

**Instituto Tecnológico de Orizaba  
División de Estudios de Posgrado e Investigación  
Maestría en Sistemas Computacionales**

“Desarrollo de una aplicación de comercio móvil

en Android”

***REPORTE DE TESIS QUE PRESENTA EL:***

I.S.C. Jorge Fernando Ambros Antemate

***PARA OBTENER EL GRADO DE:***

Maestro en Sistemas Computacionales

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***DIRECTOR DE TESIS***Dr. Giner Alor Hernández

ORIZABA, VER., MÉXICO

DICIEMBRE 2010

**ÍNDICE GENERAL**

Resumen

[Introducción 1](#_Toc263161003)

[CAPÍTULO 1 3](#_Toc263161004)

[1.1 Marco teórico. 3](#_Toc263161005)

[1.1.1 Comercio electrónico (e-commerce) 3](#_Toc263161006)

[1.1.2 Negocio electrónico (E-Business) 5](#_Toc263161007)

[1.1.3Modelos de comercio electrónico 6](#_Toc263161008)

[1.1.4 Comercio móvil (m-commerce) 8](#_Toc263161009)

[1.1.5 Sistemas operativos para dispositivos móviles 11](#_Toc263161010)

[1.1.6 Lenguajes para programar aplicaciones en dispositivos móviles 16](#_Toc263161011)

[1.1.6.1 WML (Wireless Markup Language) 16](#_Toc263161012)

[1.1.6.2 J2ME 17](#_Toc263161013)

[1.1.6.3 C++ 18](#_Toc263161014)

[1.1.6.4 Objetive-C 18](#_Toc263161015)

[1.1.6.5 .Net compact framewok 19](#_Toc263161016)

[1.1.7 Arquitectura de Software 19](#_Toc263161017)

[1.1.8 Estilos arquitectónicos 20](#_Toc263161018)

[1.1.9 Arquitectura Orientada a Servicios (SOA) 21](#_Toc263161019)

[1.1.9.1 Principios clave SOA 22](#_Toc263161020)

[1.1.9.2 Consumidores y servicios 24](#_Toc263161021)

[1.1.9.3 Como los servicios encapsulan la lógica de negocio 25](#_Toc263161022)

[1.1.10 Servicios Web y SOAP 26](#_Toc263161023)

[1.2 Planteamiento del problema 31](#_Toc263161024)

[1.3 Objetivos general y específicos 32](#_Toc263161025)

[1.4 Justificación 32](#_Toc263161026)

[CAPÍTULO 2 34](#_Toc263161027)

[2.1 Trabajos relacionados 34](#_Toc263161028)

[2.2 Análisis comparativo 40](#_Toc263161029)

[CAPÍTULO 3 43](#_Toc263161030)

[3.1 Introducción 43](#_Toc263161031)

[3.2. Representación arquitectónica 43](#_Toc263161032)

[3.3 SOA con Android 44](#_Toc263161033)

[3.4 Arquitectura de la aplicación de comercio móvil 45](#_Toc263161034)

[3.4.1 Funcionamiento de la arquitectura 48](#_Toc263161035)

[3.5 Servicios Web desarrollados 48](#_Toc263161036)

[3.5.1 Aplicación Android 50](#_Toc263161037)

[3.6 Proceso de compra electrónica 51](#_Toc263161038)

[3.6.1 Selección de producto 52](#_Toc263161039)

[3.6.2 Orden 52](#_Toc263161040)

[Conclusiones 54](#_Toc263161041)

[Glosario 55](#_Toc263161042)

[Bibliografía 57](#_Toc263161043)

**Lista de figuras**

Figura 1.1 Diferencias entre comercio electrónico por el lado de la compra y por el lado de la venta 6

Figura 1.2 Interfaz de usuario *Windows mobile* 11

Figura 1.3Interfaz de usuario *Symbian*. 12

Figura 1.4Interfaz de usuario*Android*. 13

Figura 1.5 Estructura del sistema operativo *Android*. 14

Figura 1.6 Arquitectura genérica de J2ME 18

Figura 1.7 Bajo acoplamiento entre los participantes 22

Figura 1.8 Relación entre consumidor – servicio. 25

Figura 1.9Encapsulamiento de los servicios 26

Figura 1.10 Estructura de un sobre SOAP 30

[Figura 3.1 Diagrama de la biblioteca de funciones kSOAP como parte de la aplicación Android. 44](#_Toc263161054)

[Figura 3.2 Arquitectura en capas de la aplicación de comercio móvil bajo un estilo arquitectónico SOA. 45](#_Toc263161055)

[Figura 3.3 Vista del catálogo de productos en la aplicación OsCommerce. 47](#_Toc263161056)

[Figura 3.4 Categorías disponibles 50](#_Toc263161057)

[Figura 3.5 Listado de productos por categoría 51](#_Toc263161058)

[Figura 3.6 Proceso de compra electrónica 51](#_Toc263161059)

[Figura 3.7 Selección de producto. 52](#_Toc263161060)

[Figura 3.8 Orden de compra. 53](#_Toc263161061)

**Lista de tablas**

[Tabla 1.1 Fragmento de documento WSDL. 28](#_Toc263117667)

[Tabla 1.2 Ejemplo de XSD. 29](#_Toc263117668)

[Tabla 2.1 Comparación de artículos. 40](#_Toc263117671)

[Tabla 3.1 Servicios Web desarrollados 49](#_Toc263161062)

**Resumen**

Con el surgimiento de Internet y de nuevas tecnologías las empresas buscan nuevas formas para dar a conocer sus productos o servicios, en un inicio solo estando en la tienda física el cliente adquiría los productos, posteriormente al utilizar Internet la empresa se da a conocer utilizando un portal de comercio electrónico, y actualmente, el comercio electrónico experimenta cambios, uno de estos es el denominado comercio móvil, el cual ofrece ventajas tales como la comodidad y la movilidad que tienen los teléfonos celulares, *PDAs* o *Smartphone* para realizar transacciones que antes solo estaban disponibles en las computadoras.

El presente documento consiste en un proyecto de tesis que tiene como objetivo el desarrollar una arquitectura orientada a servicios (SOA) para las aplicaciones de comercio móvil Android, al adoptarla junto con los estándares de la industria como XML, WSDL y SOAP se garantiza la interoperabilidad entre los distintos servicios desarrollados y las aplicaciones utilizando servicios Web, esto permite que los datos contenidos en el servidor de comercio electrónico se muestren directamente en una aplicación de comercio móvil, de esta forma las empresas tienen otra alternativa de ofrecer sus servicios a clientes potenciales que cuenten con un dispositivo compatible con esta aplicación. Se presentan además dos casos de estudio: el primero describe como un usuario utilizando un teléfono con el sistema operativo Android recupera los productos mediante la aplicación de comercio móvil y el segundo caso presenta los detalles del producto, la visualización del carrito de compras y el resumen de la compra realizada por el cliente.

# Introducción

El avance en las tecnologías inalámbricas incrementa el número de usuarios con teléfonos celulares, con lo cual se presenta un escenario para el desarrollo de aplicaciones de comercio electrónico para esos tipos de dispositivos. Este nuevo tipo de comercio electrónico dirigido a través de dispositivos móviles utilizando las redes de comunicación inalámbrica se denomina comercio móvil, el cual ofrece nuevas formas de intercambio de información y compras de productos o servicios, sin embargo, debido a las restricciones de un teléfono celular, este comercio opera en un ambiente un poco diferente del comercio electrónico tradicional.

En sus inicios los teléfonos celulares eran limitados en capacidad de procesamiento y memoria, esto ocasionaba que existieran ciertas limitaciones al momento de visualizar las aplicaciones o las páginas Web, por ejemplo: el contenido de las páginas Web no se mostraba igual que en una computadora de escritorio, o las aplicaciones que utilizan acceso a datos no se ejecutan correctamente o simplemente no lo hacen, lo que ocasionaba que los usuarios no las utilizarán. Pero gracias a que los teléfonos celulares incrementan en gran medida su potencia de procesamiento y capacidad de memoria, y que además se tiene una creciente colección de tecnologías de desarrollo, ahora es posible crear aplicaciones similares a las que se encuentran disponibles en las computadoras de escritorio.

El presente documento consiste en una propuesta para solucionar la problemática que se tiene en el uso y visualización de las aplicaciones de los teléfonos celulares, dichas aplicaciones ofrecen interfaces pobres a los usuarios, difícil navegación y no todas son compatibles con determinado hardware de teléfono, por lo que mientras en algunos celulares se presentan con una correcta visualización, en otros no es posible su ejecución, pero mediante el desarrollo en la plataforma *Android*, será posible el crear aplicaciones que sean más atractivas y funcionales para los usuarios.

Este documento consta de tres capítulos, en el primero se presentan los antecedentes donde se da a conocer la historia y conceptos básicos relacionados con el proyecto a desarrollar; el segundo capítulo expone una revisión de trabajos relacionados con el objetivo de conocer cómo se desarrollan proyectos similares, así como presentar una comparativa de estos trabajos para facilitar la elección de la solución a plantear; y en el tercer capítulo se presentan alternativas para desarrollar la herramienta y el porqué de la selección de la plataforma *Android* para el desarrollo de las aplicaciones.

CAPÍTULO 1

**ANTECEDENTES**

## 1.1 Marco teórico.

El presente capitulo se explican todos los conceptos básicos relacionados con el proyecto, se describe el planteamiento del problema, objetivo general, los objetivos específicos y finalmente la justificación del desarrollo.

### 1.1.1 Comercio electrónico (e-commerce)

Comercio es una actividad básica económica que involucra la compra y venta de bienes. Por ejemplo, un cliente entra a una tienda de libros, los examina, selecciona uno y paga por él. Para cumplir la necesidad del cliente, la tienda requiere llevar a cabo otras transacciones comerciales y funciones de negocios, como el manejo de la cadena de suministro, proveer soporte logístico, manejar pagos, entre otras. Al entrar a la era electrónica las transacciones comerciales, las funciones comerciales y el papeleo no se involucran en ningún contacto físico.

Por lo anterior se clasifica el comercio en: físico y electrónico. En el comercio tradicional las transacciones se realizan vía contacto físico entre humanos en un lugar físico como una tienda. Por ejemplo, si se quiere comprar un libro la persona tiene que ir a una librería ubicada en un lugar y comprar el libro con el vendedor. Por el contrario, en un sistema de comercio electrónico las transacciones se realizan vía medios electrónicos. En lugar de ir a la librería físicamente ahora la persona por medio de una página Web descarga el libro en formato digital.

El primero ejemplo de comercio electrónico es la transferencia electrónica de fondos (EFT, Electronic Funds Transfer), esto permitía que las instituciones financieras intercambiaran dinero desde una cuenta bancaria a otra, directamente sin ningún intercambio de dinero en metálico entre otras instituciones en una forma segura y eficiente. Después, fue el intercambio electrónico de datos (EDI) que facilitaba las transacciones entre empresas. Sin embargo, los primeros sistemas EDI típicamente operaban en redes especiales que fueron complejas de configurar y costosas de administrar denominadas redes virtuales privadas (VPN, Virtual Private Network), por estas razones, EDI no se emplea tan ampliamente como se esperaba. Con la llegada de las tecnologías de Internet y técnicas avanzadas de cifrado, ahora es posible aplicar el comercio electrónico sobre una red pública (Internet).

El desarrollo de la World Wide Web (WWW) acelera enormemente el desarrollo del  
comercio electrónico y amplía su alcance para cubrir diferentes tipos de aplicaciones [1].

Cuando el centro de compras es electrónico, las transacciones de negocios ocurren a través de redes de telecomunicaciones en donde compradores, vendedores y otros se involucran en la transacción –como los empleados que procesan la transacción – rara vez se ven o se conocen unos con otros además de que se encuentran físicamente en cualquier lugar del mundo.

Un modelo de tres capas se utiliza para describir el comercio electrónico, como el propuesto por Zwass [2]. Este modelo consiste en una capa de infraestructura, una de servicios, y una capa de estructuras/productos. Las tres capas adicionalmente se dividen en siete capas funcionales para ejecución de diferentes funciones. Las funciones principales proveen de:

Infraestructura técnica (Por ejemplo, Internet y WWW)

Servicio de mensajes seguros (Por ejemplo, EDI)

Soporte de servicios (Por ejemplo, pagos electrónicos)

Productos comerciales, servicios y sistemas (Por ejemplo, *e-retailing* -comercialización masiva de productos o servicios a grandes cantidades de clientes-)

Mercado electrónico (Por ejemplo, subastas en línea)

Schneider and Perry [3] ven al comercio electrónico como una manera efectiva para mejorar la cadena de valor, además de que se emplea para enlazar varias actividades funcionales (por ejemplo, producción, marketing, finanzas, entre otros) de una compañía. Este concepto de cadena de valor se extiende para unir diferentes compañías y formar una cadena de valor industrial. En general el comercio electrónico ayuda a facilitar el flujo de información a través de las cadenas de valores y para reducir los costos asociados a la transacción. También es interesante ver al comercio electrónico desde el punto de vista de relaciones. Cada negocio necesita mantener tres tipos diferentes de relación: la relación con su consumidor, con los socios (por ejemplo el proveedor), y la relación con sus empleados. El comercio electrónico provee una herramienta efectiva para construir, administrar y mejorar esas relaciones. En el contexto de comercio electrónico el primer tipo de relación no es solo vender a través de la Web, sino también manejar las relaciones con los clientes en general. Existe software especial que está disponible para este propósito. El segundo tipo de relación es acerca del manejo de la cadena de suministros por medios electrónicos. Mientras que las primeras dos relaciones son externas, la última es interna, involucra el construir una Intranet efectiva para integrar diferentes sistemas y compartir información, con estola comunicación y la productividad se mejoran.

### 1.1.2 Negocio electrónico (E-Business)

El negocio electrónico se apoya de Internet para enlazar y fortalecer los procesos de negocio, comercio electrónico, comunicación organizacional y colaboración dentro de una compañía, con sus clientes, distribuidores y otras personas involucradas. El negocio electrónico utiliza Internet, intranets y otras redes para soportar los procesos comerciales. El comercio electrónico por el contrario es la compra y venta, dar servicio y la comercialización de productos por medio de las redes de computadoras. Como el negocio electrónico incluye el proceso de transacción con proveedores y clientes existe un traslape en actividades con el comercio electrónico.

Aunque los términos ‘negocio electrónico’ y ‘comercio móvil’ generalmente se utilizan como sinónimos, la distinción entre estos recae en el amplio rango de procesos que tiene el negocio electrónico, el cual incorpora transacciones internas dentro de la organización, esto incluye transacciones relacionadas al aprovisionamiento, logística, administración de la cadena de suministros, pagos, control de existencias y órdenes de rastreo [4]. Chaffey [5] describe que el comercio electrónico es un subconjunto del negocio electrónico, donde los dos conceptos se traslapan es en la compra y venta de productos y servicios.

Cada transacción de comercio electrónico se considera desde dos perspectivas en la organización: del lado de venta y del lado de la compra

**Tipos de comercio electrónico**

Comercio electrónico por el lado de la venta: son las transacciones realizadas entre el cliente y la organización. Comercio electrónico por el lado de la compra: son las operaciones entre la organización y sus proveedores. La figura 1.1 muestra la diferencia entre los tipos de comercio electrónico.

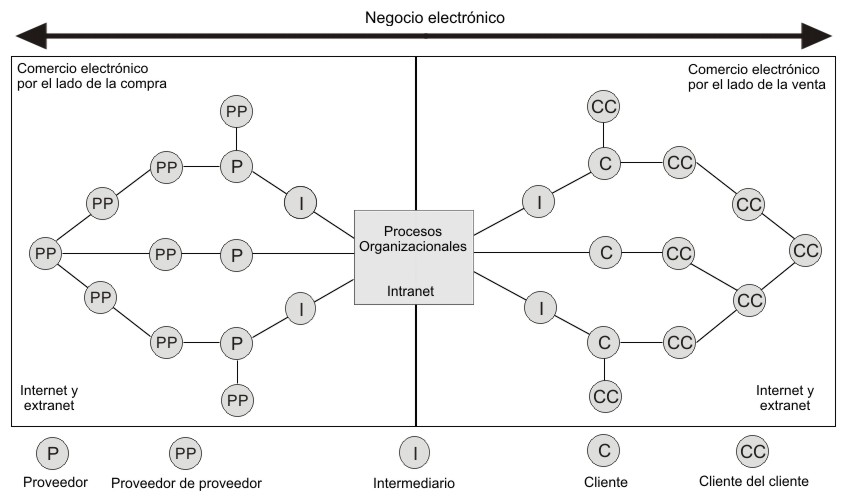


Figura 1.1 Diferencias entre comercio electrónico por el lado de la compra y por el lado de la venta

Como se observa en la figura anterior, el negocio electrónico se apoya del comercio electrónico para llevar a cabo el proceso de compra con proveedores y el proceso de la venta con los clientes.

### 1.1.3Modelos de comercio electrónico

Existen diferentes modelos de comercio electrónico dependiendo de la perspectiva de la relación entre el comprador y el vendedor. Estos modelos se utilizan para clasificar las aplicaciones de comercio electrónico, de acuerdo a la relación, se divide en cuatro categorías básicas [6]:

Negocio a consumidor (B2C, *Business to Consumer*)

En este caso, el vendedor es una organización comercial, mientras que el comprador es el consumidor. Esto simula la situación de vender al menudeo y generalmente es común llamarlo menudeo electrónico. Típicamente, las tiendas electrónicas se establecen en Internet para vender artículos a los consumidores. Amazon es un ejemplo de este tipo de comercio (www.amazon.com)

Negocio a negocio (B2B, *Business-to-business*)

En este caso, tanto el vendedor como el comprador son organizaciones comerciales. Existen tres tipos de sistemas denominados sistema orientado al comprador, sistema orientado al vendedor y mercado virtual. En muchas situaciones, esto se relaciona con el manejo de la cadena de suministros. Por ejemplo, una librería virtual requiere ordenar libros de varios editores. El proceso para ordenarlos se concluye utilizando intercambio de datos electrónicos. Ejemplo de este comercio es *General Electric’s Trading Process Network* (www.tpn.geis.com)

Consumido a consumidor (C2C, *Consumer-to-consumer*)

Esto se refiere a la situación en donde tanto el vendedor como el comprador son consumidores. Con la llegada del comercio electrónico, las subastas en línea proveen un medio efectivo para soportar el comercio electrónico entre consumidores. Mercado libre es un comercio consumidor a consumidor (www.mercadolibre.com.mx)

Consumidor a negocio (C2B, *Consumer-to-business*)

Este se considera una forma reciente de comercio, en el cual el consumidor especifica sus requisitos a una empresa, está provee el producto que cumpla con los requisitos que se solicitaron, por ejemplo, algo tan simple como un precio aceptable, o involucrar una considerable adaptación de un producto ya existente al gusto del consumidor, o inclusive la creación de un nuevo producto. Un ejemplo de este modelo en el comercio tradicional es un sastre que realiza un “traje a la medida”. La empresa Priceline ofrece una aplicación de comercio electrónico de este tipo, la cual se llama “sistema de colección en demanda” (www.priceline.com)

### 1.1.4 Comercio móvil (m-commerce)

Hablar de los inicios del comercio móvil es remontarse a 1997, cuando las dos primeras máquinas expendedoras de Coca Cola con capacidad de interactuar con teléfonos móviles se instalaron en el área de Helsinki en Finlandia. Las máquinas utilizaban mensajes de texto corto (SMS) para enviar el pago a las máquinas expendedoras. También en 1997 aparece el primer teléfono móvil basado en servicios bancarios en el banco de Merita en Finlandia, también se utilizaba SMS para llevar a cabo las operaciones, en 1998 las primeras ventas con contenido digital se realizaron como descargas para teléfonos móviles cuando el primer comercial de tonos descargables aparece en Finlandia por Radiolinja, en 1999 surgen dos plataformas comerciales de comercio móvil: la introducción de un sistema de pagos móviles realizado por *Smart* llamado *Smart Money* en Filipinas y el lanzamiento de la primera plataforma móvil en Internet por NTT DoCoMo (http://www.nttdocomo.com/) en Japón, llamada i-Mode, cerca del 2000 el comercio móvil se extendió rápidamente, en Australia se ofrecían boletos a través del teléfono celular para viajar en el tren, en Japón se ofrecieron las compras de boletos para aerolíneas[7].

El comercio móvil es cualquier transacción electrónica de bienes, productos o servicios por medio de redes de telecomunicaciones móviles, también es el surgimiento de un conjunto de aplicaciones y servicios que las personas utilizan desde sus dispositivos móviles. Es un poco diferente del comercio electrónico tradicional debido a las limitaciones que se imponen en los teléfonos o en los Asistentes Digitales Personales (PDA, *Personal Digital Assistant*). El punto clave para distinguir entre comercio móvil y comercio electrónico es el uso de un dispositivo móvil inalámbrico para acceder a un sitio de comercio Web, en lugar de utilizar un *Web browser* de una computadora personal.

**Características del comercio móvil**

La esencia del comercio móvil gira alrededor de la idea de alcanzar consumidores, proveedores y empleados sin importar en donde se encuentren localizados. El comercio móvil tiene la característica de que el usuario accede a Internet desde cualquier localización a cualquier hora (si se tiene el servicio disponible en la localidad donde se encuentra), e inclusive la capacidad de determinar la localización de la terminal móvil del usuario. El comercio móvil tiene además ciertas características las cuales no se encuentran disponibles en el comercio electrónico tradicional [8]:

Ubicuidad

Es la principal ventaja del comercio móvil. Los usuarios obtienen cualquier información en la que ellos estén interesados en cualquier momento (siempre y cuando se cuente con señal de algún proveedor de servicios) en el que lo deseen a través de los dispositivos móviles con capacidad de acceder a Internet. En las aplicaciones de comercio móvil los usuarios generalmente están ocupados en otras actividades, como viajar, mientras realizan transacciones o reciben información.

Accesibilidad

A través de los dispositivos móviles, las entidades de negocios son capaces de localizar a los consumidores en cualquier lugar, los consumidores tienen acceso a las aplicaciones de las empresas en el momento que lo requiera.

Localización

El saber la localización física en un momento en particular del usuario agrega un valor más significante al comercio móvil. Con la información de la localización, muchas de las aplicaciones basadas en localización ofrecen más prestaciones, por ejemplo, una de estas aplicaciones ayudan al usuario a localizar el restaurante más cercano, un cajero automático, o una tienda donde estén ofreciendo un producto de su preferencia.

Personalización

Los propietarios de los dispositivos móviles generalmente requieren un conjunto de aplicaciones y servicios que se encuentran en Internet, las aplicaciones de comercio móvil ofrecen la personalización para proveer información y servicios de acuerdo a un usuario en específico.

**Áreas donde también se utilizan las aplicaciones de comercio móvil**

Además de la venta de productos y/o servicios el comercio móvil cubre otras áreas, algunos ejemplos se enuncian a continuación:

Entretenimiento

El entretenimiento es la aplicación más popular en varias generaciones. El comercio móvil hace posible la descarga de juegos, imágenes, música, video a cualquier hora y en cualquier lugar.

Rastreo de inventario

La entrega justo a tiempo es algo crítico en los negocios actuales. El comercio móvil ofrece a los negocios mantener un seguimiento del movimiento del inventario y del proceso de entrega, esto mejora el servicio al cliente y la competitividad de la compañía.

Viajes

Los viáticos son costosos para los negocios. El comercio móvil ayuda a reducir los costos operacionales al proveer servicios de administración de viajes para los representantes comerciales mediante la localización de hoteles cercanos, compra de boletos, realizar ordenes de transporte, entre otros.

Tráfico

Tráfico es el movimiento (vehículos o peatones) a través de un área. Los pasajeros en vehículos o los peatones son objetos móviles, los clientes ideales para el comercio móvil. Además el control de tráfico es usualmente el mayor dolor de cabeza en muchas áreas metropolitanas. Utilizando la tecnología de comercio móvil se mejora el tráfico de diversas maneras. Por ejemplo, al contar con las capacidades de un sistema de posicionamiento global (GPS, *global positioning system*) se determina la posición del conductor y se da el aviso del estado de tráfico actual en diversas direcciones.

### 1.1.5 Sistemas operativos para dispositivos móviles

*Windows Mobile*

Es un sistema operativo desarrollado por Microsoft utilizado en teléfonos inteligentes y otros dispositivos móviles, tiene una apariencia similar a las versiones Windows de computadoras de escritorio, se basa en *Windows Embedded CE*. El sistema incluye: .*Net Compact Framework*, *Microsoft SQL Server Compact Edition*, el cual se utiliza como almacenamiento local y además se sincroniza con otras ediciones de *SQL Server 2005*.La figura 1.2 muestra la pantalla principal de *Windows mobile*.



Figura 1.2 Interfaz de usuario *Windows mobile*

*Symbian*

Este sistema operativo se desarrolla para dispositivos móviles y teléfonos inteligentes, contiene un conjunto de bibliotecas, interfaz de usuario (como lo muestra la figura 1.3), marco de desarrollo (f*ramework*) y herramientas de implementación desarrolladas por *Symbian Ltd*.



Figura 1.3Interfaz de usuario *Symbian*.

Soporta la multitarea y la ejecución de múltiples hilos. Muchos procesos se ejecutan concurrentemente, estos se comunican unos con otros y utilizan múltiples hilos que se ejecutan internamente en cada proceso. El sistema operativo tiene un sistema de archivos compatible con *Microsoft Windows* (técnicamente, un sistema de archivos *FAT32*), soporta otras implementaciones de sistemas de archivos a través de agregados. Utiliza el protocolo TCP/IP, y otras interfaces de comunicación como infrarrojos, *Bluetooh* y serial [9].

Debido a las limitantes de los teléfonos con respecto a su capacidad de almacenamiento, no se implementa la memoria virtual.

Otras características que ofrece:

* Micro-núcleo (*microkernel*). Un mínimo de funciones del sistema y datos se encuentran en el núcleo.
* Tiene soporte para máquinas virtuales: la implementación de una “computadora dentro de otra computadora”. La implementación del lenguaje de programación Java y el ambiente en tiempo de ejecución requiere que se ejecute a través de este mecanismo.
* Soporta la extensibilidad de la estructura de comunicación: Los módulos escritos implementan diversas características, desde las interfaces a nivel de usuario hasta la implementación de un nuevo protocolo para nuevos controladores de dispositivos. Esto es posible gracias al diseño del micro-núcleo, el cual permite que los nuevos módulos se introduzcan y carguen en el sistema dinámicamente.
* Diseñado principalmente con interfaces de programación de aplicaciones (API, *Application Programming Interface*) para multimedia. Los dispositivos multimedia y el contenido se manipulan por un servidor especial y por un marco de trabajo que permite que el usuario implemente nuevos módulos para el manejo de la multimedia.

*Android*

*Android* es una pila de software para dispositivos móviles que incluyen un sistema operativo, middleware, interfaz de usuario (como lo muestra la figura 1.4) y aplicaciones. El Kit de Desarrollo de Software de *Android* (SDK, *Software Development Kit*) provee las herramientas y las *APIs* necesarias para comenzar a desarrollar aplicaciones en la plataforma utilizando el lenguaje de programación Java [10].



Figura 1.4Interfaz de usuario*Android*.

Estructura de *Android*

El siguiente diagrama muestra los componentes principales del sistema operativo *Android*.

(Ver Figura 1.5).

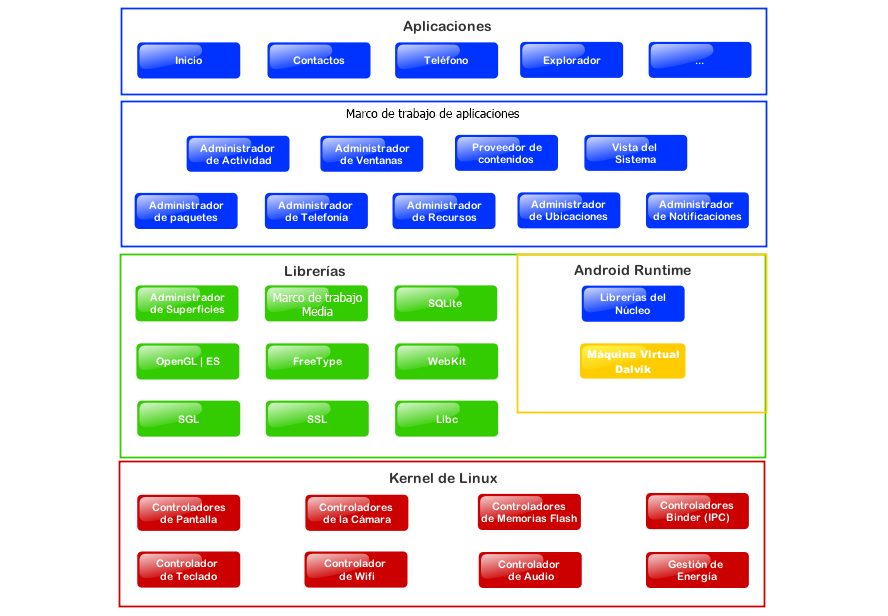


Figura 1.5 Estructura del sistema operativo *Android*.

Aplicaciones

*Android* se suministran con un conjunto de aplicaciones que incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, *Web browser*, contactos, y otros. Todas las aplicaciones se desarrollan usando el lenguaje de programación Java.

Marco de Trabajo

Los desarrolladores tienen acceso completo a las mismas APIs del marco de trabajo utilizado por las aplicaciones. La arquitectura de aplicaciones se diseñó para simplificar la reutilización de componentes, cualquier aplicación publica sus capacidades y cualquier otra aplicación hace uso de esas capacidades (sujeto a las limitaciones de seguridad impuestas por el marco de trabajo). Este mismo mecanismo permite al usuario cambiar los componentes.

Fundamentalmente todas las aplicaciones son un conjunto de servicios y subsistemas, incluyendo:

* Extensible conjunto de vistas, que se utilizan para construir una aplicación, listas, rejillas, cajas de texto, botones, e incluso un *Web browser* incrustado.
* Proveedores de Contenidos que permiten a las aplicaciones acceder a datos de otras aplicaciones (como Contactos), o para compartir sus propios datos.
* Administrador de recursos.
* Administrador de notificación que permite a todas las aplicaciones mostrar alertas personalizadas en la barra de estado.
* Administrador de actividad que gestiona el ciclo de vida de las aplicaciones.

Bibliotecas de funciones

*Android* incluye un conjunto de biblioteca de funciones de C / C ++ utilizado por diversos componentes del sistema, estas capacidades están expuestas a los desarrolladores a través del marco de trabajo. Algunas de las principales bibliotecas se enumeran a continuación:

* Biblioteca de sistema C – una implementación derivada de Distribución de Software Berkeley (BSD, Berkeley Software Distribution) de la biblioteca del sistema C estándar (libc), reajustada para dispositivos embebidos basados en Linux.
* Bibliotecas de Medios – basadas en *PacketVideo's OpenCORE*; la biblioteca soporta sonido pregrabado y grabación de muchos formatos populares de audio y video, así como archivos de imágenes estáticas, incluyendo MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
* Administrador de Superficies – administra el acceso al subsistema de pantalla y continuamente mezcla capas de gráficos de 2D y 3D de múltiples aplicaciones.
* *LibWebCore* – una moderna máquina de *Web browser* el cual potencializa en *Web browser*de *Android* y las vistas Web embebidas.
* Biblioteca de Escenas Gráficas (DGL, *Scene Graph Library*) - el motor gráfico 2D subyacente al sistema.
* Bibliotecas 3D – Una implementación basada en las interfaces de programación de aplicaciones de OpenGL ES 1.0; Estas bibliotecas utilizan tanto el hardware 3D de aceleración (si esta disponible), como el software 3D optimizado el cual se incluye en la plataforma.
* FreeType - mapa de bits y vectores para renderización de fuentes
* SQLite - un potente y ligero motor de base de datos relacionales disponibles para todas las aplicaciones.

*Android Runtime*

Cada aplicación *Android* se ejecuta en su propio proceso, con su propia instancia de la máquina virtual Dalvik, está escrita de modo que el dispositivo ejecuta múltiples máquinas virtuales de manera eficiente. La máquina virtual Dalvik ejecuta los archivos en el formato Dalvik Executable (.dex) optimizado para un mínimo de memoria.

Núcleo Linux

*Android* se basa en la versión 2.6 de Linux para el sistema de servicios básicos como la seguridad, la gestión de memoria, gestión de procesos, pila de red, y el conductor modelo.

El núcleo también actúa como una capa de abstracción entre el Hardware y el resto de la pila de Software.

### 1.1.6 Lenguajes para programar aplicaciones en dispositivos móviles

#### 1.1.6.1 WML (Wireless Markup Language)

El lenguaje de marcado inalámbrico (WML, *Wireless Markup Language*) es un lenguaje similar al Lenguaje de Marcado de Hipertexto (HTML, *Hypertext Markup Language*), utilizado para programar páginas Web para *Web browser* de dispositivos móviles, que utilizan la tecnología del Protocolo de Aplicaciones Inalámbricas (WAP, *Wireless Application Protocol*). WML es una versión reducida del lenguaje HTML que facilita la conexión a Internet de dichos dispositivos. La visualización de la página dependerá del dispositivo que se use y de la forma en que este interprete el código, ya que varían entre sí.

A diferencia de HTML, WML es más estricto, al existir un error en la escritura de las etiquetas presentará un error en vez de mostrar la página, WML utiliza tablas simples, algunos atributos existentes en HTML no existen en WML, tampoco existe la posibilidad de crear tablas dentro de tablas.

Así como HTML utiliza *Javascript* para crear acciones especiales sobre las páginas, WML utiliza *WMLScript* que es muy similar a *Javascript* pero más simple.

#### 1.1.6.2 J2ME

Es una versión de Java para dispositivos electrónicos con capacidades de procesamiento, memoria y gráficas reducidas. Contiene componentes básicos que lo diferencian de otras versiones de Java, como una máquina virtual denominada KVM (*Kilobyte Virtual Machine*, esto debido a que esta máquina virtual solo requiere de pocos *Kilobytes* de la memoria del dispositivo para funcionar) en lugar de utilizar la máquina virtual tradicional, también ofrece un recolector de basura, contiene una mínima parte de las interfaces de programación de aplicaciones de Java debido a que los dispositivos pequeños tienen capacidad de memoria reducida.

Con el tiempo la plataforma de Java ME se dividió en dos configuraciones base: una para ajustarse a los dispositivos pequeños y una que apunta hacía dispositivos más capaces como teléfonos inteligentes [11].

Arquitectura de J2ME

Debido a las diferencias que existen entre los dispositivos a los cuales se enfoca la plataforma J2ME, se definen una serie de componentes a partir delos cuales se construye una arquitectura genérica con el objetivo de conseguir un entorno de ejecución para cualquier familia de dispositivos.

Estos componentes se agrupan tal y como lo muestra la figura 1.6:

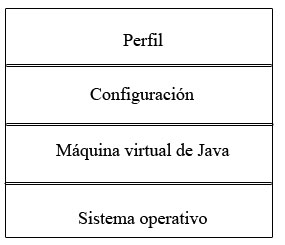


Figura 1.6 Arquitectura genérica de J2ME

Máquina Virtual Java: Se dispone de dos máquinas virtuales con diferentes características, cada una para diferentes tipos de dispositivos. Llamadas KVM y CVM (*Compact Virtual Machine*).

#### 1.1.6.3 C++

C++ es el lenguaje nativo para desarrollar aplicaciones para *Symbian*, sin embargo, el sistema operativo no utiliza una implementación estándar de C++ ya que agrega otras funcionalidades, además existen múltiples *SDKs* basados en el original, cada compañía extiende el *SDK* para realizar sus aplicaciones.

El *SDK* original contiene la documentación, archivos de cabecera y bibliotecas requeridas para el desarrollo de las aplicaciones, se encuentra disponible también un emulador basado en Windows.

#### 1.1.6.4 Objetive-C

Lenguaje utilizado principalmente en Mac OS X y el sistema operativo iPhone de Apple, utiliza programación orientada a objetos, Objetive-C se define como un pequeño conjunto de extensiones del lenguaje estándar ANSI C, se le agregaron más palabras reservadas, directivas de compilador, y nuevas sintaxis (como el envío de mensajes a objetos). La extensión es posible gracias a la biblioteca en tiempo de ejecución de Objetive-C, esto es generalmente transparente para el programador. Gracias a esta biblioteca Objetive-C es un verdadero lenguaje orientado a objetos.

Al ser una extensión de ANSI C también se utiliza código C tradicional en los programas. De hecho los programas de Objetive-C son familiares para los programadores de lenguaje C ya que inclusive utiliza la tradicional función main.

Los códigos fuentes se compilan utilizando el compilador GNU gcc estándar. El compilador reconoce los archivos fuente por la extensión .m, archivos C con la extensión .c y archivos de cabecera por la extensión .h.

#### 1.1.6.5 .Net compact framewok

.*NET Compact Framework* se enfoca al desarrollo de aplicaciones para dispositivos móviles, con reducidas características en la administración de memoria, procesos y recursos. El *framework* es parte integral de *Windows Mobile* y *Windows Embedded CE*, utiliza algunas de las bibliotecas de clases del .NET Framework completo, e incorpora bibliotecas diseñadas específicamente para dispositivos móviles esto debido a las limitantes que estos imponen, incluye también un lenguaje común en tiempo de ejecución (CLR, *common language runtime*) optimizado para los dispositivos.

*.Net Compact Framework* soporta el desarrollo en Visual Basic y Visual C#. Actualmente no existe soporte para C++.

### 1.1.7 Arquitectura de Software

A medida que se extiende el uso y crece la complejidad de aplicaciones distribuidas, los aspectos arquitectónicos del desarrollo de software están recibiendo un interés cada vez mayor en la industria del software.

En la actualidad existen varias definiciones de Arquitectura de Software, a continuación se menciona una de ellas:

Pressman la define de la siguiente forma: “La Arquitectura de Software de un sistema de programa o computación es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos.” [12]

La arquitectura concierne a un nivel de abstracción más elevado; se ocupa de componentes y no de procedimientos; de las interacciones entre esos componentes y no de las interfaces; de las restricciones a ejercer sobre los componentes y las interacciones y no de los algoritmos, los procedimientos y los tipos.

### 1.1.8 Estilos arquitectónicos

Para Pressman, un estilo describe una categoría de sistema que abarca un conjunto de componentes que realizan una función requerida por el sistema, un conjunto de conectores que posibilitan la comunicación, la coordinación y cooperación entre los componentes, las restricciones que definen como se integran los componentes para conformar el sistema, y los modelos semánticos que facilitan al diseñador el entendimiento de todas las partes del sistema. [12].

Los estilos arquitectónicos indican los tipos de componentes y conectores involucrados, patrones y restricciones de interconexión o composición entre ellos, los estilos se aplican a una gran variedad de productos con el fin de mejorar o ayudar a asegurar ciertas cualidades deseadas en los mismos.

Reynoso [13] clasifica los estilos arquitectónicos de la siguiente manera:

* Estilos de Flujo de Datos
  + Tubería y filtros
* Estilos Centrados en Datos
  + Arquitecturas de Pizarra o Repositorio
* Estilos de Llamada y Retorno
  + Modelo-Vista-Controlador (MVC)
  + Arquitecturas en Capas
  + Arquitecturas Orientadas a Objetos
  + Arquitecturas Basadas en Componentes
* Estilos Derivados
  + C2
  + GenVoca
  + REST
* Estilos de Código Móvil
  + Arquitectura de Máquinas Virtuales
* Estilos heterogéneos
  + Sistemas de control de procesos
  + Arquitecturas Basadas en Atributos
* Estilos Peer-to-Peer
  + Arquitecturas Basadas en Eventos
  + Arquitecturas Orientadas a Servicios (SOA)
  + Arquitecturas Basadas en Recursos

### 1.1.9 Arquitectura Orientada a Servicios (SOA)

Una Arquitectura Orientada a Servicios (SOA: Service Oriented Architecture) es una combinación de consumidores y servicios que colaboran entre sí, es guiada por principios y soporta diversos estándares [14], está formada por servicios independientes de aplicación los cuales son del alta cohesión, bajo acoplamiento y altamente interoperables que encapsulan los procesos de negocio, estos servicios se invocan a través de interfaces bien definidas, para llevar a cabo una comunicación se utiliza una definición formal independiente de la plataforma y del lenguaje de programación (como WSDL: Web Services Description Language). La definición de la interfaz encapsula las particularidades de una implementación, con esto se tiene una independencia del lenguaje de programación o de la tecnología de desarrollo utilizada (como PHP, Java, .NET, entre otros). Una de las características más resaltante de SOA es que está basada en contratos, donde el proveedor establece las reglas de comunicación, el transporte, y los datos de estrada y salida que se intercambian por ambas partes, con esto se logra que los componentes software desarrollados sean altamente reusables.

#### 1.1.9.1 Principios clave SOA

Una SOA eficaz implementa y adopta un importante diseño, desarrollo y principios de administración, en el contexto de SOA existen varios principios que son fundamentales para establecer una correcta colaboración entre los participantes. Los principios clave que mínimo tiene que cumplir una SOA incluye [14]:

* Bajo acoplamiento.
* Interoperabilidad.
* Reutilización.
* Descubrimiento.

**Bajo acoplamiento**

En este principio tanto el consumidor como el servicio se encuentran aislados de los cambios realizados en la tecnología y el comportamiento, el bajo acoplamiento describe una separación lógica de asuntos, esto es, que en las SOA los consumidores se separan de conexiones directas de los servicios, la intención es proteger la integridad individual de cada consumidor y servicio SOA así como evitar dependencias físicas entre ellos.

Para cumplir con éste principio, los consumidores y servicios no se comunican de manera directa, para realizar esto los participantes intercambian peticiones y respuestas utilizando mensajes, evitando con esto una conexión directa entre ellos. La figura 1.9 muestra este principio.

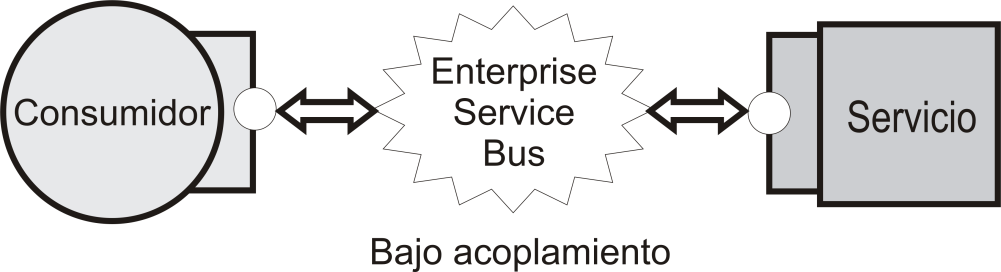


Figura 1.7 Bajo acoplamiento entre los participantes

Como se observa en la figura, los participantes se comunican vía mensajes por medio de Enterprise Service Bus (ESB), la cual es una tecnología que incluye la red, el transporte, el ruteo, la entrega de mensajes, contenido y los protocolos de comunicación, todos estos elementos son totalmente transparentes al consumidor.

Los mensajes intercambiados entre el consumidor y los servicios son ejecuciones de la interfaz del servicio. En el caso de los servicios Web, la interfaz del servicio está definida por la combinación del Lenguaje de Definición del Servicio Web (WSDL) y la definición de esquema XML (XSD), estas dos interfaces son la base para cualquier servicio Web, el diseño y el desarrollo de ellas son el punto principal del principio de bajo acoplamiento.

Un diseño de interface de servicio común es RPC-encode (Remote Procedure Call, llamado a procedimiento remoto), establece que cada operación del servicio es una función discreta que contiene uno o más parámetros, RPC es un protocolo que permite al consumidor ejecutar código en un equipo remoto sin tener que preocuparse de las comunicaciones entre ambos.

**Interoperabilidad**

Es un principio que elimina el uso de una tecnología específica. La interoperabilidad permite que el consumidor y el servicio se desarrollen en diferentes tecnologías para que colaboren y realicen el intercambio de información, por ejemplo: un servicio desarrollado en PHP en una plataforma con sistema operativo Linux con acceso a una base de datos MySQL es consumido por una aplicación desarrollada en Java o en C# que se ejecuta en una plataforma con sistema operativo Windows, las combinaciones y posibilidades son muchas, esto permite a los servicios y consumidores colaborar independientemente de la tecnología en la que se basan. Los mensajes tienen un papel importante en este principio al permitir que los participantes de la SOA colaboren e interactúen con una abstracción de la interfaz la cual aísla a los consumidores y servicios de sus características físicas o de sus especificaciones tecnológicas.

**Reutilización**

El principio de reutilización es poner especial énfasis en el desarrollo con reducción de costos. Un servicio que se desarrolla para soportar los principios de bajo acoplamiento e interoperabilidad es un buen candidato para reutilización, los consumidores utilizan la funcionalidad del servicio en una o varias aplicaciones. Cuando un servicio no provee toda la funcionalidad que requieren los consumidores se realiza una extensión del servicio más allá de su alcance original y de su funcionalidad. La extensión del las capacidades funcionales del servicio requiere el agregar nuevos datos que están contenidos en el mensaje de respuesta o modificar una operación existente, o inclusive crear nuevas operaciones. Las extensiones en un servicio existen no es una tarea trivial y depende de que tanto cumplen con los principios SOA el servicio y la interfaz.

**Descubrimiento**

Encontrar un servicio (descubrirlo) es el primero paso para la reutilización y el consumo. De nada sirve tener un servicio que tenga una alta funcionalidad si éste no es descubierto para su uso. La solución típica descubrir un servicio es un registro, el cual es similar a un catalogo o inventario de servicios, un registro incluye alguna forma de búsqueda que permite a los desarrolladores buscar por servicios creados con anterioridad los cuales son candidatos para su reutilización.

Cuando un servicio candidato es descubierto, el desarrollador evalúa si el servicio cumple con los requerimientos del consumidor, si es así, el servicio se utiliza, en caso contrario el servicio es candidato para extensión.

#### 1.1.9.2 Consumidores y servicios

Los consumidores y servicios interactúan unos con otros para solicitar y responder a solicitudes de servicios. La interacción está determinada por la interfaz del servicio, la cual es típicamente expresada como un mensaje que se mueve a través de los participantes.

Un consumidor es el que interactúa, consume y procesa el resultado del servicio, típicamente el consumidor es una pieza de software que interactúa con otra pieza de software que corresponde a un servicio.

Un servicio es un comportamiento encapsulado que está expuesto a los consumidores para resolver a una petición. Los servicios disponibles a los consumidores tienen diferente granularidad, dependiendo de está es la cantidad de información que manipulan, por ejemplo, un servicio de grado fino representa un simple comportamiento o un conjunto limitado de comportamientos, por otro lado un servicio de grano grueso es capaz de realizar combinaciones de varios servicios para resolver una petición.

Desde una perspectiva SOA los servicios son generalmente un conjunto de la funcionalidad de una aplicación.

La figura 1.7 muestra la relación entre consumidor y servicio, el consumidor realiza una petición y obtiene una respuesta que utilizada en la aplicación.

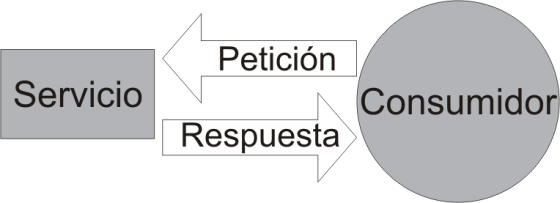


Figura 1.8 Relación entre consumidor – servicio.

#### 1.1.9.3 Como los servicios encapsulan la lógica de negocio

Para mantener su independencia los servicios encapsulan la lógica dentro de un contexto distinto, este contexto es específico de una tarea de negocio, una entidad del negocio o algún otro agrupamiento lógico [15].

Como se muestra en la figura 1.8 cuando se construyen los servicios, cada uno de ellos encapsula un proceso, un conjunto de procesos (subproceso) e inclusive toda lógica de negocio. De esta forma en lugar de realizar las aplicaciones con todo el código embebido solo utilizan los servicios que cubran las necesidades de la empresa.

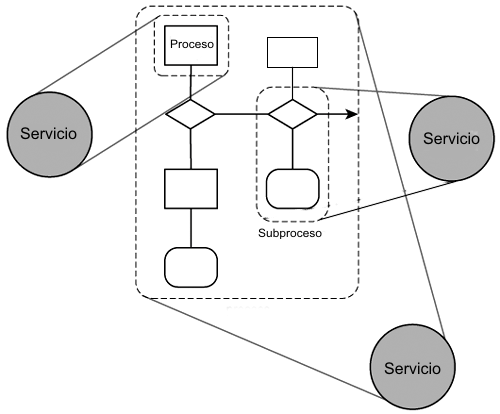


Figura 1.9Encapsulamiento de los servicios

### 1.1.10 Servicios Web y SOAP

Un servicio Web es el encargado de exponer un conjunto de funcionalidades a los consumidores, dependiendo de la granularidad del servicio las funcionalidades se describen por uno o más servicios. Los servicios Web cumplen con diversos estándares, de los cuales cuatro son el núcleo de su definición y su interfaz [15]:

* Lenguaje de descripción de servicios Web (WSDL, Web Service Description Language).
* Lenguaje de definición de esquemas XML (XSD, XML Schemas Definition Language).
* Lenguaje de marcado extensible (XML, Extensible Markup Language).
* Protocolo simple de acceso a objetos (SOAP, Simple Object Access Protocol).

Cada uno de los elementos se describen a continuación:

**WSDL:** Expone la funcionalidad del servicio como operaciones, describe el nombre, la localización y las operaciones del servicio, las entradas y salidas definidas por la interfaz. WSDL está definido por el consorcio de la World Wide Web, existen dos especificaciones, WSDL 1.1 que es reconocido e implementado a través de diversas herramientas y tecnologías, además de que se utiliza por muchas empresas para sus servicios Web, la más reciente versión WSDL 2.0 es una revisión de WSDL 1.1 e incluye diversas mejoras y extensiones, para el desarrollo del proyecto se utiliza la especificación 1.1.

Una operación definida en WSDL es una capacidad funcional del servicio, la operación le indica al servicio ejecutar la función requerida, como dar de alta un producto o recuperar el inventario, cada operación está definida por ningún, uno o varios parámetros, los parámetros describen un conjunto de información que espera el servicio para procesar la petición.

Un documento WSDL es simplemente un conjunto de definiciones, cada definición tiene un elemento raíz y definiciones internas. Los servicios se definen utilizando seis elementos principales:

* Types, proveen las definiciones de los tipos de datos utilizados para describir el intercambio de mensajes.
* Message, representan una definición abstracta de los datos a ser transmitidos, un mensaje consiste en partes lógicas, cada una de ellas está asociada con una definición dentro de algún tipo de sistema.
* PortType, es un conjunto de operaciones abstractas, cada operación se refiere a un mensaje de entrada y a uno de salida.
* Binding, especifica un protocolo en concreto y las especificaciones para el formato de los datos de las operaciones.
* Port, especifica una dirección para establecer un enlace, de esta manera se define un punto final de comunicación.
* Service, utilizado para agregar un conjunto de puertos relacionados, esto permite a un consumidor de documento WSDL determinar si se desea comunicar con un servicio en particular.

La tabla 1.1 muestra un fragmento de un documento WSDL.

Tabla 1.1 Fragmento de documento WSDL.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <wsdl:definitions  name="Tienda"  targetNamespace="http://www.tiendavirtuald.com/Tienda.wsdl"  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  xmlns:tns="http://www.tiendavirtuald.com/Tienda.wsdl"  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  xmlns:xsd1="http://www.tiendavirtuald.com/Tienda.xsd1">  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Created using Cape Clear Studio SOA Editor - http://www.capeclear.com</wsdl:documentation>  <wsdl:types>  <xsd:schema  targetNamespace="http://www.tiendavirtuald.com/Tienda.xsd1"  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  xmlns:xsd1="http://www.tiendavirtuald.com/Tienda.xsd1">  <xsd:complexType name="Categoria">  <xsd:sequence>  <xsd:element maxOccurs="1" minOccurs="1" name="idCat" type="xsd:int"/>  <xsd:element maxOccurs="1" minOccurs="1" name="nombreCat" type="xsd:string"/>  <xsd:element maxOccurs="1" minOccurs="1" name="descripcionCat" type="xsd:string"/>  </xsd:sequence>  </xsd:complexType>  </xsd:schema>  </wsdl:types>  <wsdl:message name="productoRequest"></wsdl:message>  <wsdl:message name="ProductoResponse">  <wsdl:part name="resultado" type="xsd1:Categoria"/>  </wsdl:message>  <wsdl:portType name="TiendaPortType">  <wsdl:operation name="ObtenerCategoria">  <wsdl:input message="tns:productoRequest"/>  <wsdl:output message="tns:ProductoResponse"/>  </wsdl:operation>  </wsdl:portType> |

**XSD:** Un esquema XML, se utiliza para definir y restringir el contenido de documentos XML, define los elementos que aparecen en un documento, los atributos, los elementos que son elementos hijos en la jerarquía, el orden de los elementos hijos, la cantidad de elementos hijos, los tipos de datos para los elementos y los atributos, los valores por default y fijos para los elementos y los atributos, la tabla 1.2 muestra un ejemplo de XSD de un archivo WSDL.

Tabla 1.2 Ejemplo de XSD.

|  |
| --- |
| <wsdl:types>  <xsd:schema  targetNamespace="http://www.tiendavirtual.com/servicios.xsd1"  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  xmlns:xsd1="http://www.tiendavirtual.com/servicios.xsd1">  <xsd:complexType name="categoriaProducto">  <xsd:sequence>  <xsd:element maxOccurs="1" minOccurs="1" name="idCat" type="xsd:int"/>  <xsd:element maxOccurs="1" minOccurs="1" name="nombreCat" type="xsd:string"/>  <xsd:elementmaxOccurs="1" minOccurs="1" name="descripcionCat"  type="xsd:string"/>  </xsd:sequence>  </xsd:complexType>  </xsd:schema>  </wsdl:types> |

**XML:** es la representación estructural de la instancia de un mensaje, permite que los consumidores y los servicios se comuniquen, sin importar la tecnología en la cual están desarrollados.

**SOAP:** Es un protocolo simple para el intercambio de información estructurada en un entorno distribuido y descentralizado utilizando XML, SOAP se utiliza en una gran variedad de protocolos. El marco de trabajo es diseñado para ser independiente de cualquier modelo de programación o cualquier semántica específica de alguna implementación. SOAP se basa en un paradigma de sobre que es similar a una carta que se envía utilizando el servicio postal tradicional: el sobre incluye la dirección de destino, y usualmente el tipo de mecanismo de entrega: normal, vía aérea, entrega urgente, entre otros (en SOAP esto es equivalente al protocolo de servicio y enlace), dentro del sobre se encuentra la carta o un documento (análogamente es el mensaje que se envía como una petición o una respuesta).

El sobre SOAP se estructura de la siguiente manera:

* Una cabecera (<soap:header/>) que contiene la descripción de la dirección y la información del destino.
* Cuerpo (<soap:body/>), contiene el mensaje XML codificado.
* En caso de un error SOAP tiene un área opcional para su tratamiento (<soap:fault/>), contiene información acerca del error que se retorna al consumidor.

La figura 1.10 muestra la estructura de un sobre SOAP.

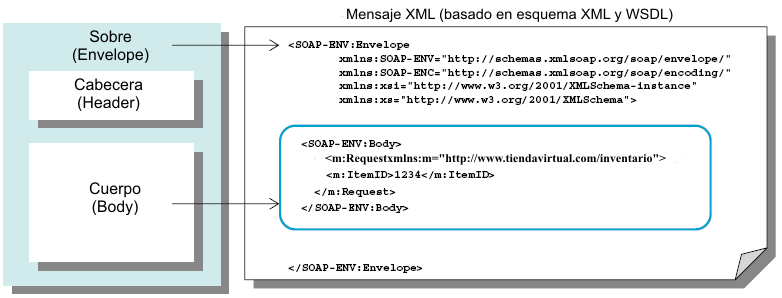


Figura 1.10 Estructura de un sobre SOAP

Fundamentalmente un servicio Web provee funcionalidad, manipula e intercambia datos en un contexto, los valores de estos datos se encapsulan en el mensaje entre etiquetas XML, estos a su vez también se encapsulan en el cuerpo SOAP el cual está dentro del sobre que se mueve entre los colaboradores.

Los estándares de los servicios Web aseguran un alto grado de interoperabilidad (todos los participantes utilizan la misma tecnología cuando se comunican con otros participantes). El uso de XML como esquema de codificación permite a los participantes producir y consumir mensajes y contenido a través de varias plataformas. Todo esto es posible gracias al cumplimiento de los estándares de Servicios Web (“WS-\*”).

Los servicios Web también se diseñan para esconder la tecnología de implementación, está capacidad de las SOA y los servicios ayudan a resolver problemas de integración, al permitir que las aplicaciones que los consumen no requieran conocer la información de la implementación técnica que está detrás del servicio Web, solo utilizan la información que está contenida en la respuesta.

## 1.2 Planteamiento del problema

En la actualidad las transacciones financieras son una necesidad, se realizan desde transferencias bancarias hasta comprar productos o servicios. Para llevar a cabo muchas de estas operaciones se utiliza Internet y se tiene que estar frente al equipo de cómputo en donde se utilizan aplicaciones desarrolladas específicamente para computadoras de escritorio, estas aplicaciones ofrecen al usuario una interfaz atractiva, fácil de utilizar y con varias prestaciones para realizar las transacciones, sin embargo esto no ocurre con las aplicaciones desarrolladas para los teléfonos celulares ya que las aplicaciones actuales de comercio móvil tienen funcionalidad limitada y una reducida gama de contenidos disponibles por lo que no cumplen con las expectativas de los usuarios y debido a esto, las aplicaciones no se utilizan.

Si bien existen diversos *Frameworks* para el desarrollo de aplicaciones para teléfonos celulares, cada uno de estos presenta restricciones, por ejemplo, muchas aplicaciones están desarrolladas en J2ME, pero debido a que los dispositivos móviles tienen una potencia de cálculo baja e interfaces de usuario pobres, es necesaria una versión específica de la máquina virtual de Java destinada a estos dispositivos, por lo que una aplicación desarrollada para un modelo de teléfono específico, no se ejecuta correctamente en otro teléfono de otra compañía.

Otro ejemplo de restricciones se presenta en las aplicaciones desarrolladas en iPhone, ya que si una empresa da a conocer sus aplicaciones primero tiene que invertir en la adquisición del kit de desarrollo y enviar la aplicación a *Apple Store*, por lo que en muchos casos el usuario final paga para utilizar dicha aplicación, además iPhone se limita solo a las capacidades que Apple ofrece en las APIs, por lo que ciertos componentes del sistema no se encuentran disponibles para el usuario.

## 1.3 Objetivos general y específicos

Objetivo general

Desarrollar una aplicación de comercio móvil bajo la plataforma *Android* con características similares a una aplicación de comercio electrónico tradicional, utilizando un acercamiento al modelo de comercio electrónico empresa a consumidor.

Objetivos específicos

* Desarrollar una arquitectura para la aplicación de comercio móvil.
* Identificar las funcionalidades y las relaciones entre los componentes que integran una aplicación comercio móvil.
* Desarrollar los servicios Web que consumirá la aplicación para mostrar los productos y/o servicios.
* Identificar y proveer prestaciones de servicios tales como compra, catálogo de productos, almacenamiento de datos del usuario, productos, compras.

## 1.4 Justificación

*Android* permite a los desarrolladores crear aplicaciones para teléfonos móviles sin las restricciones que generalmente se asocian a los *frameworks* propietarios y sacar el máximo provecho del hardware que todo dispositivo tiene que ofrecer, además se busca eliminar los problemas de incompatibilidad que actualmente existen en las aplicaciones. Ofrece también soporte nativo con muchas de las aplicaciones desarrolladas por Google (*Calendar, Maps, Talk*, entre otros) por lo que el desarrollador al utilizarlas crea aplicaciones con mayores prestaciones y gracias a los componentes que ofrece la plataforma de desarrollo las aplicaciones contarán con una interfaz atractiva al usuario y fácil de navegar.

Una aplicación desarrollada bajo *Android* se ejecuta en cualquier teléfono móvil que tenga instalado este sistema operativo y no se requieren pagar licencias como ocurre con muchas de las aplicaciones creadas en otras plataformas, esto permite a las empresas que desarrollan aplicaciones de comercio móvil las ofrezcan sin costos adicionales, a diferencia de otros *frameworks* donde el usuario termina pagando una cuota por el uso de la aplicación.

A través de la aplicación propuesta los usuarios que cuenten con un teléfono celular con el sistema operativo *Android* instalado realizarán el proceso de compra de productos y/o servicios con la ventajas de movilidad, facilidad de navegación, interfaz atractiva, entre otros, y de esta forma acercar más al usuario a la experiencia que tiene con las aplicaciones tradicionales de una computadora de escritorio.

CAPÍTULO 2

ESTADO DE LA PRÁCTICA

## 2.1 Trabajos relacionados

Desde hace aproximadamente 12 años se intenta llevar la W*orld Wide Web* a los *Smartphone*, sin embargo a pesar de los esfuerzos para crear aplicaciones no se tenía una completa experiencia debido a las limitaciones de procesamiento, memoria y herramientas para la creación de aplicaciones que existían en los teléfonos, pero esto cambió recientemente, en [16] se describe que gracias a la liberación de plataformas de desarrollo (como *Android*) para teléfonos móviles es posible crear aplicaciones para estos dispositivos, aunado a esto también se tienen otras tecnologías como *AJAX* (*Asynchronous JavaScript And XML*, JavaScript asíncrono y XML) o Web 2.0 con las cuales se desarrollarán una amplia variedad de contenido. Gracias a todo ahora los teléfonos móviles o los Smartphone se están acercando aún más a la experiencia que tienen los usuarios de escritorio.

Sin embargo a pesar de que los dispositivos móviles evolucionan tecnológicamente estos solo se utilizan para la comunicación a través de voz, envío de mensajes o para entretenimiento y no se aprovecha su tecnología para obtener otros beneficios. En [17] se describe una plataforma de cómputo distribuido de servicios, basado en el protocolo de iniciación de sesiones de servicios Web (WIP - *Web Services Session Initiation Protocol*) integrado en la plataforma *Android* para aprovechar aún más las prestaciones que ofrecen los nuevos teléfonos y de esta forma contar con servicios Web orientados a la comunicación sobre IP, se utilizan servicios Web, y comunicación en tiempo real que permite a la plataforma de computación móvil habilitar servicios de cómputo distribuido, para llevar a cabo esto se integra la herramienta gSoap desarrollada en C/C++, soporta diversas plataformas incluyendo sistemas embebidos y pequeños sistemas operativos

Las aplicaciones y servicios Web móviles generalmente no cubren las necesidades de las experiencias de usuarios de las aplicaciones de escritorio, en un inicio el modelo de Internet móvil basado en el protocolo WAP 1.0 tenía un bajo uso debido a la experiencia de navegación sobre los contenidos de Internet. En [18] se describe que gracias a la actual evolución en WAP 2.0 se tiene un gran avance en el desempeño y capacidades para mostrar contenido y una nueva experiencia Web, como por ejemplo: mejores gráficos, colores y sobre todo un acceso rápido a los servicios. Otra alternativa para desarrollar aplicaciones móviles es *AJAX*, con esto se crean aplicaciones para que sean más ricas tanto visual como interactivamente, derribando así las barreras existentes en las aplicaciones móviles.

En [19] se propone una nueva presentación para Web móvil, y es que los dispositivos móviles tienen pantallas pequeñas, interfaz de usuario limitada y poder de procesamiento, debido a estas situaciones estos dispositivos no exploran las páginas Web que se diseñan para una PC, esta nueva forma de presentación consiste en dividir la página Web, la cual se muestra en forma pequeña y por zonas de navegación, cuando el usuario al seleccione una de estas zonas el bloque original se amplia de forma automática y presentado en un tamaño que cubra la pantalla del dispositivo móvil. Gracias a esto se tiene una experiencia similar a la que tienen los usuarios de escritorio.

Muchos de los dispositivos móviles en la actualidad traen incorporados cámaras digitales las cuales aumentan cada vez más su resolución, y es común que los usuarios almacenen miles de fotos gracias a que también estos dispositivos aumentan su capacidad de almacenamiento, sin embargo aquí es donde se presenta el problema de que muchas veces los usuarios necesitan localizar de una forma más rápida y eficiente una o más imágenes, para solucionar este problema en [20] se presenta una aplicación denominada Picadomo, la cual combina la recuperación de imágenes en base a su contenido y la búsqueda de facetas, con estas características los usuarios localizan utilizando propiedades visuales (color dominante, contraste, tags que tiene la imagen, entre otros) las imágenes que necesita de una forma más sencilla y eficiente.

Debido al incremento que actualmente está registrando el uso de telefonía móvil existen alrededor de cuatro billones de estos dispositivos, por esta misma situación es de consideración el tener cuidado en ciertos aspectos como la seguridad, privacidad en la computación social móvil, cuestiones que hasta el momento no se toman en cuenta, sin embargo en [21] se analizan estos aspectos y para investigar esto se propone realizar un dispositivo para experimentar e investigar la seguridad y la privacidad, este dispositivo se basa en *Android*, debido a que es una plataforma abierta, incluye APIs y una gran integración con una amplia variedad de sensores y servicios. Gracias a esto será posible recopilar datos para solucionar la problemática antes planteada.

En [22] se describe que la próxima generación de dispositivos móviles utilizarán la tecnología W*ireless Broadband Access*, con esta tecnología se mejoran algunas aplicaciones como por ejemplo el *e-commerce*, las cuales actualmente tienen la problemática de que no se presentan de forma correcta en un dispositivo móvil debido a las restricciones descritas con anterioridad, la nueva generación de dispositivos móviles con tecnologías innovadoras de HCI (*multimodel human–computer interfaces*) y WBA realizan una revolución más allá de las soluciones de SMS que actualmente tienen algunos de los comercios.

Otra de las áreas donde se utiliza la tecnología móvil es en la educación, en donde actualmente existen herramientas de creación que diseñadas para ofrecer una rápida y eficiente producción de objetos de aprendizaje basados en Web, sin embargo estas herramientas tienen los mismos problemas de que es el mismo creador el que define e introduce el conocimiento para cualquier curso, en [23] se presenta una herramienta llamada EDUCA, la cual permite a una comunidad de autores y lectores el crear, compartir y aprender de los materiales y recursos Web en un ambiente adaptativo, esto combina métodos colaborativos, métodos *e-learning* móviles. Para ello combina técnicas de inteligencia artificial para seleccionar el mejor estilo de aprendizaje.

Siguiendo con ejemplos aplicados a la educación, en [24] se presenta un solución al problema de que los *Learning activity based standards* como el *IMS Learning Design*(IMS-LD, son estándares emergentes para especificar y ejecutar diseños de aprendizaje) se desarrollan en arquitecturas cliente / servidor, esto claramente es una problemática para los recursos limitados en los dispositivos móviles que quizás no son confiables en el acceso a internet. El sistema que presentan permite que los usuarios (los cuales tienen diferentes roles) utilicen dispositivos móviles para que interactúen unos con otros sin la necesidad de un servidor centralizado toda la comunicación se lleva a cabo mediante el envío de SMS. El sistema además se utiliza por una variedad de aplicaciones *m-learning* que requieren interacción a través de múltiples roles.

El paradigma de las aplicaciones empresariales es otra de las áreas que solo estaban disponibles para las computadoras de escritorio, pero en [25] se presenta el desarrollo de un novedoso *framework* orientado a servicios que permite a las aplicaciones móviles tener una fácil comunicación con *backends* empresariales, permitiendo intercalar la ejecución de servicios en un ambiente local y remoto, además ejecuta aplicaciones dinámicas ligeras. Al mismo tiempo este *framework* permite a las aplicaciones empresariales el controlar servicios y datos del usuario a través de sus dispositivos. Y es que las aplicaciones empresariales móviles necesitan evolucionar a un paradigma de cómputo de negocio orientado a la movilidad, en donde las funciones del negocio y las opciones de colaboración relevantes están disponibles a un gran número de usuarios móviles de la forma más apropiada.

Toshihiko Yamakami en [26] analiza algunas de las características de Web 2.0 desde una perspectiva de la evolución de Internet móvil. El autor propone un modelo de evolución de dos caminos para Web móvil 2.0. El primer camino es a través de Web móvil 1.0 y el segundo camino es a través de Web 2.0, ambos caminos tiene como alcance final Web móvil 2.0, pero se menciona que aún hay retos a vencer. Algunos de los portales móviles tienen un papel central en el Internet móvil. Sin embargo los motores de búsqueda basados en consultas aún están en un estado naciente. Además Toshihiko Yamakami analiza en [27] el actual modelo de negocio de Internet móvil de Japón con una perspectiva desde Web móvil 1.0 a Web móvil 2.0. Se analiza porque fracasó la primera generación del negocio móvil, lo cual se debió a las limitaciones que existían en los dispositivos telefónicos móviles, pero gracias a los avances tecnológicos y a la llegada de Web 2.0 surgen muchos de los nuevos modelos de negocio móviles, el cual además ofrece un nuevo paradigma para la generación de contenido, nuevas bases de datos, motores de búsqueda entre otros. Debido a las limitaciones de los teléfonos móviles (poder de procesamiento, capacidad de memoria y velocidad, tiempo de vida de la batería) estos fallan en satisfacer los requerimientos de aplicaciones complejas, pero una propuesta interesante que se describe en [28] ofrece la capacidad de tener un *middleware* capaz de soportar la reconfiguración automática, esto ofrece la ventaja de cambiar automáticamente los servicios locales y remotos al iniciar o en tiempo de ejecución de las aplicaciones móviles. Esta reconfiguración se activa por un amplio rango de parámetros estáticos y dinámicos, incluyendo preferencias de usuarios, comunicación en la red y nivel de batería. Comparado con otras soluciones similares, esta aproximación apunta a satisfacer muchos requerimientos claves en una reconfiguración automática.

En [29] se presenta un análisis relacionado con las aplicaciones que actualmente ofrecen en Australia las compañías líderes en telecomunicaciones: enfocadas al consumidor, negocios y emergencias, el estudio encontró que las aplicaciones de negocios y enfocadas al consumidor son predominantes, y en muchos casos se traslapan, otros resultados obtenidos son: mientras las aplicaciones para consumidores tienden a ser opcionales y utilizados para propósitos de conveniencia, las aplicaciones de negocios son obligatorias para empleados para obtener mejoras y beneficios organizacionales, finalmente, las aplicaciones de emergencia se identifican como las menos comunes, dada la audiencia y la naturaleza del servicio.

En [30] se analiza la competitividad entre los sistemas operativos de los Smartphone, y es que los *Smartphone* son dispositivos complicados, como si se tuviera una mini computadora, el incremento en la sofisticación de estos dispositivos trae consigo el problema de que la complejidad de su manejo por medio de software, y es aquí donde entra el sistema operativo, el cual es el corazón de *Smartphone*, ya que permite determinar las características, desempeño, seguridad y aplicaciones agregadas. El análisis menciona que actualmente los sistemas operativos dominantes son *Symbian, Apple, RIM y Microsoft*, y que existen dos puntos importantes y que marcan la diferencia en la competencia entre los SO: el creador del dispositivo y el desarrollador de las aplicaciones. Finalmente, gracias a la gran variedad de SO que se tienen actualmente se brindan oportunidades para nuevos negocios, los cuales están emergiendo debido a la necesidad de aplicaciones que demandan los usuarios en los *Smartphone*.

En [31] se presenta un estudio de las tecnologías kSOAP y J2ME Web services APIs (WSA o JSR 172) para el uso de servicios Web en dispositivos móviles, se describe además un caso de estudio con un dispositivo BlackBerry, mediante BlackBerry MDS Studio se crean aplicaciones habilitadas para comunicarse mediante un protocolo simple y compacto con servidores empresariales BlackBerry (BES) mediante servicios Web. Se presenta en [32] una SOA para dispositivos móviles mediante una implementación con COM, se agregan componentes independientes al dispositivo, en las pruebas se utiliza un teléfono Symbian S60 para demostrar la eficacia de la SOA en las aplicaciones móviles. En [33] propone un marco de trabajo SOA para servicios móviles, el objetivo es investigar como la construcción de servicios móviles se benefician del paradigma Orientado a Servicios. El marco de trabajo diseñado por componentes proporciona servicios, dando la facilidad de ser sustituidos por otras instancias o implementaciones. Una arquitectura para el consumo de servicios Web en dispositivos móviles es presentada en [34], se implementa J2ME Web Services APIs en el dispositivo para proveer el acceso a los servicios Web mediante SOAP. En [35] se presenta una arquitectura ligera móvil SOA para aplicaciones de negocio ejecutándose en dispositivos con J2ME, se presenta un marco de trabajo en donde los objetos que se envían al dispositivo móvil se serializan, comprimen y se transfieren al cliente como un mensaje RDF, la comunicación se establece mediante mensajes SOAP, todo el trabajo se realiza mediante un midlet Java que intercambia la información asíncronamente con el servidor, minimizando los datos transferidos y almacenándolos en el dispositivo. Finalmente en [36] se presenta una aplicación móvil basada en localización para enriquecer la experiencia de turistas, la aplicación utiliza los principios de una Arquitectura Orientada a Servicios con servicios Web y proveedores de datos para dar soporte a las interacciones entre turistas.

## 2.2 Análisis comparativo

Se realizó un análisis de los diferentes artículos recopilados, con lo que se realizó una tabla comparativa que contempla los siguientes puntos: el autor o autores, objetivo por el cual se desarrolla la aplicación, contexto para conocer a que área al que se enfoca la aplicación y para concluir se analizó el grado de desarrollo identificando, sólo es un prototipo o si se encuentra totalmente desarrollado. La tabla 2.1 muestra una comparativa acerca del análisis realizado a los artículos.

Tabla 2.1 Comparación de artículos.

|  |  |  |  |
| --- | --- | --- | --- |
| **Autor** | **Objetivo** | **Contexto** | **Grado de desarrollo** |
| Xiao[et al] | Mejorar y facilitar la visualización en los dispositivos | Uso cotidiano | Desarrollado |
| Hub [et al] | Recuperar de forma más eficiente imágenes almacenadas en el teléfono | Uso cotidiano | En desarrollo |
| Alazzawe [et al] | Analizar aspectos de seguridad | Seguridad | En desarrollo |
| Zatarain [et al] | Utilizar *e-learning* en los teléfonos celulares | Educación | En desarrollo |
| Zualkernan [et al] | Permite que los usuarios utilicen dispositivos móviles para que puedan interactuar con otros sin la necesidad de un servidor centralizado en los estándares emergentes para especificar y ejecutar diseños de aprendizaje. | Educación | En desarrollo |
| Ennai [et al] | Desarrollar un framework orientado a servicios que permite a las aplicaciones móviles tener una fácil comunicación con *backends* empresariales, permitiendo intercalar la ejecución de servicios en un ambiente local y remoto | Investigación | En desarrollo |
| Murarasu [et al] | Desarrollar un middleware capaz de soportar la reconfiguración automática, ofrece la ventaja de cambiar automáticamente los servicios locales y remotos al iniciar o en tiempo de ejecución de las aplicaciones móviles | Investigación | En desarrollo |
| LIU [et al] | Desarrollo de una SOA para dispositivos móviles mediante una implementación con COM. | Investigación | Desarrollado |
| Thanh [et al] | Propuesta de un marco de trabajo SOA para servicios móviles, el objetivo es investigar como la construcción de servicios móviles se benefician del paradigma Orientado a Servicios. | Investigación | En desarrollo |
| Tergujeff [et al] | Implementación de J2ME Web Services APIs en dispositivos móviles para proveer el acceso a los servicios Web mediante SOAP. | Investigación | En desarrollo |
| Natchetoi [et al] | Se presenta un marco de trabajo móvil SOA para aplicaciones de negocio ejecutándose en dispositivos con J2ME, los objetos que se envían al dispositivo móvil se serializan, comprimen y se transfieren al cliente como un mensaje RDF. | Investigación | En desarrollo |
| Paganelli [et al] | Se presenta una aplicación móvil basada en localización para enriquecer la experiencia de turistas, utilizando los principios SOA con servicios Web. | Uso cotidiano | Desarrollado |

Gracias a la revisión de diversos artículos se comprueba que la plataforma de desarrollo *Android* se utiliza en diversas áreas, el hecho de ser una plataforma *Open Source* ofrece a los desarrolladores un marco de trabajo completo para que las aplicaciones se adapten a diversos contextos, como son la vida cotidiana, educación, investigación, entre otros, así como también se presentaron diversas aplicaciones que implementan una SOA.

CAPÍTULO 3

APLICACIÓN DE LA METODOLOGÍA

## 3.1 RUP

RUP es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Tiene 3 características esenciales [37]:

1. Está dirigido por los casos de uso: orientan el proyecto a la importancia para el usuario y lo que este quiere.
2. Está centrado en la arquitectura: relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden.
3. Es iterativo e incremental: divide el proyecto en pequeños proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

Como se observa en la figura 3.1 RUP consta de cuatro fases: Inicio, Elaboración, Construcción y Transición, cada fase se subdivide a la vez en iteraciones

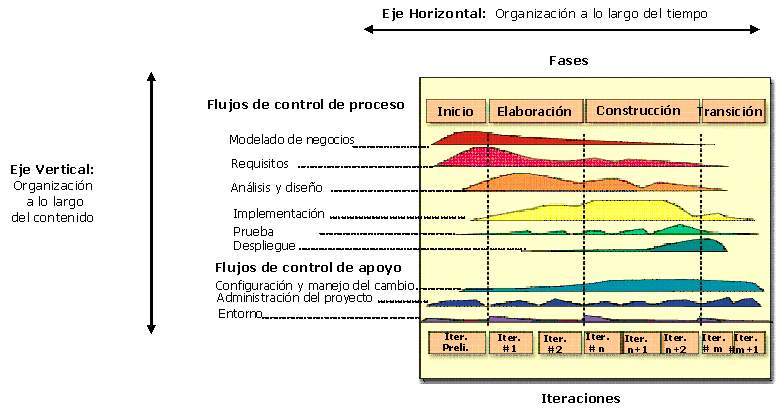


Figura 3.1 Estructura de RUP

El Proceso Unificado consta de dos dimensiones o ejes:

**Eje horizontal:** Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos.

**Eje vertical:** Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles.

Un flujo de trabajo es una relación de actividades que producen resultados observables. A continuación se describen los flujos de trabajo con sus respectivos artefactos generados para cada uno de ellos.

### 3.1.1 Flujo de trabajo de Modelado del negocio

Esta fase se centra más al funcionamiento de la empresa, sobre conocer sus procesos:

* Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado.
* Entender el problema actual en la organización e identificar potenciales mejoras.
* Asegurar que los clientes, usuarios finales, desarrolladores y otros involucrados tienen una visión común de la organización considerada.

A continuación se muestra el modelo de casos de uso del negocio, donde se describe los procesos actuales que se realizan en la empresa Dacme.

Las actividades principales de la tienda se muestran en la figura 3.2, y corresponden a la venta de productos de cómputo al público en general, el cliente solicita al empleado de mostrador los productos que desea adquirir, el empleado realiza la captura de los productos para generar un pedido y lo envía al encargado del almacén, el encargado revisa si existe la suficiente cantidad de producto para surtir el pedido, posteriormente entrega los productos al empleado de mostrador, se genera una nota de venta que se entrega al cliente para que realice el pago correspondiente al pedido y finalmente se entregan los productos y la factura al cliente.

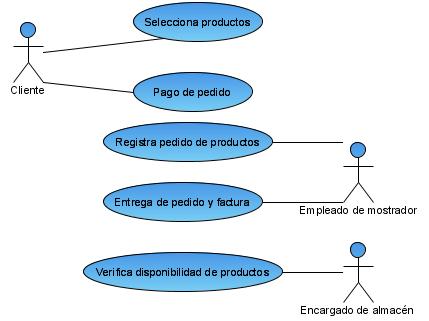


Figura 3.2 Modelado del negocio de la venta

**Actores identificados**

La figura 3.3 muestra los actores que participan en el proceso de venta.

Actores modelado de negocio.jpg

Figura 3.3 Actores identificados

**Cliente**

Es la persona que adquiere los productos que se encuentran disponibles en la tienda.

**Empleado de mostrador**

Personal encargado de atender al cliente, recibir el pedido del cliente y pedir los productos al encargado del almacén, una vez que tiene los productos captura el pedido, recibe el pago del cliente y finalmente entrega los productos.

**Encargado de almacén**

Personal encargado de verificar que exista la cantidad suficiente de productos para surtir el pedido del cliente.

**Diagrama de secuencia del proceso de venta.**

La figura 3.4 muestra el diagrama de secuencia que se realiza en el proceso de venta de productos.

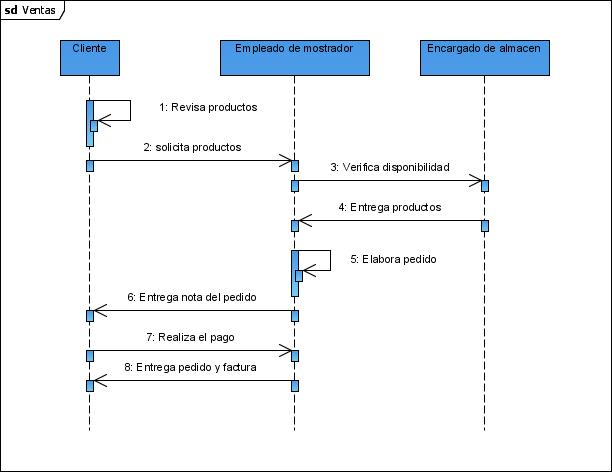


Figura 3.4 Diagrama de secuencia de venta

Diagrama de actividad

La figura 3.5 muestra el diagrama de actividad del proceso de venta que se realiza actualmente en la empresa.

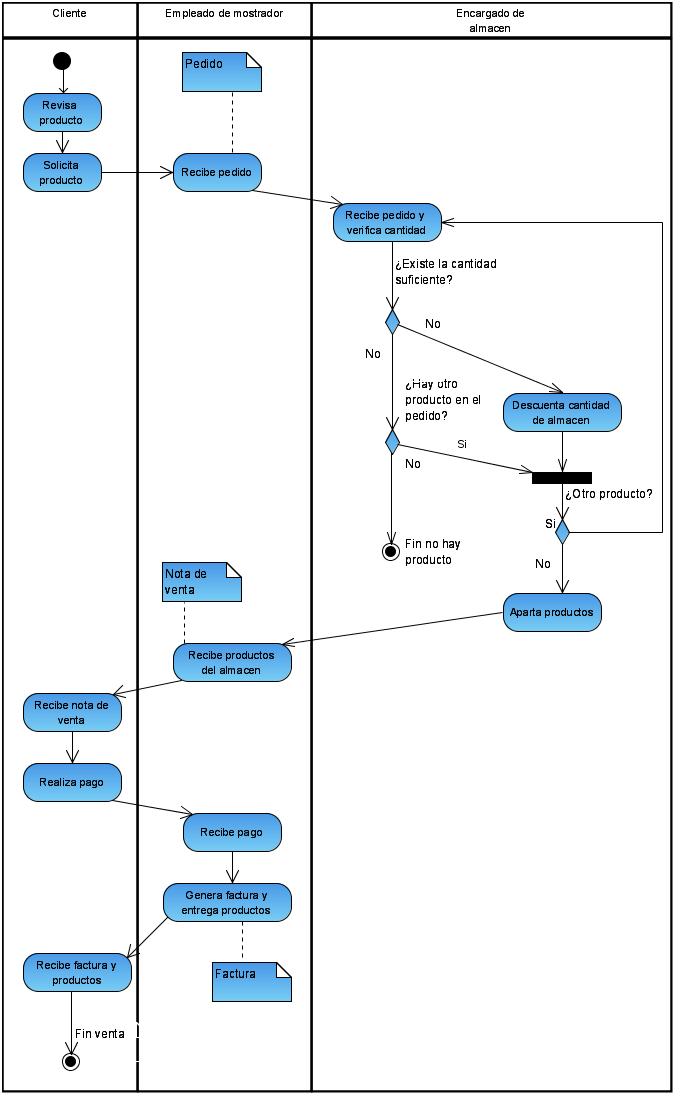


Figura 3.5 Diagrama de secuencia de la venta

Los teléfonos celulares actualmente no integran la tecnología para el uso de servicios Web (por ejemplo SOAP), debido a que muchos tienen capacidades limitadas de procesamiento y no son capaces de manipular los datos que se obtienen de un servicio Web, pero debido a los avances tecnológicos que experimentan estos dispositivos ahora son capaces de manipular más cantidad de información, en el presente capítulo se presenta el diseño de una arquitectura para el soporte de servicios Web en la aplicación de comercio móvil.

## 3.2. Representación arquitectónica

Para la realización del sistema se adopta una Arquitectura SOA por los siguientes beneficios:

* Responden más rápidamente a las condiciones cambiantes del negocio empresarial al promover y permitir la reutilización, al interconectar tecnologías existentes se evita consumir tiempo y costos en la reinvención de los componentes que integran a una aplicación.
* El sistema de comercio electrónico está formado por un conjunto de procesos que son factibles para su reutilización en otros módulos que forman parte de otros sistemas de la empresa.
* Es un estilo arquitectónico basado en estándares de desarrollo de servicios web: XML, WSDL y SOAP, con esto se garantiza la interoperabilidad entre los servicios y aplicaciones.
* Al ser débilmente acoplado ofrece una enorme flexibilidad para la definición, implementación, sustitución y evolución de los servicios que contiene.

## 3.3 SOA con Android

Los teléfonos con el sistema operativo Android no soportan directamente el tratamiento de SOAP y servicios Web, para habilitarlo se requiere que la biblioteca de funciones kSOAP[38] se incluya en el proyecto; la figura 3.1 muestra la relación entre la aplicación y kSOAP, se observa que las Actividades (las clases principales en las aplicaciones Android) se comunican directamente con la biblioteca de funciones para manejar las peticiones HTTP y procesar las respuestas a las invocaciones de los servicios Web.

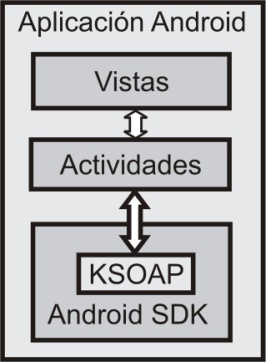


Figura 3.6 Diagrama de la biblioteca de funciones kSOAP como parte de la aplicación Android.

El sistema de comercio móvil utiliza una arquitectura Orientado a Servicios por los siguientes beneficios:

1. Al utilizar estándares de la industria como XML, WSDL y SOAP se garantiza la interoperabilidad entre los distintos servicios y las aplicaciones Android que los utilizan.
2. Los protocolos de comunicación utilizados por los servicios Web son independientes del sistema operativo, plataforma y lenguaje de programación, cualquier cambio en las tecnologías mencionadas no afecta a la aplicación de comercio móvil.
3. Los servicios Web son débilmente acoplados, lo que permite que la aplicación de comercio móvil sea fácilmente escalable, si se requiere nueva funcionalidad solo se requiere crear el servicio Web y estará disponible para su consumo en la aplicación.
4. Existe una alta reutilización al permitir que los servicios se utilicen por la aplicación de comercio móvil y por otro tipo de aplicaciones dentro y fuera de la empresa.

## 3.4 Arquitectura de la aplicación de comercio móvil

El estilo arquitectónico propuesto se estructura por capas para permitir el intercambio de datos, de esta forma cada capa proporciona sus servicios a sus capas adyacentes, en la figura 3.2 muestran estos componentes.

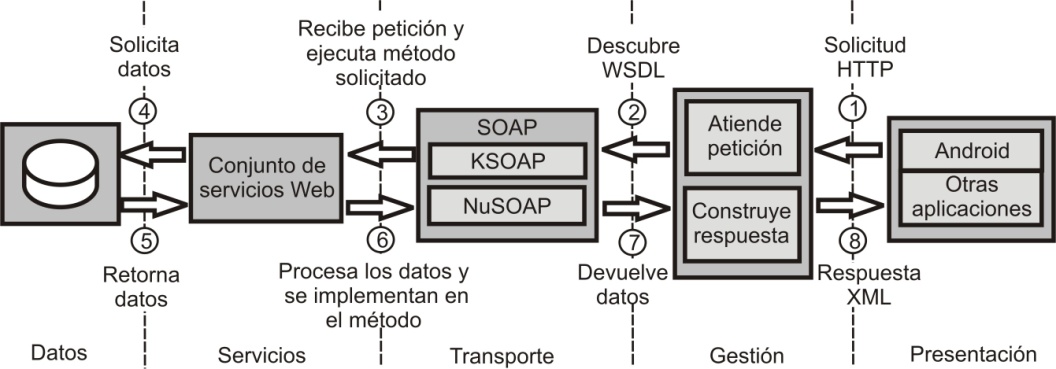


Figura 3.7 Arquitectura en capas de la aplicación de comercio móvil bajo un estilo arquitectónico SOA.

Las capas se describen a continuación:

**Presentación:** Esta capa se encarga de gestionar la entrada y salida de datos al usuario, en el teléfono Android se ejecuta una aplicación que consiste en un cliente ligero encargado de interactuar con el proveedor de servicios, la aplicación se diseña bajo el siguiente patrón:

**Modelo – Vista – Controlador (MVC)**

Es un patrón de diseño que divide una aplicación en tres módulos claramente identificables y con una funcionalidad bien definida:

* Modelo: Corresponde a la parte de las actividades (Activity) que manipulan los datos en la aplicación.
* Vista: Se encarga de mostrar al usuario la información contenida en el modelo. Las vistas corresponden a la interfaz que se presenta al usuario en el teléfono.
* Controlador: Es la parte de la actividad que se encarga de dirigir el flujo de control de la aplicación debido a mensajes externos, como datos introducidos por el usuario en el teléfono.

Una vez obtenida la respuesta XML del servicio Web el Modelo lo transforma a un objeto con la estructura anyType{key0=elemento; key1=element1; key2=element2; …;keyN-1 = elementN-1}, para extraer cada uno de los elementos se utiliza la clase SoapPrimitive y de esta forma se obtienen de manera separada las llaves junto con sus elementos y se utilizan en las actividades.

**Gestión:** Su propósito fundamental es gestionar las peticiones SOAP y proveer acceso a los documentos WSDL, además clasifica, cataloga y maneja servicios Web de forma de que sean descubiertos y consumidos por las aplicaciones.

**Transporte:** Atiende y consume servicios Web por medio de SOAP, para realizar las operaciones se utilizan las siguientes bibliotecas de funciones:

* **KSOAP:** Se emplea para los clientes desarrollados en ambientes Java para el consumo de servicios Web SOAP. Utiliza HTTP como principal mecanismo de comunicación [6], esta biblioteca de funciones debe formar parte de las aplicaciones Android.
* **NuSOAP:** Se emplea en el servidor para desarrollar servicios Web con el lenguaje PHP, está compuesto de un conjunto de clases que proveen el soporte para el desarrollo de clientes (consumidor) y de servidores (proveedor de servicios). Se basa en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1.

**Servicios:** Encargado de contener los servicios Web. Todas las transacciones de negocio se realizan mediante servicios Web (encargados de exponer toda la funcionalidad del sistema al teléfono y a otras aplicaciones). De esta forma los procesos de negocio del sistema de comercio electrónico son expuestos mediante servicios Web para ser utilizados por el sistema de comercio móvil y otros sistemas dentro o fuera de la empresa. Para desarrollar cada servicio se utilizan estándares de la industria:

* XML como el formato estándar para los datos.
* SOAP como protocolo para el intercambio de datos.
* WSDL para describir la interfaz pública de los servicios.

**Datos:** Capa encargada de gestionar el almacenamiento de los datos en la aplicación de comercio electrónico y en la de comercio móvil, se utiliza el sistema gestor de bases de datos MySQL.

**OsCommerce**

Es una solución de comercio electrónico Open Source desarrollada en PHP ejecutándose en un servidor Web Apache, la aplicación está disponible a través de una Web browser para llevar a cabo el proceso de compra de productos.

La aplicación de comercio electrónico está implementada en el servidor y es la encargada de contener los servicios Web que son empleados por la aplicación de comercio móvil mediante peticiones HTTP, utiliza para ello la biblioteca de funciones NuSOAP, como gestor de bases de datos utiliza MySQL (el cual corresponde a la capa de datos). La figura 3.3 muestra el catálogo de productos en la aplicación OsCommerce.

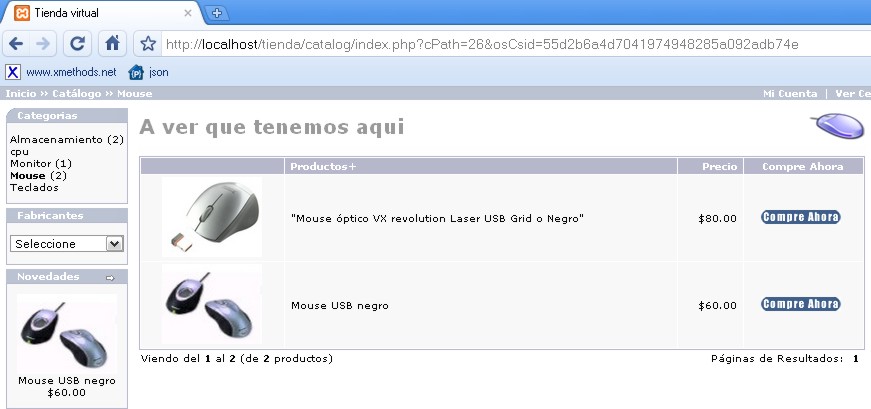


Figura 3.8 Vista del catálogo de productos en la aplicación OsCommerce.

### 3.4.1 Funcionamiento de la arquitectura

La funcionalidad de la aplicación de comercio móvil se explica a continuación, la figura 3.2 contiene encerrados entre círculos los pasos que realiza:

**Paso 1.** El cliente desde el teléfono celular ejecuta la aplicación Android, se realiza una petición HTTP por medio dela biblioteca de funciones kSOAP en donde se especifica la URL de la ubicación del WSDL.

**Paso 2.** Se establece la comunicación por medio del protocolo SOAP, la aplicación cliente descubre en el servidor el WSDL que contiene el método solicitado por la aplicación.

**Paso 3.** El servidor recibe la petición mediante la biblioteca de funciones NuSOAP, la lógica de negocio procesa la petición y ejecuta el método que corresponde a la solicitud de la aplicación cliente.

**Paso 4.** La lógica de negocio se comunica con el servidor de base de datos y ejecuta la consulta SQL que se encuentra en el método.

**Paso 5.** El servidor de base de datos recibe la petición, la procesa y retorna los registros solicitados.

**Paso 6.** El resultado obtenido del servidor de base de datos se procesa, los registros se implementan en el método por la lógica de negocio.

**Paso 7.** NuSOAP construye la respuesta y devuelve los datos por medio del servicio Web (utilizando nuevamente el protocolo SOAP) como un documento XML, en el cliente la biblioteca de funciones kSOAP es el encargado de recibir la respuesta.

**Paso 8.** El documento XML es manipulado en la Actividad, por medio de kSOAP se extraen cada una de las llaves y sus elementos para ser mostrados en las Vistas del dispositivo.

## 3.5 Servicios Web desarrollados

Los servicios Web son parte fundamental para el desarrollo de la aplicación de comercio móvil, al otorgar toda la funcionalidad para llevar a cabo el proceso de compra. La tabla 3.1 muestra las operaciones del servicio Web que consume la aplicación de comercio móvil.

Tabla 3.1 Servicios Web desarrollados

|  |  |
| --- | --- |
| Operación | Descripción |
| obtenerCategorias | Realiza la consulta para obtener las categorías de productos que se encuentran disponibles. |
| obtenerProductosPorCategoria | Realiza la consulta para obtener todos los productos disponibles de la categoría seleccionada, requiere como parámetro de búsqueda el número de categoría. |
| obtenerProducto | Realiza una consulta que obtiene todos los datos de un producto seleccionado, se requiere como parámetro el identificador del producto. |
| agregarProductoCesta | Realiza las inserciones del producto seleccionado por el cliente a la cesta, se requiere como parámetro el identificador de producto. |
| mostrarCesta | Realiza la consulta de los datos de los productos que se encuentran en la cesta para ser presentados en la pantalla del teléfono celular, requiere como parámetro cada uno de los productos seleccionados con anterioridad por el cliente. |
| actualizarCesta | Realiza actualizaciones en la cesta, aumenta o disminuye la cantidad de productos que el cliente ha seleccionado, requiere como parámetro el identificador del producto y la cantidad a actualizar. |
| limpiarCesta | Elimina los productos que se encuentran en la cesta del cliente. |
| confirmarOrden | Realiza la confirmación de la orden de los productos que el cliente ha introducido en la cesta, requiere como parámetros los identificadores de los productos y los datos del cliente. |
| insertaCliente | Captura los datos personales de un cliente cuando se da de alta en el sistema. |
| accesoCliente | Realiza la consulta para validar que es un cliente del sistema, requiere como parámetros el e-mail del cliente y su contraseña. |

### 3.5.1 Aplicación Android

El proceso de invocación de servicio Web inicia en la aplicación Android por medio de una petición HTTP, la biblioteca de funciones kSOAP toma la URL y envía la petición al servidor en donde se tiene implementada la biblioteca de funciones NuSOAP, ésta establece comunicación con la lógica de negocio y se ejecuta el método correspondiente a la búsqueda de categorías, al obtener los registros se construye la respuesta y es enviada por medio del servicio Web en formato XML a la biblioteca de funciones kSOAP de la aplicación Android, se extraen cada uno de los elementos y se presentan en las Vistas al usuario, la figura 3.4 muestra el resultado de esta invocación.

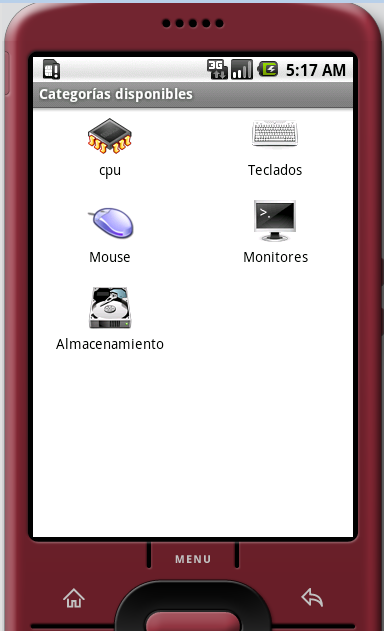


Figura 3.9 Categorías disponibles

Al seleccionar una de las categorías disponibles se realiza una segunda petición al servidor en donde nuevamente se ejecuta un método que se encarga de implementar el servicio Web, construir la respuesta y enviarla a la aplicación de comercio móvil donde finalmente se presenta en el dispositivo el resultado, la figura 3.5 presenta los productos disponible de la categoría.

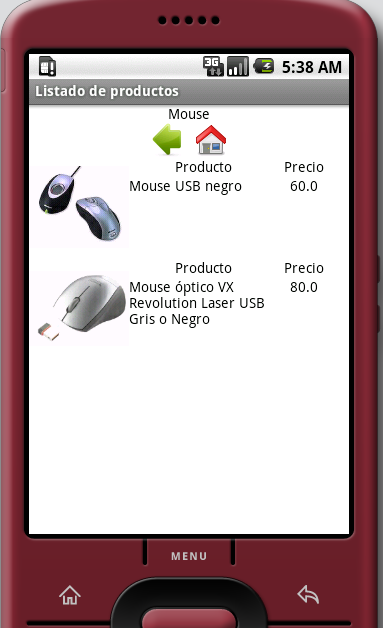


Figura 3.10 Listado de productos por categoría

## 3.6 Proceso de compra electrónica

La aplicación de comercio electrónico OsCommerce y móvil soportan parte del proceso de compra electrónica (área punteada) propuesta por Blommenstein [39], mostrada en la figura 3.6, el envío y entrega del producto, facturación y pago corresponden a procesos internos que lleva a cabo la tienda.



Figura 3.11 Proceso de compra electrónica

Se explican a continuación los subprocesos que cubren las aplicaciones de comercio electrónico y móvil.

### 3.6.1 Selección de producto

Existen tres métodos para seleccionar productos de un catálogo:

1. Navegar en un menú.
2. Búsqueda basada en características y palabras clave.
3. Que el cliente sea aconsejado por un experto.

La figura 3.7 muestra la selección de producto, encerrada en el área punteada se encuentran los pasos que cubren tanto la aplicación OsCommerce como la de comercio móvil.

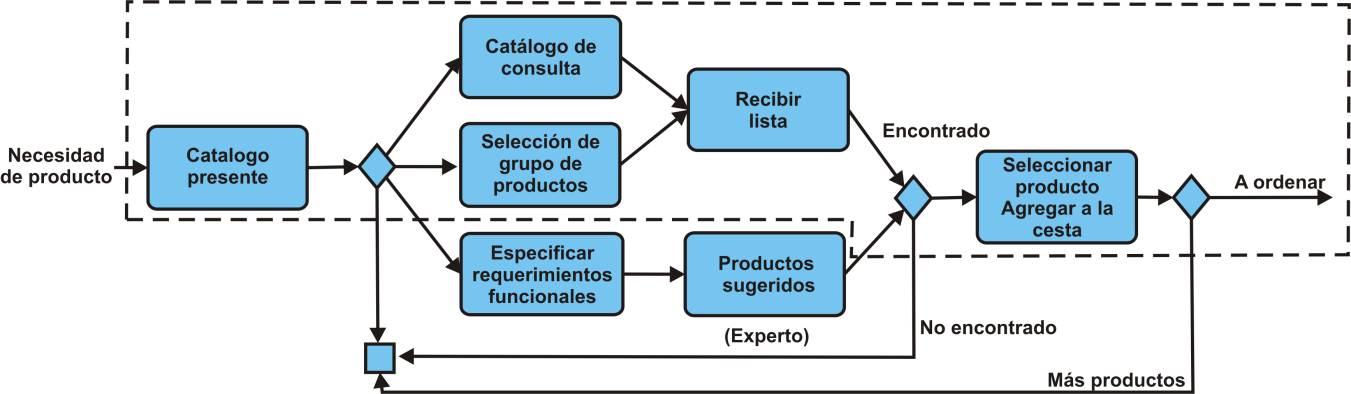


Figura 3.12 Selección de producto.

### 3.6.2 Orden

La orden de compra consiste en los siguientes escenarios:

1. El comprador ha hecho su elección y registrado lo que requiere.
2. La orden es aprobada.

La figura 3.8 muestra los pasos para la orden de compra, las aplicaciones OsCommerce y móvil solo abarca la orden de pedido (área punteada).

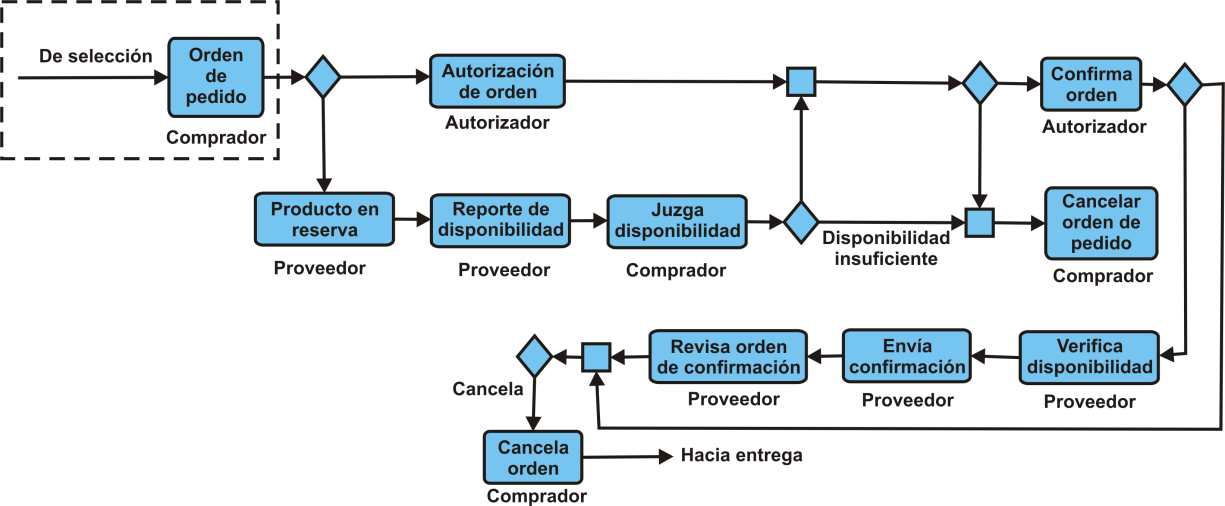


Figura 3.13 Orden de compra.

Los pasos que siguen a la orden de pedido son efectuados de manera interna en la tienda.

# Conclusiones

El presente proyecto surge a raíz de que en la actualidad están emergiendo nuevas alternativas de comercio electrónico, una de ellas es el comercio móvil, la cual abre muchas posibilidades para las empresas que dan a conocer sus productos o servicios, sin embargo debido a las restricciones que imponen los teléfonos celulares, las aplicaciones actuales de este comercio no cumplen con las expectativas de los clientes debido al bajo contenido, navegación difícil de mantener y problemas de interoperabilidad. Al desarrollar una aplicación de comercio móvil bajo la plataforma *Android* se solucionarán los problemas anteriormente descritos, así como eliminar problemas de interoperabilidad que actualmente se tienen con otras tecnologías, una vez desarrollada la aplicación, está se ejecuta con la misma presentación y funcionalidad en cualquier otro dispositivo que cuente con el sistema operativo *Android*.

Otra de las ventajas de este tipo de aplicaciones es que el usuario tenga un acercamiento a la experiencia que ofrecen las aplicaciones de comercio electrónico tradicional, las cuales se desarrollan para computadoras de escritorio. Además de que facilitará que los usuarios realicen la compra de productos o servicios desde cualquier lugar donde se tenga acceso a Internet, debido a que una de las ventajas que se tiene es la movilidad, el usuario no está sujeto a utilizar la aplicación en un lugar específico, como ocurre con las aplicaciones Web de una computadora personal de escritorio.

El desarrollar además una Arquitectura Orientada a Servicios ofrece una nueva forma de comunicación para las aplicaciones de comercio móvil mediante la invocación de servicios Web utilizando SOAP, toda la implementación se realiza solamente con bibliotecas de funciones de código abierto como son NuSOAP para el servidor y kSOAP para las aplicaciones desarrolladas en Android, obteniendo como beneficio que los datos mostrados en los teléfonos se recuperan directamente del servidor mediante la tecnología SOAP y servicios Web. Con esto se espera que las empresas que estén interesadas en el comercio móvil con teléfonos Android apliquen esta arquitectura para ofrecer a sus clientes una nueva alternativa de compra de productos / servicios.

# Glosario

**AJAX:** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Las aplicaciones se ejecutan en el cliente, (en el *Web browser* de los usuarios) mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

**IDE:** (Integrated Development Environment - Entorno Integrado de Desarrollo). Aplicación compuesta por un conjunto de herramientas útiles para un programador. Un entorno IDE es exclusivo para un lenguaje de programación o bien, se utiliza para varios. Suele consistir de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

**GPS:** Siglas en inglés de Global Positioning System, es un sistema global de navegación por satélite (GNSS) que determina en todo el mundo la posición de un objeto. Funciona mediante una red de 27 satélites (24 operativos y 3 de respaldo) en órbita sobre el globo terrestre, a 20200 km, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra.

**OpenGL ES:** Siglas en inglés de OpenGL for Embedded Systems,es una versión simplificada de la API gráfica OpenGL diseñada para dispositivos integrados, es multiplataforma. Ofrece funcionalidad completa para gráficos en 2D y 3D para PDAs, como teléfonos móviles y consolas de videojuegos. La define y promueve el Grupo Khronos, un consorcio de empresas dedicadas a hardware y software gráfico interesadas en APIs gráficas y multimedia.

***PacketVideo's OpenCORE*:** Un subsistema que provee características de manejo de multimedia, permite utilizar formatos estándares, comunicación, grabación de imágenes y video.

**PDA:** Siglas en inglés de *Personal Digital Assistant* (Asistente Digital Personal) es un pequeña computadora que ofrece la ventaja de ser un dispositivo móvil, algunas tienen capacidades de teléfono (*Smartphone*), incorporan *Web browser*, reproductores de música, lista de tareas, manejo de documentos, entre otros. Muchas PDAs acceden a Internet o intranets mediante WiFi.

**Plugin:** Es un complemento que se relaciona con una aplicación para aportarle una función nueva y generalmente muy especifica.

**SDK**: Siglas en inglés de Software Development Kit (Kit de desarrollo de software), es un conjunto de herramientas de desarrollo que permiten el crear aplicaciones para cierto sistema operativo, plataforma de hardware, sistema de cómputo, consola de videojuego, o alguna plataforma.

**Smartphone:** Dispositivo electrónico que funciona como un teléfono móvil con características similares a las de una computadora personal. Una característica importante de los teléfonos inteligentes es que incorporan un sistema operativo que permite la instalación de programas para incrementar el procesamiento de datos y la conectividad.

**SMS:** Siglas en inglés de *Short Message Service*(Servicio de Mensaje Corto). Disponible en redes digitales GSM, es enviar y recibir mensajes de texto de hasta 160 caracteres a teléfonos móviles vía el centro de mensajes de un operador de red.

**WAP:** Siglas en inglés de Wireless Application Protocol(protocolo de aplicaciones inalámbricas), es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, contiene un conjunto de protocolos de comunicaciones para normalizar el modo en que los dispositivos inalámbricos se utilizan para acceder a Internet, coreo electrónico, grupo de noticias, entre otros.

# Bibliografía

|  |  |  |
| --- | --- | --- |
| [1] | Chan H., Lee R., Dillon T., Chang E. *E-Commerce, fundamentals and Applications*. New York: John Wiley & Sons, 2001. | |
| [2] | Zwass, V, “Electronic commerce: Structures and issues”, 1996. [En línea]. Disponible: http://www.gvsu.edu/business/ijec/v1n1/p003full.html. Fecha de consulta: octubre 2009. | |
| [3] | Schneider, G. P. and Perry, J. T., *Electronic Commerce*, Course Technology, 2000. | |
| [4] | Chaffey D., Ellis-Chadwick F., Mayer R., Johnston K., *Internet Marketing: Strategy, Implementation and Practice*. Inglaterra: Pearson Education Limited, 2006. | |
| [5] | Chaffey D., *E-Business and E-Commerce Management*. Inglaterra: Financial Times/Prentice Hall, Harlow, 2006. | |
| [6] | Turban E., Lee J., King D., Chung H. *Electronic Commerce - A Managerial Perspective*, Prentice-Hall, Upper Saddle River, NJ, 2000. | |
| [7] | Wikipedia, the free encyclopedia, “*Mobile commerce*”. Septiembre 2009. [En línea] http://en.wikipedia.org/wiki/Mobile\_commerce. Fecha de consulta: Octubre 2009. | |
| [8] | Lim E., SiauK. *Advances in Mobile Commerce Technologies*. Unites States of America: Idea Group Publishing, 2003. | |
| [9] | Jipping M. J. *Smartphone Operating System Concepts with Symbian OS*. England: John Wiley & Sons, 2007. | |
| [10] | What is Android? | Android Developers, Octubre 2009. [En línea]. Disponible:  http://developer.android.com/guide/basics/what-is-android.html. Fecha de consulta: octubre 2009. | |
| [11] | Java ME Technology, 2009. [En línea]. Disponible: http://java.sun.com/javame/technology/index.jsp. Fecha de consulta: octubre 2009. | |
| [12] | Pressman R. *Software Engineering, a practitioner’s approach*. New York: McGraw Hill. 2001. | |
| [13] | Arquitectura de Software – Introducción, 2005. [En línea]. Disponible: http://download.microsoft.com/download/4/F/F/4FF88340-43CC-4C5B-8E50-09002969D0DD/20051122-ARC-BA.ppt. Fecha de consulta: Mayo 2010. | |
| [14] | Bean J. SOA and Web Services Interface Design. United States of America: Elsevier Inc. 2010. | |
| [15] | Erl T. Service-Oriented Architecture: Concepts, Technology, and Design. Pearson Education. 2005 | |
| [16] | Reynolds F., “Web 2.0–In Your Hand”. PERVASIVE computing. 2009. 86–88 | |
| [17] | Chou W., Li L. “WIPdroid–A Two-way Web Services and Real-time Communication Enabled Mobile Computing Platform for Distributed Services Computing”. 2008 IEEE International Conference on Services Computing. 2008. 205-212. | |
| [18] | Galindo L., Salvachúa J. “Overview of the new user centered mobile applications”. The Second International Conference on Next Generation Mobile Applications, Services, and Technologies. 2008. 167 – 173 | |
| [19] | Xiao Y., Tao Y., Li Q.. “A New Mobile Web Presentation with Better User Experience”. 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing. 2009. 137 – 141 | |
| [20] | Hub A., Blank D., Henrich A., Müller W. “Picadomo: Faceted Image Browsing for Mobile Devices”. 2009 Seventh International Workshop on Content-Based Multimedia Indexing. 2009. 249 – 254 | |
| [21] | Alazzawe A., Alazzawe A., Wijesekera D., Dantu R. “A Testbed for Mobile Social Computing”. | |
| [22] | Jeng I., Chang A., y Wang Y. “Plug into the Online Database and Play Mobile Web 2.0”. IT Pro. 2008. 34- 38 | |
| [23] | Zatarain R., Barrón M., Sandoval G., Urías E., Osorio M., Reyes C. “EDUCA: A Web 2.0 Collaborative, Mobile and E-Learning Authoring System”. 2009 Ninth IEEE International Conference on Advanced Learning Technologies. 2009. 287- 289. | |
| [24] | Zualkernan I., Nikkhah S., Al-SabahM. “A Lightweight Distributed Implementation of IMS LD on Google’s Android Platform”. 2009 Ninth IEEE International Conference on Advanced Learning Technologies. 2009. 59 – 63 | |
| [25] | Ennai A., BoseS. “MobileSOA: A Service Oriented Web 2.0 Framework for Context-Aware, Lightweight and Flexible Mobile Applications”. | |
| [26] | Yamakami T., “MobileWeb 2.0: Lessons from Web 2.0 and Past Mobile Internet Development”. 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07). 2007. | |
| [27] | YamakamiT. “A Turning Point Analysis from Mobile Web 1.0 to Mobile Web 2.0: Lessons from Mobile Content Start-up Company Growth and Downturn in Japan”. 7th International Conference on Mobile Business. 2008. 164 – 170. | |
| [28] | Florindor A., Magedanz T. “Mobile Middleware Solution for Automatic Reconfiguration of Applications”. 2009 Sixth International Conference on Information Technology: New Generations. 2009. 1049 – 1055. | |
| [29] | Abbas R., Michael K., Michael M.G., Aloudat A.. “The Current State of Commercial Location-based Service Offerings in Australia”. 2009 Eighth International Conference on Mobile Business. 2009. 188 – 195 | |
| [30] | Lin F., Ye W. “Operating System Battle in the Ecosystem of Smartphone Industry”. 2009 International Symposium on Information Engineering and Electronic Commerce. 2009. 617 – 621. | |
| [31] | Kozel T., Slaby A., “Mobile devices and Web services”. 7th WSEAS International Conference on APPLIED COMPUTER SCIENCE. 2007. | |
| [32] | LIU, Z., GAO, X., “SOA Based Mobile Device Test”. 2009 Second International Conference on Intelligent Computation Technology and Automation. 2009. | |
| [33] | Thanh D., Jorstad I., “A Service-Oriented Architecture Framework for Mobile Services”. Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/ELearning on Telecommunications Workshop. 2005. | |
| [34] | Tergujeff T., Haajanen J., Leppänen J., Toivonen., “Mobile SOA: Service Orientation on Lightweight Mobile Devices”. 2007 IEEE International Conference on Web Services. 2007. | |
| [35] | Natchetoi Y., Kaufman V., Shapiro A., “Service-Oriented Architecture for Mobile Applications”. SAM’08 ACM. 2008. | |
| [36] | Paganelli F., Parlanti D., Francini N., Giuli N., "A SOA-Based Mobile Guide to Augment Tourists' Experiences with User-Generated Content and Third-Party Services," iciw, pp.435-442, 2009 Fourth International Conference on Internet and Web Applications and Services, 2009. |
| [37] | I. Jacobson, G. Booch, J. Rumbaugh. *El proceso unificado de desarrollo de software.* Madrid: Pearson Education. 2000 |
| [38] | ksoap2-android, agosto 2009. [En línea]. Disponible: http://code.google.com/p/ksoap2-android/.Fecha de consulta: octubre 2009. |
| [39] | Blommnestein F. “Procedures for electronic purchasing”. Project Group. 2000 |
|  |  |