diseño de sistemas necesita enfatizar las necesidades humanas más que las de la máquina. Ver figura 3.3.10



**Figura 3.3.10 Logo Rubi**

La tabla 3.3.11 Muestra el comparativo de los lenguajes de desarrollo

	JSP	PHP	Rubi
Características	<ul style="list-style-type: none"> <li>• JSP utiliza etiquetas y operadores similares a la norma PHP y ASP idiomas, pero la sintaxis del código se basa en el lenguaje Java.</li> <li>• El lenguaje JSP crea servlets y ejecuta rápido los servlets.</li> <li>• Crear páginas del lado del servidor.</li> <li>• Multiplataforma.</li> <li>• Código bien estructurado.</li> <li>• Integridad con los módulos de Java.</li> <li>• La parte dinámica está escrita en Java.</li> </ul>	<ul style="list-style-type: none"> <li>• Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.</li> <li>• Es considerado un lenguaje fácil de aprender.</li> <li>• El código fuente escrito en PHP es invisible al navegador web y al cliente.</li> <li>• Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.</li> </ul>	<ul style="list-style-type: none"> <li>• Orientado a objetos</li> <li>• Existe diferencia entre mayúsculas y minúsculas.</li> <li>• Múltiples expresiones por líneas, separadas por punto y coma “,”</li> <li>• Dispone de manejo de excepciones.</li> <li>• Librerías de extensiones dinámicamente si el (Sistema Operativo) lo permite.</li> <li>• Dinámico</li> </ul>

	<ul style="list-style-type: none"> <li>• Permite la utilización de servlets.</li> </ul>	<ul style="list-style-type: none"> <li>• Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).</li> <li>• Posee una amplia documentación</li> <li>• Es software libre.</li> <li>• Permite aplicar técnicas de programación orientada a objetos.</li> </ul>	<ul style="list-style-type: none"> <li>• Entiende expresiones regulares</li> <li>• Multiplataforma</li> <li>• Recolector de basura inteligente</li> <li>• Sintaxis flexible</li> <li>• Sobre carga de operadores</li> <li>• Puede cargar librerías de extensiones dinámicamente si el (Sistema Operativo) lo permite.</li> <li>• Portátil</li> </ul>
Ventajas	<ul style="list-style-type: none"> <li>• Las páginas son más convenientes para los desarrolladores de Java , que no es una plataforma de Microsoft</li> <li>• le permite utilizar las mismas funciones que otras páginas, como la conexión a una base de datos , el análisis de XML y el uso de estilos CSS .</li> <li>• es también compatible con los servidores de Internet de Linux , por lo que puede</li> </ul>	<ul style="list-style-type: none"> <li>• Se necesita instalar un servidor web.</li> <li>• Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.</li> <li>• La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.</li> <li>• La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.</li> <li>• Dificulta la</li> </ul>	<ul style="list-style-type: none"> <li>• Permite desarrollar soluciones a bajo Costo.</li> <li>• Software libre.</li> <li>• Multiplataforma.</li> <li>• Bastante accesible</li> </ul>

	utilizarlo con los servicios Web Apache.	modularización. <ul style="list-style-type: none"> <li>• Dificulta la organización por capas de la aplicación.</li> </ul>	
Desventajas	<ul style="list-style-type: none"> <li>• los desarrolladores no crean activamente nuevas bibliotecas o add-ons para el idioma.</li> <li>• Si tiene que trabajar con las nuevas tecnologías, es posible que tenga problemas para encontrar las bibliotecas necesarias para el lenguaje.</li> </ul>	<ul style="list-style-type: none"> <li>• Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no necesariamente impide que el código sea examinado.</li> <li>• Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel, sin embargo este inconveniente se puede minimizar con técnicas de caché tanto en archivos como en memoria.</li> <li>• Las variables al no ser tipificadas dificulta a los diferentes IDEs para ofrecer asistencias</li> </ul>	<ul style="list-style-type: none"> <li>• No soporta polimorfismo de funciones (sobrecarga)</li> <li>• No es software fuerte</li> <li>• Interpretado = Lento</li> </ul>

		para el tipificado del código, aunque esto no es realmente un inconveniente del lenguaje en sí.	
--	--	---	--

**Tabla 3.3.11 comparativo lenguajes de desarrollo**

El sistema empleara JSP, HTML, CSS y JavaScript para que el usuario tenga un a ambiente manejable. Ver figura 3.3.12



JBoss es un servidor de aplicaciones Java EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible la máquina virtual de Java. JBoss Inc., Boss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de e-business. Combinando una arquitectura orientada a servicios SOA, con una licencia GNU de código abierto, JBoss AS puede ser descargado, utilizado, incrustado y distribuido sin restricciones por la licencia. Ver logo en la figura 3.3.13.

Las características destacadas de JBoss incluyen:

- Producto de licencia de código abierto sin coste adicional.
- Cumple los estándares.
- Confiable a nivel de empresa
- Incrustable, orientado a arquitectura de servicios.

- Flexibilidad consistente
- Servicios del middleware para cualquier objeto de Java.
- Soporte completo para JMX.



**Figura 3.3.13 Logo JBoss**

jQuery, es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada. jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados.<sup>2</sup> jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Ver logo en la figura 3.3.14.

Sus características:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.

- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.5



**Figura 3.3.14 Logo jQuery**

Finalmente, el software que se emplea tiene ventajas y desventajas, pero es confiablemente seguro y puede interactuar con bastante información.

### **3.4 Separación por módulos utilizando la metodología MVC**

Como se mostró anteriormente, la metodología MVC es una gran herramienta para el desarrollo de sistemas, en este capítulo se representará el sistema en un diagrama de bloques y posteriormente los módulos representados por bloques serán representados usando la metodología MVC.

En las figuras 3.4.1.1, 3.4.1.2 y 3.4.1.3 se podrá observar el diagrama de Flujo del sistema.

En las figuras 3.4.2.1, 3.4.2.2, 3.4.2.3 y 3.4.2.4 se podrá observar el diagrama MVC del sistema.

### 3.4.1 Diagrama de Flujo del sistema

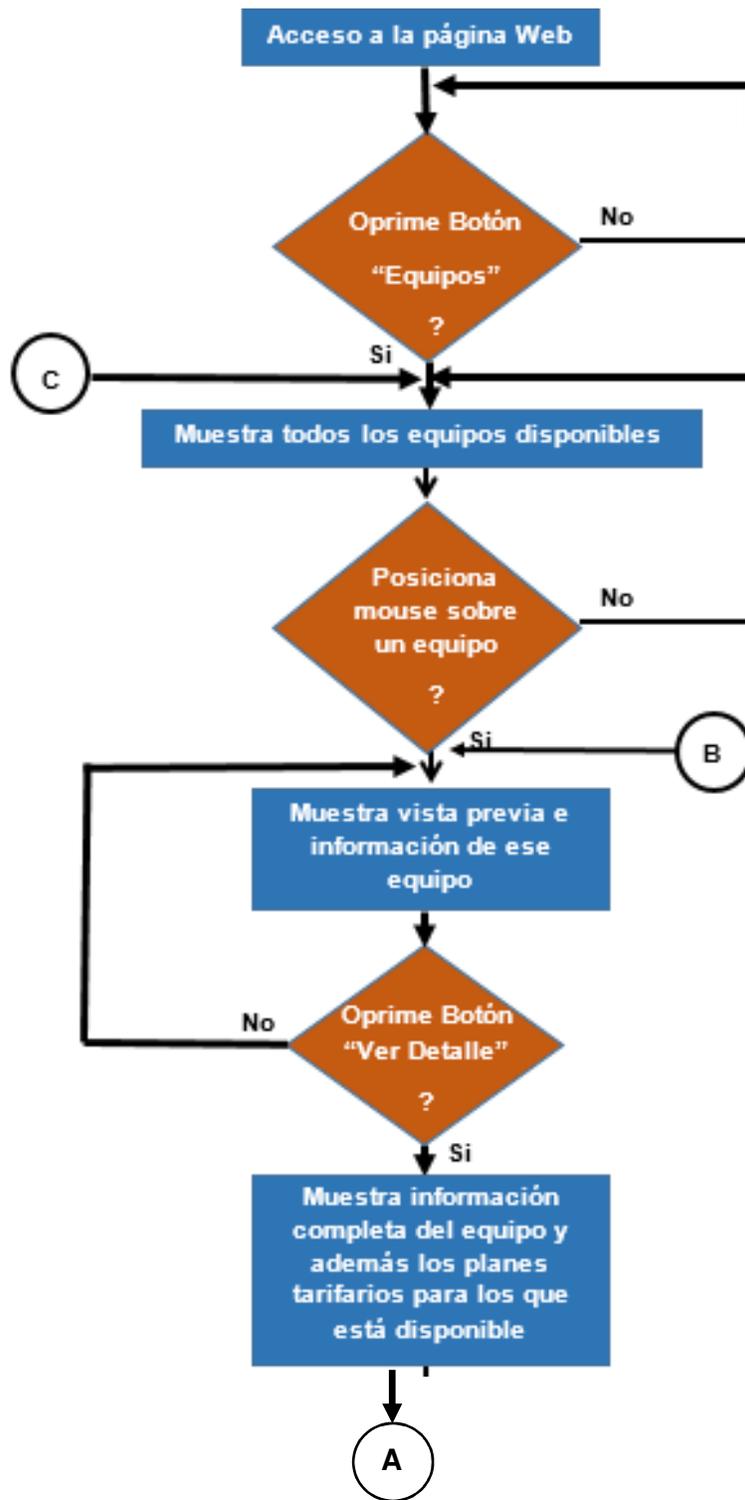


Figura 3.4.1.1 Diagrama de flujo del sistema (primera parte)

Diagrama de Flujo del sistema (Continuación)

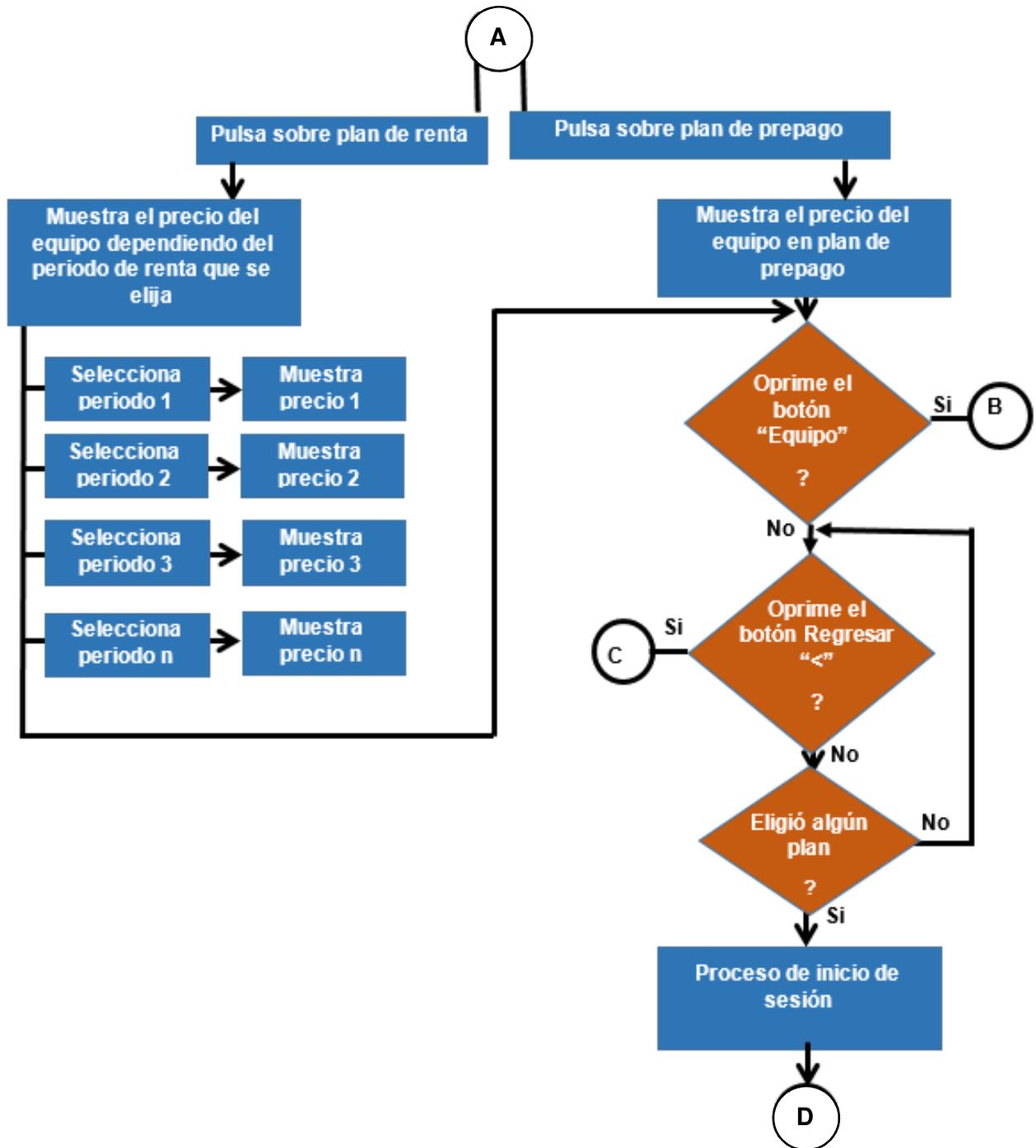


Figura 3.4.1.2 Diagrama de flujo del sistema (segunda parte)

Diagrama de flujo del sistema (Continuación)

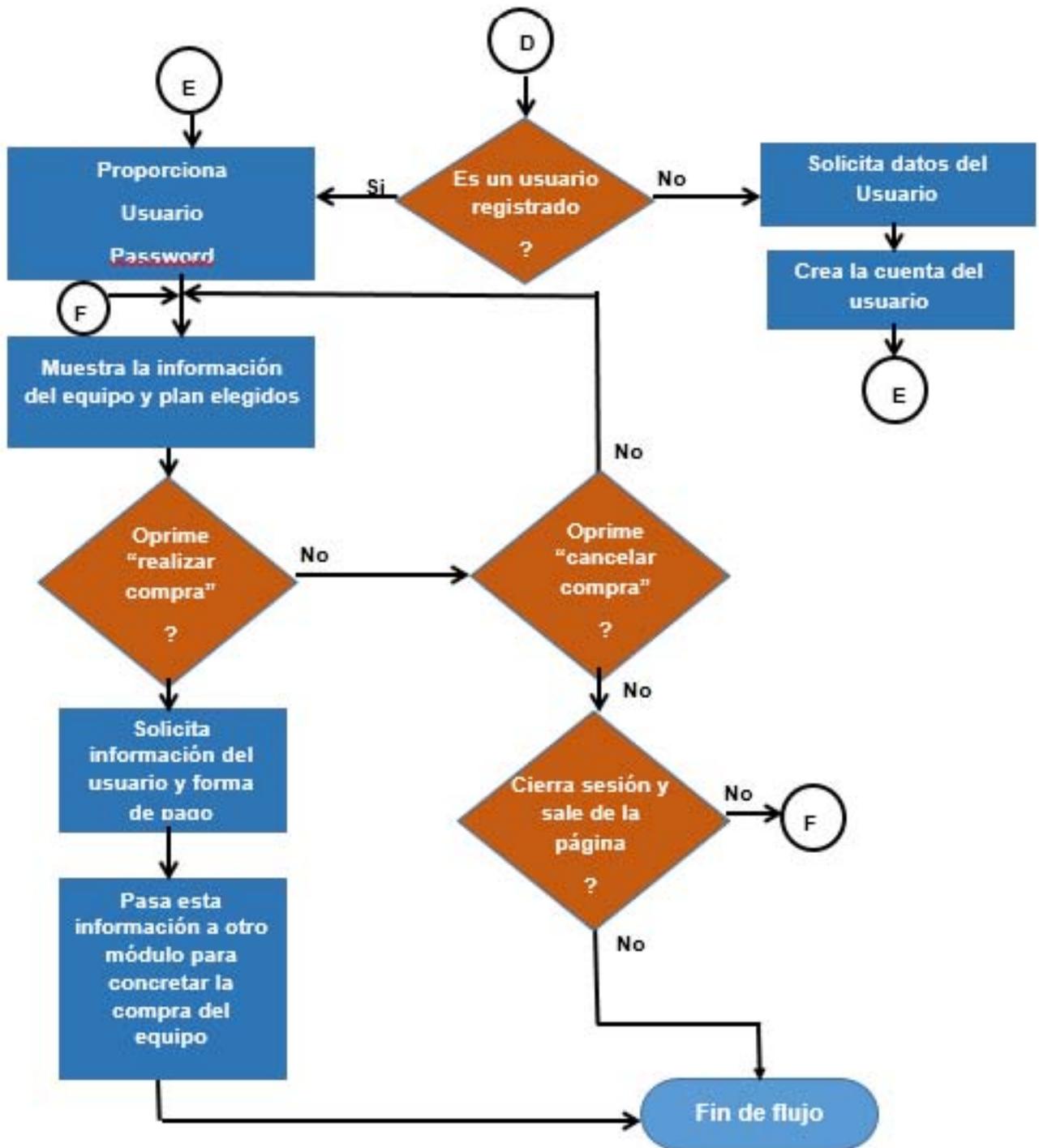


Figura 3.4.1.3 Diagrama de flujo del sistema (tercera parte)

3.4.2 Diagrama MVC del sistema

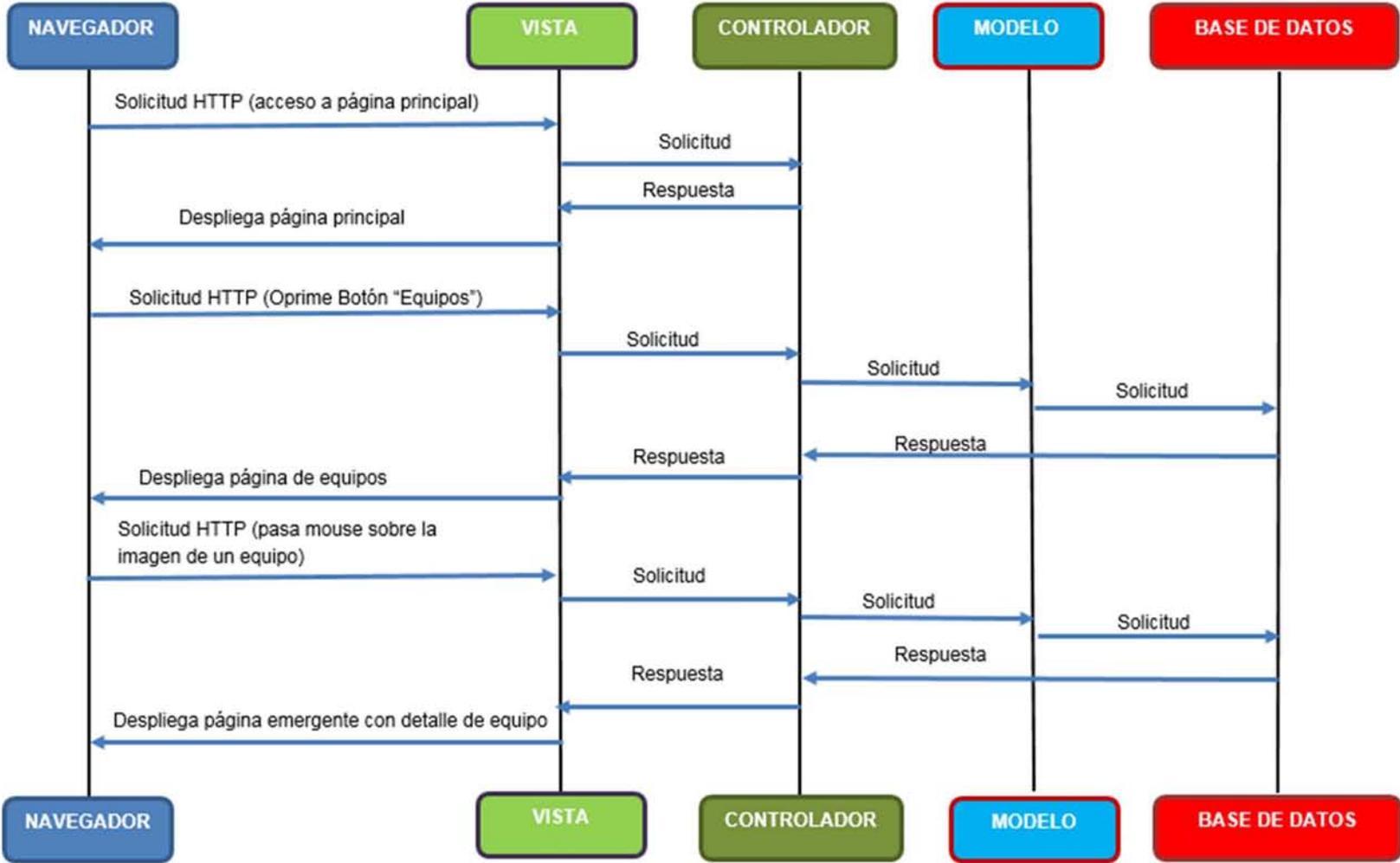


Figura 3.4.2.1 Diagrama MVC del sistema (primera parte)

Diagrama MVC del sistema (Continuación)

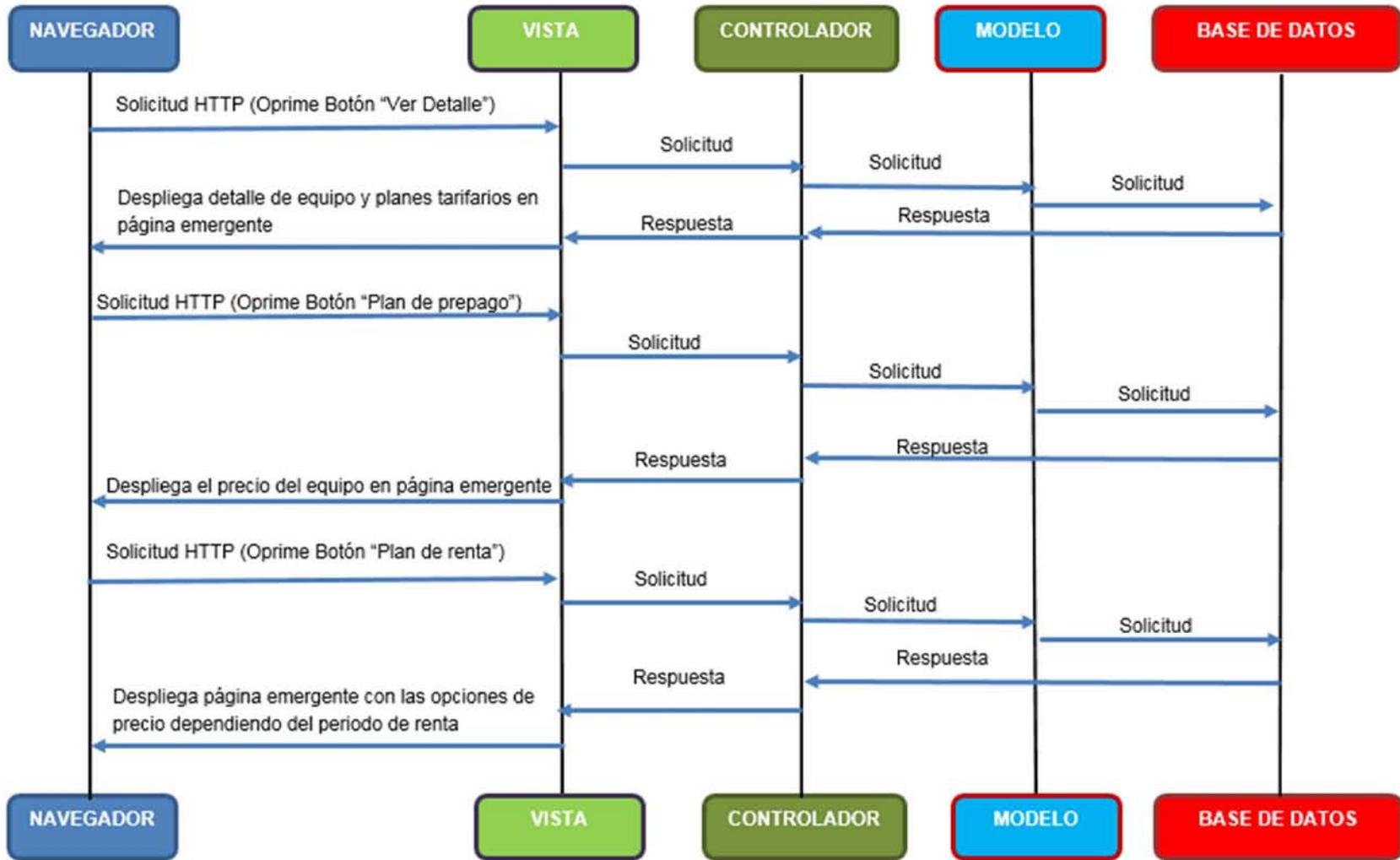


Figura 3.4.2.2 Diagrama MVC del sistema (segunda parte)

Diagrama MVC del sistema (Continuación)

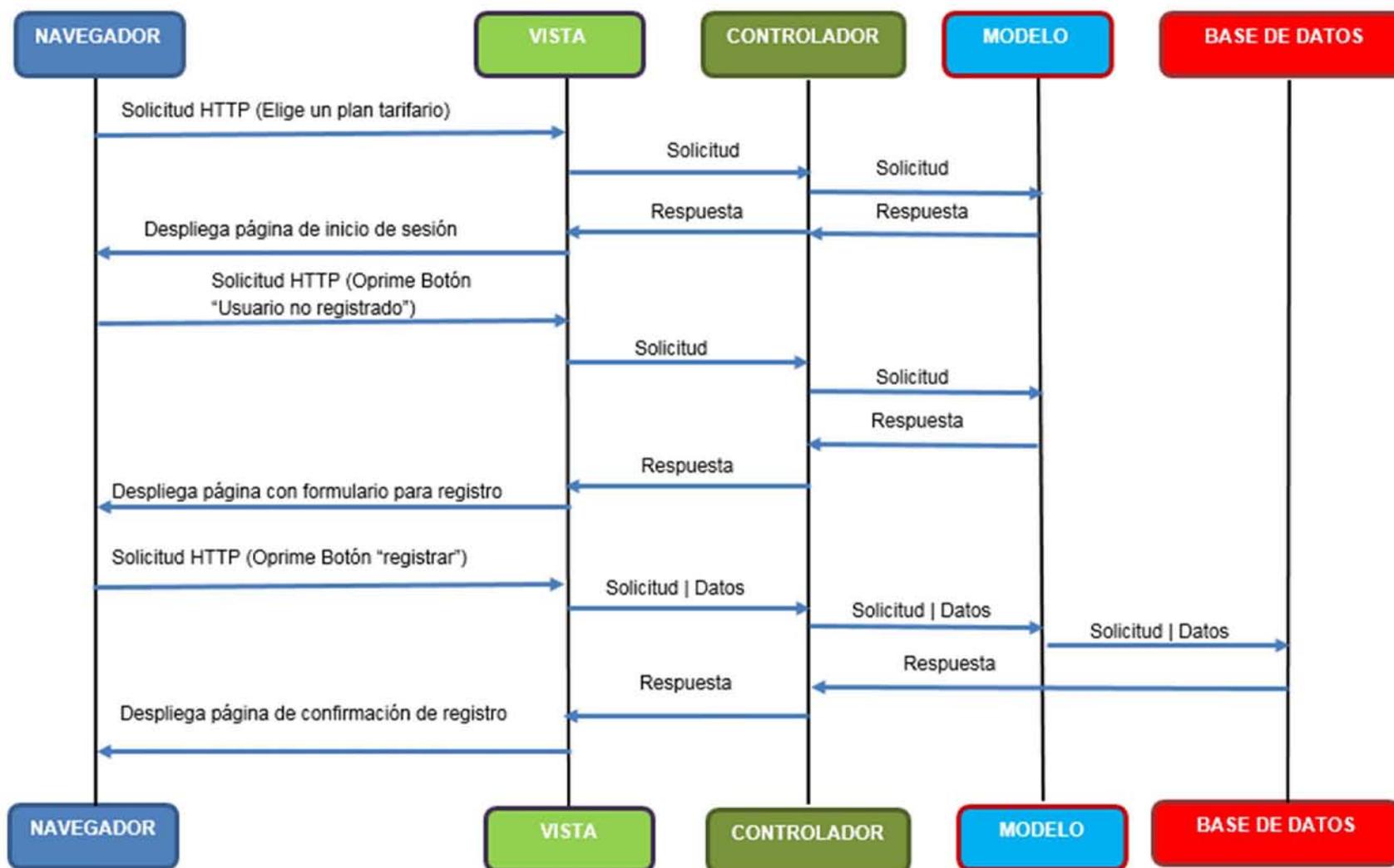


Figura 3.4.2.3 Diagrama MVC del sistema (tercera parte)

Diagrama MVC del sistema (Continuación)

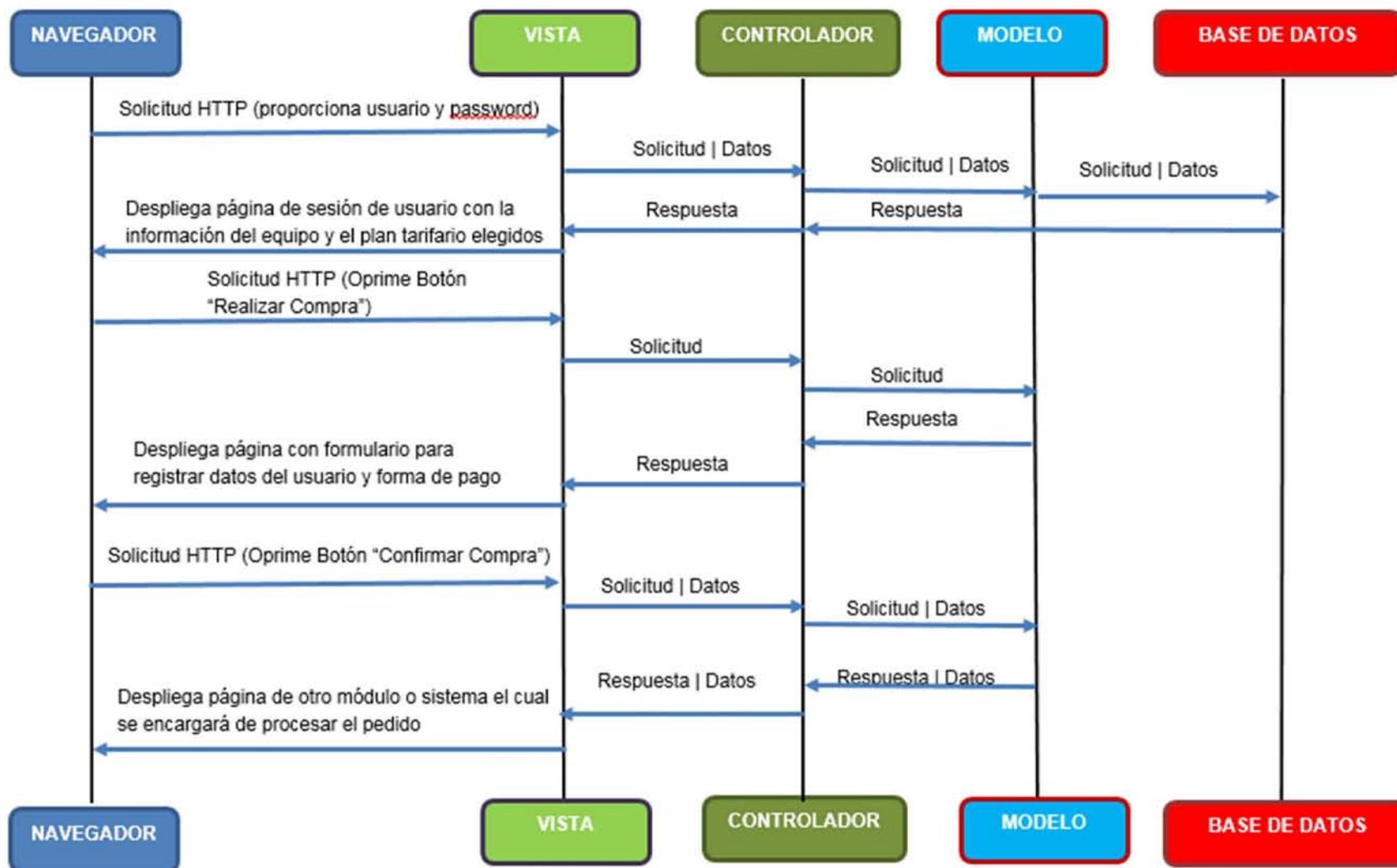


Figura 3.4.2.4 Diagrama MVC del sistema (cuarta parte)

### 3.5 Diseño del Front - end a nivel de bloques sugerido al usuario.

Para lograr un mejor entendimiento con el usuario es importante hacerle ver de forma esquemática como lucirá el Front - end de la solución, en las figuras 3.5.1, 3.5.2 y 3.5.3, se presentan los esquemas de dicha solución.

#### Pantalla Inicial:

Se muestra el banner de promociones, una lista de los equipos más vendidos y se definen tanto el top de la pantalla integrando la selección de equipos, el carrito de compras y la búsqueda por marca y modelo del teléfono. En la figura 3.5.1 se muestra el diseño propuesto.

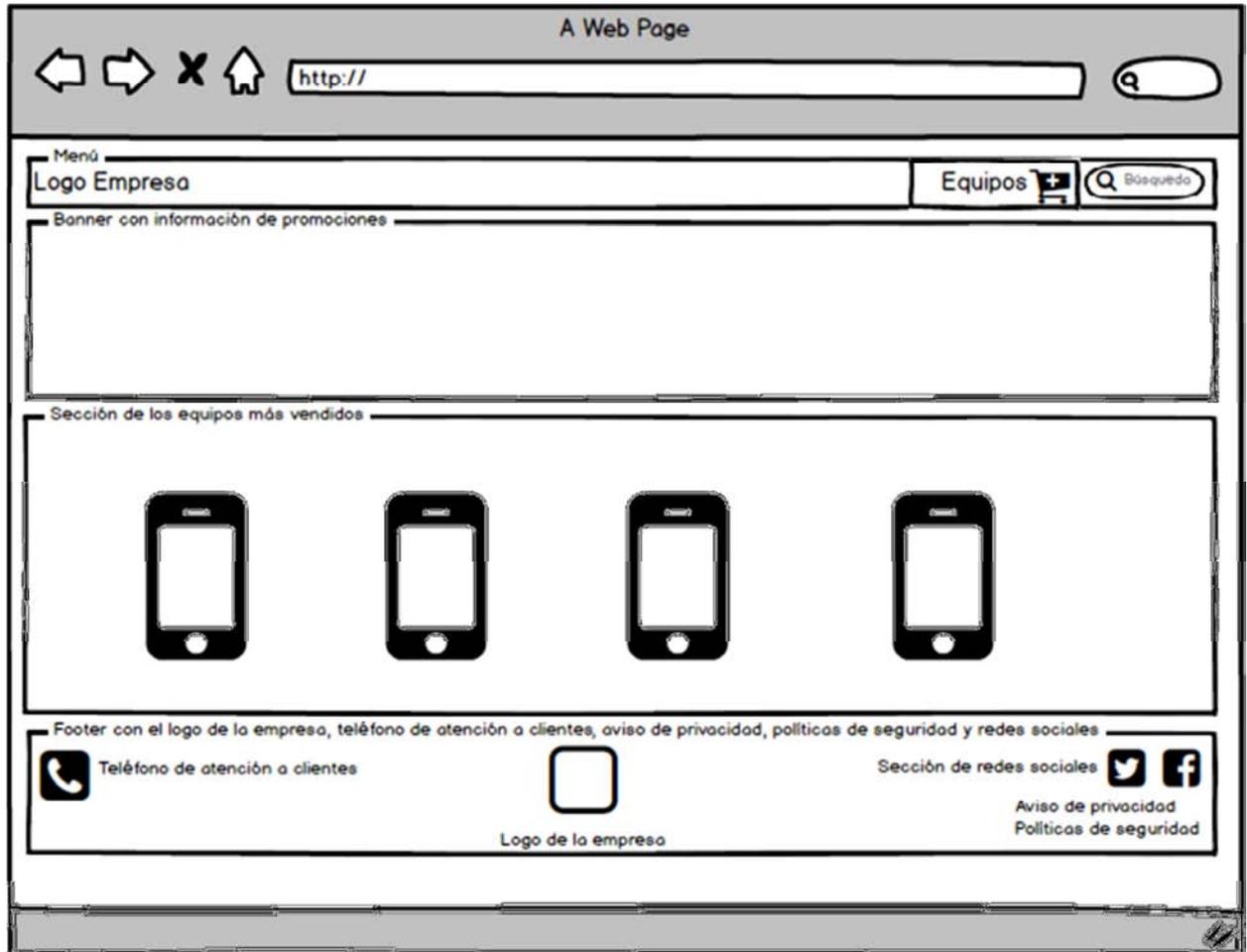


Figura 3.5.1 Pantalla Inicial

### Pantalla de selección y filtrado de equipo:

Esta selección se podrá realizar por marca, sistema operativo y precio, así como también la realización de zoom de los equipos para mostrar más o menos equipos en la pantalla. Vea figura 3.5.2.

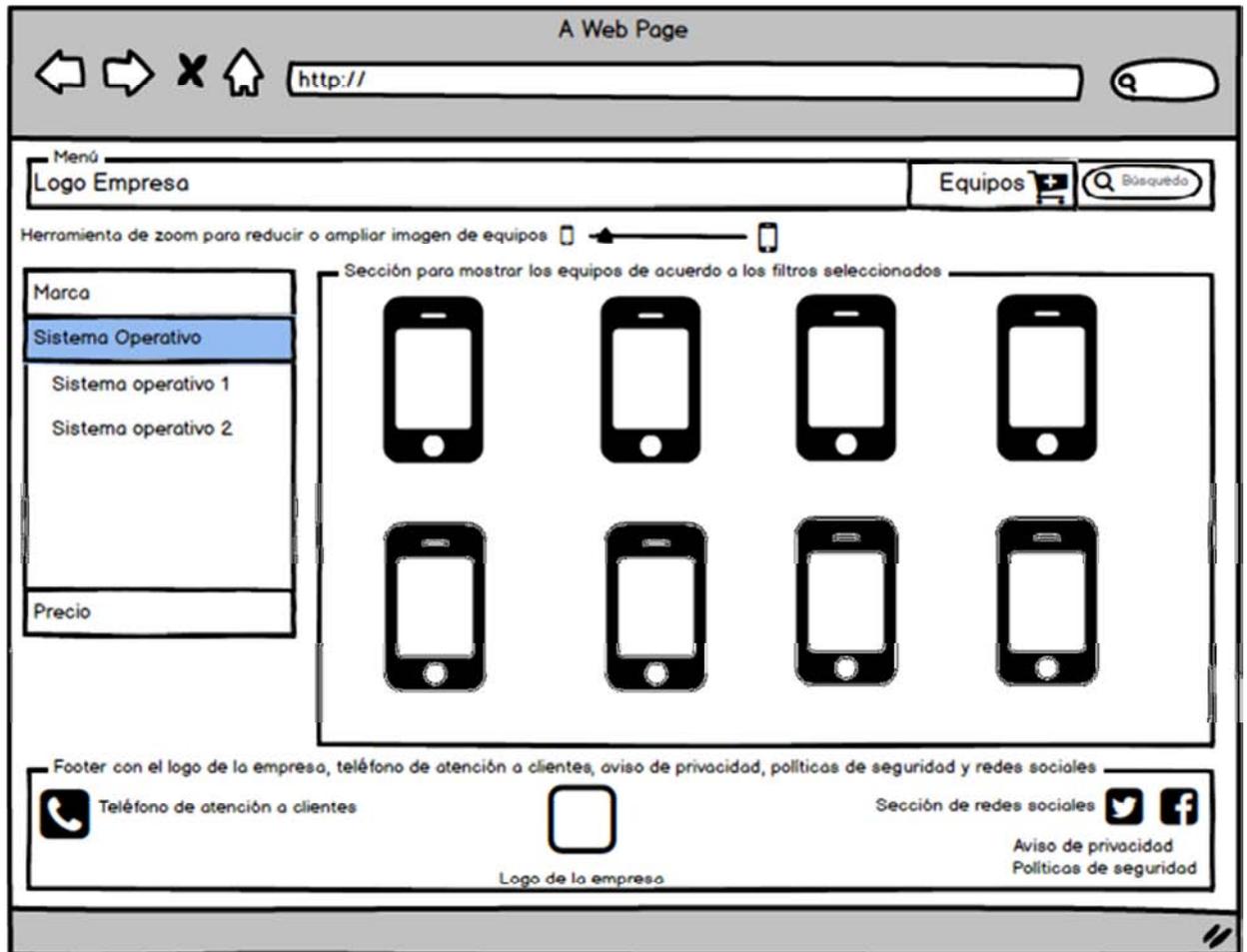


Figura 3.5.2 Pantalla de selección y filtrado de equipo.

### Pantalla de las características del equipo seleccionado:

En esta pantalla se muestran las características principales del equipo, colores en los que está disponible, datos del plan en el que se vende el equipo y otros planes. Vea figura 3.5.3.

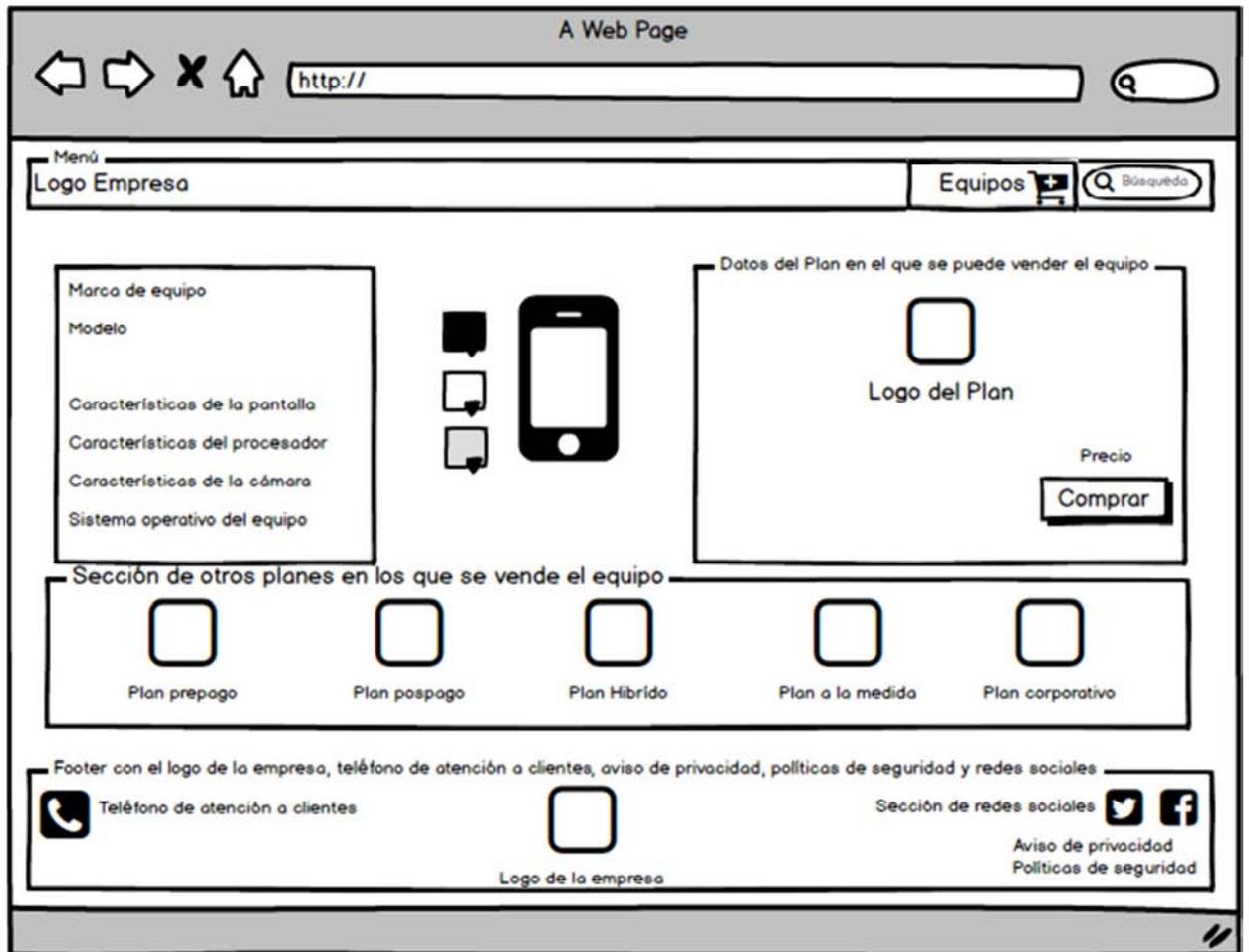


Figura 3.5.3 Pantalla características de equipos.



# **CAPÍTULO**

## **IV**



## Capítulo 4 Diseño y construcción de la aplicación

### 4.1. Modelo esencial ambiental del sistema.

#### 4.1.1 Diagrama de contexto.

El diagrama de contexto muestra a través de flujos de datos las interacciones existentes entre los agentes externos y el sistema, sin describir en ningún momento la estructura del sistema de información.

Un diagrama de contexto se puede desarrollar con el uso de dos tipos de bloques de construcción:

- Entidades (Actores): cuadros etiquetados; uno en el centro representando el sistema, y varios cuadros alrededor para los factores externos.
- Relaciones: líneas etiquetadas entre las entidades y el sistema.

Para diseñar un diagrama de contexto se pueden utilizar distintos tipos de dibujo para representar entidades externas, los cuales pueden usar óvalos, figuras de línea, imágenes, autoformas o cualquier otra representación que presente un significado.

Un diagrama de contexto también puede enlistar las clasificaciones de las entidades externas lo que añade claridad al nivel de participación de las entidades con respecto al sistema. Esas categorías incluyen:

- Activo: entidades externas dinámicas que frecuentemente inician eventos para lograr algún objetivo o propósito.
- Pasivo: Entidades externas estáticas que no interactúan frecuentemente con el sistema.
- Cooperativo: Entidades externas predecibles que son utilizadas por el sistema para alcanzar algún resultado deseado.

- Autónoma (Independiente): Entidades externas que están separadas del sistema, pero que lo afectan indirectamente, a través de restricciones impuestas o influencias similares.

Este diagrama permite lograr una vista a alto nivel de lo que el sistema ayudará a resolver, por lo que no es necesario representar todos los flujos de datos del sistema en él, a continuación se mencionan las características que lo conforman:

- Considerar únicamente los flujos de datos que permitan visualizar las interacciones generales asociadas directamente con el objetivo principal del sistema.
- Usar flujos de datos compuestos para agrupar a aquellos que sean similares.
- Los terminadores son todos aquellos actores, llámese sistemas, organizaciones y personas con los que interactúa el sistema.
- Los repositorios de datos que el sistema comparte con dichos terminadores.
- La información que recibe el sistema como entrada y que debe ser procesada.
- La información generada por el sistema y que es entregada.

El diagrama de contexto del sistema se muestra en la Figura 4.1.1.1. Puede observar de forma general la relación del sistema con los diferentes procesos.

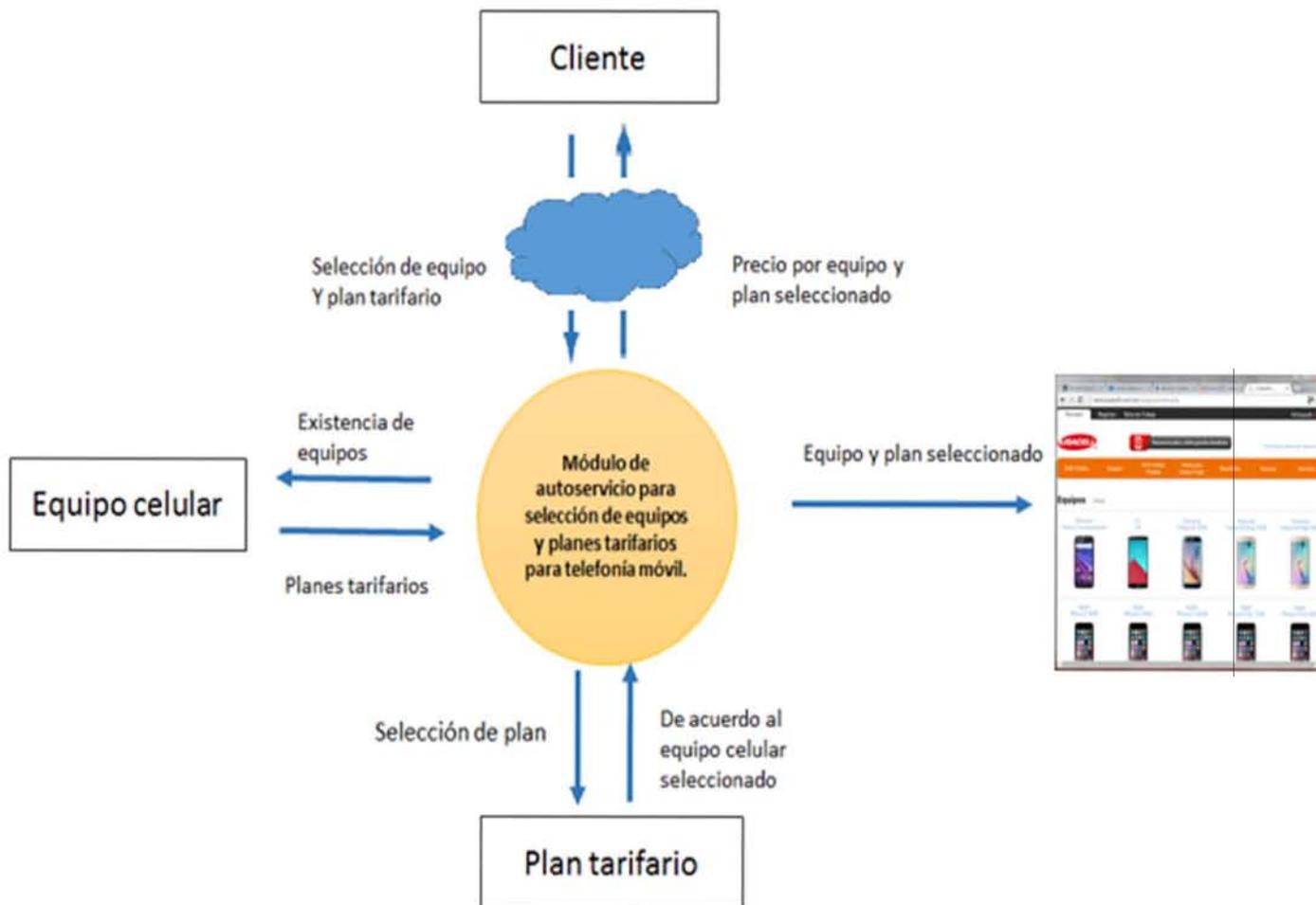


Figura 4.1.1.1 Diagrama de Contexto del sistema

## 4.1.2 Diagrama Entidad - Relación

Este diagrama es una descripción conceptual de las estructuras de datos y sus relaciones. Denominado por sus siglas como E-R (Entity Relationship), es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema así como sus interrelaciones y sus propiedades.

El diagrama E-R consta de los siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombos, que representan relaciones.
- Líneas, que unen atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones.
- Elipses dobles, que representan atributos multivalorados.
- Elipses discontinuas, que denotan atributos derivados.
- Líneas dobles, que indican participación total de una entidad en un conjunto de relaciones.

Los tipos de relaciones que pueden representarse en este tipo de diagramas se muestran en la figura 4.1.2.1

RELACIÓN	SÍMBOLOGÍA
<b>Relación uno a uno (1:1, 1/1):</b> Una entidad del tipo A solo se puede relacionar con un registro de la entidad del tipo B, y viceversa.	
<b>Relación uno a Muchos (1:n, 1/∞):</b> Significa que una entidad del tipo A puede relacionarse con cualquier cantidad de registros de la entidad B, y una entidad del tipo B solo puede estar relacionada con un registro de la entidad del tipo A.	
<b>Relación Muchos a Muchos (n:n, ∞/∞):</b> Establece que cualquier cantidad de registros de la entidad "A" pueden estar relacionados con cualquier cantidad de registros de la entidad B y viceversa.	

Figura 4.1.2.1 Tipos de relaciones

La figura 4.1.2.2 muestra el diagrama E/R del Módulo de autoselección de equipos y planes tarifarios para telefonía móvil.

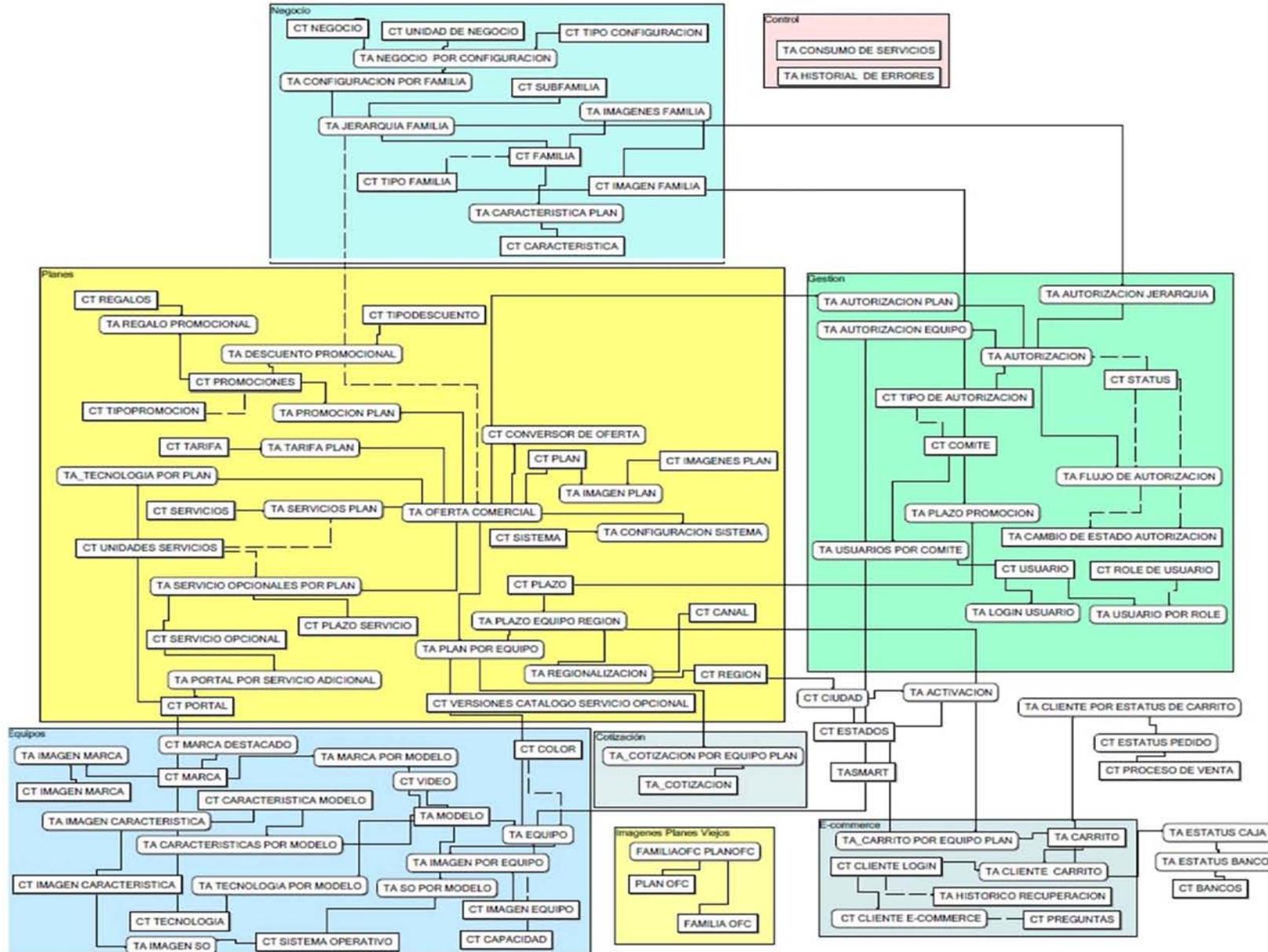


Figura 4.1.2.2 Diagrama Entidad-Relación del Sistema

### 4.1.3 Diagramas de casos de uso, procesos y clases.

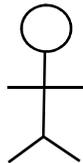
#### Casos de uso

El diagrama de casos de uso representa la operación del sistema por parte de un cliente o usuario (también llamado actor); de igual forma, representa el tipo y el orden de los elementos que interactúan en dicho sistema.

Un diagrama de casos de uso consta de los siguientes elementos:

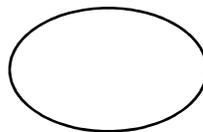
- Actor.
- Casos de Uso.
- Relaciones de Uso, Herencia y Comunicación.

**Actor:** Es la acción que un usuario asume con respecto al sistema y no necesariamente representa a una persona en particular, más bien la labor que realiza al operar el sistema. El actor se representa en el diagrama como lo muestra la siguiente Figura 4.1.3.1



**Figura 4.1.3.1 Actor**

**Caso de Uso:** Es una tarea u operación específica que se realiza tras una orden de algún agente externo, ya sea por la petición de un actor o bien porque se invocó desde otro caso de uso y lo representaremos por una elipse, como lo muestra la Figura 4.1.3.2



**Figura 4.1.3.2 Caso de uso**

## Relaciones:

- **Asociación:** Es el tipo más básico de relación, indica la invocación por parte de un actor o desde un caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple tal como se muestra en la Figura 4.1.3.3



Figura 4.1.3.3 Asociación

- **Dependencia:** Es una relación muy particular entre clases, en la cual una clase depende de otra, la representaremos con una flecha punteada como la que se muestra en la Figura 4.1.3.4



Figura 4.1.3.4 Dependencias

- **Generalización:** Este tipo de relación es muy utilizado y que cumple una doble función dependiendo de su estereotipo, que puede ser de Uso (<<uses>>) o de Herencia (<<extends>>). Dicha relación se denota con una flecha simple representada por Figura 4.1.3.5

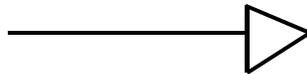
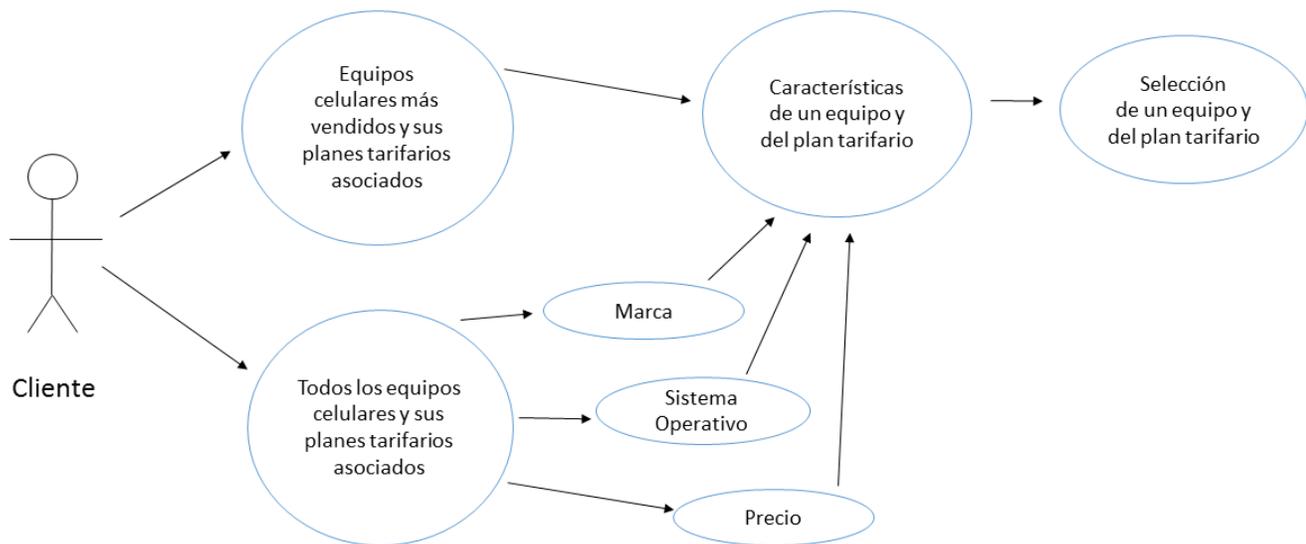


Figura 4.1.3.5- Generalización

**extends:** Es utilizado cuando un caso de uso es similar a otro.

**uses:** Se emplea cuando se tiene un conjunto de características que son similares en varios casos de uso a la descripción de la característica.

La figura 4.1.3.6 muestra el diagrama de caso de uso general del módulo de autoservicio para selección de equipos y planes tarifarios para telefonía móvil.



**Figura 4.1.3.6 Diagrama de caso de uso general**

### **Equipos más vendidos**

Cuando el usuario ingresa al portal del módulo de autoselección de equipos y planes tarifarios la primera pantalla que observa son los equipos más vendidos, con el modelo, la marca y el costo aproximado del equipo. Al seleccionar un equipo podrá ver una tabla con los diferentes colores de equipos y las distintas capacidades de memoria y el plan tarifario prepago. Si está interesado por un paquete se va directo al plan tarifario de Pospago que tiene asignados beneficios. Ver figura 4.1.3.7

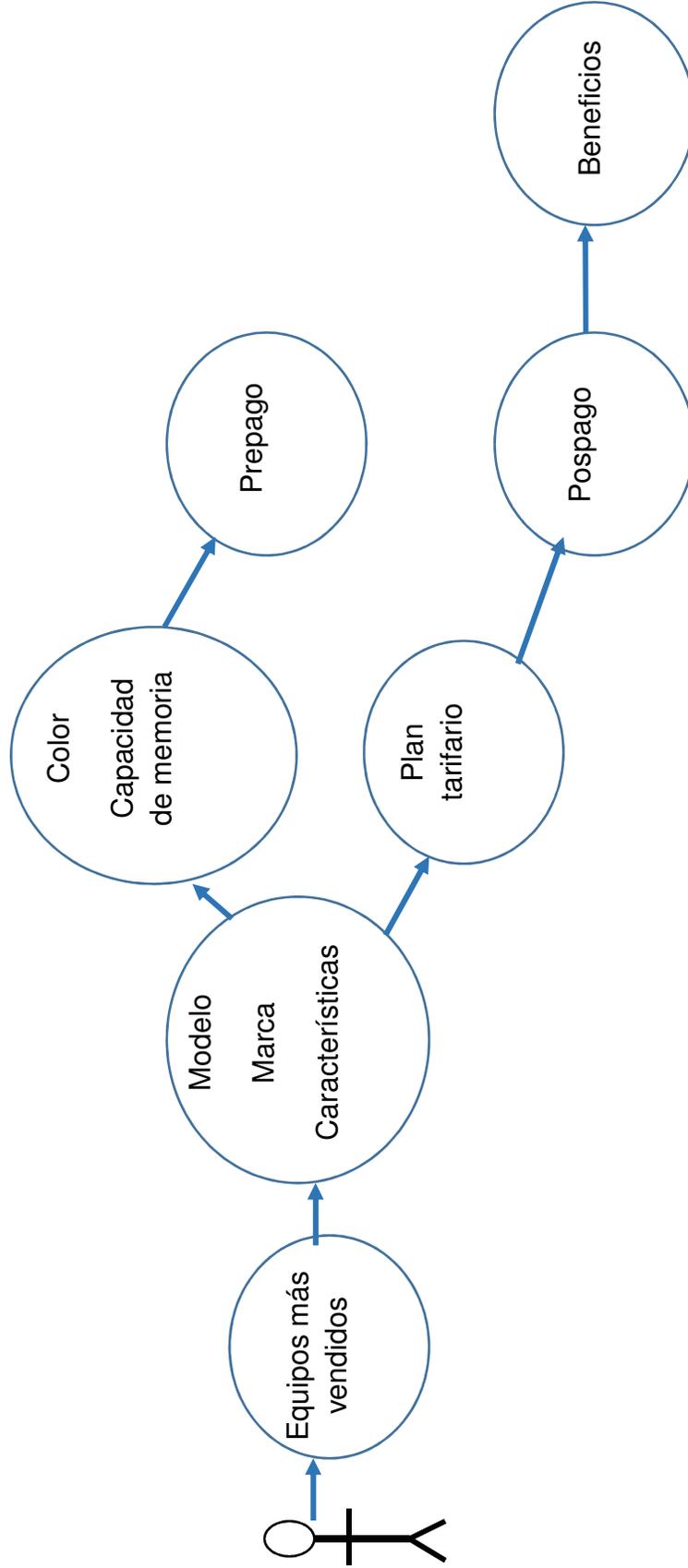


Figura 4.1.3.7 Casos de uso de “equipos más vendidos”

## Todos los equipos

Cuando el usuario ingresa al portal del módulo de autoselección de equipos y planes tarifarios no quiere saber cuáles son los teléfonos más vendidos, y selecciona la opción de equipos. Se muestran tres filtros: Marca, Sistema operativo y Costo, dentro del filtro de costo, podrá ver los equipos con el costo que esté dispuesto a pagar. También tiene la opción de ver todos los equipos que puede contratar con el Plan de Prepago o Pospago. Ver figura 4.1.3.8

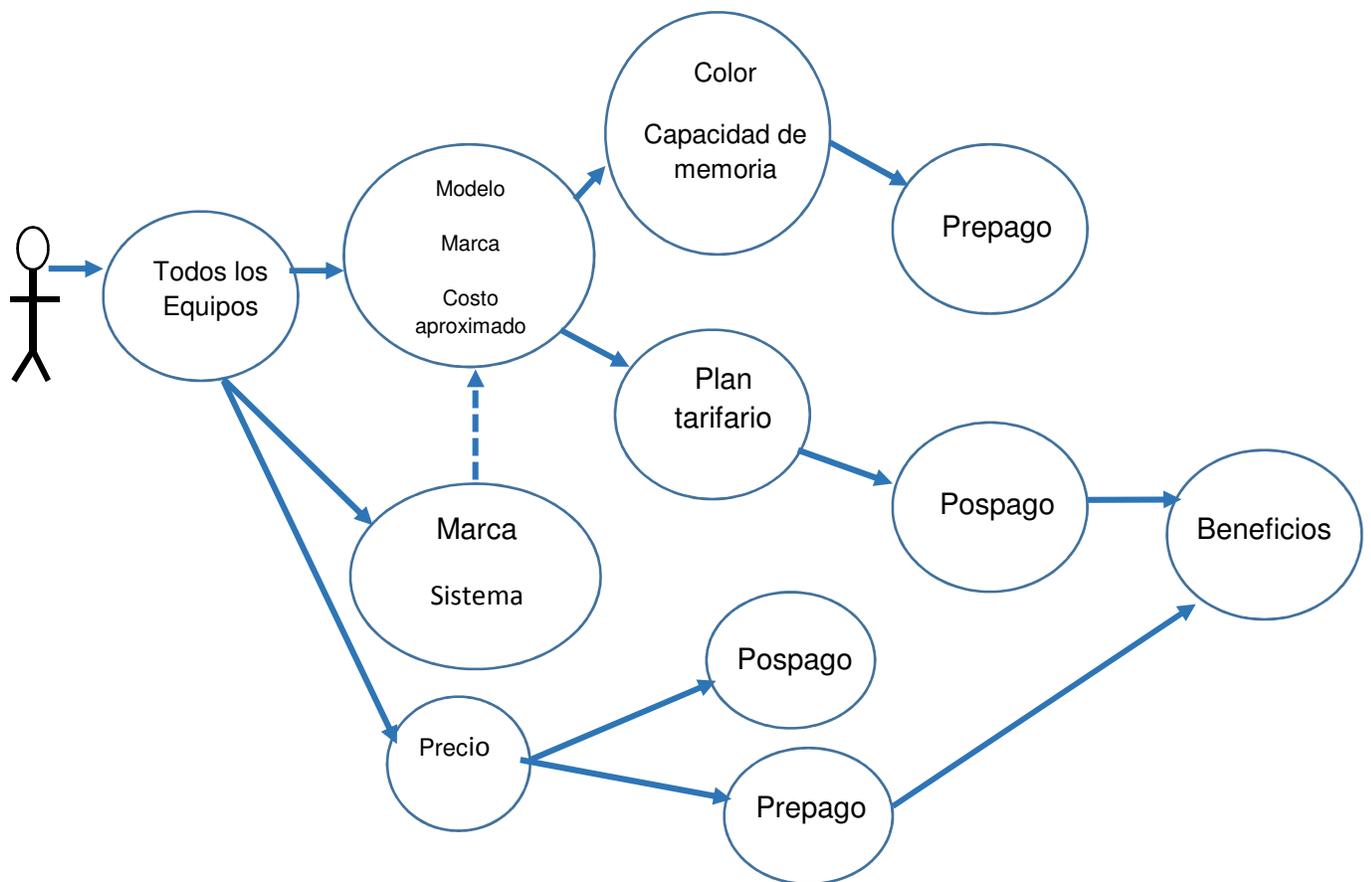
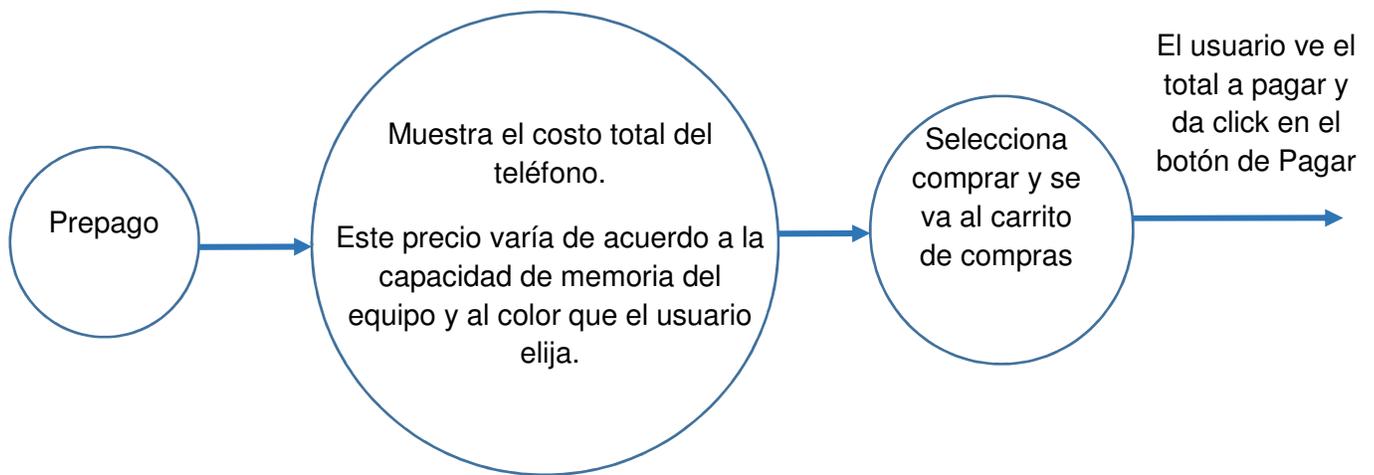


Figura 4.1.3.8 Casos de uso de “todos los equipos”

### Plan Tarifario de Prepago

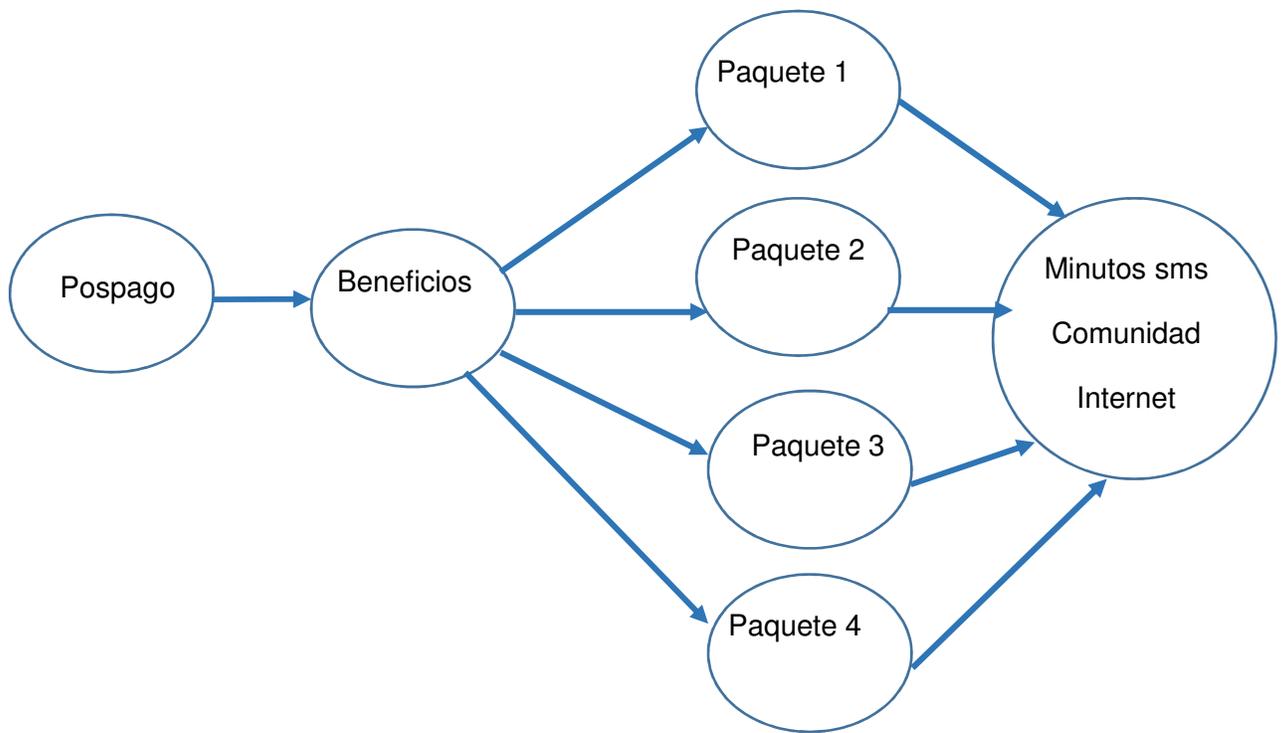
El plan tarifario de Prepago da el costo total del equipo, el costo varía de acuerdo al color del equipo que el usuario seleccione y a la capacidad de memoria que el usuario elija. Si el usuario está de acuerdo con las especificaciones del color y capacidad de memoria, selecciona comprar y se envía al carrito del súper para proceder con la compra del equipo. Ver figura 4.1.3.9



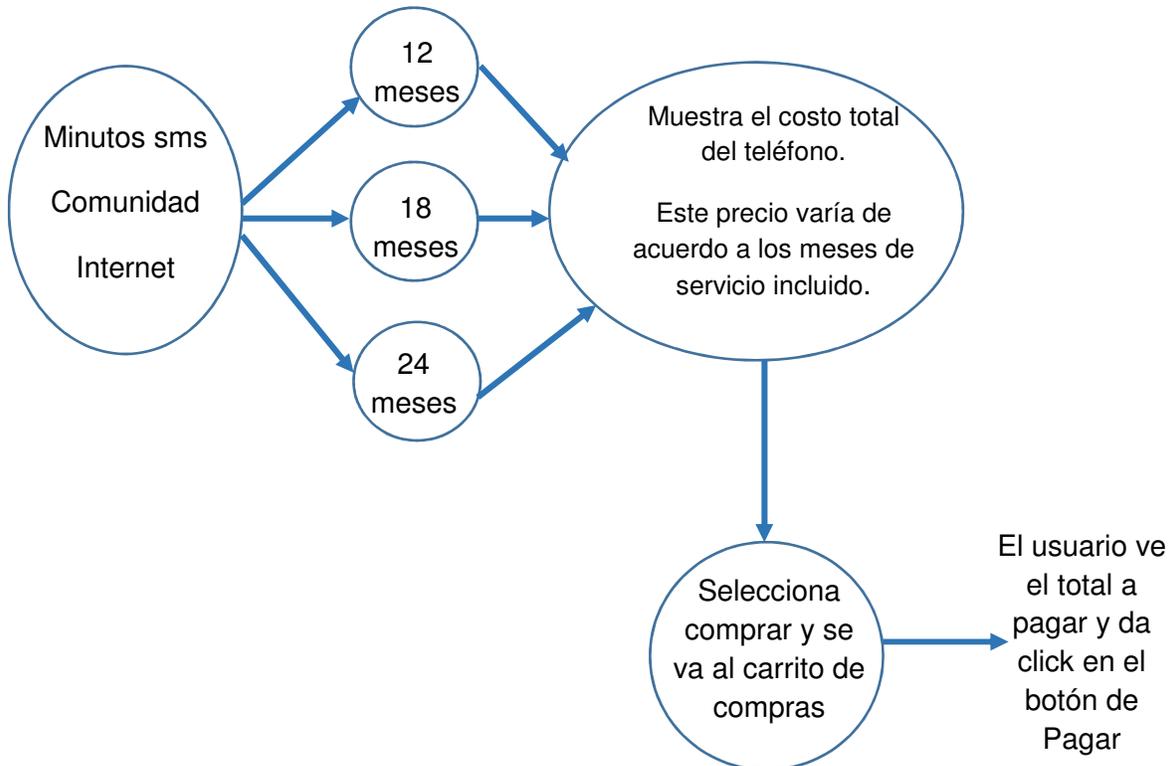
**Figura 4.1.3.9 caso de uso al seleccionar el Plan Prepago**

### Plan tarifario Pospago.

El usuario elige el plan de Pospago el cual tiene beneficios agrupados en paquetes, los cuales incluyen minutos SMS, llamadas e internet. El usuario tiene la opción de pagar a 12, 18 y 24 meses. Una vez elegido el paquete y los meses de pago se muestra el costo mensual del teléfono. Y procede la compra del equipo. Ver figuras 4.1.3.10 y 4.1.3.11



**Figura 4.1.3.10 caso de uso al seleccionar el Plan de Pospago**



**Figura 4.1.3.11 caso de uso al seleccionar el Plan de Pospago**

### **4.1.3.2 Diagrama de Procesos**

Este diagrama representa gráficamente la secuencia de actividades que se realizan al operar el sistema; estas actividades o procesos se identifican con símbolos de acuerdo a su naturaleza, dentro de estos símbolos se incluye una breve descripción de dicho proceso.

De esta forma, se ofrece una descripción visual de las actividades a desarrollar para proporcionar al sistema las entradas necesarias para que después de procesarlas, el sistema entregue los resultados que se esperan.

La creación de un diagrama de procesos, facilita la visualización y la comprensión de los mismos, centrándose en sus aspectos específicos y apreciando las interrelaciones con otros procesos y subprocesos.

Asimismo, definen los límites de un proceso que a veces no son tan evidentes si no se definen sus proveedores (entradas) y los otros procesos con los que interactúa al seguir una secuencia ordenada que nos lleve a obtener la salida o resultado requerido.

Dadas las características que se mencionan, este tipo de diagramas constituyen una excelente referencia para establecer mecanismos de control y medición de los diferentes procesos que constituyen un sistema al mismo tiempo que facilitan el estudio y la aplicación de medidas de corrección y/o de optimización.

En la figura 4.1.3.2.1 se muestra la vista general del diagrama de procesos del sistema

Las figuras 4.1.3.2.2, 4.1.3.2.3, 4.1.3.2.4 y 4.1.3.2.5 muestran los detalles de cada uno de los procesos que se desprenden del diagrama general.

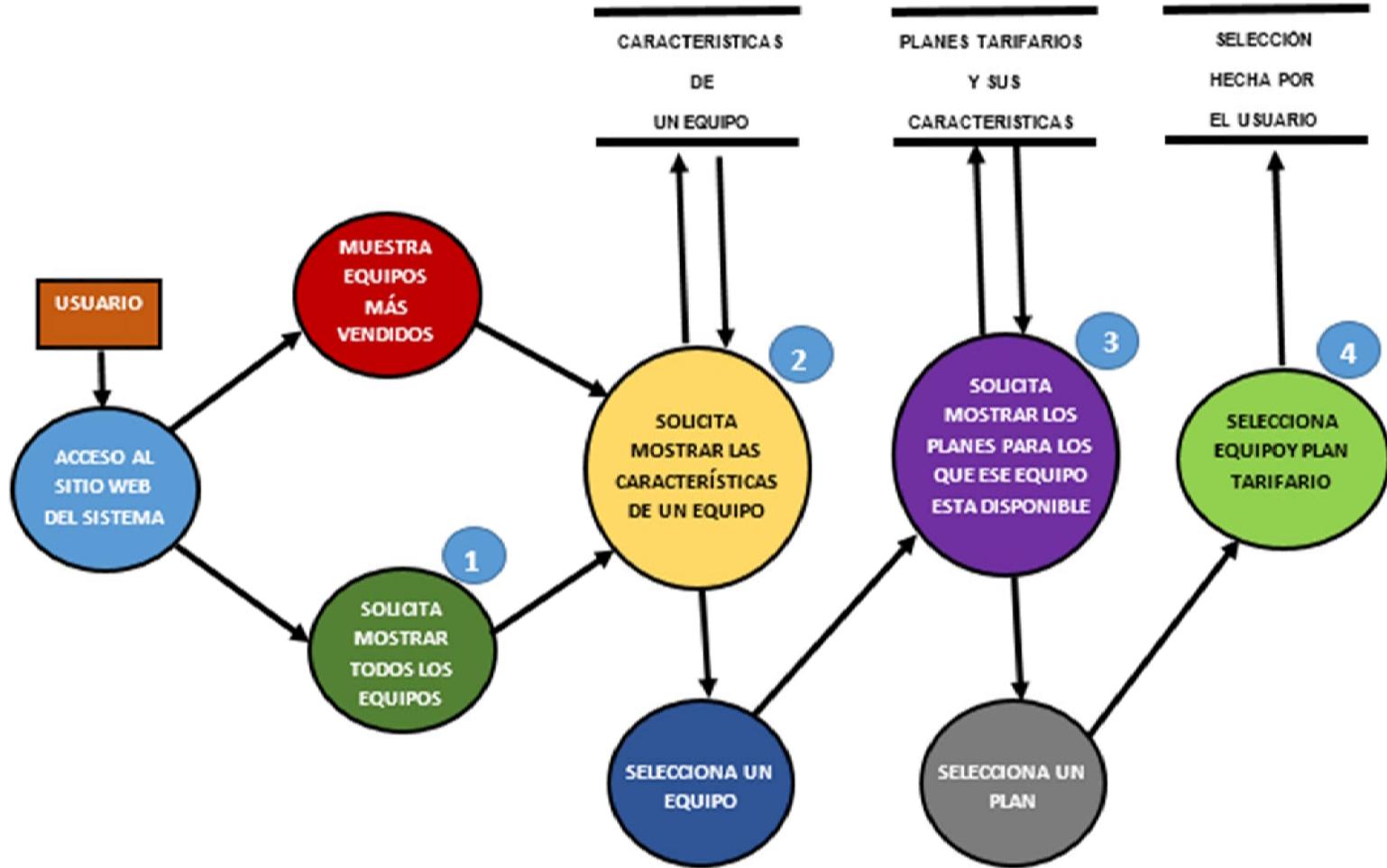


Figura 4.1.3.2.1 diagrama de procesos del sistema (vista general)

Diagrama de Procesos del Sistema (Detalle del Proceso “Solicita mostrar todos los

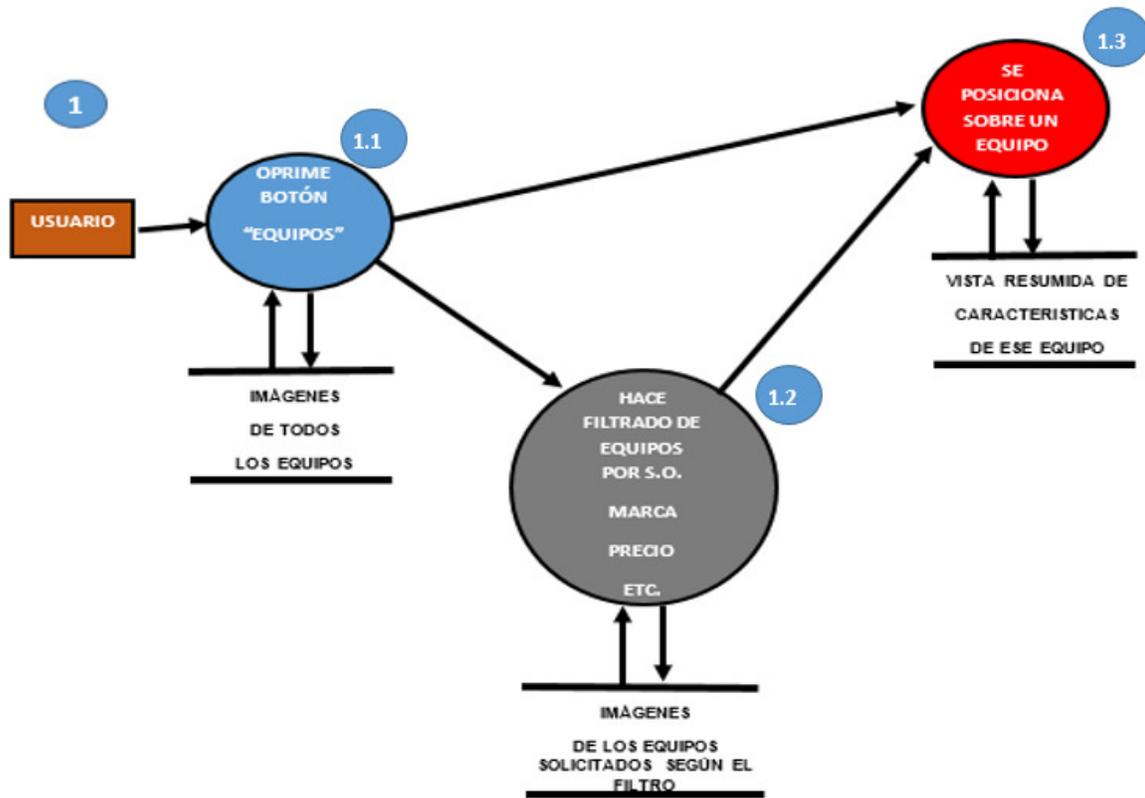


Figura 4.1.3.2.2 diagrama de procesos del sistema  
(detalle del proceso “Solicita mostrar todos los equipos”)

equipos”)

Diagrama de Procesos del sistema (Detalle del proceso “Solicita mostrar las características de un equipo”)

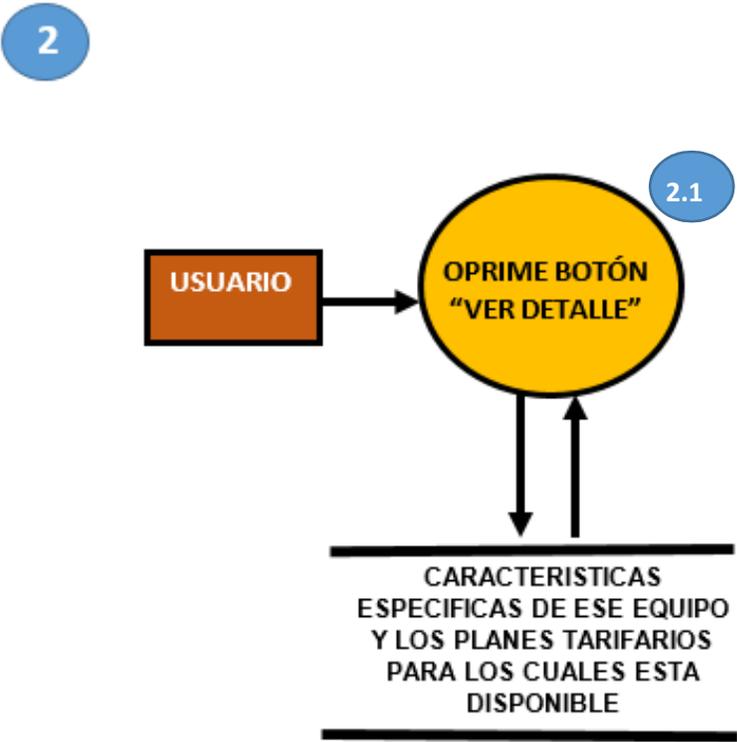


Figura 4.1.3.2.3 diagrama de procesos del sistema  
(detalle del proceso “Solicita mostrar las características de un equipo”)

Diagrama de Procesos del Sistema (Detalle del proceso “Solicita mostrar los planes para los que este equipo esta disponible”)

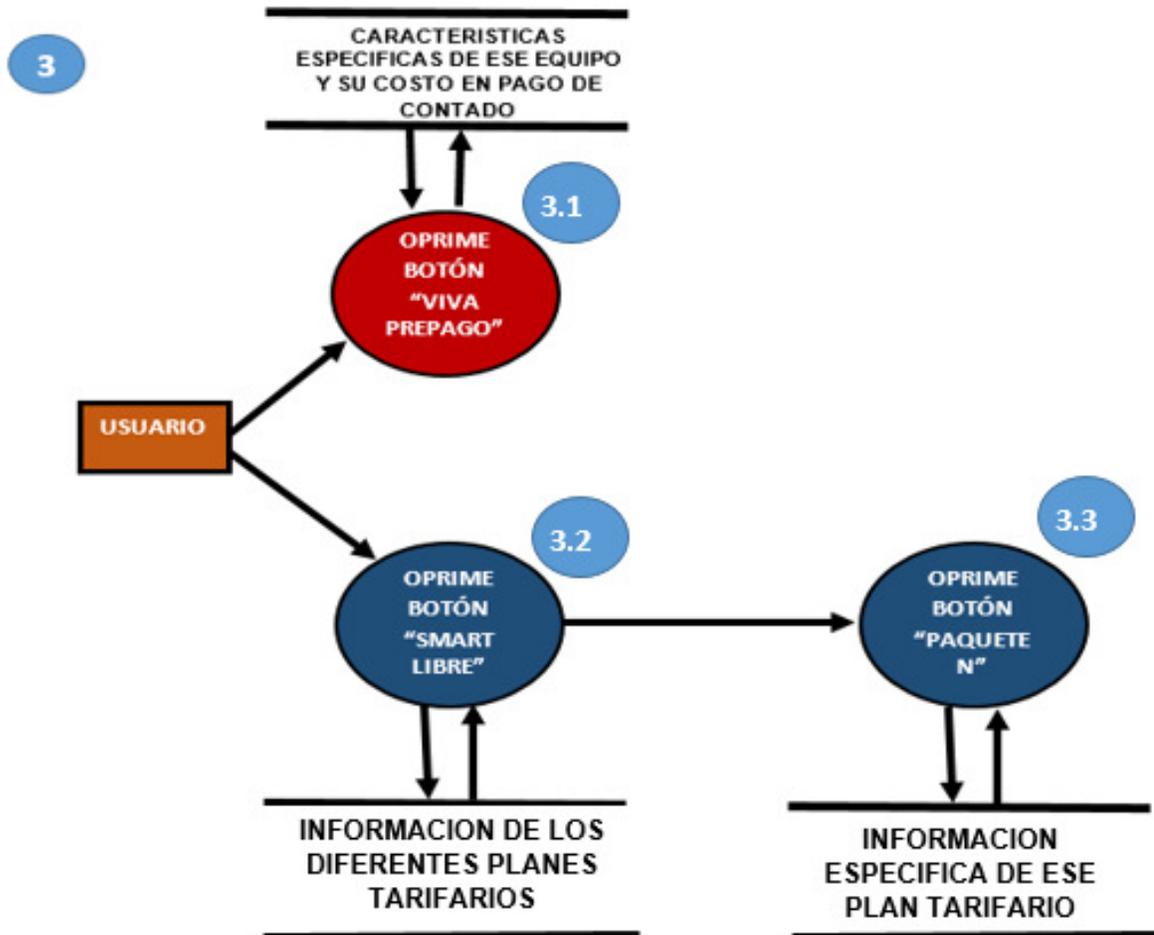


Figura 4.1.3.2.4 diagrama de procesos del sistema

(detalle del proceso “Solicita mostrar los planes para los que este equipo esta disponible”)

Diagrama de Procesos del Sistema (Detalle del proceso “Selecciona equipo y plan tarifario”)

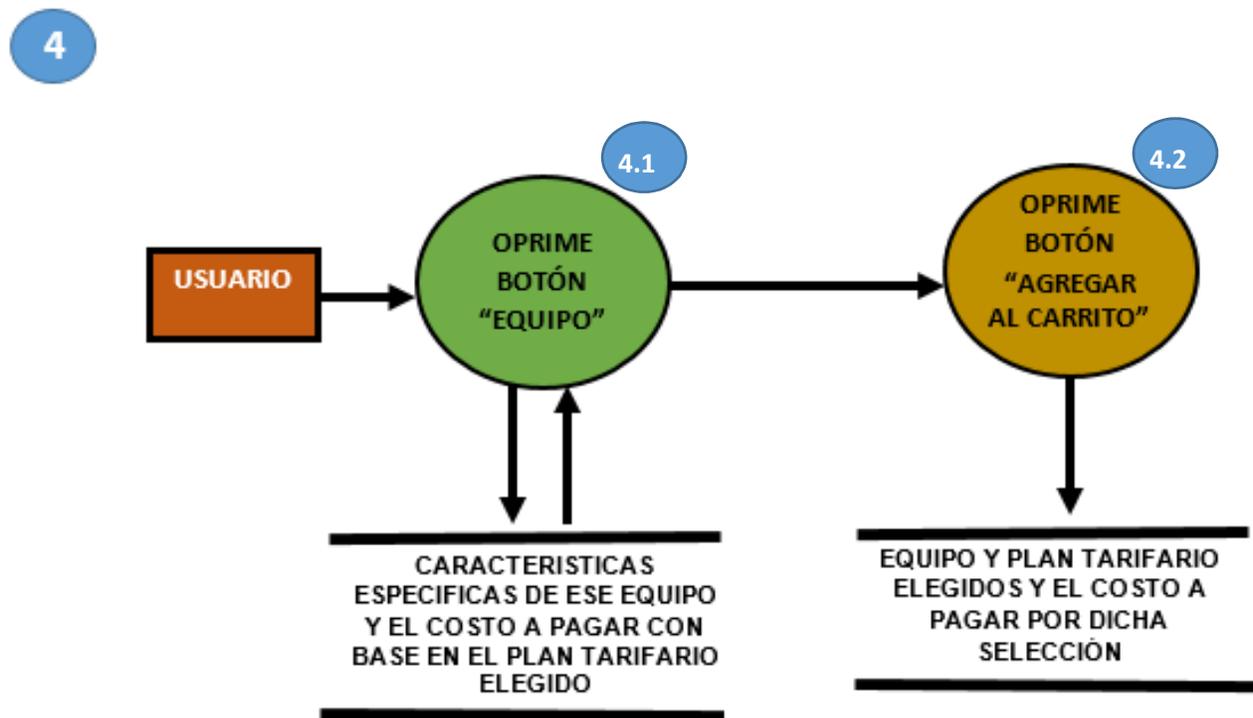


Figura 4.1.3.2.5 diagrama de procesos del sistema  
(detalle del proceso “Selecciona equipo y plan tarifario”)

## Diagrama de clases

El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación.

- La clase define el ámbito de definición de un conjunto de objetos.
- Cada objeto pertenece a una clase.
- Los objetos se crean por instanciación de las clases.

Cada clase se representa en un rectángulo con tres compartimientos como se muestra en la figura 4.1.3.8

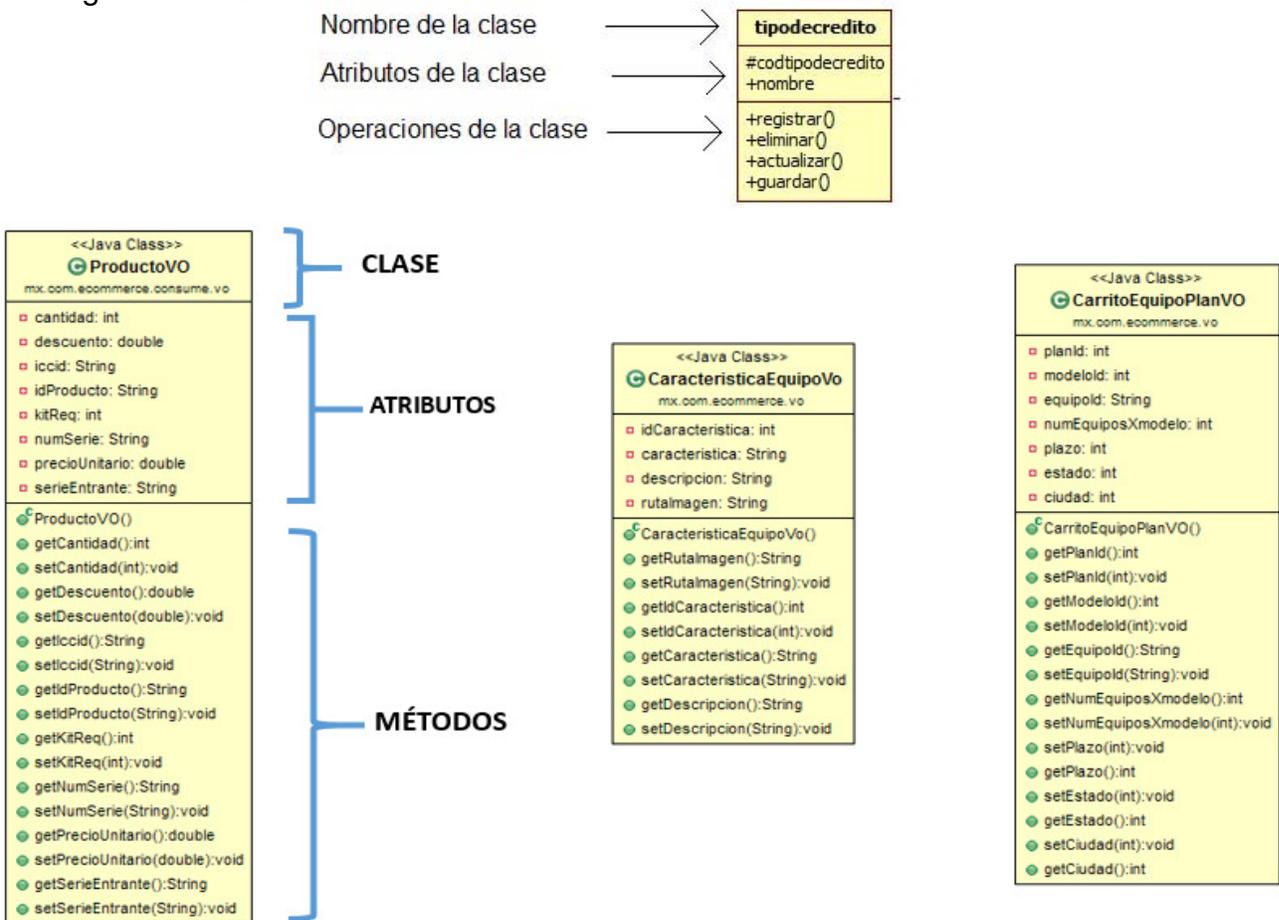


Figura 4.1.3.8 Diagrama de clase

Los diagramas de clase son importantes no solo para la visualización, especificación y documentación del modelo estructural, pero también para la construcción de sistemas ejecutables. Ingeniería hacia adelante e ingeniería inversa. En la figura 4.1.3.9 se muestra el diagrama de clases del Módulo de autoselección de equipos y planes tarifarios para telefonía móvil.

#### **4.1.4 Diccionario de datos**

En diccionario de datos es una manera de documentar una bases de datos dónde se deposita la información como nombre, tipo y descripciones, también encontramos información relacionada con los diferentes tipos de registros, privilegios de los usuarios y estadísticas.

Entre sus funciones se pueden encontrar las siguientes:

- Describir las distintas características de cada uno de los módulos la Base de Datos.
- Proveer una descripción de cada tabla y explicar el propósito de la misma.
- Muestra la estructura de cada tabla, incluyendo sus campos y sus tipos de datos, todos los índices y todas las vistas.
- Describir reglas, como pueden ser el uso de valores no nulos o valores únicos.

En particular, todos los diversos esquemas, (externo, conceptual e interno), se almacenan físicamente en el diccionario, tanto en forma fuente como en forma objeto. Un diccionario amplio incluirá también las referencias cruzadas que indican, por ejemplo qué partes de datos utiliza cada programa, qué informes necesita cada departamento, etc. De hecho, el diccionario puede integrarse a la base de datos que describe, y, por tanto, incluir su propia descripción.

La figura 4.1.4.1 muestra un extracto del diccionario de datos utilizado en la construcción del sistema, un ejemplo de las tablas utilizadas.

Tablas	
Nombre	Acronimo
Catalogo Bancos	CTCPPBNC
Catálogo de características de familia	CTCPPCARFAM
Catálogos de características de Modelos telefonicos	CTCPPCARMDL
Catalogo de Ciudades	CTCPPCIUDAD
Catalogo de clientes ecommerce	CTCPPCLNECO
catalogo de login de Cliente	CTCPPCLNLGN
Catalogo de Colores para los equipos	CTCPPCLR
Catalogo de Comites en el sistema	CTCPPCMT
Catálogo de canales de venta	CTCPPCNL
Catálogo de Conversion de la oferta comercial	CTCPPCNVOFT
Catalogo de Capacidad de memoria	CTCPPCPD
Catalogo de estados	CTCPPESTADO
Catálogo de familias	CTCPPFAM
Catálogo de familias de Oferta Comercial	CTCPPFAMOFC
Catalogo de imagenes de características de modelos de teléfonos	CTCPPIMGCAR
Catálogo de Imagenes por equipo	CTCPPIMGEQP
Catálogo de Imagenes de Familia	CTCPPIMGFAM
Catálogo de Imagenes de Marcas de equipos telefonicos	CTCPPIMGMRC
Catálogo de Imagenes por plan	CTCPPIMGPLN
Catálogo de Marcas de ModelosTelefonicos	CTCPPMRC
Catálogo de marcas destacadas	CTCPPMRC DST
Catálogo de negocio	CTCPPNGC
Catálogo de planes	CTCPPPLN
Catálogo de planes de Oferta comercial	CTCPPPLNOFC

**Tabla 4.1.4.1 Ejemplo de las tablas utilizadas**

La figura 4.1.4.2 muestra los campos que componen una de las tablas, así como su acrónimo, el tipo de dato, su longitud, un breve comentario y si el campo es una llave primaria o foránea

Campo(s) of "CTCPPBNC" Table						
Nombre	Acronimo	Tipo de dato	Longitud	Comentario	Es PK	Es FK
CLAVE SUBROGADA DE BANCO	FIBANCOID	INTEGER	0	Id banco	Yes	No
CLAVE DE BANCO OFC	FCCDGBANCO	VARCHAR2(50)	50	Codigo del banco	No	No
NOMBRE BANCO	FCNOMBREBANCO	VARCHAR2(255)	255	Nombre del banco	No	No
OPERADOR BANCO	FCCIAOPER	VARCHAR2(50)	50	Codigo de operacion	No	No
REGISTRO ACTIVO	FIREGACTIVO	INTEGER	0	Auditoria: Registro Activo	No	No
FECHA DE ALTA	FDALTA	TIMESTAMP		Auditoria: Fecha Alta	No	No
FECHA CAMBIO DE ESTADO	FDCAMSTAT	TIMESTAMP		Auditoria: Cambio de Estatus	No	No
USUARIO	FIUSUARIO	INTEGER	0	Auditoria: Usuario	No	No
IP EQUIPO	FCIPEQUIPO	VARCHAR2(15)	15	Auditoria: IP Equipo	No	No
USUARIO BASE DE DATOS	FCUSERDB	VARCHAR2(30)	30	Auditoria: Usuario Base de Datos	No	No
FECHA MODIFICACION BASE DE DATOS	FDMODDB	DATE		Auditoria: Fecha de Modificación Base de Datos	No	No

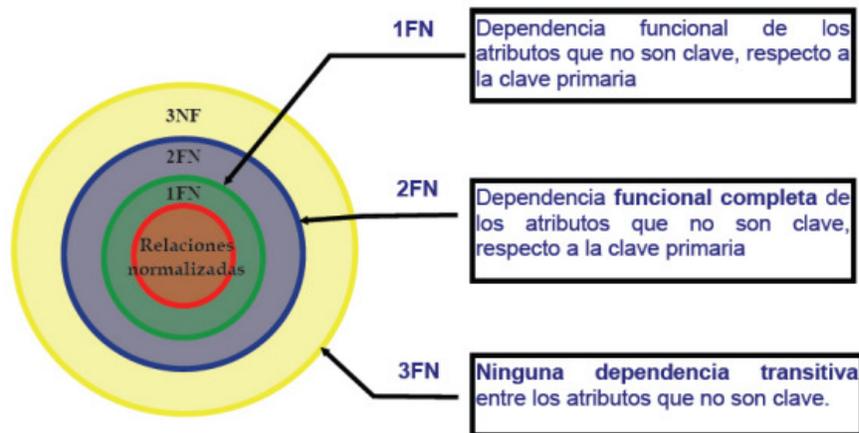
**Tabla 4.1.4.2 Ejemplo de los campos que componen una tabla y sus características**

#### **4.1.5 Normalización de la base de datos**

El objetivo del diseño de las bases de datos relacionales es la generación de un conjunto de esquemas relacionales que permitan almacenar información sin redundancias innecesarias, pero que también permitan recuperar fácilmente esa información. Un enfoque es el diseño de esquemas que se hallen en una forma normal adecuada. La normalización es el proceso de agrupar a los campos de datos en tablas que representan a las entidades y sus relaciones. La razón de la normalización es asegurar que el modelo conceptual de la base de datos funcionará. Esto no significa que una estructura no normalizada no funcionará, sino que puede causar algunos problemas cuando los programadores de aplicación traten de modificar la base de datos para insertar, actualizar o eliminar datos.

El objetivo principal de la normalización es utilizar las técnicas de descomposición sin pérdidas como ayuda en el diseño de bases de datos relacionales. El objetivo principal del proceso de normalización es reducir la redundancia y, por lo tanto, evitar problemas con operaciones de actualización.

Las formas normales son una serie de restricciones que se definen sobre las estructuras relacionales para evitar anomalías al efectuar adiciones, eliminaciones o actualizaciones de tuplas. Con el fin de conseguir que una relación cumpla con una forma normal se efectúa un proceso de descomposición. Ésta implica dividir los atributos de una relación en dos subconjuntos (posiblemente con una intersección no vacía) sin que por ello se pierda alguna información contenida en la relación original. Como se observa en la siguiente figura 4.1.5.1



**Figura 4.1.5.1 Formas normales**

Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- No establece restricciones artificiales en la estructura de los datos.
- Claridad. Debido a que la representación de la información es a través de tablas simples.
- Mínima redundancia. La información no estará duplicada innecesariamente
- Flexibilidad: La información que necesitan los usuarios se puede obtener de las tablas mediante las operaciones del álgebra relacional.
- Facilidad de uso: Los datos están agrupados en tablas que identifican claramente un objeto.
- Seguridad. Consultar o actualizar información es mucho más sencillo de realizar.
- Mejora la independencia de datos. Los programas son independientes de las estructuras, al aumentar la base de datos añadiendo nuevos atributos o nuevas tablas no se afecta a los programas que las manejan.

Uno de los conceptos fundamentales en la normalización es el de dependencia funcional. La dependencia funcional es una noción semántica donde cada dependencia

es una clase especial de regla de integridad y representa una relación de uno a muchos.

La normalización se lleva a cabo en una serie de pasos, cada paso corresponde a una forma normal que tiene ciertas propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto y más fuerte y por lo tanto, son menos vulnerables a las anomalías de actualización.

Las Formas normales son las siguientes:

### **Primera Forma Normal 1FN**

Una tabla se encuentra en primera forma normal si y solo si todas las columnas de todas las tablas contienen solo valores atómicos, es decir un único valor. Establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

### **Segunda Forma Normal 2FN**

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la llave primaria de la tabla para identificarlos.

Una vez alcanzado el nivel de la Segunda Forma Normal, se controlan la mayoría de los problemas de lógica. Podemos insertar un registro sin un exceso de datos en la mayoría de las tablas.

### **Tercera Forma Normal 3FN**

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave.

Cuando las tablas están en la Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria, y no deben haber datos repetidos. Esto provee un esquema limpio y elegante, que es fácil de trabajar y expandir.

### **Cuarta Forma Normal 4FN**

La 4NF se asegura de que las dependencias multivaluadas independientes estén correcta y eficientemente representadas en un diseño de base de datos. La 4NF es el siguiente nivel de normalización después de la forma normal de Boyce-Codd (BCNF).

Una tabla está en 4NF si y solo si esta en Tercera forma normal o en BCNF (Cualquiera de ambas) y no posee dependencias multivaluadas no triviales. La definición de la 4NF confía en la noción de una dependencia multivaluada. Una tabla con una dependencia multivaluada es una donde la existencia de dos o más relaciones independientes muchos a muchos causa redundancia; y es esta redundancia la que es suprimida por la cuarta forma normal.

La forma normal de Boyce-Codd (BCNF) ocurre si una tabla está en tercera forma normal y además todo determinante es una clave candidata. En base de datos un atributo determinante es un atributo del que depende funcionalmente de manera completa algún otro atributo.

### **Quinta Forma Normal 5FN**

La quinta forma normal se refiere a dependencias que son extrañas. Tiene que ver con tablas que pueden dividirse en subtablas, pero que no pueden reconstruirse.

Una tabla esta en Quinta Forma Normal (5FN) o Forma Normal de Proyección-Unión si está en 4FN y las únicas dependencias que existen son las dependencias de unión de una tabla con sus proyecciones relacionándose entre las distintas proyecciones mediante la clave primaria o cualquier clave alternativa.

La 5FN se emplea cuando en una misma tabla tenemos mucha información redundante, con pocos atributos o cuando una tabla posee una gran cantidad de atributos y se hace por ello inmanejable.

Para conseguir que una tabla 4FN con gran cantidad de atributos esté en 5FN, se parte la tabla original en tantas tablas como se desee, teniendo cada una de ellas en común con las demás los campos que forman la clave primaria en la tabla original.

Básicamente, las reglas de normalización están encaminadas a eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas.

A continuación se muestra el proceso de normalización de la base de datos del sistema utilizando una muestra de los datos manejados.

La Tabla 4.1.5.2 muestra el ejemplo de los datos sin normalizar

Catálogo de Clientes Ecommerce (CTCPPCLNECO)														
ID_Orden	ID_Cliente	NOMBRE	Ap_paterno	Ap_materno	Calle	Número	Colonia	Delegación	ID_Tel	Modelo	Memoria Interna	Sistema Operativo	Fecha de alta	Plan
0001	C0001	Benito	Chacón	León	Rio Churubusco	2301	agrícola oriental	Iztacalco	T0001	Xperia 3	32 G	Android	30 marzo 2015	Prepago
0002	C0002	Laura	barrera	Rangel	Palmas	709	Química	Nezahualcóyotl	T0002	IPhone 5	16G	iOS	3 febrero 2015	Pospago
0002	C0002	Laura	barrera	Rangel	Palmas	709	Química	Nezahualcóyotl	T0012	IPhone 6	16 G	iOS	3 febrero 2015	Pospago
0003	C0003	Liliana	Valdés	Brito	Rio Mixcoac	75-A	Los morales	Lomas de Chapultepec	T0003	IPhone 5	8G	iOS	4 noviembre 2014	Pospago
0003	C0003	Liliana	Valdés	Brito	Rio Mixcoac	75-A	Los morales	Lomas de Chapultepec	T0009	IPhone 4S	16 G	iOS	4 Noviembre 2014	Pospago
0004	C0004	Arturo	Alcántara	Calderón	Magdalena mixiuca	45	del valle	Chalco	T0004	Moto G	64 G	Android	4 enero 2015	Prepago
0005	C0005	Iliana	Sánchez	Poulett	lago azul	23	cerrada Coyoacán	Coyoacán	T0005	Galaxy S6	32 G	Android	5 marzo 2015	Pospago
0005	C0005	Iliana	Sánchez	Poulett	lago azul	23	cerrada Coyoacán	Coyoacán	T0005	Galaxy S6	32 G	Android	5 marzo 2015	Pospago
0005	C0005	Iliana	Sánchez	Poulett	lago azul	23	cerrada Coyoacán	Coyoacán	T0011	IPhone 4S	16 G	iOS	5 marzo 2015	Prepago
0005	C0005	Iliana	Sánchez	Poulett	lago azul	23	cerrada Coyoacán	Coyoacán	T0022	Xperia Z4	32 G	Android	5 marzo 2015	Pospago
0006	C0006	Beatriz	Roldán	Tapia	Sierra candela	111 p4	Los morales	Álvaro Obregón	T0006	Xperia 3	32 G	Android	3 abril 2015	Prepago

Tabla 4.1.5.2 Tabla CTCPPCLNECO sin normalizar

## Aplicando la Primera Forma Normal (1FN)

Para que la tabla se encuentre en 1FN debe de tener campos atómicos y no puede tener grupos repetitivos de datos. Se identifican los campos que tengan varios valores para una misma clave (grupos repetitivos). Después, se eliminan estos campos de la tabla anterior y se crea una nueva tabla; la clave principal de esta nueva tabla estará formada por la clave de la tabla anterior más la clave que le corresponda a la nueva. Las tablas 4.1.5.3 y 4.1.5.4 muestran los datos en su primera forma normal

CTCPPCLNECO-orden								
ID_Orden	ID_Cliente	NOMBRE	Ap_paterno	Ap_materno	Calle	Número	Colonia	Delegación
0001	C0001	Benito	Chacón	León	Rio Churubusco	2301	agrícola oriental	Iztacalco
0002	C0002	Laura	barrera	Rangel	Palmas	709	Química	Nezahualcóyotl
0003	C0003	Liliana	Valdés	Brito	Rio Mixcoac	75-A	Los morales	Lomas de Chapultepec
0004	C0004	Arturo	Alcántara	Calderón	Magdalena mixiuca	45	del valle	Chalco
0005	C0005	Iliana	Sánchez	Poulett	lago azul	23	cerrada Coyoacán	Coyoacán
0006	C0006	Beatriz	Roldán	Tapia	Sierra candela	111 p4	Los morales	Álvaro Obregón

**Tabla 4.1.5.3 Tabla en primera forma normal (Grupos no repetitivos)**

CTCPPCLNECO-orden-detalle						
ID_Orden	ID_Tel	Modelo	Memoria Interna	Sistema Operativo	Fecha de alta	Plan
0001	T0001	Xperia 3	32 G	Android	30 marzo 2015	Prepago
0002	T0002	IPhone 5	16G	iOS	3 febrero 2015	Pospago
0002	T0012	IPhone 6	16 G	iOS	3 febrero 2015	Pospago
0003	T0003	IPhone 5	8G	iOS	4 noviembre 2014	Pospago
0003	T0009	IPhone 4S	16 G	iOS	4 Noviembre 2014	Pospago
0004	T0004	Moto G	64 G	Android	4 enero 2015	Prepago
0005	T0005	Galaxy S6	32 G	Android	5 marzo 2015	Pospago
0005	T0005	Galaxy S6	32 G	Android	5 marzo 2015	Pospago
0005	T0011	IPhone 4S	16 G	iOS	5 marzo 2015	Prepago
0005	T0022	Xperia Z4	32 G	Android	5 marzo 2015	Pospago
0006	T0006	Xperia 3	32 G	Android	3 abril 2015	Prepago

**Tabla 4.1.5.4 Tabla en primera forma normal (Grupos repetitivos)**

Aplicando la Segunda Forma Normal (2FN)

**Segunda Forma Normal (2FN)** Dependencia Funcional. Una relación está en 2FN si está en 1FN y si los atributos que no formen parte de ninguna clave dependen de forma completa de la clave principal, es decir que no existan dependencias parciales.

Tomando en cuenta lo anterior, se puede observar que la tabla 4.1.5.3 ya se encuentra en 2FN, sin embargo, en la tabla 4.1.5.4 se observa que los campos “Modelo”, “Memoria interna” y “Sistema Operativo” tienen dependencia parcial solamente con la llave “ID\_Tel”.

Se procede entonces a aplicar la segunda normal y se obtienen las tablas 4.1.5.5 y 4.1.5.6 las cuales ya no tienen dependencias parciales de la llave primaria (que en este caso es una llave compuesta por los campos “ID\_Orden” y “ID\_Tel”)

CTCPPCLNECO-orden-detalle			
ID_Orden	ID_Tel	Fecha de alta	Plan
0001	T0001	30 marzo 2015	Prepago
0002	T0002	3 febrero 2015	Pospago
0002	T0012	3 febrero 2015	Pospago
0003	T0003	4 noviembre 2014	Pospago
0003	T0009	4 Noviembre 2014	Pospago
0004	T0004	4 enero 2015	Prepago
0005	T0005	5 marzo 2015	Pospago
0005	T0005	5 marzo 2015	Pospago
0005	T0011	5 marzo 2015	Prepago
0005	T0022	5 marzo 2015	Pospago
0006	T0006	3 abril 2015	Prepago

**Tabla 4.1.5.5 Segunda forma Normal de la tabla CTCPPCLNECO-orden-detalle**

CTCPPCLNECO-Tel			
ID_Tel	Modelo	Memoria Interna	Sistema Operativo
T0001	Xperia 3	32 G	Android
T0002	IPhone 5	16G	iOS
T0012	IPhone 6	16 G	iOS
T0003	IPhone 5	8G	iOS
T0009	IPhone 4S	16 G	iOS
T0004	Moto G	64 G	Android
T0005	Galaxy S6	32 G	Android
T0005	Galaxy S6	32 G	Android
T0011	IPhone 4S	16 G	iOS
T0022	Xperia Z4	32 G	Android
T0006	Xperia 3	32 G	Android

**Tabla 4.1.5.6 Segunda Forma Normal datos con dependencia parcial a la llave compuesta**

Aplicando la Tercera Forma Normal (3FN)

**La tercera forma normal (3NF)** es una forma normal usada en la normalización de bases de datos. Indica que una tabla está en 3NF si y solo si las dos condiciones siguientes se mantienen:

- La tabla está en la segunda forma normal (2NF)
- Ningún atributo no-primario de la tabla es dependiente transitivamente de una clave primaria

De lo anterior, se puede observar que en la tabla 4.1.5.3 los campos que componen el nombre y la dirección del cliente, dependen tanto del campo "ID\_Orden" como del campo "ID\_Cliente". Las tablas 4.1.5.5 y 4.1.5.6 ya están en la tercera forma normal conforme a la definición anterior

Se procede entonces a aplicar la tercera forma normal y se obtienen las tablas 4.1.5.7 y 4.1.5.8 para hacer que los campos que componen el nombre y la dirección del cliente dependan

CTCPPCLNECO-orden	
ID_Orden	ID_Cliente
0001	C0001
0002	C0002
0003	C0003
0004	C0004
0005	C0005
0006	C0006

**Tabla 4.1.5.7 Tercera forma normal de la tabla CTCPPCLNECO-orden**

CTCPPCLNECO-cliente							
ID_Cliente	NOMBRE	Ap_paterno	Ap_materno	Calle	Número	Colonia	Delegación
C0001	Benito	Chacón	León	Rio Churubusco	2301	agrícola oriental	Iztacalco
C0002	Laura	barrera	Rangel	Palmas	709	Química	Nezahualcóyotl
C0003	Liliana	Valdés	Brito	Rio Mixcoac	75-A	Los morales	Lomas de Chapultepec
C0004	Arturo	Alcántara	Calderón	Magdalena mixiuca	45	del valle	Chalco
C0005	Iliana	Sánchez	Poulett	lago azul	23	cerrada Coyoacán	Coyoacán
C0006	Beatriz	Roldán	Tapia	Sierra candela	111 p4	Los morales	Alvaro Obregón

**Tabla 4.1.5.8 3FN de los datos que dependen solamente de la llave ID\_Cliente**

CTCPPCLNECO-orden-detalle			
ID_Orden	ID_Tel	Fecha de alta	Plan
0001	T0001	30 marzo 2015	Prepago
0002	T0002	3 febrero 2015	Pospago
0002	T0012	3 febrero 2015	Pospago
0003	T0003	4 noviembre 2014	Pospago
0003	T0009	4 Noviembre 2014	Pospago
0004	T0004	4 enero 2015	Prepago
0005	T0005	5 marzo 2015	Pospago
0005	T0005	5 marzo 2015	Pospago
0005	T0011	5 marzo 2015	Prepago
0005	T0022	5 marzo 2015	Pospago
0006	T0006	3 abril 2015	Prepago

**Tabla 4.1.5.5 3FN de la tabla CTCPPCLNECO-orden-detalle**

CTCPPCLNECO-Tel			
ID_Tel	Modelo	Memoria Interna	Sistema Operativo
T0001	Xperia 3	32 G	Android
T0002	IPhone 5	16G	iOS
T0012	IPhone 6	16 G	iOS
T0003	IPhone 5	8G	iOS
T0009	IPhone 4S	16 G	iOS
T0004	Moto G	64 G	Android
T0005	Galaxy S6	32 G	Android
T0005	Galaxy S6	32 G	Android
T0011	IPhone 4S	16 G	iOS
T0022	Xperia Z4	32 G	Android
T0006	Xperia 3	32 G	Android

**Tabla 4.1.5.6 3FN de la tabla CTCPPCLNECO-Tel**

## 4.2 Construcción de la base de datos

Para construir la base de datos se requieren de varios pasos:

- Instalación de Oracle 11g
- Creación de la base de datos
- Como crear las tablas
- Instalación de Oracle

La instalación del software de Oracle se realiza a través de Oracle Universal Installer. Antes de instalar se tienen que revisar los siguientes puntos:

1 Gb de memoria

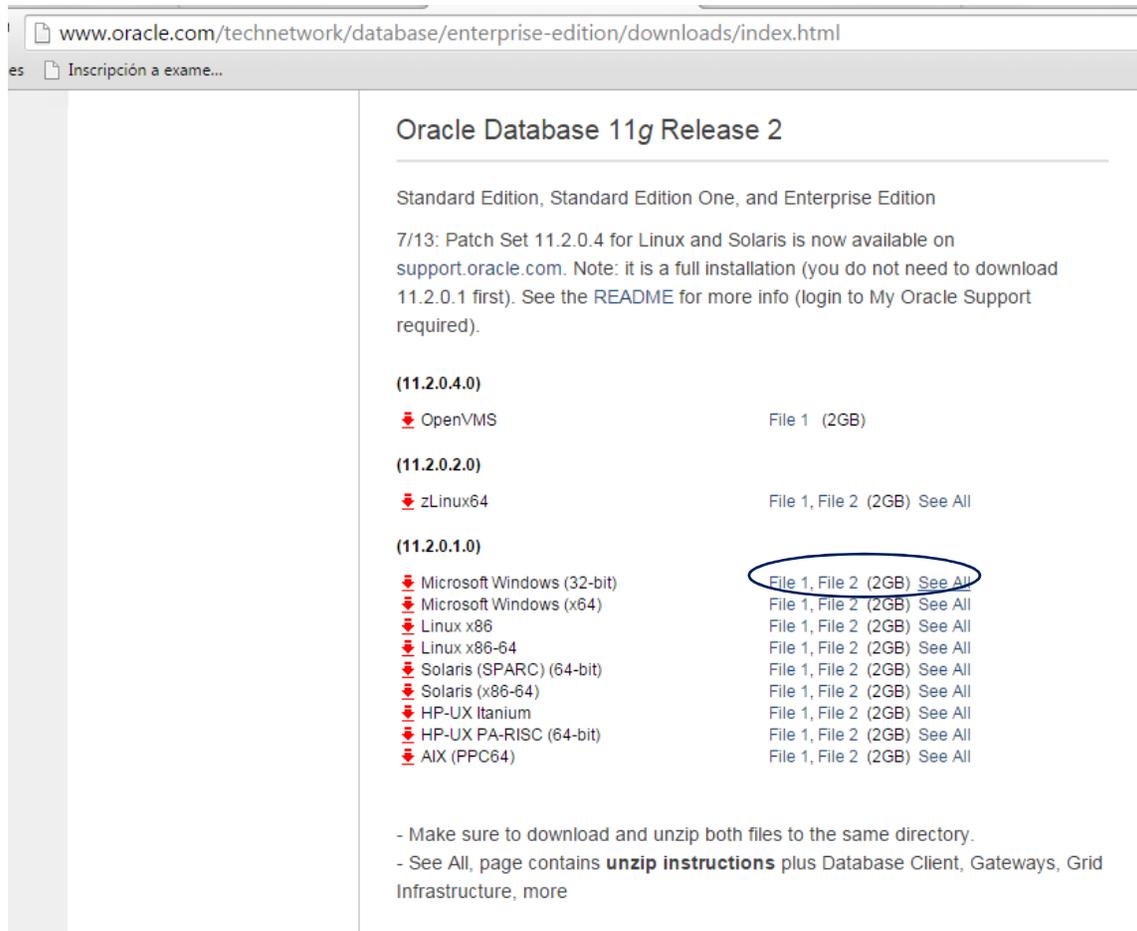
Service packs y parches de SO

Sistema de ficheros adecuados

Para el desarrollo de la base de datos se descarga el software Oracle 11g de la siguiente dirección de internet:

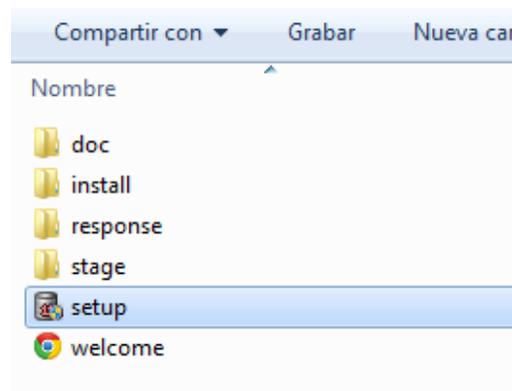
<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112010-win32soft-098987.html>

Para proceder con la instalación de Oracle, se descargan los dos archivos ejecutables de la base de datos, el sistema operativo utilizado en nuestro módulo de autoservicio para selección de equipos y planes tarifarios para telefonía móvil es Microsoft Windows 32-bit, se descargan los dos archivos File1 y File2 ver figura 4.2.1:



**Figura 4.2.1 Descarga de los dos archivos de Oracle 11g**

Se descomprimen los archivos y se le da doble clic al archivo ejecutable “setup”, ver figura 4.2.2



**Figura 4.2.2 Inicia la configuración de Oracle 11g**

El software Oracle inicia la instalación, ver figura 4.2.3

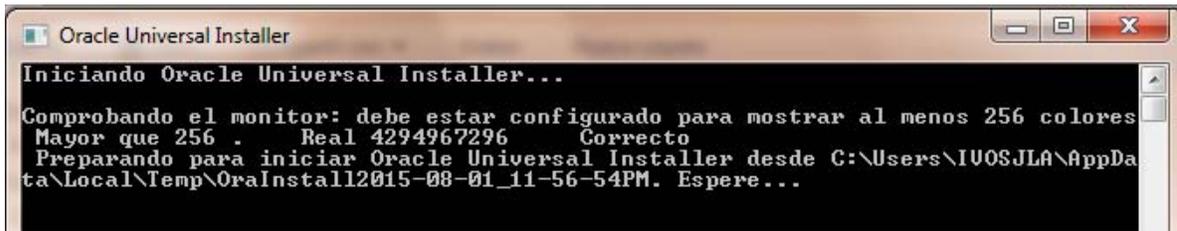


Figura 4.2.3 Proceso de instalación

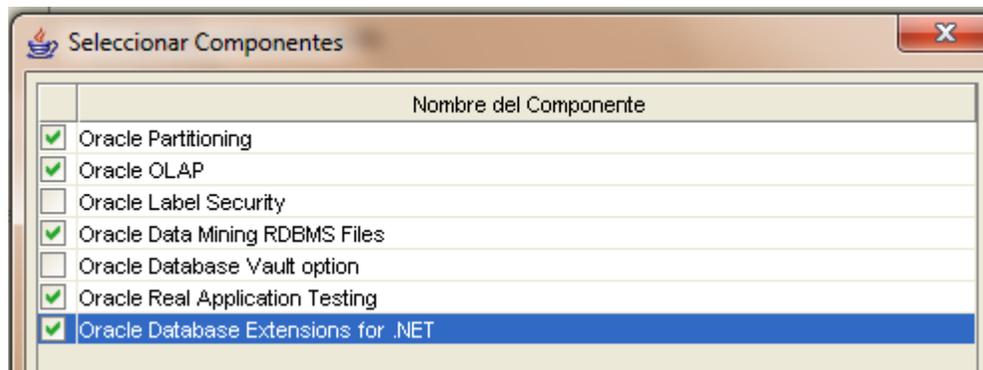
Seleccionamos la opción de crear y configurar base de datos como se muestra en la figura 4.2.4



Figura 4.2.4 Crear base de datos

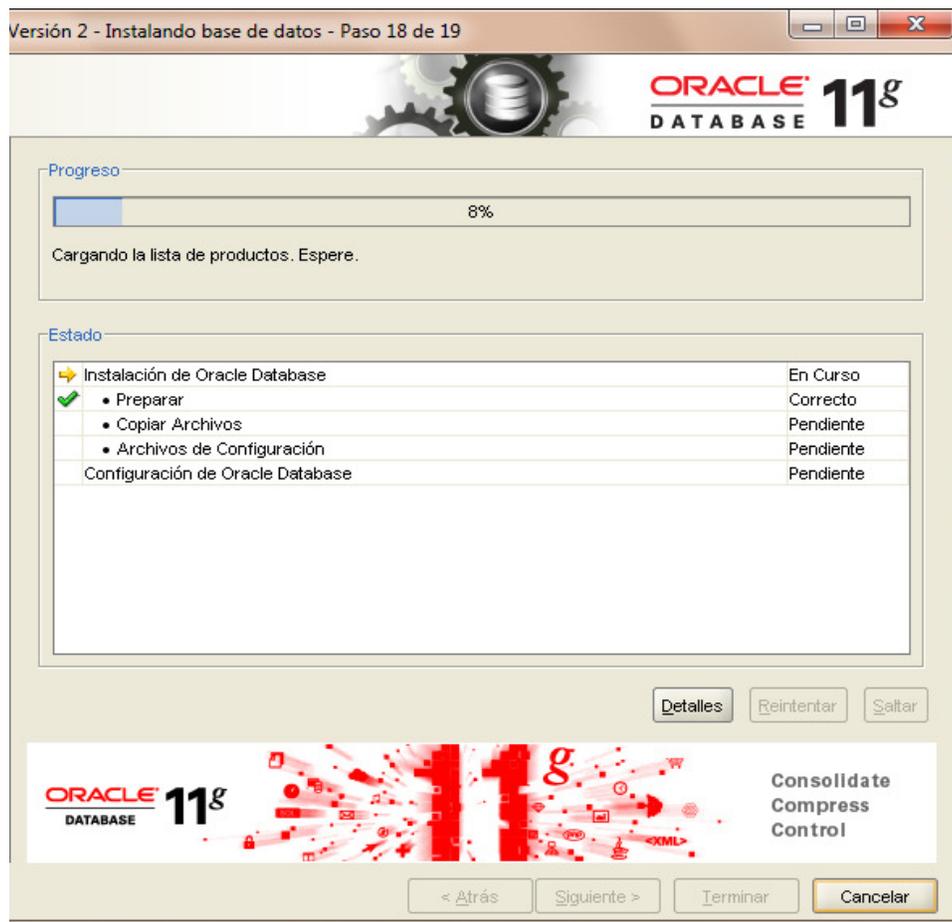
La base de datos a instalar es la opción de servidor y el tipo de instalación es de "instancia única". El tipo de instalación es "avanzada", el idioma es español e inglés.

Para la edición de la base de datos a instalar se selecciona "Enterprise Edition" y en opciones se activa la casilla "oracle database extensions for .NET", ver figura 4.2.5



**Figura 4.2.5 Edición de base de datos**

Oracle da inicio con la instalacion de todos los componentes, prepara, copia los archivos y configura la base de datos de Oracle, ver figura 4.2.6.

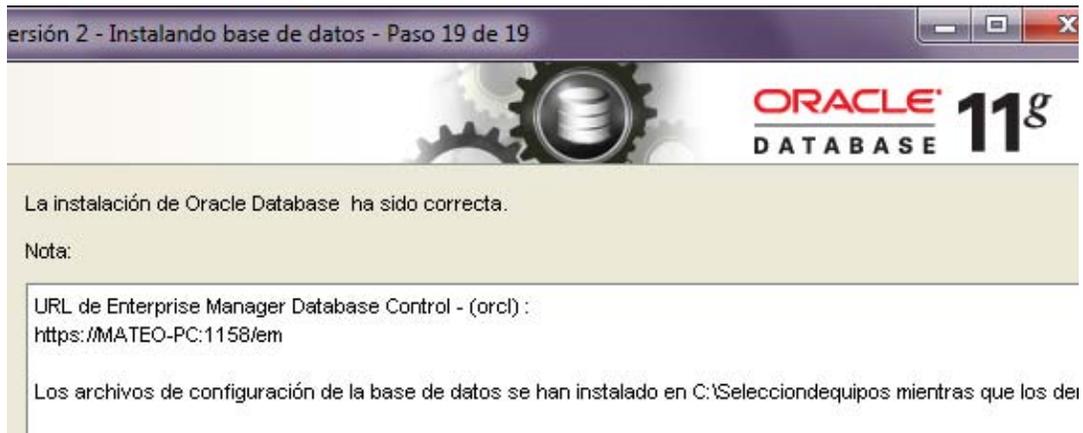


**Figura 4.2.6 Inicia proceso de instalación de Oracle 11g**

La instalación

es

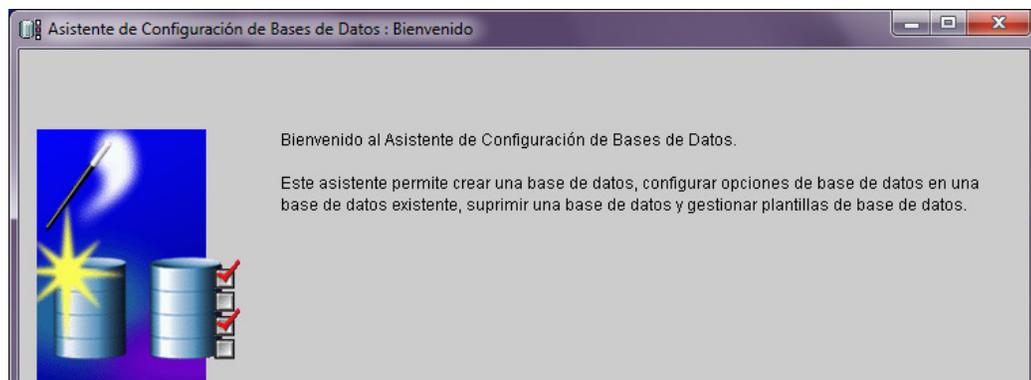
exitosa del software de Oracle 11g.ver figura 4.2.7



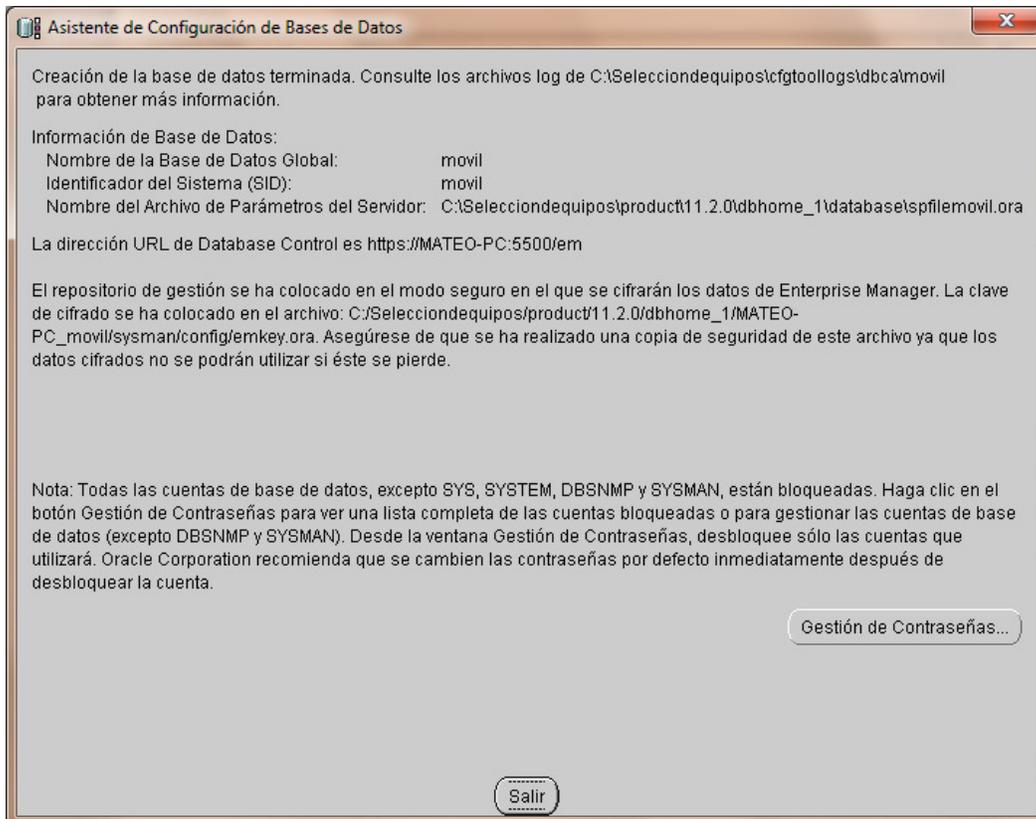
**Figura 4.2.7 Termina la instalación de Oracle**

- Configuración de la base de datos

Se abre el asistente para la configuración de la base de datos y se crea la base de datos, ver figura 4.2.8 y 4.2.9



**Figura 4.2.8 Creacion de base de datos**



**Figura 4.2.9 Creacion de base de datos**

Una tabla en las bases de datos, se refiere al tipo de modelado de datos, donde se guardan los datos recogidos por un programa. Una tabla es utilizada para organizar y presentar información. Las tablas se componen de filas y columnas de celdas que se pueden rellenar con textos y gráficos, se componen de dos estructuras:

- Registro: es cada una de las filas en que se divide la tabla, cada registro contiene datos de los mismos tipos que los demás registros.
- Campo: es cada una de las columnas que forman la tabla. Contienen datos de tipo diferente a los de otros campos. En el ejemplo anterior, un campo contendrá un tipo de datos único, como una dirección, o un número de teléfono, un nombre, etc.

Para crear tablas, empleamos el sistema Erwin. En el módulo de autoselección de equipos de telefonía móvil y planes tarifarios, crearemos la tabla CTCCPARMDL (Catálogos de características de Modelos telefónicos), se da clic en el menú File, new, table, ver la figura 4.2.10

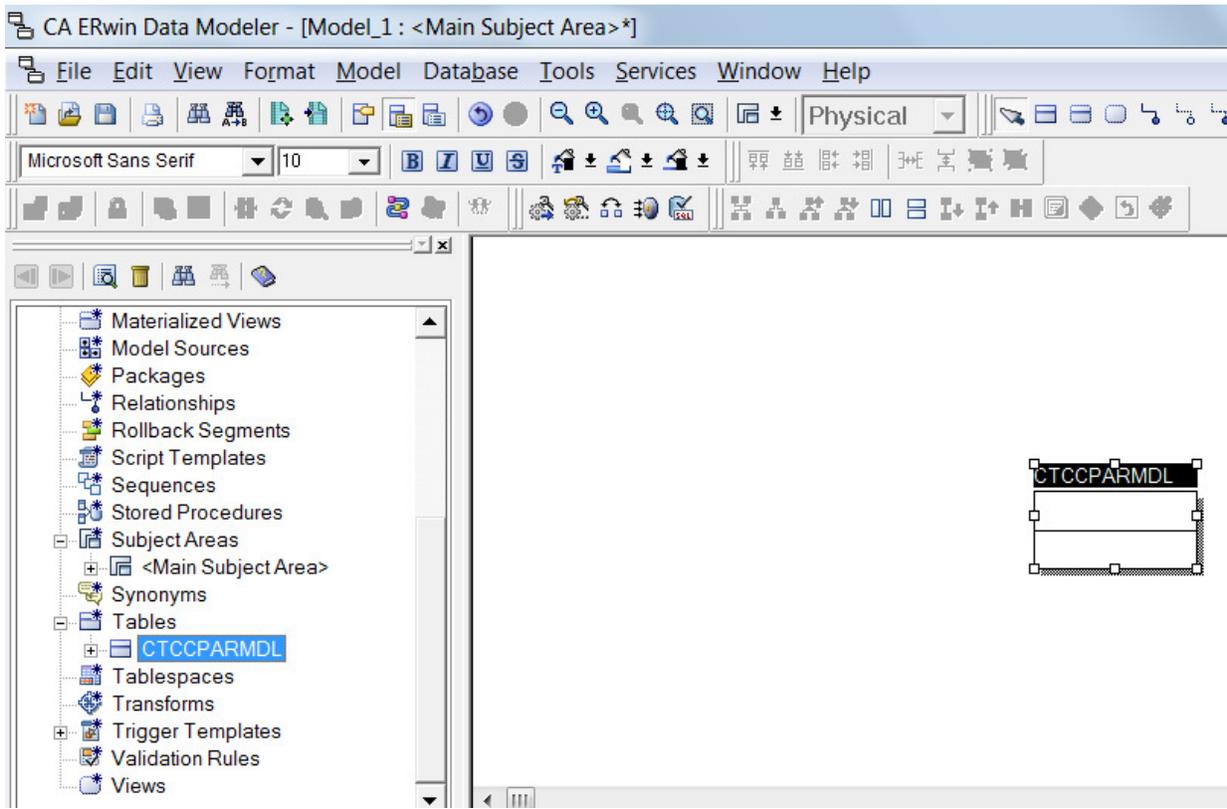


Figura 4.2.10 creación de una tabla

Se observan las relaciones que tiene la tabla con otras dos tablas del sistema. Ver figura 4.2.11

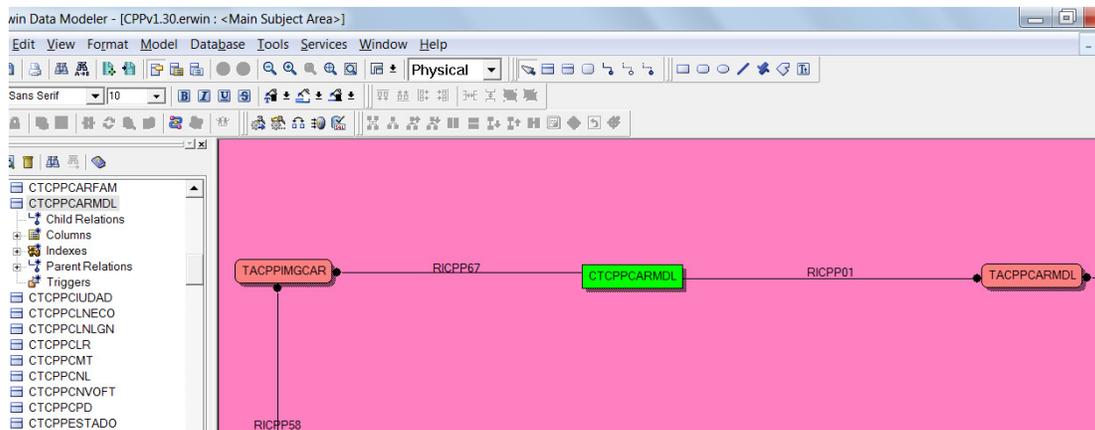
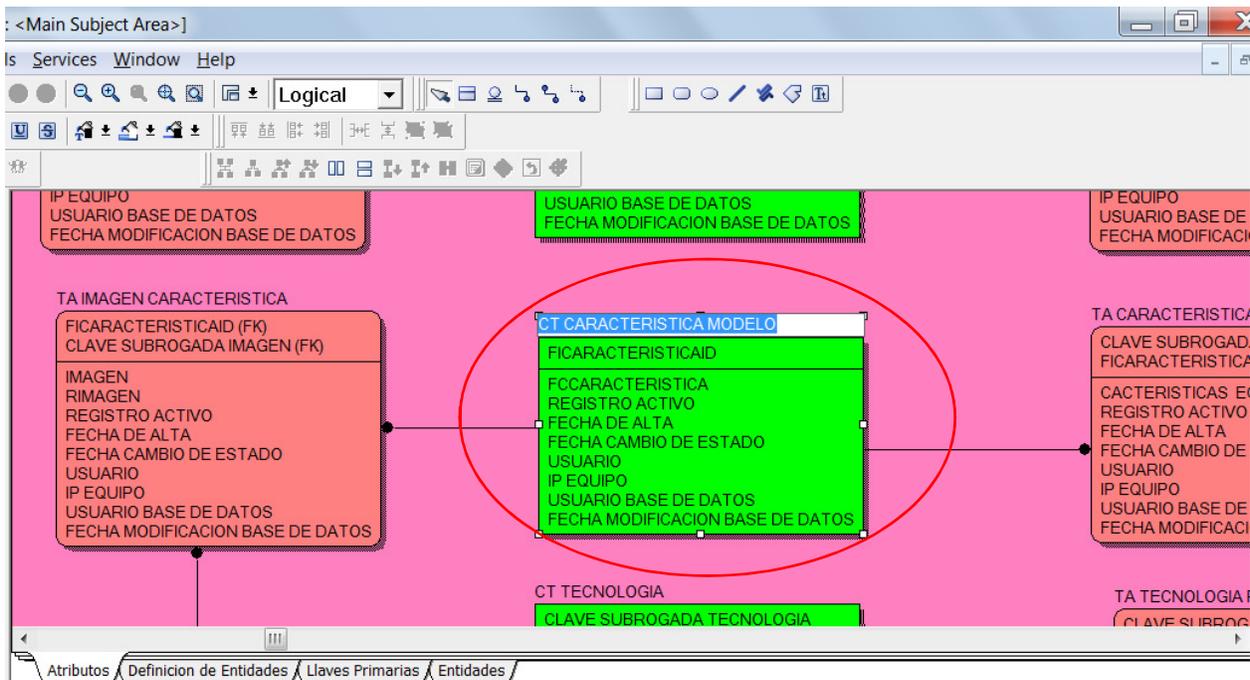


Figura 4.2.11 relación de la tabla CTCCPARMDL

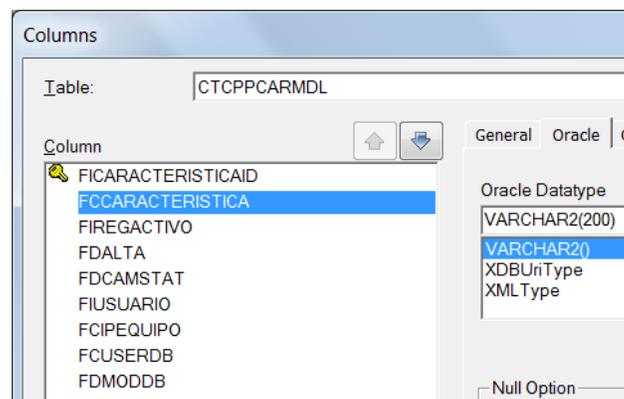
A continuación se muestra la figura 4.2.12 con los campos de la tabla CTCPPCARM DL.



**Figura 4.2.12 Campos de la tabla CTCPPCARM DL**

## COLUMNAS

La columna muestra los tipos de datos que contiene la tabla CTCPPCARM DL, como pueden ver en las siguientes figuras 4.2.13 a la figura 4.2.17, los campos que se tienen son: características del equipo, si está activo el móvil, fecha de alta, imagen del equipo y fecha de modificación.



**Figura 4.2.13 Características de los campos de la tabla CTCPPCARM DL**

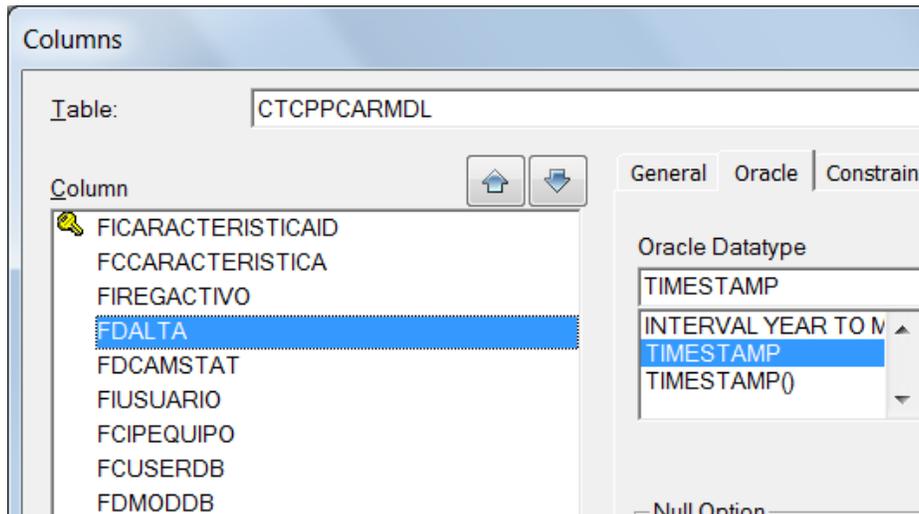


Figura4.2.14

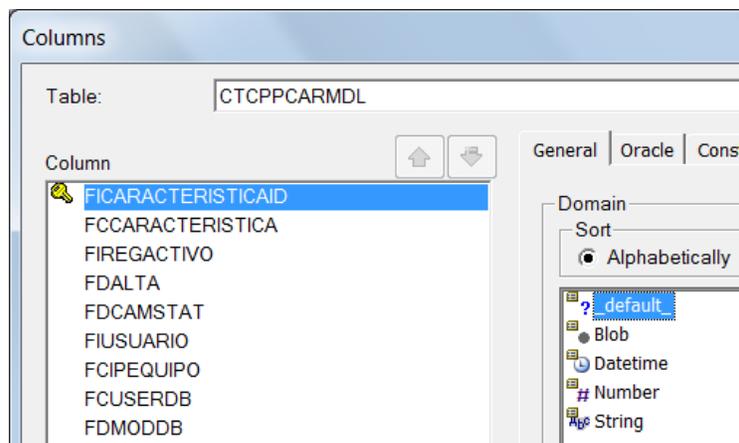


Figura4.2.15

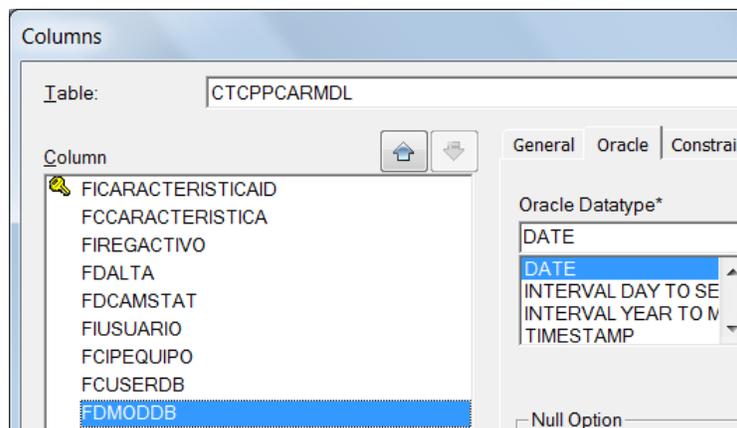


Figura 4.2.16

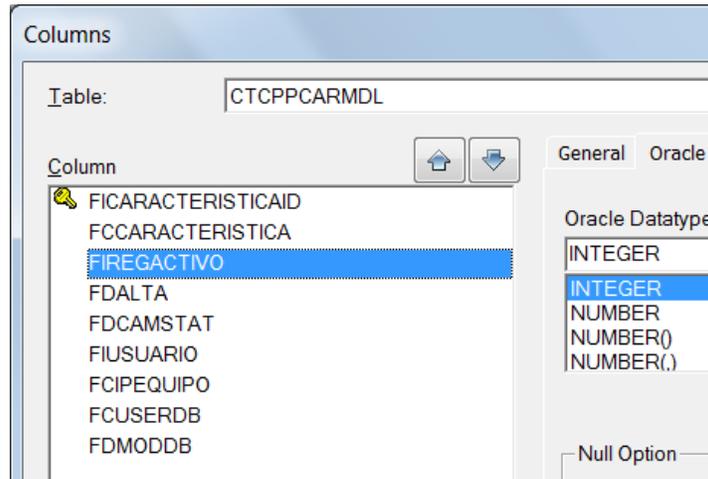


Figura 4.2.17

## INDEX

El índice de una base de datos es una estructura de datos que mejora la velocidad de las operaciones, es un identificador único de cada fila de una tabla y permite un acceso rápido a los registros de una tabla en una base de datos. En la siguiente figura 4.2.18, se tiene el índice PKCTCPPCARACTERISTICAEQUIPO.

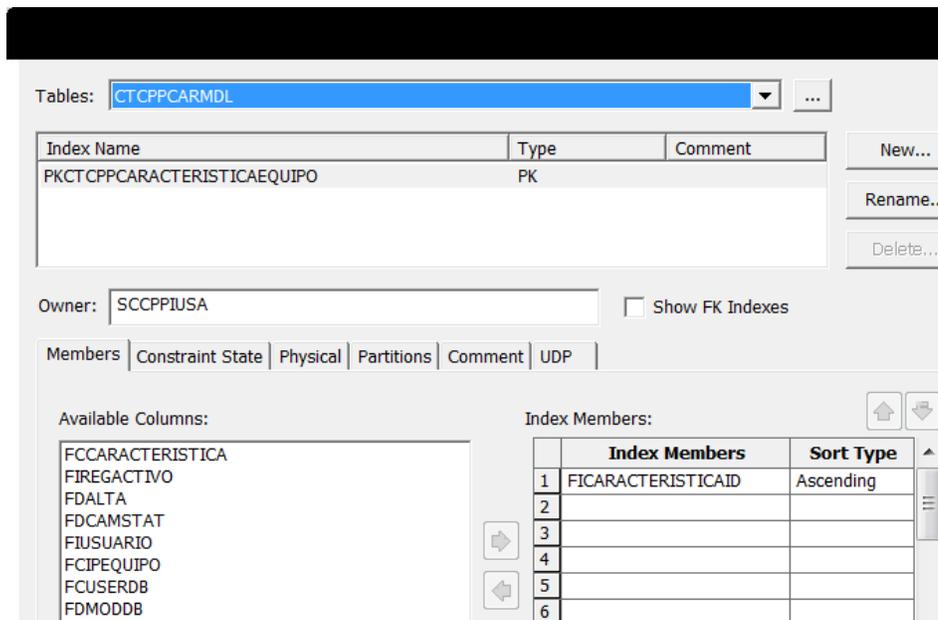


Figura 4.2.18 índice de la tabla CTCCPARMDL

## Llaves primarias

Es un campo, o grupo de campos que identifica en forma única un registro. Ningún otro registro puede tener la misma llave primaria. La llave primaria se utiliza para distinguir un registro con el fin de que se pueda tener acceso a ellos, organizarlos y manipularlos. Ver figura 4.2.19

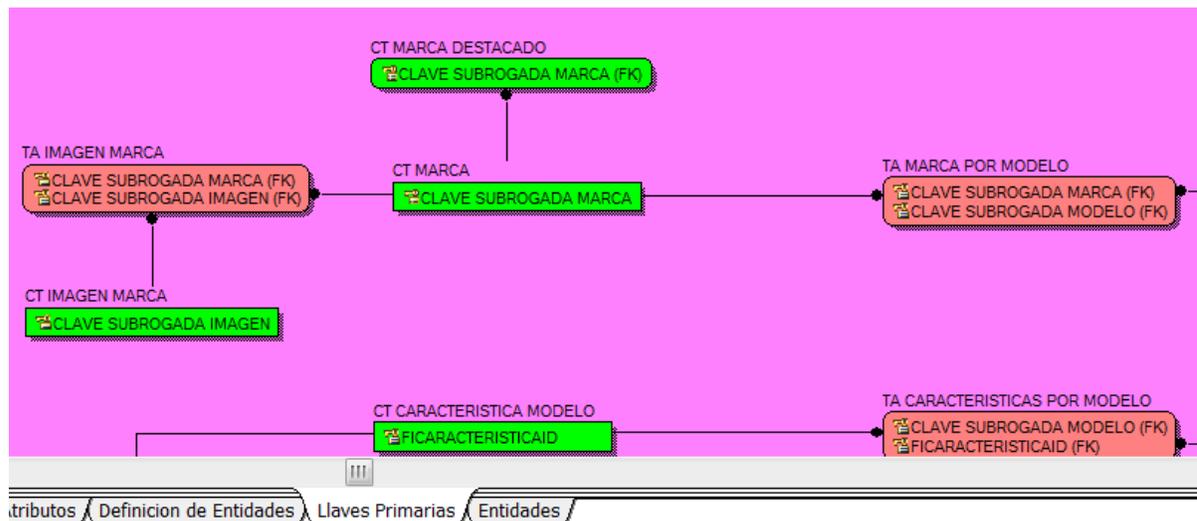


Figura 4.2.19 llave primarias

## 4.3 Construcción de front-end y sitio Web

Para la construcción del front, es necesario el uso de algunas herramientas y realizar algunas configuraciones como la creación de variables de entorno, lo más importante es que todas estas herramientas son software libre, por lo que solamente es necesario descargar las aplicaciones desde los sitios y realizar las configuraciones que se describen a continuación:

1. Descargar **Jboss EAP versión 6.1.0 GA** del siguiente enlace:

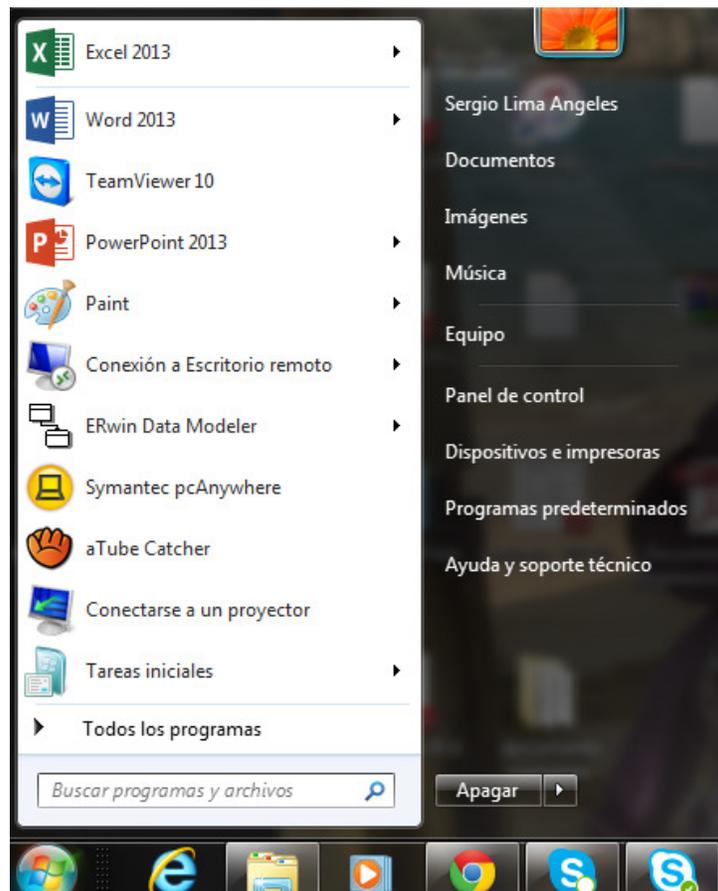
<http://www.jboss.org/download-manager/file/jboss-eap-6.1.0.GA.zip>

2. Descargar e instalar **Java SE Development Kit versión 6.0.45** del siguiente enlace:

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR>

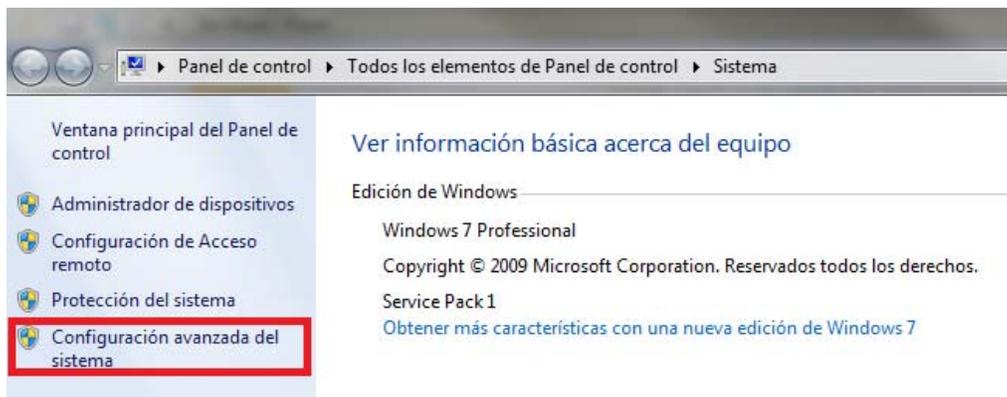
3. Primero instalar el JDK y posteriormente configurar la variable de entorno **JAVA\_HOME** como se detalla a continuación:

1. Para ello es necesario ir a las propiedades del equipo como se ve en la figura 4.3.1



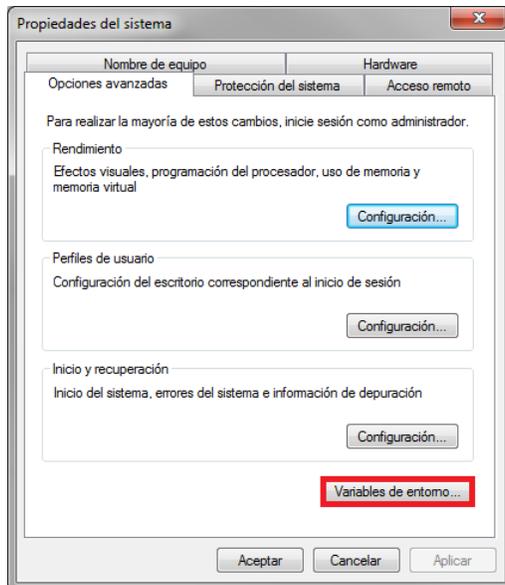
**Figura 4.3.1 Propiedades**

1. Es necesario ir a la configuración avanzada del sistema como se ve en la figura 4.3.2



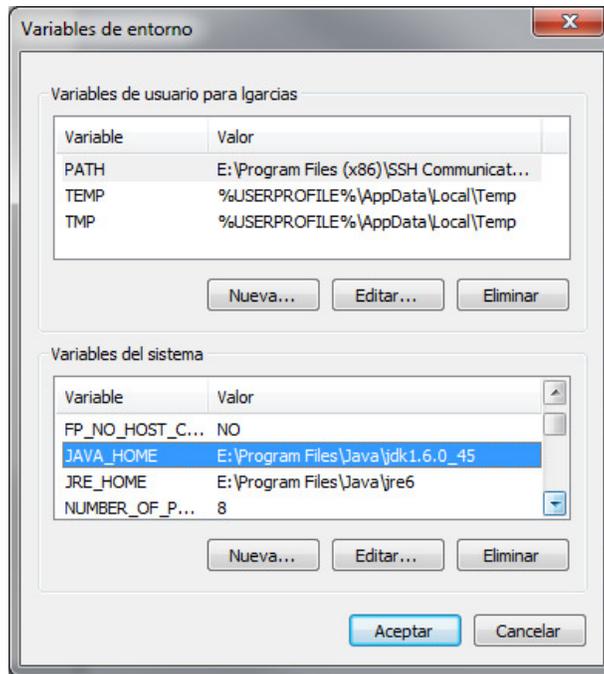
**Figura 4.3.2 Configuración avanzada**

2. En la pantalla de propiedades seleccionar variables de entorno como se muestra en la figura 4.3.3



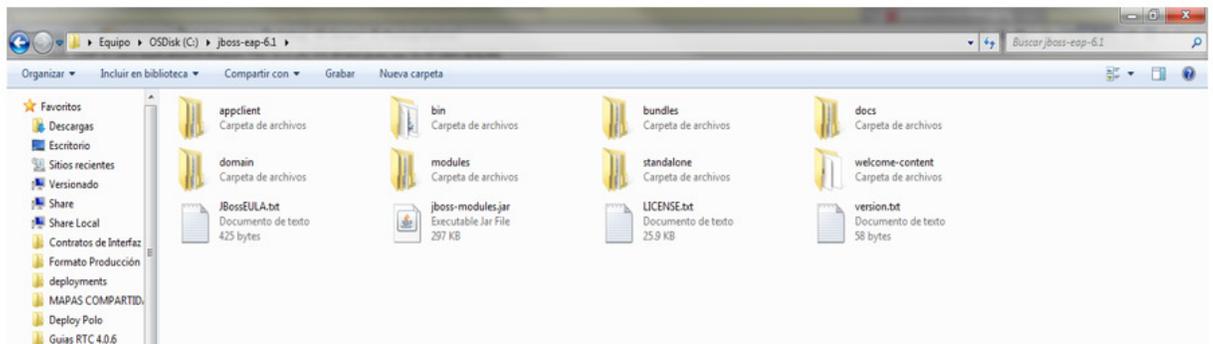
**Figura 4.3.3 Variables de entorno**

1. En la sección de variables de entorno crear la variable JAVA\_HOME, con la ruta de la instalación del JDK como se muestra en la figura 4.3.4



**Figura 4.3.4 Variable JAVA\_HOME**

3. Posteriormente copiar el contenido del archivo descargado jboss-eap-6.1.0.GA.zip en el disco raíz preferentemente, como se aprecia en la figura 4.3.5.



**Figura 4.3.5 Carpeta jboss-eap-6.1.0.GA**



7. La dirección IP es la dirección del servidor donde se están ejecutando los servicios
8. Para obtener el puerto escucha del WS se sumara el Offset más el puerto por default HTTP (8080).



Respecto a la construcción del sitio web, se puede observar que las propuestas vistas en el capítulo 3 en la sección 3.5 Diseño de Front-end a nivel de bloques se cumplieron satisfactoriamente como se muestra a continuación:

- En la pantalla inicial se propuso una sección donde se cuente con el logo de la empresa, un link para acceder a los equipos, un carrito de compra y el botón de búsqueda, así mismo un banner con las promociones, una sección de equipos más buscados y un footer con el logo de la empresa, teléfonos de contacto, aviso de privacidad y redes sociales. Como vemos en la imagen 4.3.9 se integró lo propuesto en el sistema.

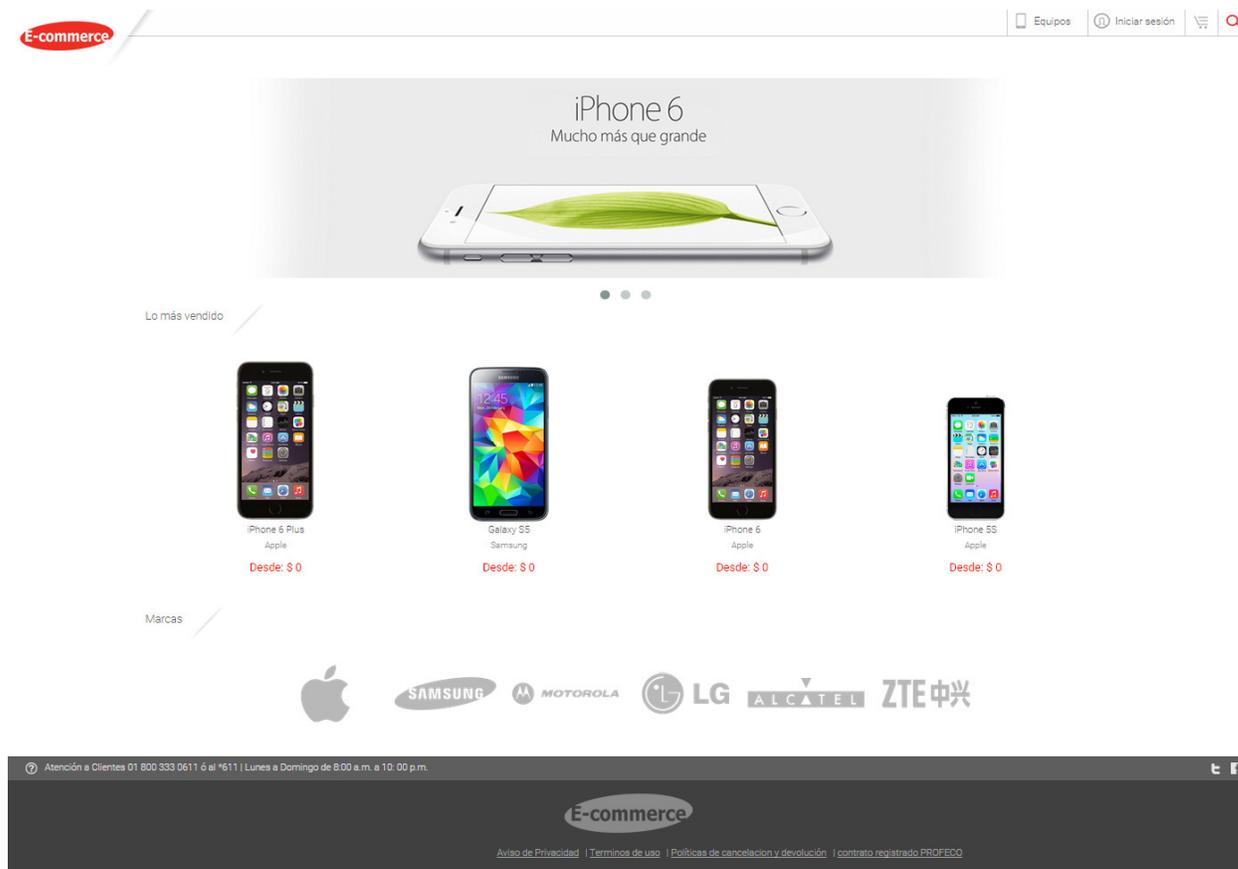
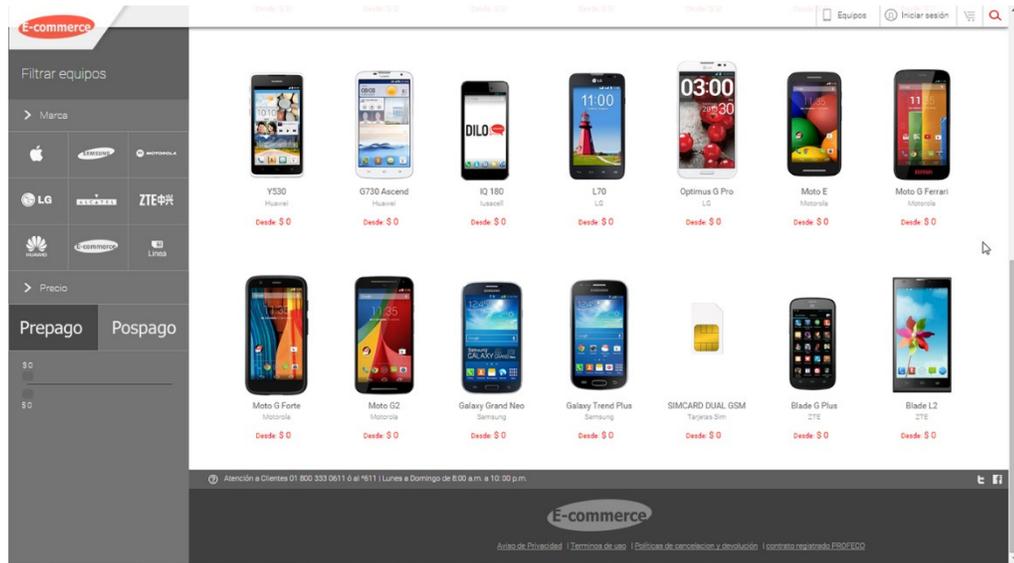


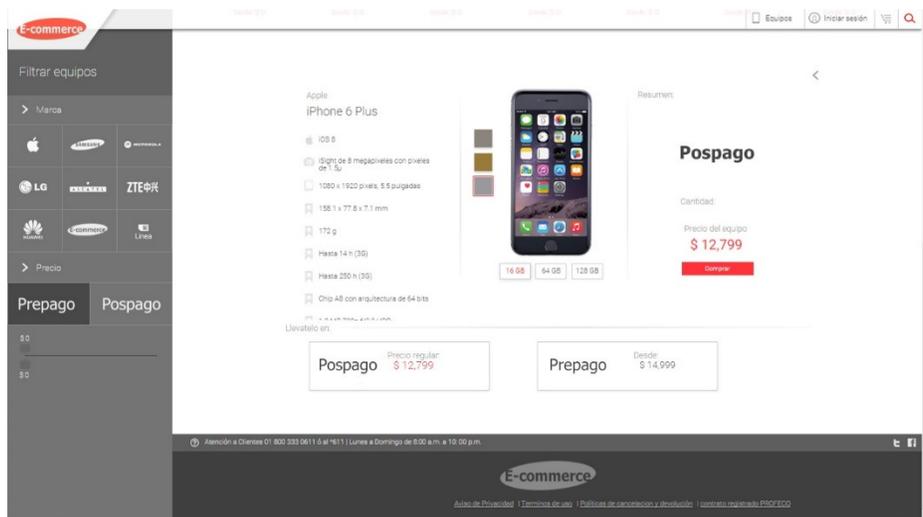
Figura 4.3.9 Pantalla de inicio

- En la pantalla de selección y filtrado de equipo, se propuso el uso de filtros como, marca y precio, así como una herramienta para realizar el zoom de las imágenes de los equipos y como se observa en la figura 4.3.10 se cumplió el requerimiento.



**Figura 4.3.10 Pantalla de selección y filtrado de equipo**

- Finalmente en la figura 4.3.11, se puede apreciar el cumplimiento de lo propuesto para esta sección, que es el mostrar las características principales del equipo, colores en los que está disponible y la información del plan en el que se puede vender el equipo.



**Figura 4.3.11 Características del equipo y plan tarifario**

## **4.4 aplicación de pruebas y mantenimiento**

### **4.4.1 Aplicación de pruebas al sistema**

La etapa de pruebas al sistema implica la operación del mismo en condiciones controladas; estas condiciones incluyen la evaluación del tipo de entradas que se le van a proporcionar al sistema (pasar el mouse o dar clic sobre imágenes, dar clic sobre ligas, introducción mediante el teclado de datos del usuario, etc)

La operación del sistema en la etapa de pruebas se debe de dar tanto en condiciones normales (haciendo que el usuario introduzca datos válidos y correctos, que espere a que el sistema le responda en tiempo para poder realizar la siguiente actividad, etc), como en condiciones anormales (haciendo que el usuario introduzca datos erróneos o en un formato no admitido, que no espere a que el sistema le responda antes de realizar otra actividad, etc); todo esto con el objetivo de corroborar que el sistema entrega los resultados esperados y además poder encontrar posibles fallas o errores.

A continuación se muestran los diferentes tipos de pruebas que se le pueden aplicar a un sistema.

- Prueba de caja negra. Se denomina caja negra a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tomar en cuenta el diseño y funcionamiento interno que le hace posible entregar dichas salidas. Dicho de otro modo, la prueba de caja negra nos ayudará a determinar qué es lo que hace el sistema sin importar cómo lo hace.

Las pruebas de caja negra se centran principalmente en lo que se desea obtener del sistema; es una manera de encontrar casos específicos en los que el sistema siempre va a funcionar o a fallar. Este tipo de pruebas se limitan a que el usuario (también llamado sujeto de prueba o tester) proporcione las entradas necesarias para observar las salidas que proporciona el sistema, salidas que se pueden comprobar y verificar.

- Prueba de caja blanca. Este tipo de pruebas, también conocidas como pruebas estructurales, se centran en los procedimientos, funciones y demás elementos que hacen posible que el sistema entregue los resultados que de él se esperan. Por lo anterior, el diseño de este tipo de pruebas se basa en gran medida en el código fuente del sistema o módulo que se va a probar.

Para llevar a cabo las pruebas de caja blanca, el usuario elige distintos valores de entrada con el fin de evaluar cada uno de los posibles flujos de ejecución del sistema para después observar y evaluar los resultados de salida.

A pesar de que el enfoque de este tipo de pruebas permite aplicarlas para cubrir una gran variedad de casos de uso, podrían pasar por alto partes del sistema que aún no estén terminadas o bien especificaciones o requisitos por parte del usuario que no se estén cumpliendo.

- Prueba de integración. Este tipo de pruebas, también llamadas pruebas integrales son aquellas que se realizan para corroborar que todos los elementos (Ya sean funciones, rutinas, incluso módulos) de un sistema, funcionan e interactúan de forma correcta cuando se ejecutan en la secuencia que se tiene diseñada.

Por lo anterior, este tipo de pruebas sólo puede ejecutarse una vez que los elementos individuales han superado cada uno sus pruebas correspondientes para después poder integrarlos y probarlos como un conjunto

- Prueba de validación. El objetivo de este tipo de pruebas es comprobar que el sistema cumple con las especificaciones bajo las cuales fue diseñado e implementado. Dicho de otro modo, este tipo de pruebas determinarán si el sistema hace lo que el usuario solicitó que hiciera, si cumple con los requerimientos iniciales.

Dentro de las pruebas de validación, existen dos tipos basados en el entorno en el cual se llevan a cabo

- Prueba alfa. Se realizan en un entorno controlado y diseñado por el personal de desarrollo y de sistemas. En este tipo de pruebas, el usuario opera el sistema de forma natural y es observado por el personal de desarrollo quienes toman nota de los posibles problemas y/o fallas que se pudieran presentar.
  - Prueba beta. Estas pruebas se llevan a cabo en el ambiente de producción, es decir, en el entorno normal de operación del sistema. A diferencia de las pruebas alfa, el personal de desarrollo no está presente y no controla las condiciones en las cuales se opera el sistema. En este caso, es el mismo usuario quien registra los problemas o fallas que pudo observar y le informa al respecto al personal de desarrollo.
- Prueba de estrés. En este caso, el objetivo es obtener datos sobre el comportamiento del sistema cuando es operado por múltiples usuarios al mismo tiempo y cada uno de estos usuarios, a su vez, utilizan el sistema de tal forma que exigen de él respuestas inmediatas y sus características más elaboradas, generando carga hasta el punto de hacerlo inutilizable.

Estas pruebas ayudan a realizar el dimensionamiento del hardware y la configuración de la plataforma operativa (Sistema Operativo, Servidor de Aplicaciones, Servidor de Base de Datos, etc) necesarios para la correcta operación del sistema.

Las pruebas de estrés son de las últimas que se le deben realizar al sistema ya que por su carácter poco realista, la situación de carga simulada, muy probablemente no se de en la operación real.

A continuación, en las figuras 4.4.1.1, 4.4.1.2 y 4.4.1.3 se muestra la matriz de las pruebas hechas al sistema.

### Matriz de pruebas hechas al sistema

Prueba	Tipo	Resultado	Comentario
Acceder a la página del sistema mediante un navegador	Caja negra   Alfa	Aprobado	
Oprime botón "Equipos" para desplegar todos los equipos en existencia	Caja negra   Alfa	Aprobado	
Pasar el puntero del mouse sobre un equipo para desplegar una vista previa de sus características	Caja negra   Alfa	Aprobado	
En la pantalla de vista previa de las características del equipo, oprimir el botón "Ver detalle" para desplegar toda la información de ese equipo	Caja negra   Alfa	Aprobado	La información detallada del equipo aparece en forma parcial, no se despliega completa.
En la vista detallada de las características del equipo, oprimir el botón "Viva Prepago" para desplegar el precio del equipo en plan de prepago	Caja negra   Alfa	Aprobado	
En la vista detallada de las características del equipo, oprimir el botón "Smart Libre" para desplegar las características de los planes de renta para los cuales está disponible el equipo	Caja negra   Alfa	Aprobado	
En la vista de las características de los planes, oprimir la combinación de botones "Paquete N" + "N meses" y corroborar que del lado derecho inferior de la pantalla se actualiza el costo correspondiente así como las características del plan elegido	Caja negra   Alfa	Aprobado	

Figura 4.4.1.1 Matriz de pruebas al sistema (primera parte)

### Matriz de pruebas hechas al sistema (continuación)

Prueba	Tipo	Resultado	Comentario
En la vista de las características de los planes, oprimir el botón "Equipo" para regresar a la vista de las características del equipo	Caja negra   Alfa	Aprobado	
En la vista de las características del equipo, oprimir el botón "<" para regresar a la vista de todos los equipos disponibles	Caja negra   Alfa	Aprobado	
En la vista de todos los equipos disponibles, del lado izquierdo, en la sección de "Marca" oprimir uno de los botones de la marca de equipos y corroborar que del lado derecho de la pantalla se muestran resaltados únicamente los teléfonos correspondientes a la marca seleccionada	Caja negra   Alfa	Aprobado	
En la vista de todos los equipos disponibles, del lado izquierdo en la sección "Precio" oprimir el botón "Smart Libre", seleccionar el rango de precios y corroborar que del lado derecho de la pantalla se muestran resaltados únicamente los teléfonos cuyo costo entra en el rango de precios seleccionado	Caja negra   Alfa	Aprobado	
Oprime botón "Equipos" para desplegar todos los equipos en existencia	Validación	Aprobado	El usuario pudo observar desplegados todos los equipos que ofrece la compañía

Figura 4.4.1.2 Matriz de pruebas al sistema (segunda parte)

**Matriz de pruebas hechas al sistema (continuación)**

<b>Prueba</b>	<b>Tipo</b>	<b>Resultado</b>	<b>Comentario</b>
Pasar el puntero del mouse sobre un equipo para desplegar una vista previa de sus características	Validación	Aprobado	Al pasar el puntero del mouse sobre la imagen de uno de los equipos, se despliega una ventana donde el usuario puede observar las características principales del equipos
En la pantalla de vista previa de las características del equipo, oprimir el botón "Ver detalle" para desplegar toda la información de ese equipo	Validación	No aprobado	La información detallada del equipo aparece en forma parcial, no se despliega completa.
Desplegar los costos por tipo de plan	Validación	Aprobado	En las características de un equipo, al oprimir el botón ya sea del plan de prepago o de uno de los planes de renta, se actualiza el costo del equipo
Agregar un equipo al carrito de compras	Validación	Aprobado	Una vez que el usuario ha elegido un equipo y un plan, al oprimir el botón "Agregar al carrito" se muestra un número "1" junto al ícono del carrito de compras

**Figura 4.4.1.3 Matriz de pruebas al sistema (tercera parte)**

#### 4.4.2 Mantenimiento del sistema

El mantenimiento del sistema es la modificación del mismo con el fin de corregir errores, mejorar el rendimiento o bien agregarle funcionalidades. El mantenimiento es una de las fases del ciclo de vida de un sistema y se lleva a cabo después de haber realizado las pruebas correspondientes y analizado el resultado de dichas pruebas.

A continuación se describen los diferentes tipos de mantenimiento que se pueden llevar a cabo a un sistema

- **Mantenimiento Preventivo.** Este tipo de mantenimiento está enfocado a mejorar las características del sistema, sin alterar sus especificaciones iniciales. Suele implementar sentencias que comprueban la validez de los datos de entrada, reestructuración del código para mejorar su legibilidad, o bien, incluir comentarios, etc.
- **Mantenimiento Correctivo.** Su objetivo es localizar y eliminar errores o comportamientos no deseados del sistema observados en la etapa de pruebas.

Los errores o comportamientos no deseados de un sistema se pueden agrupar en las siguientes categorías

- **Errores de programación.** Inconsistencias en el diseño y/o implementación de funciones o rutinas
- **Errores de documentación.** Se dan cuando el manual que se le entrega al usuario para la operación del sistema no corresponde con la funcionalidad real del mismo
- **Errores de procesamiento.** Ocurren cuando la salida que entrega el sistema, no es la deseada por el usuario

- Errores de rendimiento. No es un error como tal, sin embargo, representa un problema para la operación ya que se presenta cuando el tiempo de respuesta del sistema es demasiado elevado
- Mantenimiento Perfectivo. Está enfocado a elevar la calidad del sistema; se logra mediante la reescritura de código para hacerlo más fácil de mantener, más eficiente. Se lleva a cabo también si se requiere modificar el modelo de datos, cambiar las vistas del front-end para mejorar o hacer más amigable su operación.
- Mantenimiento Aumentativo. Se da cuando es necesario adicionar un módulo o una nueva funcionalidad al sistema a fin de satisfacer los requerimientos del usuario.
- Mantenimiento Vertical. En el caso de que se requiera migrar el sistema a un nuevo servidor o conjunto de servidores, dependiendo de cómo se haya llevado a cabo la implementación, se debe llevar a cabo lo que se llama mantenimiento vertical, el cual ayudará a determinar si el sistema continuará operando normalmente e incluso mejorará su rendimiento al migrarlo a los nuevos equipos.
- Mantenimiento Horizontal. Este tipo de mantenimiento ayuda a determinar si el sistema va a continuar operando en condiciones óptimas o incluso va a mejorar su rendimiento al migrarlo de plataforma operativa (Sistema Operativo, Servidor de Aplicaciones, Servidor de Bases de Datos)

#### **4.5 Generación de reportes.**

En el ámbito de la informática, los reportes son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios.

El reporte, de esta forma, confiere una mayor utilidad a los datos. Los reportes tienen diversos niveles de complejidad, desde una lista o enumeración hasta gráficos mucho más desarrollados.

De la misma forma, gracias a los reportes cualquier persona puede proceder a realizar un resumen de datos o a clasificar estos en grupos determinados. Por todo ello, se entiende que estos documentos sean tan importantes en cualquier empresa ya que gracias a ellos cuenta con sus propias bases de datos, realiza recopilaciones de las ventas acometidas en un periodo determinado.

Un reporte es un Documento, generado por el Sistema, que nos presenta de manera Estructurada y/o Resumida, datos relevantes guardados o generados por la misma aplicación de tal manera que se vuelvan útiles para los fines que la Escuela o al Programa Escuelas de Calidad convengan.

El reporte para el caso concreto de la aplicación es medio que sirve para comunicar información relevante, sirve para dar mayor información sobre un tema en específico, como lo es el saber que equipos de telefonía celular son los más vendidos y con que planes tarifarios, como se muestra en la figura 4.5.1. donde se muestran 4 de los equipos más vendidos ordenados de mayor a menor en cuanto a cantidad vendida. Con ello el cliente puede ir directamente a uno de estos equipos para revisar sus características, precio, planes tarifarios en los que se incluye, evitando con ello el revisar cada uno en la vista equipos y evitando con ello que se pierda tiempo en la elección.

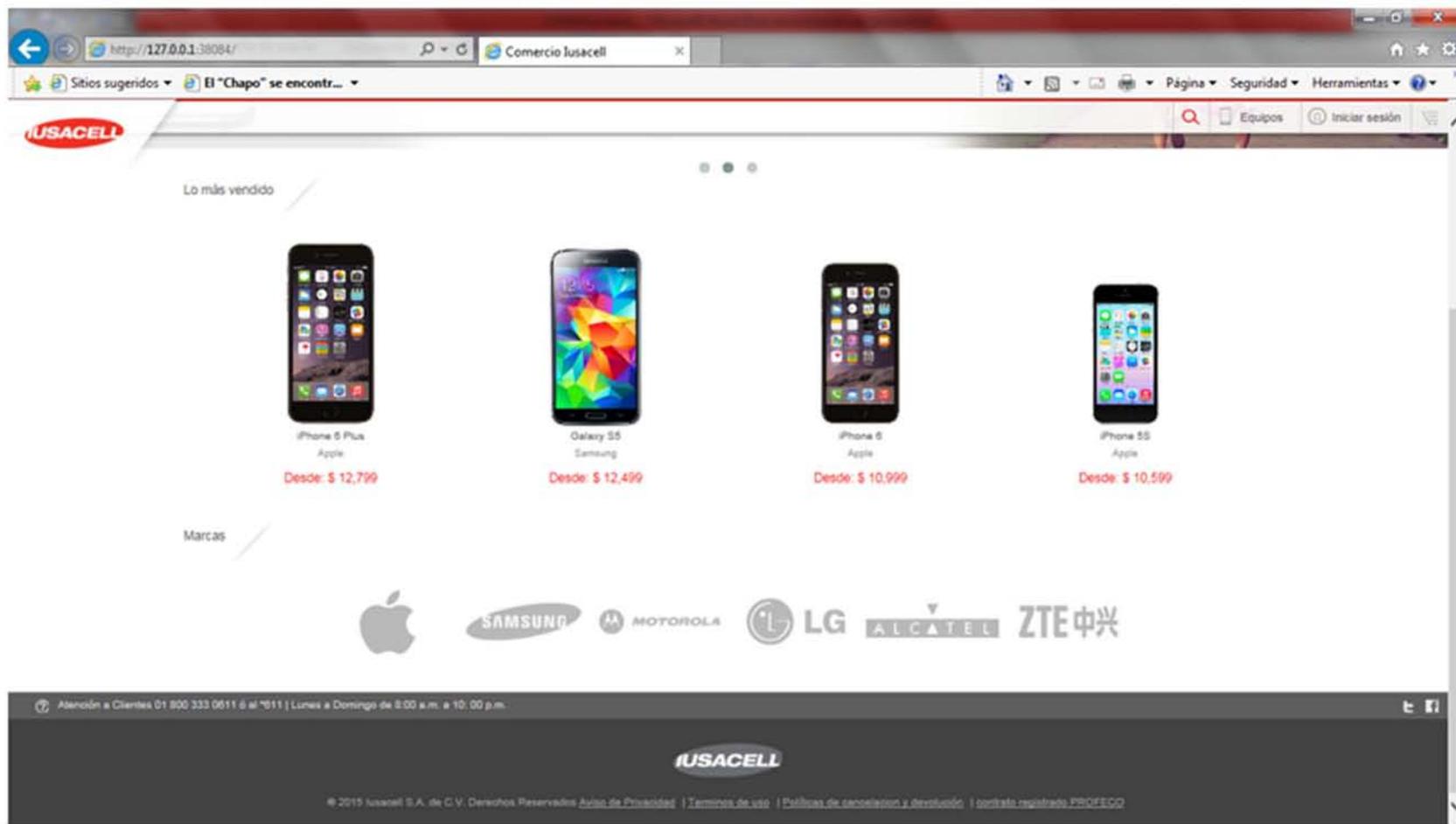


Figura 4.5.1. Equipos más vendidos

Por otro lado en el sistema al dar click en el botón equipo, parecen todos los equipos igualmente ordenados de acuerdo a que equipo se ha vendido más, ordenados de forma descendente, lo cual permite orientar al cliente para la elección de un dispositivo que sea acorde con la popularidad del mercado y que a su vez permita visualizar los datos mínimos para su elección , como los son: Modelo, Marca y Precio, como se muestra en la Figura 4.5.2

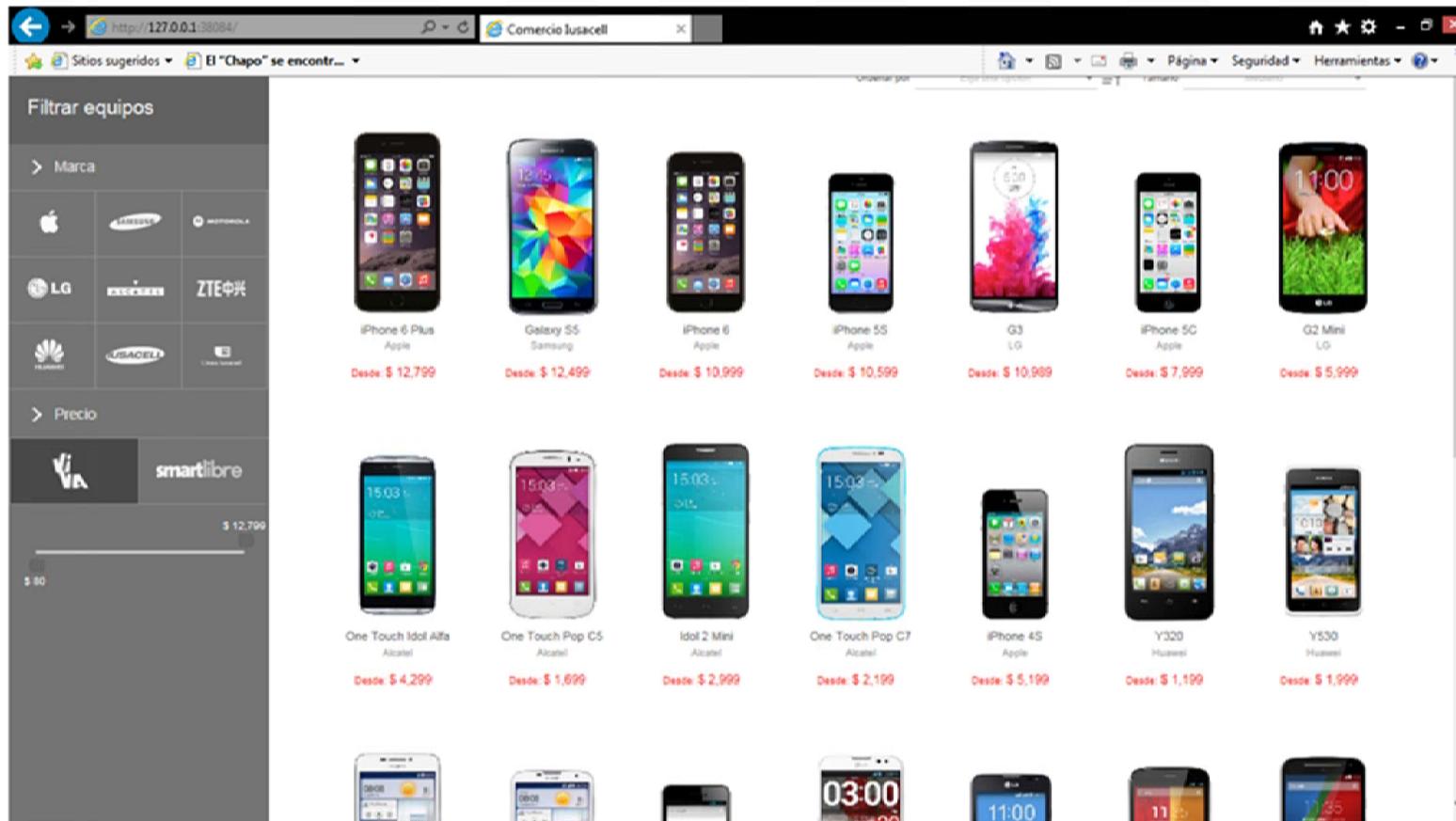


Figura 4.5.2 Elección de equipos

# **CONCLUSIONES**



## CONCLUSIONES

A continuación, se muestran las conclusiones a las que se llegaron después del desarrollo de este trabajo de tesis

- El desarrollo e implementación del sistema cumplió con los requerimientos del cliente para brindar una interfaz web amigable que resolvió el problema del usuario al permitirle seleccionar un equipo celular y un plan de pago sin tener que acudir físicamente a una tienda.
- El utilizar herramientas de software libre, combinadas con software propietario, permiten implementar una solución altamente confiable y con costo muy accesible
- Los elementos utilizados para el desarrollo del sistema, fueron los óptimos ya que permiten que el usuario acceda y ejecute el sistema en su navegador sin tener que instalar manualmente ningún componente adicional.
- Al realizar el análisis, diseño e implementación del sistema comprobamos que los conocimientos adquiridos en las materias impartidas en la carrera de Ingeniería en Computación nos dieron los conocimientos necesarios para llevar a cabo estas actividades.
- La culminación de este proyecto nos permitió aplicar los conocimientos teóricos adquiridos, así como también el aprovechar la experiencia acumulada en nuestra vida laboral y mediante esta combinación aportamos lo mejor de nosotros para crecimiento profesional y el desarrollo mismo de las empresas en la que laboramos.
- Con el desarrollo de este sistema se mejora la experiencia al cliente permitiéndole la elección de un equipo y un plan de manera intuitiva y sencilla, afianzando el interés de la empresa en la mejora continua del servicio de atención al cliente y permitiendo una reducción en los tiempos de respuesta para la selección de los equipos.
- Los reportes que el sistema emite reflejan la situación actual de los equipos más vendidos y facilitan la toma de decisiones.

- Se reducen las filas en la sucursal para la elección de equipos y planes permitiendo un mayor desahogo en las actividades del ejecutivo.
- El desarrollo del sistema se logró de forma exitosa ya que al ser una interfaz web amigable y confiable para el usuario, le permite adquirir el equipo de su agrado y el plan tarifario adecuado a sus ingresos.
- El sistema se diseñó dinámicamente, permitiendo que el usuario se sienta seguro de mover los botones a su conveniencia y tener una mejor vista de las consultas a los equipos, siendo atractivo a cualquier usuario.
- El sistema puede ir creciendo e interactuar con otras aplicaciones administrativas dentro de la empresa.
- El adquirir los conocimientos durante la carrera, me sirvieron de base para poder entender el requerimiento del cliente y ofrecer la mejor solución a su negocio.

# **BIBLIOGRAFÍA**

## Bibliografía

- [1] Elmasri Ramez, "Fundamentos de Sistemas de Bases de Datos", ADDISON-WESLEY.
- [2] David Arnow y Gerald Weiss, "Introducción a la programación con Java", ADDISON-WESLEY.
- [3] Dave Taylor, "Creating Cool HTML 4 web pages", IDG BOOKS.
- [4] Doug Lowe y Barry Build, "Fully updated for java 6, Java for dummies", Wiley Publishing Inc.
- [5] John Cameron, "JSP & JSD, the Jackson approach to software development", IEEE Computer Society Press.
- [6] Gary Hansen, "Diseño y administración de Bases de Datos", Prentice Hall Iberia.
- [7] David Kruglinski, "Sistemas de Administración de Bases de Datos", Mc-Graw-Hill.
- [8] <http://www.altonivel.com.mx/41847-como-escoger-un-plan-de-telefonía-celular.html>, link consultado el día 12 de Junio de 2015
- [9] <http://www.smartblog.es/2012/01/como-elegir-un-smartphone-consideraciones-ante-la-compra-de-un-terminal/>, link consultado el día 12 de junio de 2015
- [10] [http://sg.com.mx/revista/42/tecnología-movil-y-arquitectura#.VcAU7\\_N\\_NHx](http://sg.com.mx/revista/42/tecnología-movil-y-arquitectura#.VcAU7_N_NHx), link consultado el 13 de junio de 2015
- [11] [https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca\\_de\\_JavaScript](https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript), link consultado el día 19 de junio de 2015
- [12] <http://librosweb.es/libro/javascript/>, link consultado el día 19 de junio de 2015
- [13] <https://books.google.com.mx/books?id=gsxVpvEC4iUC&printsec=frontcover&q=javascript&hl=es&sa=X&ei=9GEVfjhHZLjoASYI5GoBA&ved=0CDkQ6AEwBA#v=onepage&q=javascript&f=false>, link consultado el día 19 de junio de 2015
- [14] <http://edumatica.ing.ula.ve/edumatica/Teleclases/Tecniweb/Ingeniería%20Web/Teleclase/Ejecución/Prácticas/JavaScript/Páginas/CaracterísticasGenerales.htm>, link consultado el 20 de Junio de 2015
- [15] [https://librosweb.es/libro/css/capitulo\\_1/breve\\_historia\\_de\\_css.html](https://librosweb.es/libro/css/capitulo_1/breve_historia_de_css.html), link consultado el 20 de Junio de 2015

- [16]<http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>, link consultado el 21 de Junio de 2015
- [17]<http://www.desarrolloweb.com/articulos/831.php>, link consultado el 21 de Junio de 2015
- [18]<http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>, link consultado el 21 de Junio de 2015
- [19][http://www.ecured.cu/index.php/Arquitectura\\_Cliente\\_Servidor](http://www.ecured.cu/index.php/Arquitectura_Cliente_Servidor), link consultado el 21 de Junio de 2015
- [20]<https://www.oracle.com/servers/index.html>, link consultado el 25 de Junio de 2015
- [21]<https://es.wikipedia.org/wiki/Cliente-servidor>, link consultado el 21 de Junio de 2015
- [22][http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/marquez\\_a\\_bm/capitulo5.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf), link consultado el 21 de Junio de 2015
- [23]<http://es.ccm.net/contents/148-entorno-cliente-servidor>, link consultado el 21 el Julio 2015
- [24]<http://es.ccm.net/contents/147-redes-arquitectura-cliente-servidor-en-3-niveles>, link consultado el 21 de Junio de 2015
- [25] <https://es.opensuse.org/Apache>, link consultado el 30 de junio de 2015
- [26]<http://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>, link consultado el 30 de junio de 2015
- [27][https://en.wikipedia.org/wiki/Comparison\\_of\\_application\\_servers](https://en.wikipedia.org/wiki/Comparison_of_application_servers), link consultado el 30 de junio de 2015
- [28]<https://prezi.com/9ulrsdett7ov/servidor-web-cherokee/>, link consultado el 30 de junio de 2015
- [29] <https://raiolanetworks.es/blog/nginx/>, link consultado el 30 de junio de 2015
- [30]<http://www.gridmorelos.uaem.mx/~mcruz//cursos/miic/sybase.pdf>, el link se consultó el 30 de Junio de 2015
- [31]<http://www-01.ibm.com/software/mx/informix/everywhere.html>, link consultado el 30 de Junio de 2015

- [32] <https://es.wikipedia.org/wiki/PHP>, link consultado el 02 de Julio de 2015
- [33] <https://es.wikipedia.org/wiki/JBoss>, link consultado el 02 de Julio de 2015
- [34] <https://es.wikipedia.org/wiki/JQuery>, link consultado el 03 de Julio de 2015
- [35] [http://www.ecured.cu/index.php/Arquitectura\\_en\\_Capas](http://www.ecured.cu/index.php/Arquitectura_en_Capas), link consultado el 4 de Julio de 2015
- [36] [https://es.wikipedia.org/wiki/Cliente\\_liviano](https://es.wikipedia.org/wiki/Cliente_liviano), link consultado el 04 de Julio de 2015
- [37] <https://prestashop5estrellas.wordpress.com/2010/03/29/el-patron-mvc-modelo-vista-controlador/> Consultado el 07 de julio de 2015
- [38] <http://thelozu.blogspot.mx/2013/07/modelo-vista-controlador.html> Consultado el 07 de julio de 2015
- [39] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122> Consultado el 08 de julio de 2015
- [40] <http://www.slideshare.net/mobile/juanvladimir87/bdoo-ventajas-y-desventajas> Consultado el 09 de julio de 2015
- [41] <https://juan0989.wordpress.com/2011/02/12/caracteristicas-de-las-bases-de-datos-objetos-relacionales/1> Consultado el 09 de Julio de 2015
- [42] [http://es.m.wikipedia.org/wiki/Base\\_de\\_datos\\_relacional](http://es.m.wikipedia.org/wiki/Base_de_datos_relacional) Consultado el 09 de julio de 2015
- [43] <https://www.lsi.us.es/docs/doctorado/tesis/Memoria-Tesis-MTGomez.pdf> Consultado el 09 de julio de 2015

# APÉNDICE

## **Organigrama genérico de una compañía de telefonía móvil y descripción de sus responsabilidades.**

Para la operación correcta de un empresa de telefonía celular es necesario contar con recursos tanto humanos como materiales que coadyuven en posicionar a la empresa como líder del mercado, para ello es necesario subdividir el trabajo en áreas para agilizar los procesos de negocio. A continuación se detalla cómo está organizada nuestra empresa en México:

### **Presidencia**

El presidente de la empresa es la máxima autoridad inmediata. Se encuentra ubicado entre los niveles más altos de la organización y sus funciones son las dirigir y controlar el funcionamiento de la compañía, representar a la compañía en todos los negocios y contratos con terceros en relación con el objeto de la sociedad, también convoca y preside las reuniones de la Junta Directiva. El Presidente es el representante Legal de la Coalición Empresarial, para todos los actos judiciales y extrajudiciales.

### **Vicepresidente de servicio al cliente**

Responsable de la estrategia y ejecución para brindarle un excelente servicio al cliente en cuentas corporativas/elite, call centers, tiendas, distribuidores y en línea.

### **Vicepresidente de Mercadotecnia**

Responsable del desarrollo de productos y ofertas, percepción del consumidor, de administrar las marcas que operan en México de manera interactiva y digital, y de toda la publicidad. Además es responsable de patrocinios y de generar estrategias de mercado para ambos segmentos de prepago y pospago.

### **Vicepresidente Ventas y distribución Pospago y prepago**

Responsable de las ventas y distribución de productos y servicios de pospago y prepago, así como de las ventas a negocios pequeños, medianos y grandes.

### **Vicepresidente de Legal, regulatorio y cumplimiento**

Responsable de todos los asuntos legales y de cumplimiento. También responsable de la administración de asuntos regulatorios en todas las áreas de negocio.

### **Vicepresidente Tecnología de la información**

Responsable del desarrollo y la administración de los sistemas que apoyan a todos los segmentos de negocios, redes y servicios. También es responsable de la estrategia de seguridad de la información, estándares y desarrollo de plataformas.

### **Vicepresidente Recursos humanos**

Responsable de supervisar recursos humanos y las relaciones laborales, garantizar una buena comunicación entre todos los niveles de la organización, elaborar y controlar el proceso de reclutamiento, selección, ingreso e inducción del personal, a fin de asegurar la elección de los candidatos más idóneos para los puestos, supervisar y revisar los procesos de nómina a fin de garantizar el depósito oportuno de los empleados y asignados de la empresa, planificar y supervisar el plan de formación de personal.

### **Vicepresidente de Finanzas**

Responsable de dirigir la planeación financiera, contabilidad, impuestos, auditorías, administración de inversiones, finanzas, relaciones con inversores y accionistas, y bienes raíces, ambiente, salud y protección empresarial.

### **Gerencia de Operaciones e ingeniería de la red**

Responsable del diseño, ingeniería, construcción y operación de las redes de telefonía móvil, administración de redes IP, operación, monitorización y optimización, soporte técnico, configuración y puesta en servicio de sistemas.

### **Gerencia de Desarrollo**

Responsable de la ejecución de los desarrollos de sistemas considerando los más altos estándares de desarrollo, pruebas y aseguramiento de la calidad, asignación de recursos para la obtención de resultados. Asegurar que los riesgos estratégicos se consideren de manera apropiada y ayudar al directorio a entender los riesgos de una empresa y formular los planes de acción que se requieran.

### **Gerencia de Planeación de negocios**

Responsable planeación de negocios administración y responsabilidad de los planes de negocio que son claves para lograr los objetivos tácticos y estratégicos.

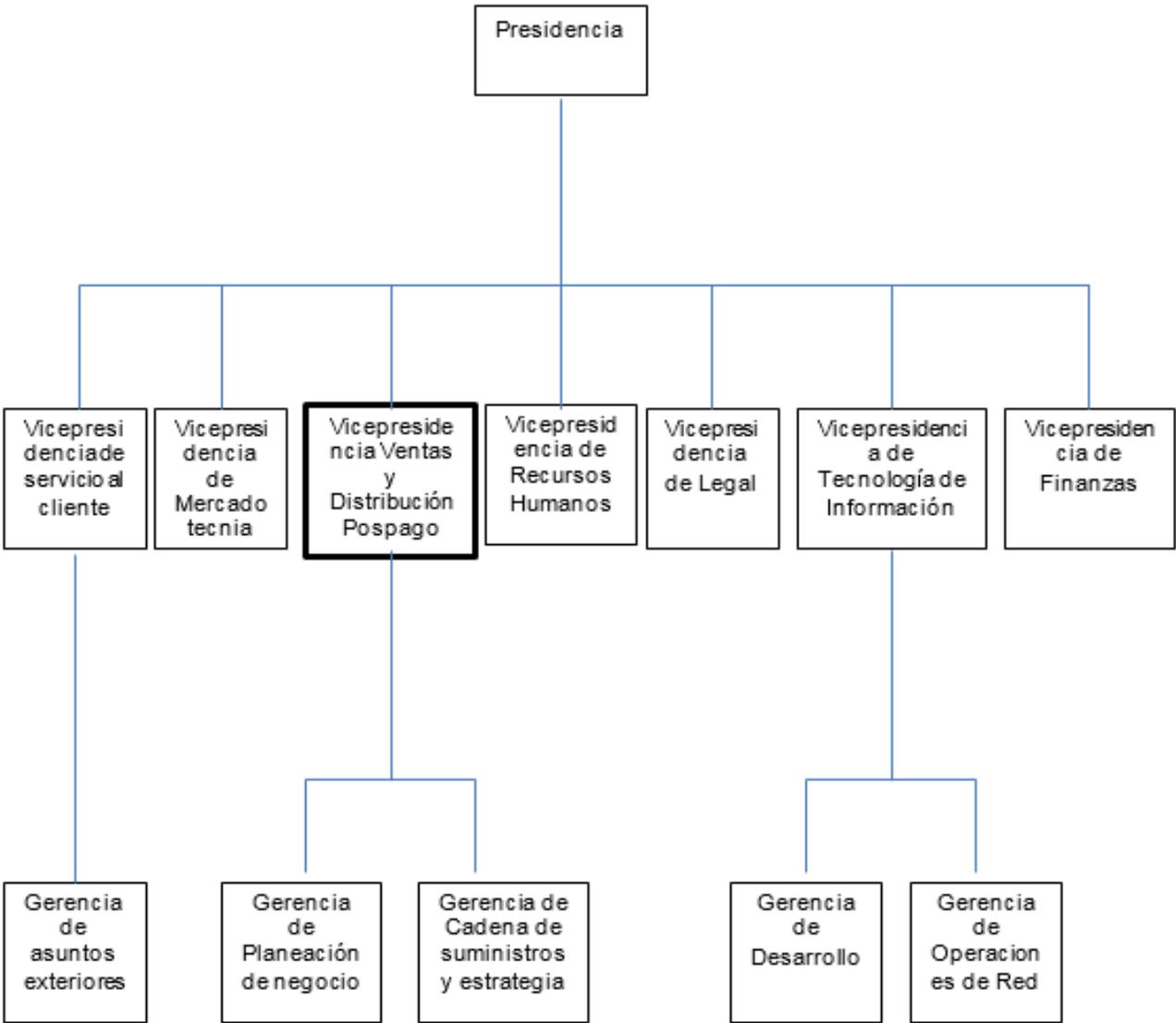
### **Gerencia Cadena de suministros y estrategia**

Responsable de las negociaciones, contratos, planeación del inventario, compras, logística y control de calidad de la cadena de suministros, así como de la administración de proyectos y programas estratégicos entre organizaciones.

### **Gerencia de Asuntos exteriores**

Responsable de las campañas de asuntos públicos que apoyan la estrategia de la empresa y otros objetivos de políticas públicas importantes en México.

En la figura 1.4.1 se muestra el organigrama general y se resalta el área al que va dirigido el desarrollo.



**Figura 1.4.1 Organigrama**

## **Conceptos mercadológicos que influyen en el usuario en la compra de un teléfono celular**

Actualmente existe una gran variedad de opciones para los usuarios de teléfonos celulares y teléfonos inteligentes (smartphones); esta variedad se compone de diversos fabricantes, de los cuales cada uno de ellos ofrece varios modelos y en la mayoría de los casos, cada modelo cuenta con varias opciones en cuanto a capacidades, prestaciones y por supuesto, costos.

De lo anterior podemos observar que los fabricantes requieren hacer grandes inversiones tanto de dinero como de tiempo y recursos humanos para el diseño, planeación e implementación de campañas publicitarias que atraigan a los consumidores a comprar sus productos y actualmente no basta sólo con esa primera compra, las compañías están en una lucha cada vez más fuerte por crear fidelidad por parte del usuario, es decir, que la siguiente vez que vaya a comprar un teléfono o a renovar el que ya tiene, elija el modelo más reciente de la gama que ya utiliza o incluso que invierta en comprar un teléfono de una gama más alta, obviamente de su misma marca.

Para diseñar e implementar las mencionadas campañas publicitarias, las áreas de mercadotecnia de los fabricantes o bien las agencias de publicidad que estos contratan para dicho fin, echan mano de conceptos mercadológicos que hacen más atractivo su producto para el consumidor, además claro de explotar las características propias del teléfono que están promocionando.

A continuación, se mencionarán algunos conceptos mercadológicos que utilizan los fabricantes para influenciar a su favor la decisión del usuario al momento de comprar un teléfono.

### **Moda**

Los fabricantes utilizan a artistas de moda (actores, actrices, músicos, cantantes, etc.) para promocionar sus productos, las estrategias más recientes pretenden

que la persona en cuestión aparezca en público “casualmente” portando e incluso utilizando el teléfono de la compañía que lo contrata para tal fin.

Un ejemplo claro de esta estrategia se dio en una entrega de los premios Oscar, en la cual la presentadora de dichos premios apareció en varios momentos del evento portando un teléfono de cierta marca y el momento clave fue cuando “espontáneamente” invitó a varios actores y actrices de renombre mundial a que se tomaran una “selfie” (fotografía echa por la misma persona que aparece en la imagen).

En la transmisión en vivo del evento y sus posteriores repeticiones, que por cierto es uno de los eventos más vistos a nivel mundial, las cámaras muestran claramente el teléfono con el que se realizó la fotografía grupal; obviamente la empresa fabricante de dicho teléfono pagó una suma multimillonaria a los organizadores del evento, a la televisora que lo transmitió y a la conductora que hizo posible esta campaña publicitaria.

### **Estatus**

Sucede sobre todo con los teléfonos de gama alta, con grandes prestaciones y por supuesto, precios más elevados. En este caso lo que hacen las compañías, aunque no abiertamente, es promocionar el teléfono como un producto que no cualquier persona puede tener. El diseño de los teléfonos, con colores como el dorado, con acabados finos y de líneas suaves, sugieren que se trata de un producto exclusivo, costoso, orientado a personas con fuerte poder adquisitivo.

Nuevamente, las empresas echan mano de personajes públicos y famosos, quienes ostentan estos teléfonos en eventos o en sus redes sociales, reforzando la idea de que estos equipos son para gente con un estatus o nivel de vida alto.

### **Libertad**

Recientemente, las empresas han enfocado sus campañas a la libertad que proporciona al usuario el poseer uno de sus teléfonos. Graban promocionales en el campo, en un lago, en la playa, en la ciudad, dando la idea de que en cualquiera

de estas situaciones y en muchas otras, el teléfono tiene una característica (la calidad de la recepción, la capacidad de la cámara, la facilidad de uso, el acceso e integración con redes sociales, incluso en algunos casos la resistencia del teléfono al agua o a condiciones extremas de uso) que el usuario podrá explotar y disfrutar si compra dicho teléfono.

### **Diseño y Tecnología**

Por último, pero no menos importante, están las características tecnológicas y de diseño del teléfono como tal. En este caso el fabricante explota en sus promocionales la característica o características que hacen del teléfono especial o único.

El diseño del teléfono como tal, los materiales y acabados con los que se fabrica, son un elemento muy importante, además de los colores en los que se ofrece. La tendencia actual es el uso de aluminio que hace que el teléfono sea elegante y resistente a la vez.

La capacidad de la cámara que incorpora el teléfono y las facilidades y prestaciones que proporciona para su uso, actualmente también son una característica que influye mucho en la decisión del consumidor.

Otro elemento muy importante es el tamaño y la resolución de la pantalla del teléfono; actualmente, irónicamente la funcionalidad de hacer y recibir llamadas es una parte mínima del uso que se le da a un teléfono, ahora el usuario lo utiliza para ver fotos y videos, jugar videojuegos, crear y editar documentos e imágenes, etc. Por lo cual se requiere de una pantalla con muy buenas prestaciones para poder llevar a cabo estas actividades.

Todos los elementos mencionados, los fabricantes los entrelazan de formas cada vez más ingeniosas para crear y difundir sus campañas publicitarias en medios impresos, redes sociales, anuncios en televisión, espectaculares fijos o electrónicos en las grandes ciudades, etc.