



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS E INFORMÁTICA**

AUTOR: PAZMIÑO PINTO, SANTIAGO DAVID

**TEMA: DESARROLLO E IMPLEMENTACIÓN DE UN PLUGIN DE GOOGLE
WALLET PARA PAGOS ONLINE UTILIZANDO SOFTWARE OPEN SOURCE**

DIRECTOR: ING. GALÁRRAGA, FERNANDO

CODIRECTOR: ING. ÑACATO, GERMÁN

SANGOLQUÍ, OCTUBRE 2013

CERTIFICACIÓN

Certificamos que el presente trabajo, fue realizado en su totalidad por el señor Santiago David Pazmiño Pinto, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA, bajo nuestra supervisión.

Sangolquí, Septiembre de 2013

Ing. Fernando Galárraga

DIRECTOR

Ing. Germán Ñacato

CODIRECTOR

DECLARACIÓN

Yo, Santiago David Pazmiño Pinto, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido presentado previamente para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en el presente documento.

La Universidad de Las Fuerzas Armadas ESPE, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

Santiago David Pazmiño Pinto

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, **SANTIAGO DAVID PAZMIÑO PINTO**, autorizo a la Universidad de las Fuerzas Armadas ESPE la publicación, la publicación en la biblioteca virtual de la Institución, del trabajo “**DESARROLLO E IMPLEMENTACIÓN DE UN PLUGIN DE GOOGLE WALLET PARA PAGOS ONLINE UTILIZANDO SOFTWARE OPEN SOURCE**”, cuyo contenido, ideas y criterios son de mí exclusiva responsabilidad y autoría.

Sangolquí, Octubre del 2013

Santiago David Pazmiño Pinto

DEDICATORIA

Dedico este trabajo a mis padres, porque son ellos la razón y la fuerza que me ha traído por este largo y esperanzador camino y culminar mi carrera de ingeniería.

A mi mamá por ser siempre una luz en los momentos difíciles, por ser ella la que me encaminó a seguir ésta senda.

A mi papá por el esfuerzo y el sacrificio que ha hecho, para permitirme llegar a este día en que culmino mis estudios universitarios.

SANTIAGO DAVID PAZMIÑO PINTO

AGRADECIMIENTO

Agradezco a mis padres por haber trabajado todos estos años, que me han brindado una educación de calidad, con valores y responsabilidades.

A mi Director de Tesis Ing. Fernando Galárraga, por el apoyo brindado desde el inicio del proyecto en las aulas de clase, por su ayuda en los momentos difíciles y su dedicación al proyecto.

A mi Codirector de Tesis Ing. Germán Ñacato, por sus consejos, por su tiempo y su paciencia.

A mis profesores que han aportado todos y cada uno de alguna manera con sus conocimientos para este proyecto.

A mi querida Carrera de Ingeniería en Sistemas e Informática, por haber sido mi segundo hogar durante todos estos años de educación superior.

Muchísimas Gracias.

SANTIAGO DAVID PAZMIÑO PINTO

Índice General

| | | |
|-------|---|----|
| 1 | CAPÍTULO I GENERALIDADES | 1 |
| 1.1 | ANTECEDENTES | 2 |
| 1.2 | PLANTEAMIENTO DEL PROBLEMA..... | 3 |
| 1.3 | JUSTIFICACIÓN E IMPORTANCIA | 4 |
| 1.4 | OBJETIVOS..... | 5 |
| 1.4.1 | OBJETIVO GENERAL..... | 5 |
| 1.4.2 | OBJETIVOS ESPECÍFICOS..... | 5 |
| 1.5 | ALCANCE | 5 |
| 2 | CAPÍTULO II MARCO TEÓRICO | 6 |
| 2.1 | FORMA DE PAGO GOOGLE WALLET..... | 6 |
| 2.1.1 | Acerca de Google Wallet [1]..... | 6 |
| 2.1.2 | Servicio Online de Google Wallet [1]..... | 6 |
| 2.1.3 | Métodos de pago aceptados [2]..... | 7 |
| 2.1.4 | Cobertura [2] | 7 |
| 2.1.5 | Protección contra fraude [3]..... | 8 |
| 2.1.6 | Merchant ID y Merchant Key [4]..... | 10 |
| 2.1.7 | Costos de Transacción del vendedor [5] | 11 |
| 2.1.8 | Integración de Google Wallet Checkout [6] | 12 |
| 2.1.9 | Google Wallet Checkout API [7] | 14 |

| | | |
|--------|---|----|
| 2.1.10 | Pruebas en el Sandbox [8]..... | 17 |
| 2.2 | GESTOR DE CONTENIDOS JOOMLA [9]..... | 19 |
| 2.2.1 | Joomla Framework..... | 20 |
| 2.2.2 | Joomla Extension Directory [10] | 22 |
| 2.2.3 | Componentes [10]..... | 23 |
| 2.2.4 | Módulos [10]..... | 23 |
| 2.2.5 | Plugins [10]..... | 24 |
| 2.3 | SOLUCIÓN E-COMMERCE VIRTUEMART [11] | 24 |
| 2.3.1 | Características de Virtuemart [12] | 25 |
| 2.3.2 | Características principales pre-instaladas [12]..... | 25 |
| 2.3.3 | Nuevas Características Versión 2 [12]..... | 26 |
| 2.3.4 | Información técnica acerca de Virtuemart 2 [13] | 27 |
| 2.3.5 | Lenguaje de Programación Web PHP [14] | 27 |
| 2.3.6 | Extensible Markup Language XML [15]..... | 28 |
| 2.4 | METODOLOGÍA DE DISEÑO DE HIPERMEDIA ORIENTADA A OBJETOS (OOHDM) [16] | 28 |
| 2.4.1 | Concepto | 28 |
| 2.4.2 | Características principales..... | 29 |
| 2.4.3 | Etapas de la Metodología OOHDM..... | 29 |
| 2.4.4 | UML [17] | 38 |

| | | |
|--------|---|----|
| 3 | CAPÍTULO III DISEÑO DE LA APLICACIÓN | 39 |
| 3.1 | Determinación de requerimientos SRS. Estándar IEEE 830..... | 39 |
| 3.1.1 | Introducción | 39 |
| 3.1.2 | Descripción Global..... | 40 |
| 3.1.3 | Funciones del Producto | 42 |
| 3.1.4 | Características del Usuario..... | 43 |
| 3.1.5 | Restricciones | 43 |
| 3.1.6 | Suposiciones y dependencias | 43 |
| 3.1.7 | Requisitos Futuros..... | 44 |
| 3.1.8 | Requisitos Específicos | 44 |
| 3.1.9 | Interfaces Externas | 46 |
| 3.1.10 | Requisitos de Rendimiento | 46 |
| 3.1.11 | Restricciones de Diseño | 46 |
| 3.1.12 | Atributos del sistema..... | 47 |
| 3.1.13 | Atributos del Software | 47 |
| 3.2 | OOHDM..... | 48 |
| 3.2.1 | Determinación de Requerimientos, Identificación de Roles y Tareas | 48 |
| 3.2.2 | Especificación de escenarios..... | 48 |
| 3.2.3 | Modelo y especificación de casos de uso..... | 50 |
| 3.2.4 | Especificación UIDs..... | 58 |

| | | |
|-------|---|--------------------------------------|
| 3.2.5 | Diseño Conceptual | 60 |
| 3.2.6 | Diseño Navegacional | 61 |
| 3.2.7 | Diseño de Interfaces Abstracta..... | 62 |
| 4 | CAPÍTULO IV IMPLEMENTACIÓN Y PRUEBAS | 63 |
| 4.1 | Implementación | 63 |
| 4.2 | Pruebas | 64 |
| 4.2.1 | Plan de Pruebas | 64 |
| 4.2.2 | Casos de Prueba | 68 |
| 4.2.3 | Reporte de Errores e Inconsistencias | 72 |
| 4.2.4 | Control de Calidad | 72 |
| 5 | CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES | 77 |
| 5.1 | Conclusiones | 77 |
| 5.2 | Recomendaciones | 78 |
| 6 | BIBLIOGRAFÍA | 79 |
| | Trabajos citados..... | 80 |
| | ANEXOS | ¡Error! Marcador no definido. |

Listado de Tablas

| | |
|--|----|
| Tabla 2.1 Requerimientos Solución e-commerce Virtuemart..... | 27 |
| Tabla 3.1 Diccionario Caso de Uso Instalar Plugin | 51 |
| Tabla 3.2 Diccionario Caso de Uso Configurar Plugin | 53 |
| Tabla 3.3 Diccionario Caso de Uso Publicar Plugin..... | 55 |
| Tabla 3.4 Diccionario Caso de Uso Seleccionar Plugin | 57 |
| Tabla 4.1 Calendarización..... | 67 |
| Tabla 4.2 Responsables Pruebas | 67 |
| Tabla 4.3 Caso de Prueba CS-001..... | 68 |
| Tabla 4.4 Caso de Prueba CS-002..... | 69 |
| Tabla 4.5 Caso de Prueba CS-003..... | 70 |
| Tabla 4.6 Caso de Prueba CS-004..... | 71 |
| Tabla 4.7 Reporte de Errores e Inconsistencias | 72 |

Listado de Figuras

| | |
|--|----|
| Figura 2.1 Diagrama del trabajo conjunto de las tres capas (el framework, la aplicación CMS, y las extensiones)..... | 21 |
| Figura 2.2 Fases de Desarrollo OOADM..... | 30 |
| Figura 2.3 UID en UML..... | 32 |
| Figura 2.4 Ejemplo de Diseño Conceptual | 33 |
| Figura 2.5 Ejemplo Contexto Navegacional | 35 |
| Figura 2.6 Ejemplo Diagrama de Vista de Datos Abstractos (ADV) | 36 |
| Figura 3.1 Caso de Uso, Instalar Plugin..... | 50 |
| Figura 3.2 Caso de Uso, Configurar Plugin | 52 |
| Figura 3.3 Caso de Uso, Publicar Plugin | 54 |
| Figura 3.4 Caso de Uso, Seleccionar Plugin | 56 |
| Figura 3.5 Caso de Uso Instalar Plugin: UID | 58 |
| Figura 3.6 Caso de Uso Configurar Plugin: UID | 58 |
| Figura 3.7 Caso de Uso Publicar Plugin: UID | 59 |
| Figura 3.8 Caso de Uso Seleccionar Forma de Pago: UID | 59 |
| Figura 3.9 Diseño Conceptual..... | 61 |
| Figura 3.10 Diseño Navegacional | 61 |
| Figura 3.11 ADV Configuración Plugin | 62 |

Listado de Figuras de Anexos

| | |
|--|--------------------------------------|
| Figura A.1 Xampp Setup | ¡Error! Marcador no definido. |
| Figura A.2 Selección componentes..... | ¡Error! Marcador no definido. |
| Figura A.3 Selección Carpeta Destino..... | ¡Error! Marcador no definido. |
| Figura A.4 Inicio Instalación..... | ¡Error! Marcador no definido. |
| Figura A.5 Progreso Instalación..... | ¡Error! Marcador no definido. |
| Figura A.6 Instalación Finalizada | ¡Error! Marcador no definido. |
| Figura A.7 Inicio de Servicios Xampp..... | ¡Error! Marcador no definido. |
| Figura A.8 Xampp Control Panel..... | ¡Error! Marcador no definido. |
| Figura A.9 Sitio de Administración MySQL..... | ¡Error! Marcador no definido. |
| Figura A.10 Creación Base de Datos | ¡Error! Marcador no definido. |
| Figura A.11 Creación Usuario Base de Datos | ¡Error! Marcador no definido. |
| Figura A.12 Lista de Usuarios Base de Datos | ¡Error! Marcador no definido. |
| Figura A.13 Creación Carpeta Raíz del Sitio..... | ¡Error! Marcador no definido. |
| Figura A.14 Extracción Instalador Joomla..... | ¡Error! Marcador no definido. |
| Figura A.15 Selección Idioma Instalación Joomla | ¡Error! Marcador no definido. |
| Figura A.16 Recopilación Información Servidor..... | ¡Error! Marcador no definido. |
| Figura A.17 Licencia Joomla | ¡Error! Marcador no definido. |
| Figura A.18 Conexión a la Base de Datos | ¡Error! Marcador no definido. |
| Figura A.19 Configuración FTP | ¡Error! Marcador no definido. |
| Figura A.20 Configuración Administración del Sitio Joomla; | ¡Error! Marcador no definido. |
| Figura A.21 Finalización Instalación Joomla | ¡Error! Marcador no definido. |
| Figura A.22 Panel de Administración Joomla | ¡Error! Marcador no definido. |

| | |
|--|--------------------------------------|
| Figura A.23 Extension Manager Joomla..... | ¡Error! Marcador no definido. |
| Figura A.24 Instalación Exitosa Virtuemart | ¡Error! Marcador no definido. |
| Figura A.25 Administrador de Templates..... | ¡Error! Marcador no definido. |
| Figura A.26 Configuración Componente Virtuemart | ¡Error! Marcador no definido. |
| Figura B.1 Creación Cuenta Google | ¡Error! Marcador no definido. |
| Figura B.2 Configuración Cuenta Vendedor | ¡Error! Marcador no definido. |
| Figura B.3 Merchant ID y Merchant Key | ¡Error! Marcador no definido. |
| Figura B.4 Google Wallet Sandbox | ¡Error! Marcador no definido. |
| Figura C.1 Selección Plugin Google Wallet | ¡Error! Marcador no definido. |
| Figura C.2 Instalación Correcta Plugin Google Wallet | ¡Error! Marcador no definido. |
| Figura C.3 Activación Plugin de Pago Google Wallet | ¡Error! Marcador no definido. |
| Figura C.4 Creación Nueva Forma de Pago..... | ¡Error! Marcador no definido. |
| Figura C.5 Configuración Plugin de Pago Google Wallet ... | ¡Error! Marcador no definido. |
| Figura D.1 Página de Inicio Tienda Virtual | ¡Error! Marcador no definido. |
| Figura D.2 Agregar Productos al Carrito | ¡Error! Marcador no definido. |
| Figura D.3 Carrito de Compras..... | ¡Error! Marcador no definido. |
| Figura D.4 Información de Facturación y Envío | ¡Error! Marcador no definido. |
| Figura D.5 Selección Método de Envío y Forma de Pago ... | ¡Error! Marcador no definido. |
| Figura D.6 Botón de Pago Google Wallet | ¡Error! Marcador no definido. |
| Figura D.7 Sitio Google Wallet con Información del Pedido; | ¡Error! Marcador no definido. |
| Figura D.8 Selección Método de Pago en Google Wallet ... | ¡Error! Marcador no definido. |
| Figura D.9 Mensaje Pago Finalizado Google Wallet..... | ¡Error! Marcador no definido. |

RESUMEN

El crecimiento del mercado de internet así como la aparición de innumerables tiendas virtuales, ha generado un espacio para la aparición de billeteras virtuales, en las cuales los usuarios pueden guardar su información financiera para ser usada de manera segura para pagos en internet. El presente proyecto se realizó en base a la oportunidad de integrar la billetera virtual Google Wallet con la solución e-commerce Virtuemart (que funciona en conjunto con Joomla), a través de un plugin de pago, para ofrecer a los usuarios de la solución una fácil manera de agregar Google Wallet a sus métodos de pago aceptados. La metodología usada en el proyecto es OOHDM (Metodología de Desarrollo Hipermedia Orientada a Objetos) la cual, gracias a sus fases y a su orientación a objetos, ofrece un marco ideal para el desarrollo del plugin. Finalmente se creó un plugin de fácil instalación, a través del gestor de extensiones de Joomla; de fácil configuración, en los métodos de pago de Virtuemart; y de fácil uso por parte de los clientes de la tienda virtual.

Palabras Clave: Plugin, Google Wallet, Joomla, Virtuemart, OOHDM.

ABSTRACT

The expansion of the internet market and the apparition of countless virtual stores, have developed a space for virtual wallets, in which the users can store their financial information in order to use this to make payments trough internet in a safe way. This project was developed to pursue an opportunity to integrate de virtual wallet Google Wallet and the e-commerce solution Virtuemart (that works along with the CMS Joomla), trough a payment plugin, to offer this solution to users who want to easily add Google Wallet to their accepted payment methods. The methodology used in this Project is OOHDM (Object Oriented Hypermedia Design Methodology) in which, thanks to its phases and to its object oriented model, offers an ideal frame to the development of the plugin. Finally the plugin created has an easy installation method, trough the Joomla Extension Manager; with easy configuration, in Virtuemart Payment Methods; and easy use for the virtual store clients.

Keywords: Plugin, Google Wallet, Joomla, Virtuemart, OOHDM.

CAPÍTULO I GENERALIDADES

El vertiginoso avance de los sistemas de información y comunicaciones en el mundo, ha afectado de diversas formas a cada aspecto del día a día de las personas, uno de esos aspectos ha sido el económico, esto en el sentido que ahora se tiene numerosos servicios financieros, así como la posibilidad de realizar pagos y compras a través de la web, la aparición y desarrollo de este comercio electrónico ha traído grandes cambios a lo que habitualmente era la forma de realizar comercio en las empresas.

Es en esta nueva forma, que las empresas buscan tener su espacio en la web, en el cual vender sus productos directamente al cliente, de esta forma expandiendo el mercado de su empresa a nivel mundial, así como también un cambio en el modelo de distribución, hacen que el comercio electrónico marque un cambio en la manera de vender, ya sea que sea una empresa con presencia solamente en la web o como una que tiene sólo un mercado tradicional.

Ahora tomando en cuenta que cada empresa cuente con su tienda virtual, significa que cada una tendría su propio registro de clientes, su propio manejo de cuentas bancarias, números de tarjeta y otras formas de pago. Es esta una gran cantidad de información financiera de cada uno de sus usuarios.

Para los usuarios especificar su información en cada uno de estos sitios, tanto su información personal como financiera, no es un proceso que quieran realizar múltiples veces, mas aún no tienen la confianza de escribir su información financiera en cada sitio,

ya que no conocen la seguridad del mismo, aún menos qué tratamiento se le dará a su información.

En este punto es donde se centra una gran lucha entre las grandes empresas de la web, la de la gestión de transacciones financieras y la de los datos de millones de personas. Empresas como eBay, Google y Apple, además de otras, son las que están en busca de su espacio en el mercado de gestión de información financiera. eBay a través de su división de pago PayPal ostenta gran parte de este mercado de gestión, pero la entrada de Google a este espacio, inicialmente con su producto Google Checkout, actualmente Google Wallet, ha dado como resultado una aceleración en la búsqueda de obtener el dominio sobre el mercado de gestión de información financiera, como también de formas de pago a través del móvil.

1.1 ANTECEDENTES

Google Checkout, disponible en Estados Unidos desde el 28 de enero de 2006, fue un servicio de procesamiento de pago online provisto por Google, su finalidad era simplificar el proceso de pago de compras online. Los usuarios guardaban su información financiera y dirección de envío en sus cuentas de Google, con lo cual podían realizar compras a través del botón de Google Checkout.

Para el 19 de septiembre de 2011, Google Checkout fue reemplazado por Google Wallet, el cual es visto como una mejor competencia para PayPal.

Joomla es un sistema de gestión de contenidos, el cual permite construir sitios y aplicaciones web. Muchos de sus aspectos, incluyendo su facilidad de uso así como su

escalabilidad, han hecho de Joomla uno de los software de sitios web más populares del mundo, Joomla es un software open source.

Virtuemart es una solución de e-commerce open source, para ser usada como extensión en un sistema de gestión de contenidos como Joomla, es la solución para ventas online más popular para Joomla, con más de 2.5 millones de descargas.

1.2 PLANTEAMIENTO DEL PROBLEMA

El problema surge debido a la carencia de un plugin actualizado para la forma de pago Google Wallet que facilite la implementación de la misma en tiendas virtuales basadas en la solución, open source de e-commerce Virtuemart, la cual trabaja conjuntamente con un sistema gestor de contenidos llamada Joomla.

Con este plugin se pretende solucionar la deficiencia que presentan los plugin disponibles en cuanto a configuración y manejabilidad de la forma de pago Google Wallet y todos sus parámetros.

1.3 JUSTIFICACIÓN E IMPORTANCIA

Actualmente, la solución de e-commerce Virtuemart se encuentra en su versión 2.0.16, por lo cual muchos de los cambios de la versión anterior, especialmente en lo referente al nuevo sistema de plugins, basado en la estructura de plugins de Joomla, han resultado en la necesidad de realizar una migración de los plugins de las versiones anteriores a la versión actual, o de crear nuevos plugins en base a la nueva estructura.

Al desarrollar este trabajo se obtendrá un plugin actualizado para la forma de pago de Google Wallet, lo cual influirá en una mayor difusión de la misma para los usuarios de Virtuemart.

También se documentará paso a paso como desarrollar un plugin para Virtuemart siguiendo los lineamientos de estructura propuestos por los desarrolladores de Virtuemart.

Este trabajo demostrará que el enfoque de desarrollo de software en componentes, es el adecuado al reutilizar código y crear componentes personalizados de acuerdo a las necesidades.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar e implementar un plugin de Google Wallet para pagos online, mediante la utilización de PHP y los lineamientos de Joomla y Virtuemart para el desarrollo de plugins, para disponer de esta forma de pago online dentro de la última versión de Virtuemart.

1.4.2 OBJETIVOS ESPECÍFICOS

- Investigar los lineamientos para el desarrollo de plugins dentro de Virtuemart y Joomla.
- Desarrollar un plugin de Google Wallet de acuerdo a los lineamientos establecidos por Google.
- Implementar el plugin con todas sus funcionalidades, dentro de una tienda virtual de prueba, usando Virtuemart.

1.5 ALCANCE

El plugin a desarrollarse es un componente que será parte de la solución de e-commerce Virtuemart, y de esta manera proveer a los usuarios de Joomla de una nueva forma de pago conocida como Google Wallet.

CAPÍTULO II MARCO TEÓRICO

2.1 FORMA DE PAGO GOOGLE WALLET

2.1.1 Acerca de Google Wallet [1]

Google Wallet es una cartera virtual que almacena la información relacionada con los pagos, el historial de transacciones y las ofertas, entre otras muchas cosas. Todo está sincronizado en la nube, lo que le permite tener siempre una cartera disponible independientemente del dispositivo que use, ya sea en una tienda física o virtual. Puede acceder a Google Wallet descargando la aplicación para móviles Google Wallet o utilizando el servicio online.

2.1.2 Servicio Online de Google Wallet [1]

El servicio online de Google Wallet es un servicio gratuito que le permite utilizar su cartera en la Web. Este servicio almacena la información de su tarjeta de crédito en su cuenta de Google de forma segura para que no tenga que introducir su información de envío y de facturación cada vez que haga una compra online. Cuando finalice una compra online con vendedores que acepten pagos mediante Google Wallet, puede pagar rápidamente iniciando sesión en Google Wallet.

Tenga en cuenta que si anteriormente era usuario de Google Checkout, ahora lo es de Google Wallet. Este servicio está disponible en muchos países

2.1.3 Métodos de pago aceptados [2]

Google Wallet acepta la mayoría de tarjetas de débito y de crédito, entre las que se incluyen VISA, MasterCard, American Express y Discover. Los compradores introducen la información de su tarjeta de débito o crédito cuando se registran por primera vez en Google Wallet y pueden seleccionar su método de pago preferido durante el pago.

2.1.4 Cobertura [2]

Google Wallet Checkout está disponible para comerciantes de EE.UU. y del Reino Unido. Los comerciantes de Google Wallet Checkout de EE.UU. deben tener una dirección y una cuenta bancaria en este país, además de los siguientes elementos:

- Un número de la Seguridad Social (opcional) y una tarjeta de crédito válida.
- Un número de identificación fiscal federal/número de identificación del empresario.

Los compradores de muchos otros países fuera de Estados Unidos y el Reino Unido también se pueden registrar para obtener el servicio y realizar compras mediante Google Wallet. Como sus compras se procesarán siempre en la moneda que corresponda a su dirección (USD para los vendedores de EE.UU. o libras esterlinas para los vendedores del Reino Unido), las tarjetas de crédito de los vendedores proporcionarán una conversión monetaria perfecta.

Los comerciantes de Alemania, Argentina, Australia, Austria, Bélgica, Brasil, Canadá, Corea del Sur, Dinamarca, España, Estados Unidos, Finlandia, Francia, Hong

Kong, Irlanda, Israel, Italia, Japón, México, Noruega, Nueva Zelanda, Países Bajos, Polonia, Portugal, Reino Unido, República Checa, Rusia*, Singapur, Suecia, Suiza y Taiwán* también pueden utilizar Google Wallet Checkout para vender aplicaciones en Google Play.

2.1.5 Protección contra fraude [3]

Google ayuda a los vendedores que usan Google Wallet Checkout a combatir el fraude en varias maneras, resultando en las tasas de fraude más bajas del mercado.

2.1.5.1 Política de Pago Garantizado

La política de pago garantizado es un beneficio único que ofrece a los vendedores protección contra reembolsos no garantizados, permitiéndoles a los vendedores aceptar órdenes de alto riesgo. Para todas las transacciones elegibles, Google reembolsará completamente al vendedor de cualquier reembolso que haya hecho un comprador resultante de algún reclamo por compras no autorizadas y la no recepción de las compras.

2.1.5.2 Como determinar la elegibilidad

La elegibilidad de una orden para el Pago Garantizado, está basada en una compleja tecnología para detección contra fraude. Las órdenes inelegibles no son

necesariamente fraudulentas. Dado que los vendedores conocen mejor su negocio, pueden procesarlas a su discreción.

2.1.5.3 Criterios Adicionales

Aunque una orden sea elegible para el Pago Garantizado, también se debe completar los siguientes criterios para que sea procesado apropiadamente:

- La orden debe ser por objetos tangibles que cumplan las políticas de contenido de Google (las órdenes por objetos no tangibles no son elegibles para Pago Garantizado).
- El vendedor debe enviar la orden a la dirección del comprador (como esta especificado en el Centro para Vendedores de Google Checkout) de acuerdo al método de envío y tiempo de entrega acordado cuando la orden fue hecha.
- Para transacciones menores a USD \$250, el vendedor debe proveer un número de seguimiento del envío o alguna otra prueba del envío con dirección de envío detallada.
- Para transacciones mayores a USD \$250, el vendedor debe proveer una prueba del envío con la firma del comprador y dirección de envío detallada.
- El vendedor debe mostrar una clara política de devolución en su sitio web.
- El vendedor debe cumplir los términos de servicio de Google.
- El reembolso resultante debe ser a un reclamo por un pago no autorizado o por ítems no recibidos. Reembolsos por reclamos de mercadería defectuosa, ítems diferentes a los descritos o cargos duplicados no están cubiertos.

- Al recibir un requerimiento de parte de Google por documentación adicional, el vendedor debe proveer esta información hasta en diez días laborables.

Si un comprador desea hacer cambios a una orden, el vendedor debería guiar al comprador a cancelar la orden y realizar una nueva, esto si el vendedor desea su calificación por el Pago Garantizado.

2.1.5.4 Política de Resolución de Reembolsos

La política de resolución de reembolsos es la protección básica contra reembolsos disponible para los vendedores en todas las transacciones de Google Checkout. Google evaluará todos los reembolsos que el vendedor reciba y, cuando sea posible, peleará contra los mismos. La cuenta de checkout del vendedor no será debitada por el reembolso si Google puede disputar el reembolso de manera satisfactoria con la documentación de soporte que el vendedor deberá proveer.

2.1.6 Merchant ID y Merchant Key [4]

2.1.6.1 Merchant ID

Merchant ID es un código numérico único que Google Wallet Checkout asigna a cada negocio.

2.1.6.2 *Merchant Key*

Merchant Key es un código único que ayuda a mantener una comunicación segura con Google. Google y el vendedor usaran este código para autenticar y verificar la integridad de cualquier mensaje que intercambien. El vendedor así mismo necesitara su Merchant Key para usarlo ya sea en el carrito de compras pre integrado con Google Wallet o en otras soluciones e-commerce.

2.1.7 **Costos de Transacción del vendedor [5]**

Los costos por procesar transacciones están determinados por el volumen de ventas durante el mes en curso menos cualquier reembolso y son reiniciados el día 5 de cada mes.

El vendedor puede verificar el costo actual por transacciones en la pestaña Pagos de su cuenta Google Wallet Checkout. El costo por transacción de una orden en particular aparece en la página de Detalle de Órdenes. En la tabla 2.1 se muestra el costo por transacción por volumen de ventas.

Tabla 2.1 Costo por Transacción por Volumen de Ventas

| | |
|---|--------------------------------------|
| Ventas mensuales menores a \$3,000 | 2.9% + \$0.30 por transacción |
| \$3,000 - \$9,999.99 | 2.5% + \$0.30 |
| \$10,000 - \$99,999.99 | 2.2% + \$0.30 |
| \$100,000 o mas | 1.9% + \$0.30 |

Órdenes de compradores que residan fuera del país en el cual se encuentra registrada la cuenta de Google Wallet Checkout del vendedor, serán cargadas con 1% adicional al costo de transacción.

2.1.8 Integración de Google Wallet Checkout [6]

2.1.8.1 Soluciones sin firma digital

Órdenes realizadas a través de botones BuyNow, Google shopping cart o el api HTML no usan autenticación mediante firma digital porque no existe una buena forma en que Google pueda mantener la clave requerida de manera segura usando los métodos de integración antes mencionados.

Cuando una orden es emitida sin una firma, Google incluirá un mensaje de advertencia en el correo de notificación de la orden y en los detalles de la orden, en la bandeja de entrada de Órdenes realizadas. Google sugiere revisar estas órdenes antes de ser procesadas, esto para asegurar que los precios por los ítems sean válidos.

2.1.8.2 Vendiendo con Facturas por email

El vendedor puede aceptar por teléfono y otras formas fuera de línea al enviar facturas por email desde el Centro para Vendedores de Google Wallet Checkout. Las facturas contendrán un mensaje personalizado acerca del pago solicitado, más un

vínculo para que los clientes realicen sus pagos al logearse en sus cuentas de Google Wallet.

2.1.8.3 Vendiendo con botones BuyNow

Un botón BuyNow es una imagen que se muestra a lado de objetos en un sitio web para los cuales se quiera ofrecer Google Wallet como método de pago. Cuando los compradores den click en el botón BuyNow, serán redirigidos inmediatamente al proceso de pago teniendo que ver un resumen de lo que están comprando. Después de que los compradores hayan confirmado su pago, la orden será mostrada en el Centro para Vendedores de Google Wallet Checkout para ser procesado por el vendedor.

Botones BuyNow permiten al vendedor ofrecer un solo objeto a la vez, si se planea vender más de un objeto a la vez, Google Shopping Cart será una mejor elección.

Las transacciones con el botón BuyNow no pueden exceder los \$10,000. Transacciones con precios altos también pueden incluir un alto escrutinio para mantener un ambiente seguro para vendedores y clientes.

2.1.8.4 Vendiendo con Google Shopping Cart

Google Shopping Cart es un método fácil y rápido para añadir la funcionalidad de una cesta de compras a una página web, permitiendo a los compradores seleccionar múltiples objetos para ser comprados al ser añadidos individualmente al carro de compras.

Cuando los compradores den click al botón añadir a la cesta junto a los ítems de un sitio web, ese ítem se añadirá a la cesta de compras de Google. La cesta se abre instantáneamente en el sitio web para mostrar los objetos que se han seleccionado para comprar.

Una vez que los compradores estén listos para pagar sus órdenes, un link en la cesta de compras los guiará al flujo de pago de Google Wallet, donde podrán revisar y completar el pago.

2.1.9 Google Wallet Checkout API [7]

Google Checkout Application Programming Interface (API) es un protocolo de Google que permite a los servidores de los vendedores intercambiar información directamente con Google Wallet Checkout. Una integración con la API permite a los vendedores enviar carritos de compras desde sus sitios web hacia Google Wallet Checkout, además de realizar el cobro a tarjetas de crédito, reembolsar órdenes y realizar otras actividades de procesamiento de órdenes de manera programada, en vez de manualmente en el Centro para Vendedores.

Una integración con la API es la elección apropiada para vendedores con soluciones e-commerce existentes que necesiten acceso a características mas avanzadas o desean automatizar el procesamiento de las órdenes y las áreas de manejo.

2.1.9.1 Opciones de integración de la API

Después de que el vendedor se ha registrado para una cuenta de Google Wallet Checkout, puede elegir entre los siguientes dos tipos de integraciones:

- 1) Solamente el carrito de compras
- 2) El carrito de compras más el sistema de proceso de órdenes con Checkout.

2.1.9.1.1 Integración de carrito de compras (Integración nivel uno)

Una integración solo de carrito de compras es la elección apropiada si se desea empezar a recibir pagos de Google Wallet de manera más rápida.

Si se realiza la integración a este nivel:

- Se puede especificar tablas de impuestos y costos de envío que serán enviados automáticamente a Google.
- Las notificaciones de nuevas órdenes se realizaran a través de email.
- Se procesarán las órdenes desde el Centro para Vendedores de Google Checkout.
- No se tendrá acceso a algunas características, como la habilidad de procesar cargos parciales.

Si se planea implementar el API pero no de manera integral con el sistema de procesamiento de pagos con Checkout, Google recomienda implementar el HTML API.

2.1.9.1.2 Integración con procesamiento de órdenes (Integración de nivel dos)

Una integración con procesamiento de órdenes es la elección apropiada si se desea invertir un mayor tiempo en el proceso de integración de manera que se pueda automatizar completamente el manejo de órdenes.

Si se integra en este nivel:

- Se puede especificar tasas más complejas de impuestos y envíos, las cuales serán enviadas automáticamente a Google en respuesta a la información de envío provista por el comprador.
- Se puede aceptar códigos y cupones de promoción.
- Se pueden procesar órdenes parciales.
- Se notificara al vendedor de nuevas órdenes realizadas a través de una redirección a una URL definida por el vendedor.

2.1.9.2 Interfaces API

Los vendedores pueden elegir entre dos tipos de implementación para Google Wallet Checkout:

2.1.9.2.1 HTML APIs

Permite a los vendedores intercambiar información con Google Wallet Checkout usando pares de nombre/valor en HTML forms. Los vendedores también pueden usar HTML API para enviar pares de nombre=valor vía solicitud servidor a servidor HTTP POST. Implementaciones HTML son particularmente apropiadas para vendedores

pequeños que no desean generar código XML. Órdenes en implementaciones HTML no pueden ser firmadas digitalmente, de manera que los vendedores que integren el HTML API y no emitan solicitudes servidor a servidor deberían revisar las órdenes manualmente.

2.1.9.2.2 XML APIs

Implementaciones con XML APIs se recomienda para vendedores que necesitan disponer de órdenes firmadas digitalmente antes de enviarlas a Google. Implementaciones XML también son recomendadas para vendedores que desean ofrecer cupones o descuentos y para vendedores que planean integrar Google Wallet Checkout con su sistema de facturación y proceso de órdenes.

Para integrar solo el carrito de compras usando el HTML API, se debe especificar los ítems, impuestos e información de envío en un form HTML como valores escondidos.

Para completar una integración de nivel 1 con Google Wallet Checkout usando el XML API, se debe generar un carrito de compras XML y enviarlo en un mensaje HTML form o usar un envío servidor a servidor.

2.1.10 Pruebas en el Sandbox [8]

Google Checkout provee dos sistemas de procesamiento de órdenes: el ambiente sandbox y el ambiente de producción. El ambiente sandbox permite a los vendedores

probar la integración sin usar transacciones reales. Para evitar cualquier confusión no se debe presentar a los compradores ningún botón que los redirija al sandbox.

El ambiente de producción permite realizar el cobro de los fondos a un comprador cuando realiza una orden. Se usara el ambiente de producción una vez que se haya completado las pruebas en el sandbox y se esté listo para procesar transacciones reales.

2.1.10.1 Acceso al Sandbox

Para empezar a usar el sandbox:

- Se debe crear una cuenta de comprador en <https://sandbox.google.com/checkout>
- Se debe crear una cuenta de vendedor en <https://sandbox.google.com/checkout/sell>

Se debe crear cuentas separadas en el sandbox para el comprador y vendedor, porque no se puede comprar ítems a una misma cuenta.

El proceso de registro en el sandbox es idéntico al proceso de registro en el ambiente de producción. Sin embargo, dado que el sandbox es un sistema de pruebas, no se hace necesario ingresar tarjetas de crédito reales, información de seguro social u otra información. Solo se necesitará proveer una dirección de email válida, que se convertirá en el nombre de usuario de la cuenta de sandbox.

Crear una cuenta en el ambiente sandbox no crea automáticamente una cuenta en el ambiente de producción, y viceversa. Se debe registrar en cada ambiente por separado.

La cuenta de sandbox tendrá un número diferente de Merchant ID y Merchant Key que la cuenta de producción. Los vendedores deben asegurarse de usar la combinación apropiada de ID/Key en su integración.

2.1.10.2 Pasando a producción

Después de haber completado la fase de pruebas en el sandbox, se debe pasar a producción para empezar a procesar órdenes reales.

Para pasar a producción:

- Se debe cambiar la URLs que se usa en el envío de `sandbox.google.com/checkout` a `checkout.google.com`
- Se debe cambiar el valor del Merchant ID a uno asociado a una cuenta de producción.
- Se debe cambiar el valor del Merchant Key a uno asociado a una cuenta de producción.

2.2 GESTOR DE CONTENIDOS JOOMLA [9]

Un sistema gestor de contenidos (CMS) es un software que permite crear y manejar páginas web fácilmente, al separar la creación del contenido de las mecánicas requeridas para ser presentado en la web.

En un sitio, el contenido es guardado en una base de datos. La vista es creada por un template. Joomla es un cms basado en la web que permite al contenido y los datos ser creados y compartidos de manera colaborativa.

Joomla permite a las personas con o sin conocimientos técnicos de codificación, la posibilidad de tener sitios web dinámicos que pueden ser fácilmente manejados.

2.2.1 Joomla Framework

Los frameworks de software se refieren a la arquitectura de un programa. Los Frameworks son eficientes porque permiten reutilizar código como parte del diseño.

Entender el Joomla! framework no es necesario para la mayoría de usuarios del CMS, pero es la fundamento integral que hace al CMS trabajar tan bien. El framework es un set base de librerías de software y patrones de diseño sobre los que corre el CMS, y es la parte de Joomla que lo convierte en fácilmente extensible.

Joomla! 1.5 introdujo el Joomla! framework como el primer paso para separar los elementos del framework de la aplicación CMS. Esto se realizo para permitir al framework soportar múltiples aplicaciones, no solo el CMS.

La figura 2.1 ilustra como el framework soporta la aplicación CMS y como esta aplicación se puede agrandar con extensiones.

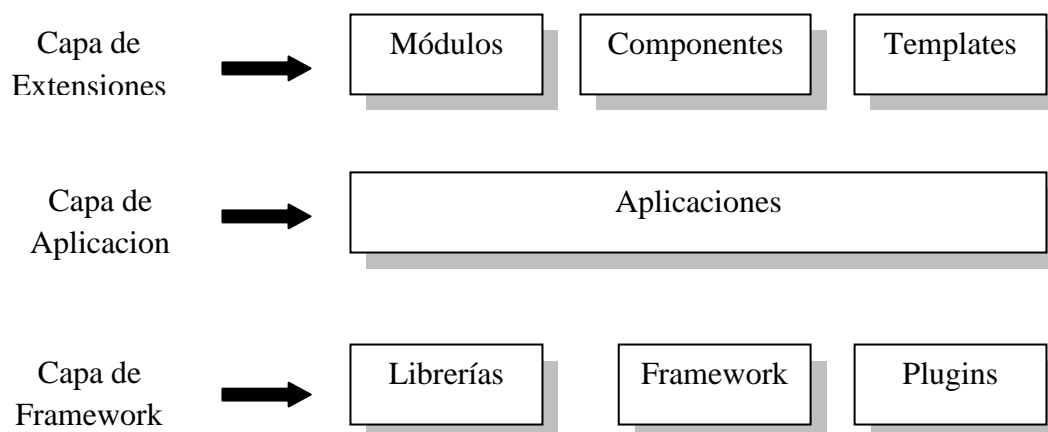


Figura 2.1 Diagrama del trabajo conjunto de las tres capas (el framework, la aplicación CMS, y las extensiones)

Framework, librerías, y plugins en conjunto son la capa fundamental. Las librerías son colecciones de código reusable que realizan una función específica y que pueden ser reusadas por diferentes aplicaciones independientes y el framework. Las librerías permiten a los datos ser usados y compartidos de manera conjunta y modular.

Los Plugins son usados para extender la funcionalidad en el framework. Aplicaciones son programas que corren usando la funcionalidad del framework. La aplicación Joomla! CMS corre sobre el Joomla! framework. La separación de la capa de aplicación de la capa de framework permite a los desarrolladores crear otras aplicaciones que corran usando la funcionalidad del framework.

Módulos, componentes y templates son ítems que extienden a las aplicaciones. Los módulos, por ejemplo, son una forma de presentar información. Componentes son una forma de organizar y guardar información, y los templates son una forma de organizar visualmente la presentación de información.

2.2.2 Joomla Extension Directory [10]

Joomla! Extension Directory (JED) es un portal de información centralizado (<http://extensions.joomla.org>) acerca de extensiones GNU GPL que han sido creadas para Joomla!

Las extensiones ayudan a expandir la funcionalidad de Joomla! El CMS y el framework, y muy probablemente si se necesita de alguna funcionalidad específica que no venía incluida en la instalación estándar de Joomla, ya se ha creado una extensión para ello.

El otro servicio que provee JED es una plataforma de soporte para la comunidad de desarrolladores GNU GPL, comerciales o no comerciales, ya que su contribución a la comunidad Joomla es imperativamente importante para el crecimiento y uso de Joomla.

El directorio en sí está construido de manera que las extensiones son agrupadas por la funcionalidad que proveen.

Un listado contendrá generalmente información acerca de la extensión, qué hace, y señalará si es un módulo, plugin, o componente. También mostrara el nombre del desarrollador o el nombre de la compañía, la fecha de ingreso al directorio, incluirá así

mismo la última fecha en la que la extensión fue actualizada por el desarrollador así como los links de descarga. Algunos desarrolladores incluyen links a demos, opciones de soporte, o documentación.

2.2.3 Componentes [10]

Son mini-aplicaciones que se integran con el framework o el CMS. Contienen su propia específica funcionalidad, su propia estructura de base de datos, y sus propios aspectos de presentación. Ejemplos de componentes son Web Forms Application Suite, una galería de fotos o un sistema de descarga de documentos, o el componente Web Links encontrado en la instalación por default de Joomla.

2.2.4 Módulos [10]

Son elementos de presentación. Toman información que ya existe y la presentan en paquetes visuales que se pueden colocar en el template de manera flexible. Los módulos también ayudan a expandir componentes al permitir a la información de un componente ser presentada visualmente.

Ejemplos de módulos incluyen un slide show de fotos o la ultima foto añadida. La instalación por default de Joomla tiene un módulo de últimas noticias que extiende al componente de contenido para mostrar una lista de los últimos artículos añadidos al sitio.

2.2.5 Plugins [10]

Son pequeñas porciones de código que trabajan detrás de la presentación para asistir a cómo se comporta algo. Mientras una página está siendo cargada, los plugins buscarán en la página y, basados en que es lo que está siendo enviado al navegador, actuar de acuerdo a las instrucciones con las cuales a sido diseñado o programado a hacer. Ejemplos de plugins podrían insertar el código de un proveedor de analytics como Google o Wopra, podrían poner un borde o sombra alrededor de un ítem de manera automática, o también podrían ser el editor de contenido del sitio.

Joomla tiene varios plugins por default como e-mail cloak el cual evita que las direcciones de e-mail presentadas en el sitio sean leídas por spambots.

2.3 SOLUCIÓN E-COMMERCE VIRTUEMART [11]

Virtuemart es una solución e-commerce open source (un componente o plugin) para el sistema gestor de contenidos Joomla. Ambos son distribuidos bajo la licencia GNU General Public License, lo cual significa que son gratis para descargar y usar.

Joomla! y Virtuemart están escritos en PHP y desarrollados para un fácil uso en un ambiente PHP / MySQL. Virtuemart puede ser usado como un carrito de compras o en modo Catálogo.

Es necesario tener instalado Joomla (como el Framework), entonces instalar y configurar Virtuemart.

2.3.1 Características de Virtuemart [12]

Virtuemart ofrece muchas características desde su instalación estándar, pero se puede extender fácilmente la funcionalidad de Joomla y Virtuemart con plugins, componentes, templates y módulos, gratis o comerciales.

Las extensiones están disponibles en <http://extensions.virtuemart.net/extensions-virtuemart-2>

2.3.2 Características principales pre-instaladas [12]

- Productos y categorías ilimitadas.
- Panel de administración basado en la web.
- Envíos altamente configurables.
- Completo control y seguimiento de inventario.
- Diferentes grupos de clientes (Ventas, empleados, o grupos personalizados).
- Múltiples imágenes y archivos (como datos técnicos y volantes) por producto.
- Atributos de producto (como tamaño y color) pueden ser añadidos al producto.
- Tipos de producto para clasificación (como auto, moto o álbum de música).
- Múltiple muestra de precio por producto (basado en la cantidad y/o basado en el grupo de cliente).
- Muestra de precio flexible (formato de número y moneda, con o sin impuestos).

- Rápida conversión entre diferentes monedas usando tasas actuales.
- Estadísticas de compra con resúmenes de nuevos clientes, nuevas compras.
- Manejo de órdenes, con funcionalidades de historial, notificaciones a clientes y edición de orden.
- Reportes básicos: ítems vendidos, ganancias mensuales /anuales.

2.3.3 Nuevas Características Versión 2 [12]

Virtuemart Versión 2.0 y sus subsecuentes actualizaciones han añadido las siguientes características avanzadas:

- Compatibilidad con Joomla 2.5
- Nuevos plugins de pago: PayZen, Authorize.Net, SystemPay, Moneybookers, Authorize.net, Klarna, and Heidelpay.
- Nuevo sistema de facturación incorporado en el núcleo: después de la confirmación de la orden de pago, una notificación es enviada directamente al cliente mediante mail.
- Librería PDF Library añadida en “All-In-One Installer”.
- Títulos de página modificables por categoría y producto.
- Rápida sincronización de medios.
- Migración de la mayoría de información.

- Vista para vendedor añadida, extendida con presentación para, acerca y contacto.
- Página de información legal.

2.3.4 Información técnica acerca de Virtuemart 2 [13]

A diferencia de versiones anteriores, Virtuemart 2 está escrito en código orientado a objetos usando Joomla framework, por lo cual ahora tiene los mismos requisitos de sistema que Joomla; tal como se indica en la tabla 2.1.

Tabla 2.1 Requerimientos Solución e-commerce Virtuemart

| Software | Recomendado | Mínimo |
|-------------------|-------------|--------|
| PHP | 5.3+ | 5.2.4+ |
| MySQL Database | 5.0.4+ | 5.0.4+ |
| Apache Web Server | 2.x+ | 2.x+ |

2.3.5 Lenguaje de Programación Web PHP [14]

PHP, acrónimo de "*PHP: Hypertext Preprocessor*", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

2.3.6 Extensible Markup Language XML [15]

Describe una clase de objetos de datos llamados documentos XML y parcialmente describe el comportamiento de programas que procesan estos documentos. XML es un perfil de aplicación o forma restringida de SGML, Standard Generalized Markup Language [ISO 8879]. Por construcción, los documentos XML son parte documentos SGML.

Los documentos XML están constituidos de unidades de almacenamiento llamadas entidades, las cuales contienen datos analizados o sin analizar. Los datos analizados están conformados de caracteres, algunos de los cuales forman información de caracteres, y algunos de los cuales forman un markup. Un Markup codifica una descripción de la estructura lógica y de la estructura de almacenaje de el documento. XML provee un mecanismo para imponer restricciones en la estructura de almacenaje y en la estructura lógica.

2.4 METODOLOGÍA DE DISEÑO DE HIPERMEDIA ORIENTADA A OBJETOS (OOHDM) [16]

2.4.1 Concepto

OOHDM es una metodología orientada a objetos y ampliamente aceptada para el desarrollo de aplicaciones hipertexto particularmente de aplicaciones web.

Se basa en dos premisas fundamentales:

- La utilización de casos de uso para la fácil captura y definición de requisitos por parte de los usuarios no expertos; de tal manera que los analistas posteriormente puedan generar los diagramas de interacción de usuario (UIDs) o módulos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz.
- Comenzar el diseño del sistema con el fin de que las necesidades de interacción del sistema sean las deseadas por los usuarios.

2.4.2 Características principales

- Apropiaada para un comportamiento complejo. OOHDm provee las mejores herramientas para controlar el desarrollo de una aplicación que tiene un complejo comportamiento.
- Separación del diseño con respecto al desarrollo. Permite que la complejidad del desarrollo de software sea menor y ya que esta ocurre a diferentes niveles.

2.4.3 Etapas de la Metodología OOHDm

OOHDm es una metodología orientada a objetos, comprende cinco etapas tal como se puede visualizar en la figura 2.1.

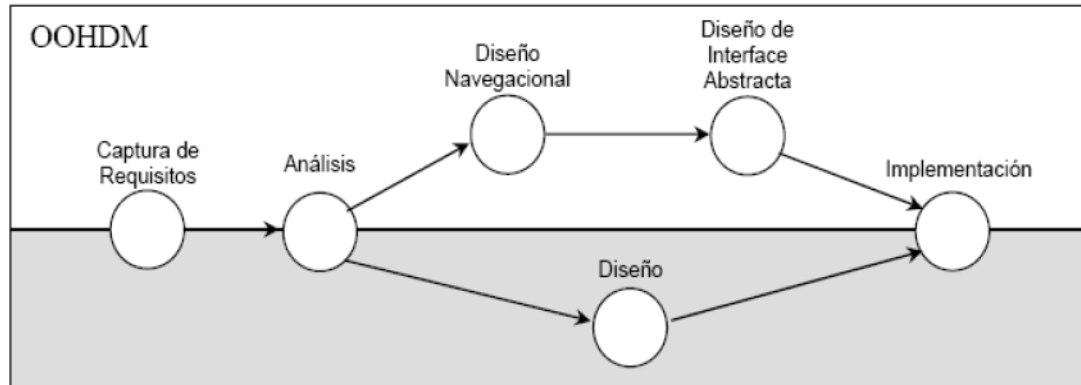


Figura 2.2 Fases de Desarrollo OOADM

OOADM propone el desarrollo de aplicaciones web a través de cinco etapas donde se combinan notaciones gráficas UML (Lenguaje de Modelado Unificado) con otras propias de la metodología que son:

- Definición de requerimientos
- Diseño o Modelo Conceptual
- Diseño Navegacional
- Diseño de Interfaces Abstractas
- Implementación

En cada etapa se crean un conjunto de modelos orientados a objetos que describen un diseño particular. El éxito de esta metodología es la clara identificación de los tres diferentes niveles de diseño en forma independiente de la aplicación.

2.4.3.1 Definición de Requerimientos

Se basa en los diagramas de casos de uso, los cuales son diseñados por escenarios con la finalidad de obtener de manera clara los requerimientos y acciones del sistema y de los usuarios.

OOHDM propone dividir esta etapa en 5 sub etapas:

- **Identificación de Roles y Tareas.** Se determina los posibles roles que podrían tomar los potenciales usuarios de la aplicación, ya que los usuarios pueden tomar diferentes roles importantes en cada intercambio de información con el sistema.
- **Especificación de escenarios.** Los escenarios son descripciones narrativas de cómo la aplicación será utilizada. En esta sub-etapa, cada usuario deberá especificar los escenarios que describen su tarea.
- **Especificación de casos de uso.** Un caso de uso es la descripción a detalle de las actividades y procesos necesarios para utilizar la aplicación.
- **Especificación de UIs.** Los UIs permiten representar de forma gráfica, rápida y sencilla los casos de uso generados en la etapa anterior. Para obtener UIs desde un caso de uso, la secuencia de información intercambiada entre el usuario y el sistema debe ser identificada y organizada en las interacciones, como se indica en la figura 2.3.

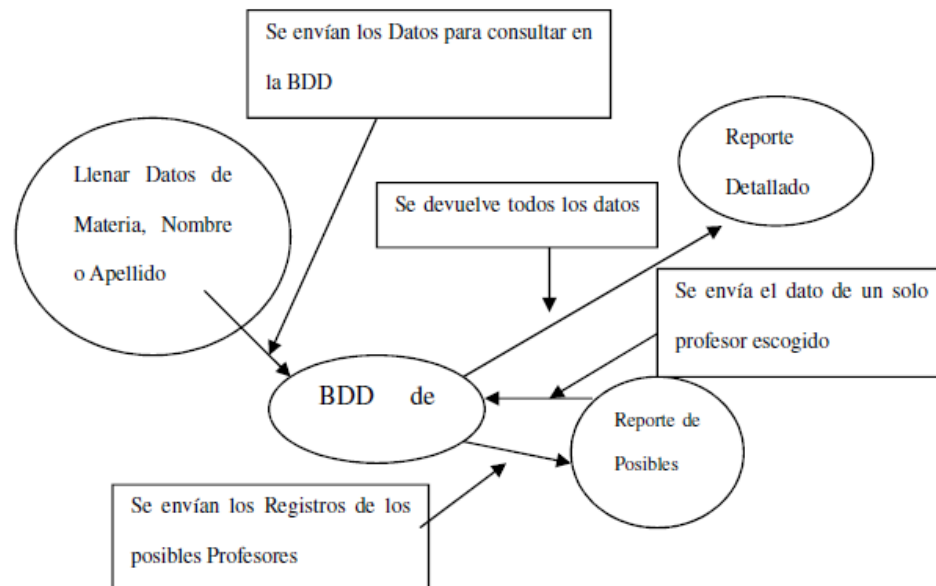


Figura 2.3 UID en UML

- **Validación de Casos de Uso y UIDs.** El desarrollador deberá interactuar con los usuarios para validar los casos de uso y UIDs obtenidos.

2.4.3.2 Diseño Conceptual

Está representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos; es decir formado por clases, relaciones y subsistemas. Este diseño se establece de acuerdo a los UIDs establecidos.

Las clases son descritas y usadas como en los modelos orientados a objetos tradicionales; conectadas por relaciones y los objetos son instancias de las clases.

Las clases son usadas en el diseño navegacional para derivar nodos y las relaciones para construir enlaces o vínculos.

Si la aplicación crece, el diseño conceptual debe estar preparado para ser extendido, en la figura 2.4 se muestra un ejemplo de diseño conceptual.

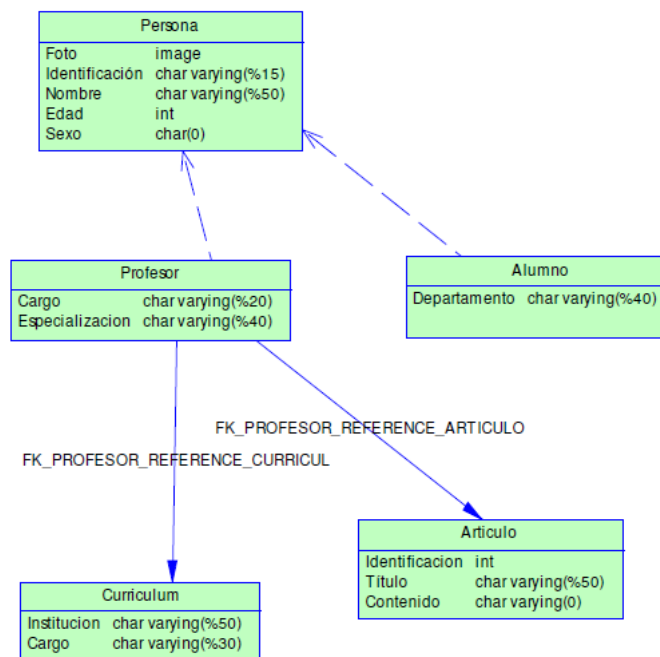


Figura 2.4 Ejemplo de Diseño Conceptual

2.4.3.3 Diseño Navegacional

Un modelo navegacional es construido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios.

El diseño navegacional es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales.

- **Esquema de Clases Navegacionales.** Las posibles vistas del hiper documento se generan a través de las llamadas clases navegacionales, que son tipos predefinidos de clases conformados por: nodos, enlaces, anclas y estructuras de acceso a los índices o recorridos guiados, que representan los posibles caminos de acceso a los nodos.

Nodos.

1. Contenedores de información básicos.
2. Se definen como vistas orientadas a objetos de las clases conceptuales definidas en la fase de diseño conceptual.
3. Contiene atributos simples y enlaces.

Enlaces.

1. Identifican relaciones.
2. Implementan las relaciones definidas en el esquema conceptual.
3. Las clases de los enlaces especifican sus atributos, comportamiento y los objetos fuente y destino.
4. Representan las posibles formas de comenzar la navegación.

Estructuras de Acceso.

1. Actúan como índices o diccionarios.
2. Útiles para ayudar al usuario final a encontrar la información deseada.

- **Esquema de Contextos Navegacionales.** Un esquema de contexto navegacional es el que organiza el espacio navegacional en conjuntos convenientes o sub-espacios que pueden ser recorridos en orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada que es obtener la información requerida, en la figura 2.5 se muestra un ejemplo de contexto navegacional.

Contexto Navegacional

1. Es un conjunto de nodos, enlaces, clases de contextos y otros contextos navegacionales.
2. Inducido de clases navegacionales.
3. Se definen por extensión o enumerando sus miembros.
4. Un índice o un tour guiado definen contextos de navegación.

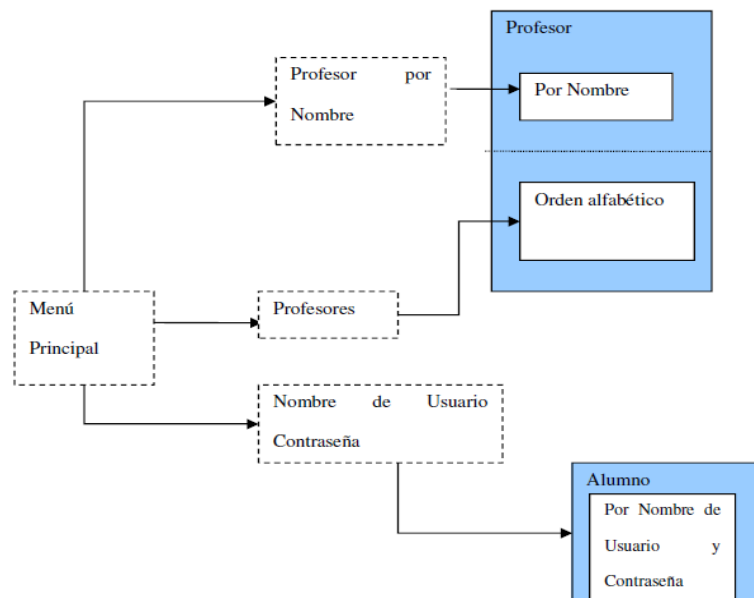


Figura 2.5 Ejemplo Contexto Navegacional

2.4.3.4 *Diseño de Interfaces Abstractas*

En esta etapa se define la forma en la cual los objetos navegacionales pueden aparecer, es decir, especifica la estructura y el comportamiento de la interface del sistema hipermedia con el usuario. Este modelo es abstracto, por lo tanto independiente de la implementación final del sistema.

Este diseño se basa en tres diagramas que se complementan entre sí y son:

- **Diagrama de Vista de Datos Abstractos (ADV).** Este diagrama incluye una vista (ADV) por cada clase navegacional (nodo, enlace o estructura de acceso) que fue establecida durante la fase de Diseño Navegacional.

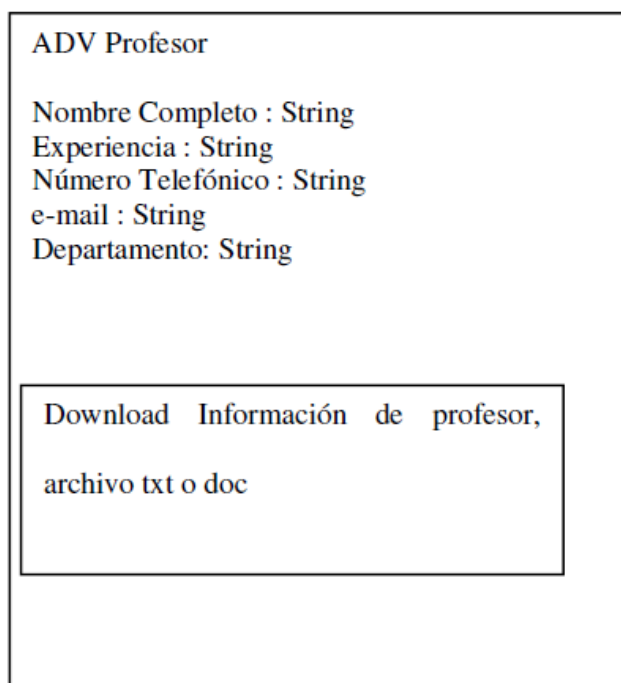


Figura 2.6 Ejemplo Diagrama de Vista de Datos Abstractos (ADV)

- **Diagrama de Configuración.** En este diagrama se representan:
 1. Los eventos externos provocados por el usuario, como clic o doble clic del rato que maneja un ADV.
 2. Los servicios que ofrece el ADV.
 3. Las relaciones estáticas entre las ADVs.

- **Diagrama de Estado.** Se representa el comportamiento dinámico del sistema hipermedia mediante el establecimiento de un diagrama de transición de estados para cada ADVs, en el que se reflejan los posibles estados por los que puede pasar cada objeto de la interface (oculto, desactivado, ampliado, reducido, normal) y los eventos que originan los cambios de estado.

2.4.3.5 Implementación

En esta fase se debe implementar el diseño, ya que hasta el momento todos los modelos fueron construidos en forma independiente de la plataforma de implementación y por tanto, se ha identificado la información que será mostrada, se cuenta con una idea básica de cómo se verán las interfaces y se tiene un completo conocimiento del dominio del problema.

Para comenzar con la implementación se deberá elegir donde se almacenara los objetos y con qué lenguaje o herramienta se desarrollara las interfaces; para después continuar con la creación de las mismas.

2.4.4 UML [17]

Unified Model Language (UML) permite especificar, visualizar y documentar modelos de sistemas de software, incluyendo su estructura y diseño, en un forma que incluya todos estos requerimientos. (se puede usar UML para modelado de negocios y modelado de otros sistemas que no sean software). Usando cualquiera de las numerosas herramientas basadas en UML existentes, se pueden analizar requerimientos futuros de cualquier aplicación y diseñar una solución que resuelva estos requerimientos, representando los resultados usando trece tipos de diagramas estándar con UML 2.0's.

Se puede modelar cualquier tipo de aplicación, funcionando sobre cualquier tipo de combinación de hardware, sistema operativo, lenguaje de programación, y red, en UML.

Construido sobre conceptos fundamentales de orientación a objetos incluyendo clases y operaciones, es un complemento natural para lenguajes y ambientes orientados a objetos, pero también se puede usar para modelar aplicaciones con lenguajes no orientados a objetos.

CAPÍTULO III DISEÑO DE LA APLICACIÓN

3.1 Determinación de requerimientos SRS. Estándar IEEE 830.

3.1.1 Introducción

a) Propósito

El fin de este SRS, es especificar cómo se desenvolverá el proyecto y a que límites llegará.

Va dirigido al administrador del sistema y a los potenciales usuarios.

b) Alcance

- El plugin de Google Wallet para pagos on-line, será una forma de pago disponible para la solución e-commerce Virtuemart, dentro del CMS Joomla, que permitirá al usuario disponer de esta forma de pago desarrollada por Google.
- El plugin dispondrá de las opciones de configuración ya sea para el ambiente de pruebas Sandbox de Google o para el ambiente de producción para uso final.
- Se podrá instalar como una extensión más del CMS a través del manejador de extensiones de Joomla.
- Se podrá configurar desde el administrador de formas de pago de Virtuemart

c) **Definiciones, Siglas y Abreviaciones**

- Joomla, CMS, que ayuda con la administración del portal.
- Virtuemart, solución e-commerce para ser usada como extensión dentro del CMS Joomla
- OOHDM, Object Oriented Hypermedia Design Methodology
- CMS, Content Management System
- Google Wallet, billetera virtual desarrollada por Google.

3.1.2 Descripción Global

a) **Perspectiva del Producto**

El plugin servirá para proporcionar una nueva forma de pago a la solución e-commerce Virtuemart.

El plugin dependerá y solo podrá ser utilizado dentro de la solución e-commerce Virtuemart dentro de una instalación de Joomla.

b) **Interfaces del sistema**

El plugin deberá ser realizado siguiendo los lineamientos para programación de plugins de pago recomendados por los desarrolladores de Virtuemart, para su correcto funcionamiento dentro de la solución e-commerce.

c) Interfaces con el Usuario

El plugin al funcionar dentro del CMS Joomla, mantendrá la estructura que tiene Joomla, para sus extensiones, pero específicamente la presentación de las formas de pago de Virtuemart.

d) Interfaces con el hardware

El plugin al funcionar dentro del CMS Joomla, tendrá los mismos requerimientos de hardware que el CMS, así mismo se deberá tener una conexión a internet ya que las pruebas en el sandbox de Google, como el funcionamiento en producción necesitan el acceso a los servidores de Google.

e) Interfaces con el software

Como se ha venido señalando, principalmente la interfaz de software que tendrá el plugin es la solución e-commerce Virtuemart.

f) Interfaces de comunicaciones**Protocolo TCP/IP**

El Internet Protocol es un protocolo no orientado a la conexión, usado en el origen como en el destino para la comunicación de datos a través de una red de paquetes conmutados.

TCP, es el protocolo que garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.

Protocolo HTTPS

HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

3.1.3 Funciones del Producto

El plugin realizará lo siguiente:

- Tomará la información generada por el proceso de checkout de Virtuemart, para enviar esa información a Google para el pago.
- Redireccionará al usuario al sitio de Google Wallet, al final del proceso de checkout.

- Permitirá dos ambientes de uso de la forma de pago, esto es en ambiente de pruebas en el Sandbox de Google y ambiente de producción.

3.1.4 Características del Usuario

- El administrador deberá conocer con anterioridad la creación de cuentas de Google Wallet para vendedor, tanto en el sandbox como para producción.
- El administrador deberá conocer como instalar y configurar Joomla y Virtuemart.
- El usuario deberá tener una cuenta de Google Wallet activa.

3.1.5 Restricciones

- Lenguajes y tecnologías en uso: HTML, PHP, XML.
- El acceso a los usuarios lo controlarla plataforma de alojamiento de la aplicación (JOOMLA).
- Los protocolos de comunicación: TCP/IP, HTTPS.

3.1.6 Suposiciones y dependencias

El cliente podrá acceder desde cualquier sistema operativo, siempre que tenga un browser, ya sea Internet Explorer, Firefox, Chrome, etc.

3.1.7 Requisitos Futuros

Actualización del plugin, según cambios de versiones futuras de la forma de pago y de la solución e-commerce.

3.1.8 Requisitos Específicos

a) Requerimientos Funcionales

Requerimiento 1. Instalación de plugin

Descripción

Este proceso sirve para instalar el plugin dentro de la plataforma, a través del manejador de extensiones.

Entradas

Plugin en formato reconocido por el manejador de extensiones.

Salidas

Mensaje de instalación correcta del plugin.

Requerimiento 2. Configuración plugin

Descripción

Este proceso sirve para configurar el plugin dentro de la solución e-commerce, a través del link de métodos de pago.

Entradas

Información solicitada por el plugin para su correcto funcionamiento.

Salidas

Mensaje de configuración correcta del plugin.

Requerimiento 3. Publicación del plugin en las formas de pago

Descripción

Este proceso sirve para publicar el plugin dentro de la solución e-commerce, en el paso de selección de método de pago del checkout.

Entradas

Selección de publicación.

Salidas

Mensaje de publicación correcta del plugin.

Requerimiento 4. Uso de la forma de pago

Descripción

El usuario podrá acceder a esta forma de pago en el checkout una vez haya sido configurada y publicada por el administrador.

Entradas

Selección de la forma de pago, en el proceso de checkout.

Salidas

Botón para pago de Google Wallet.

b) Requerimientos No Funcionales

- Red de internet, banda ancha.
- Explorador, se probará en Google Chrome y Firefox.
- La seguridad está restringida a la plataforma Joomla.

3.1.9 Interfaces Externas

Virtuemart, la solución e-commerce sobre la cual se usará el plugin, administrara las funciones del mismo, como configuración, publicación y uso.

3.1.10 Requisitos de Rendimiento

Al ser una forma de pago para la solución e-commerce podrá ser accedido por varios usuarios a la vez (conurrencia), sin que haya un corte en el servicio.

3.1.11 Restricciones de Diseño

No existen restricciones de diseño.

3.1.12 Atributos del sistema

- El plugin se ejecutará conjuntamente con la solución e-commerce Virtuemart, sobre el CMS Joomla, misma que puede ser usada en cualquier servidor web que cumpla los requerimientos mínimos para su instalación.
- Los usuarios podrán acceder al plugin a través del CMS

3.1.13 Atributos del Software

- a) **Fiabilidad.** Al estar basado en la web, la fiabilidad del plugin dependerá del desarrollo, el cual se basa en tecnologías nuevas y metodologías como OOADM, también dependerá de la velocidad de conexión entre el servidor y el cliente.
- b) **Disponibilidad.** El software será desarrollado con el fin de que varios usuarios puedan acceder a la forma de pago.
- c) **Seguridad.** El plugin dependerá de la seguridad del CMS, ya que mediante este se tendrá acceso al mismo para configuración y uso.
- d) **Mantenimiento.** El software estará pensado para que cualquier persona que tenga conocimientos en programación de plugins para Joomla y Virtuemart pueda modificarlo.
- e) **Portabilidad.** El plugin podrá instalarse en Joomla v1.5 o posterior y Virtuemart 2.

3.2 OOHDM

3.2.1 Determinación de Requerimientos, Identificación de Roles y Tareas

Roles

- **Administrador.** Será el encargado de instalar y configurar el plugin dentro de la solución e-commerce.
- **Usuario.** Será la persona que podrá seleccionar y usar la forma de pago.

Tareas

- El administrador instala el plugin.
- El administrador configura el plugin.
- El administrador publica el plugin.
- El usuario selecciona la forma de pago.
- El usuario usa la forma de pago.

3.2.2 Especificación de escenarios

a) Instalación del plugin

El administrador deberá tener el conocimiento de cómo instalar extensiones en Joomla a través del instalador de extensiones.

b) Configuración del plugin

El administrador deberá llenar toda la información requerida por el plugin para su correcto funcionamiento.

c) Publicación del plugin

El administrador publica el plugin para su uso como forma de pago en la solución e-commerce.

d) Selección de la forma de pago

El usuario podrá seleccionar la forma de pago, publicada por el administrador, para su uso en el paso de selección de forma de pago, en el proceso de checkout de la solución e-commerce.

e) Uso de la forma de pago

El usuario será redireccionado al final del proceso de checkout, si ha seleccionado la forma de pago del plugin, a la página de la forma de pago para terminar su transacción.

3.2.3 Modelo y especificación de casos de uso

A continuación el Caso de Uso Instalar Plugin, se muestra en la figura 3.1

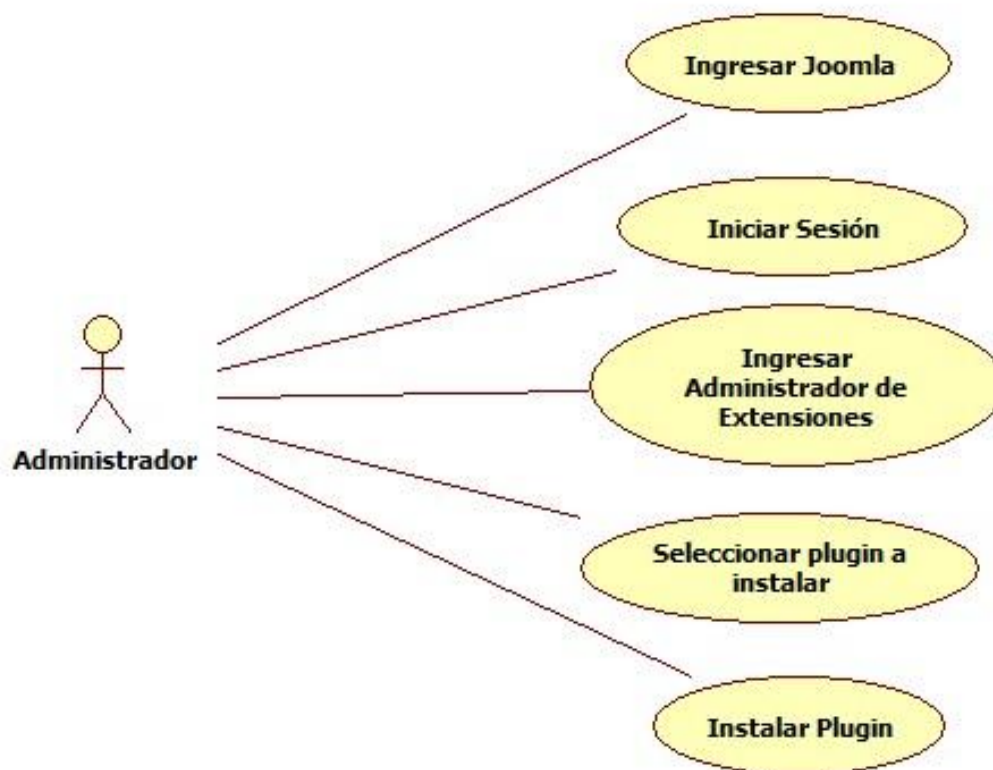


Figura 3.1 Caso de Uso, Instalar Plugin

La especificación del caso de uso, Instalar Plugin, se muestra en la tabla 3.1

Tabla 3.1 Diccionario Caso de Uso Instalar Plugin

| Detalle: | | | |
|--------------------------|--|--|----------|
| Nombre del CDU | Instalar Plugin | | |
| Estado: | Propuesto | Versión 1.0 | Fase 1.0 |
| Autor: | Santiago Pazmiño | | |
| Actor: | Administrador | | |
| Objetivo: | El administrador instala el Plugin de la forma de pago | | |
| Escenarios: | | | |
| Flujo Básico | Actor | Eventos | |
| | 1. Ingresa a página de administración Joomla | 2. Se muestra la página de administración | |
| | 3. Inicia sesión | 4. Se muestra el panel de administración | |
| | 5. Accede al administrador de extensiones | 6. Se muestra el panel de administración de extensiones | |
| | 7. Se selecciona ruta donde se encuentra el plugin | 8. Se muestra el plugin seleccionado | |
| | 9. Se pulsa el botón subir e instalar | 10. Se muestra mensaje de Plugin instalado correctamente | |
| Flujo Alternativo | 1. Ingresa mal usuario o clave | 2. Joomla no permite el ingreso | |
| | 3. Intenta instalar un Plugin no compatible | 4. Joomla no permite la instalación | |
| Pre Condiciones | El usuario debe tener permisos de administrador | | |
| Post Condiciones | El usuario a instalado correctamente el Plugin | | |

El caso de uso Configuración del Plugin se muestra en la figura 3.2

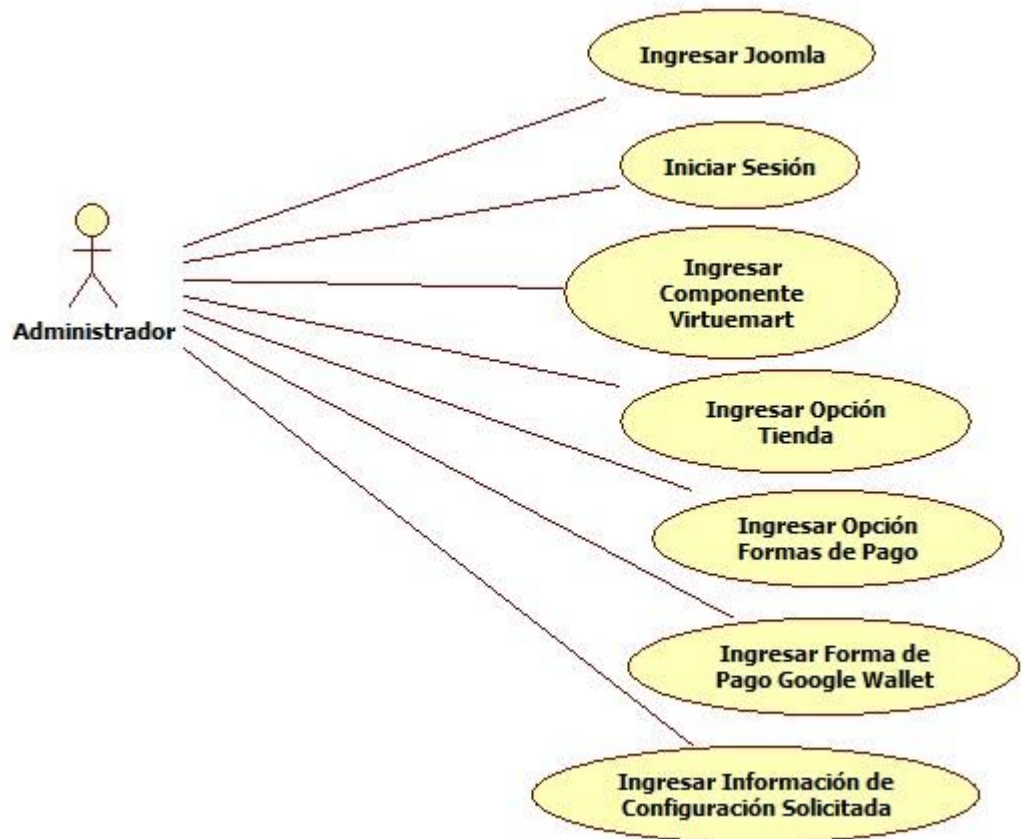


Figura 3.2 Caso de Uso, Configurar Plugin

La especificación del caso de uso, Configurar Plugin, se muestra en la tabla 3.2

Tabla 3.2 Diccionario Caso de Uso Configurar Plugin

| Detalle: | | | |
|--------------------------|--|--|----------|
| Nombre del CDU | Configurar Plugin | | |
| Estado: | Propuesto | Versión 1.0 | Fase 1.0 |
| Autor: | Santiago Pazmiño | | |
| Actor: | Administrador | | |
| Objetivo: | El administrador configura el Plugin de la forma de pago | | |
| Escenarios: | | | |
| Flujo Básico | Actor | Eventos | |
| | 1. Ingresar a página de administración Joomla | 2. Se muestra la página de administración | |
| | 3. Inicia sesión | 4. Se muestra el panel de administración | |
| | 5. Ingresar al componente Virtuemart | 6. Se muestra el componente Virtuemart | |
| | 7. Ingresar a la opción Tienda | 8. Se muestra el contenido de la opción Tienda | |
| | 9. Ingresar a la opción formas de pago | 10. Se muestran las formas de pago instaladas | |
| | 10. Ingresar forma de pago Google Wallet | 11. Se muestra panel de configuración de Google wallet | |
| | 12. Ingresar información de configuración requerida por la forma de pago | 13. Se guarda información ingresada | |
| Flujo Alternativo | 1. Ingresar mal usuario o clave | 2. Joomla no permite el ingreso | |
| | 3. Ingresar información de configuración incompleta o incorrecta | 4. Se muestra mensaje de error de configuración | |
| Pre Condiciones | El usuario debe tener permisos de administrador | | |
| Post Condiciones | El usuario a configurado correctamente el Plugin | | |

El caso de uso Publicación del Plugin se muestra en la figura 3.3

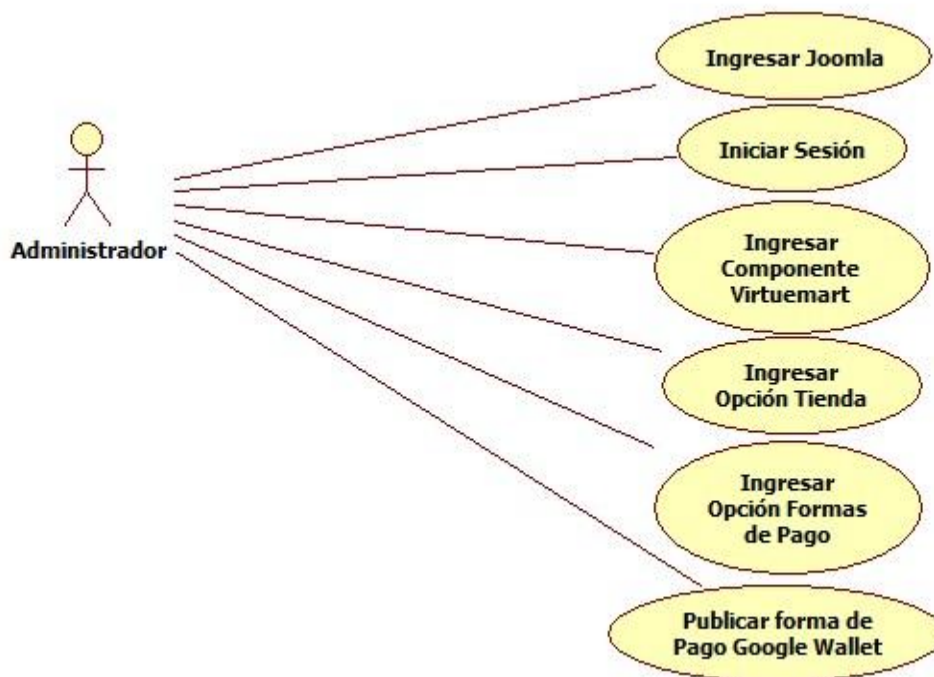


Figura 3.3 Caso de Uso, Publicar Plugin

La especificación del caso de uso, Publicar Plugin, se muestra en la tabla 3.3

Tabla 3.3 Diccionario Caso de Uso Publicar Plugin

| Detalle: | | | |
|--------------------------|--|---|----------|
| Nombre del CDU | Publicar Plugin | | |
| Estado: | Propuesto | Versión 1.0 | Fase 1.0 |
| Autor: | Santiago Pazmiño | | |
| Actor: | Administrador | | |
| Objetivo: | El administrador publica el Plugin de la forma de pago | | |
| Escenarios: | | | |
| Flujo Básico | Actor | Eventos | |
| | 1. Ingresa a página de administración Joomla | 2. Se muestra la página de administración | |
| | 3. Inicia sesión | 4. Se muestra el panel de administración | |
| | 5. Ingresa al componente Virtuemart | 6. Se muestra el componente Virtuemart | |
| | 7. Ingresa a la opción Tienda | 8. Se muestra el contenido de la opción Tienda | |
| | 9. Ingresa a la opción formas de pago | 10. Se muestran las formas de pago instaladas | |
| | 10. Selecciona forma de pago Google Wallet | 11. Se muestra opciones de selección | |
| | 12. Publica la forma de pago | 13. La forma de pago se encuentra disponible en la tienda | |
| Flujo Alternativo | 1. Ingresa mal usuario o clave | 2. Joomla no permite el ingreso | |
| | 3. No se selecciona forma de pago a publicar | 4. No se muestra opciones de selección | |
| Pre Condiciones | El usuario debe tener permisos de administrador | | |
| Post Condiciones | El usuario a publicado correctamente el Plugin | | |

El caso de uso Selección del Plugin se muestra en la figura 3.4

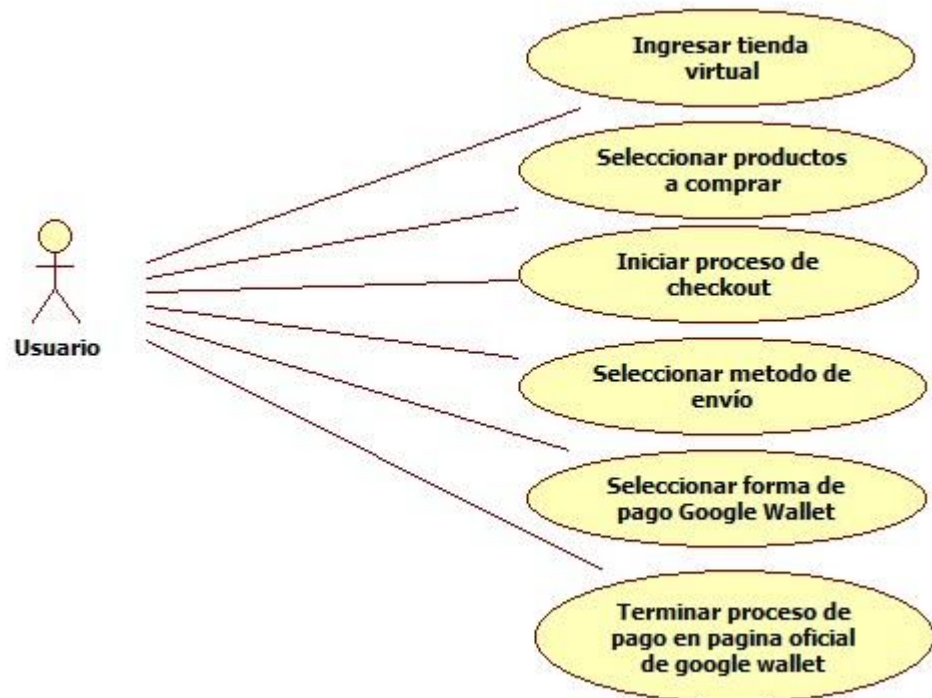


Figura 3.4 Caso de Uso, Seleccionar Plugin

La especificación del caso de uso, Seleccionar Plugin, se muestra en la tabla 3.4

Tabla 3.4 Diccionario Caso de Uso Seleccionar Plugin

| Detalle: | | | |
|--------------------------|--|--|----------|
| Nombre del CDU | Seleccionar Plugin | | |
| Estado: | Propuesto | Versión 1.0 | Fase 1.0 |
| Autor: | Santiago Pazmiño | | |
| Actor: | Usuario | | |
| Objetivo: | El usuario selecciona la forma de pago | | |
| Escenarios: | | | |
| Flujo Básico | Actor | Eventos | |
| | 1. Ingresa a página de tienda virtual | 2. Se muestra la página de la tienda virtual | |
| | 3. Inicia sesión | 4. Inicia sesión correctamente | |
| | 5. Selecciona productos a comprar | 6. Los productos se agregan al carrito de compras | |
| | 7. Inicia proceso de checkout | 8. Se muestra todos los productos contenidos en la cesta | |
| | 9. Selecciona método de envío | 10. Se agrega valor de método de envío a la cesta | |
| | 10. Selecciona forma de pago Google Wallet | 11. Se prepara la información a ser enviada a la forma de pago Google Wallet | |
| | 12. Redirección a la página de la forma de pago | 13. Se termina proceso de pago | |
| Flujo Alternativo | 1. Ingresa mal usuario o clave | 2. Virtuemart no permite el ingreso | |
| | 3. No se selecciona productos a comprar | 4. Cesta vacía no se inicia proceso de checkout | |
| Pre Condiciones | El usuario debe estar registrado en la tienda virtual | | |
| Post Condiciones | El usuario ha seleccionado y usado correctamente el plugin | | |

3.2.4 Especificación UIDs

A continuación el UID para el Caso de Uso Instalar Plugin, se muestra en la figura 3.5

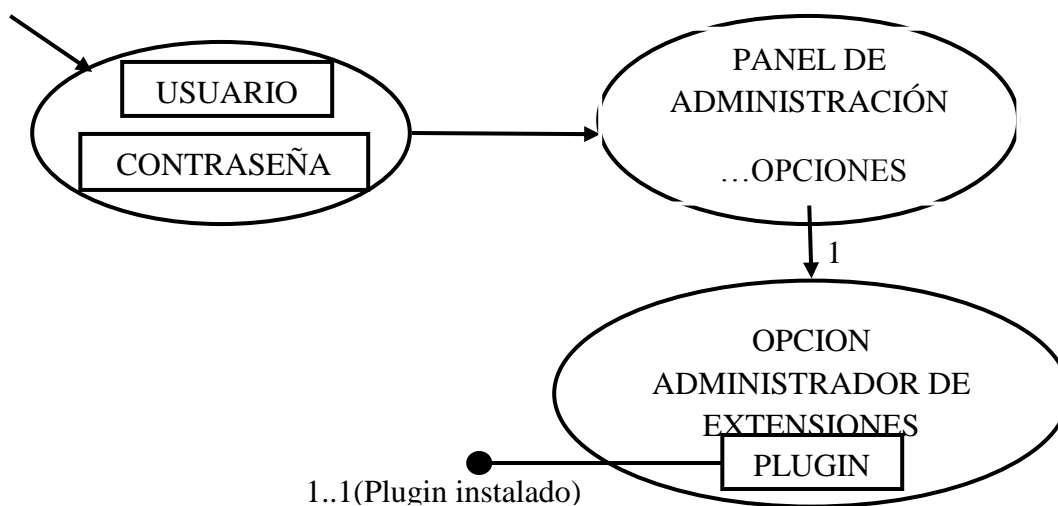


Figura 3.5 Caso de Uso Instalar Plugin: UID

En la figura 3.6 se muestra el UID para el Caso de Uso Configurar plugin

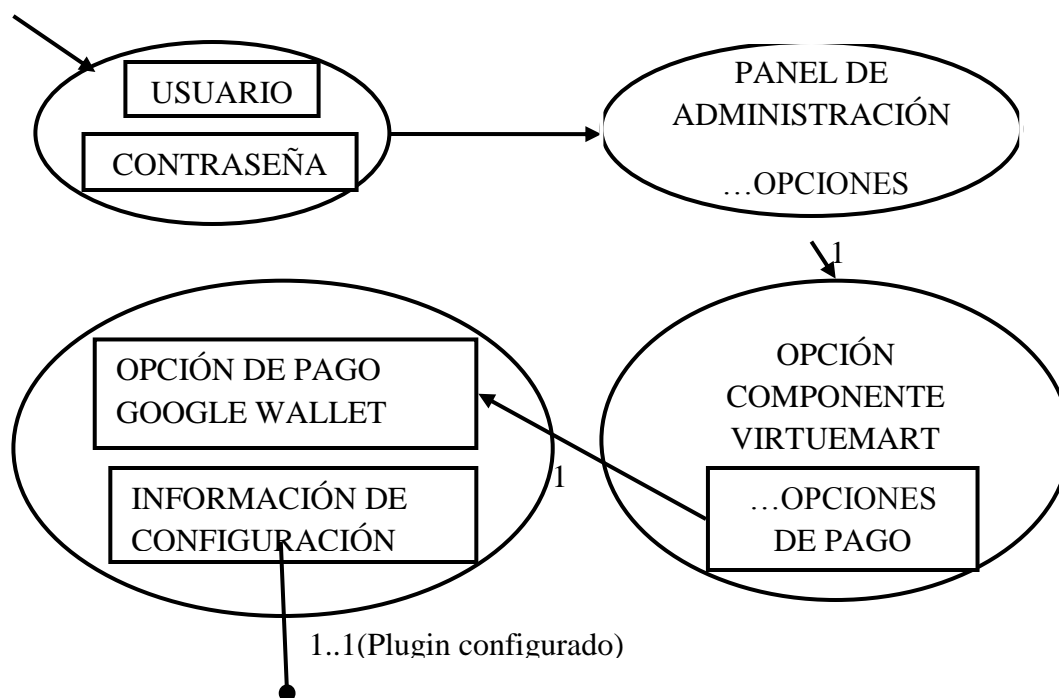


Figura 3.6 Caso de Uso Configurar Plugin: UID

En la figura 3.7 se muestra el UID para el Caso de Uso Publicar plugin

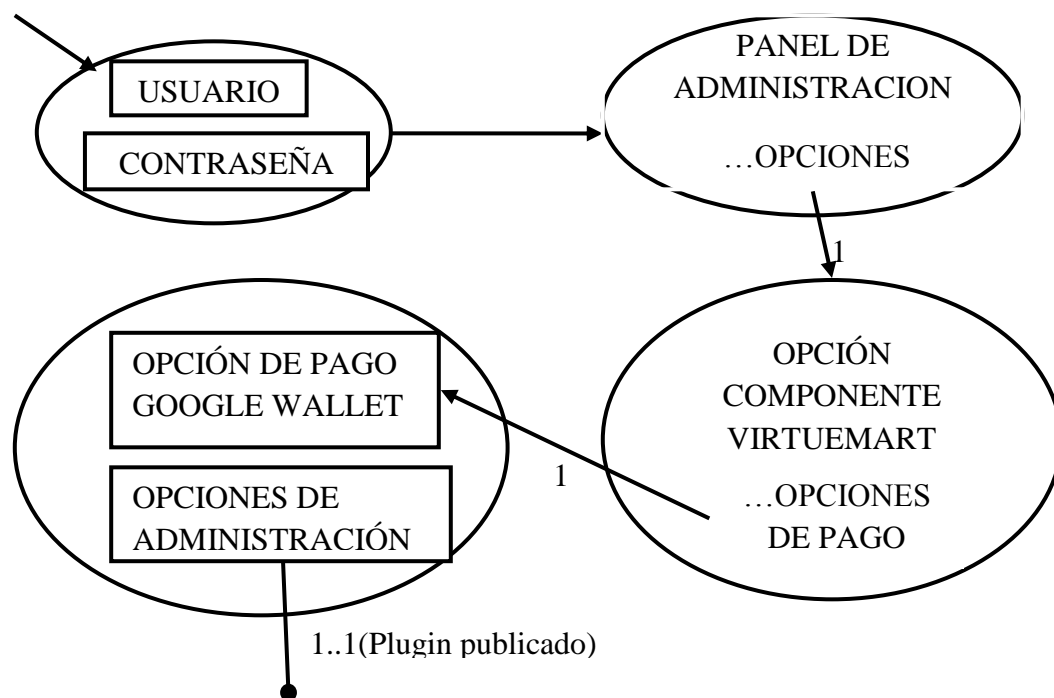


Figura 3.7 Caso de Uso Publicar Plugin: UID

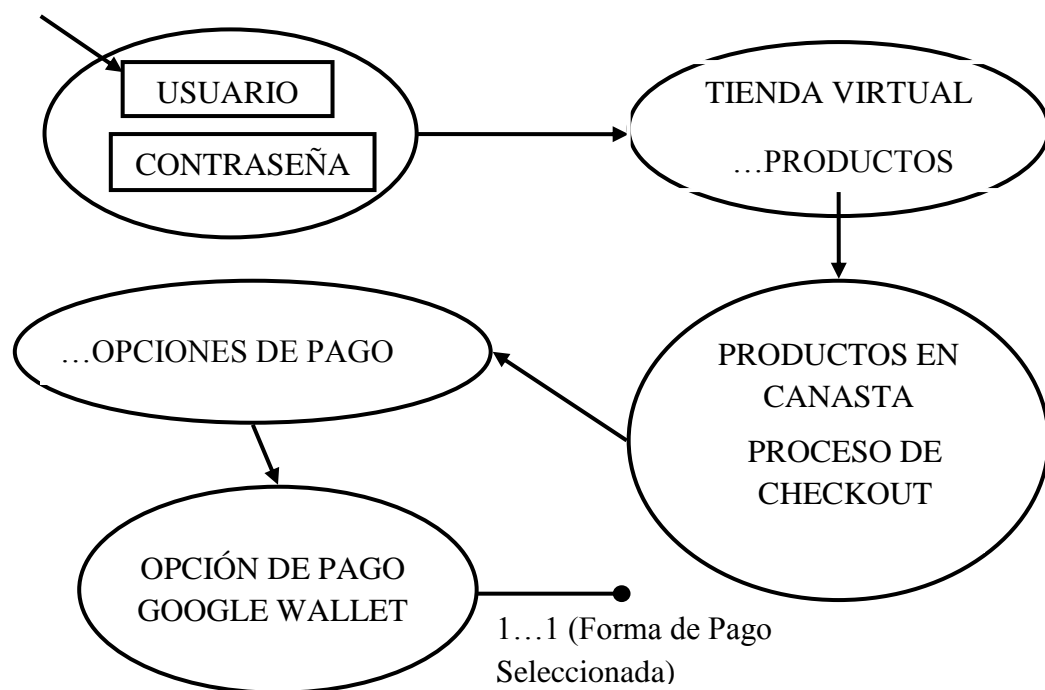


Figura 3.8 Caso de Uso Seleccionar Forma de Pago: UID

3.2.5 Diseño Conceptual

El diseño conceptual de la aplicación se basa en los UIDs especificados, pero además para el plugin a desarrollarse se basará también en las clases relacionadas con el plugin, dentro del sistema.

Para este caso tomaremos en cuenta las clases para plugins de pago en Virtuemart, de estas clases heredaremos funciones y atributos para nuestra clase de Google Wallet.

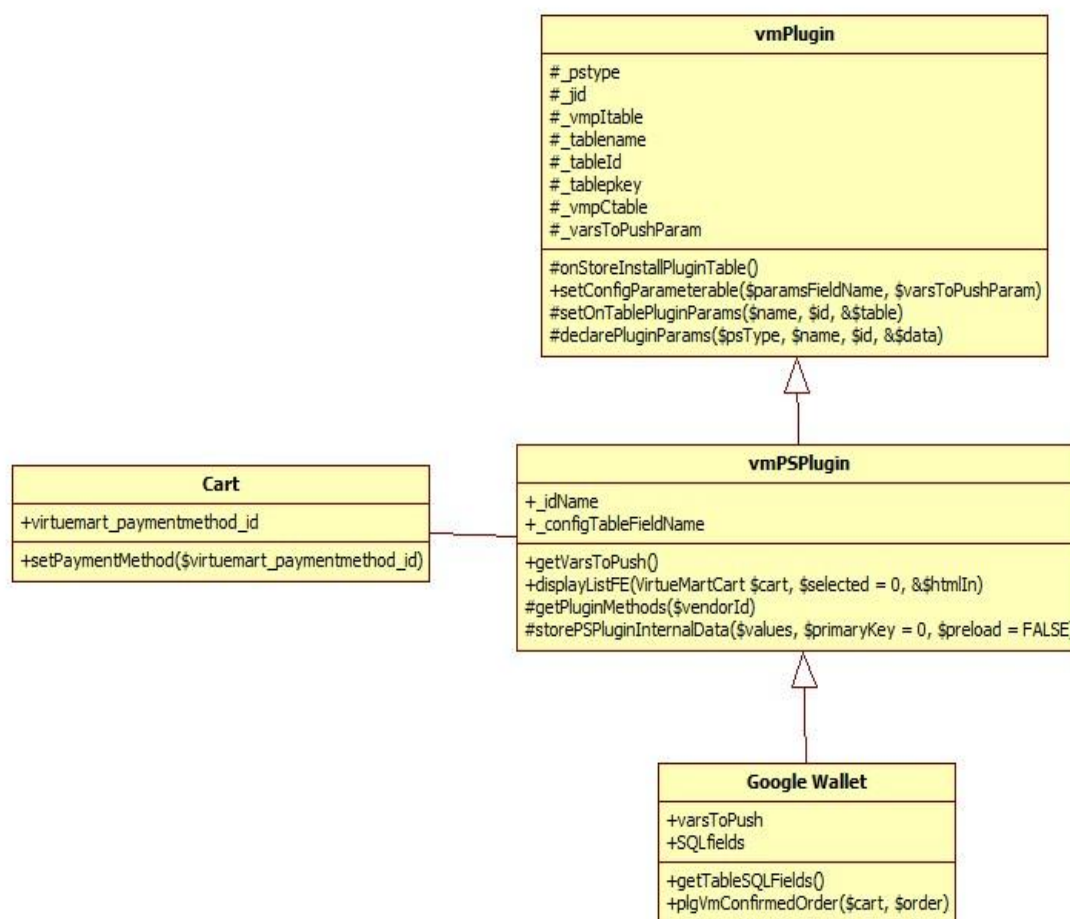


Figura 3.9 Diseño Conceptual

3.2.6 Diseño Navegacional

La aplicación del diseño navegacional se muestra en la figura 3.10

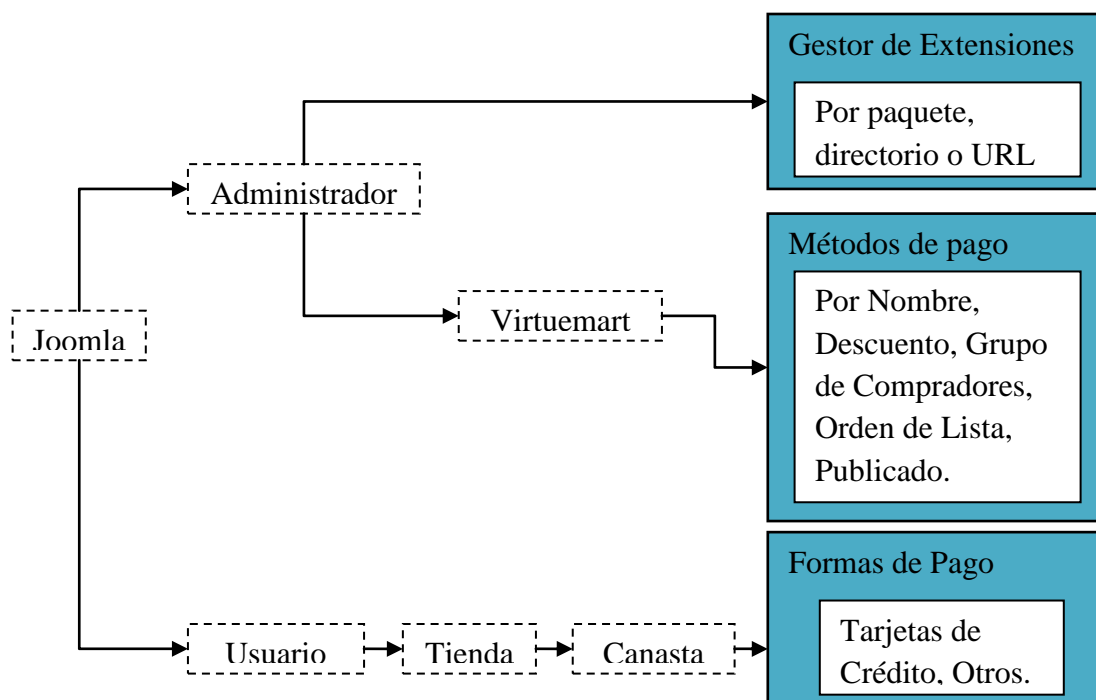


Figura 3.10 Diseño Navegacional

3.2.7 Diseño de Interfaces Abstracta

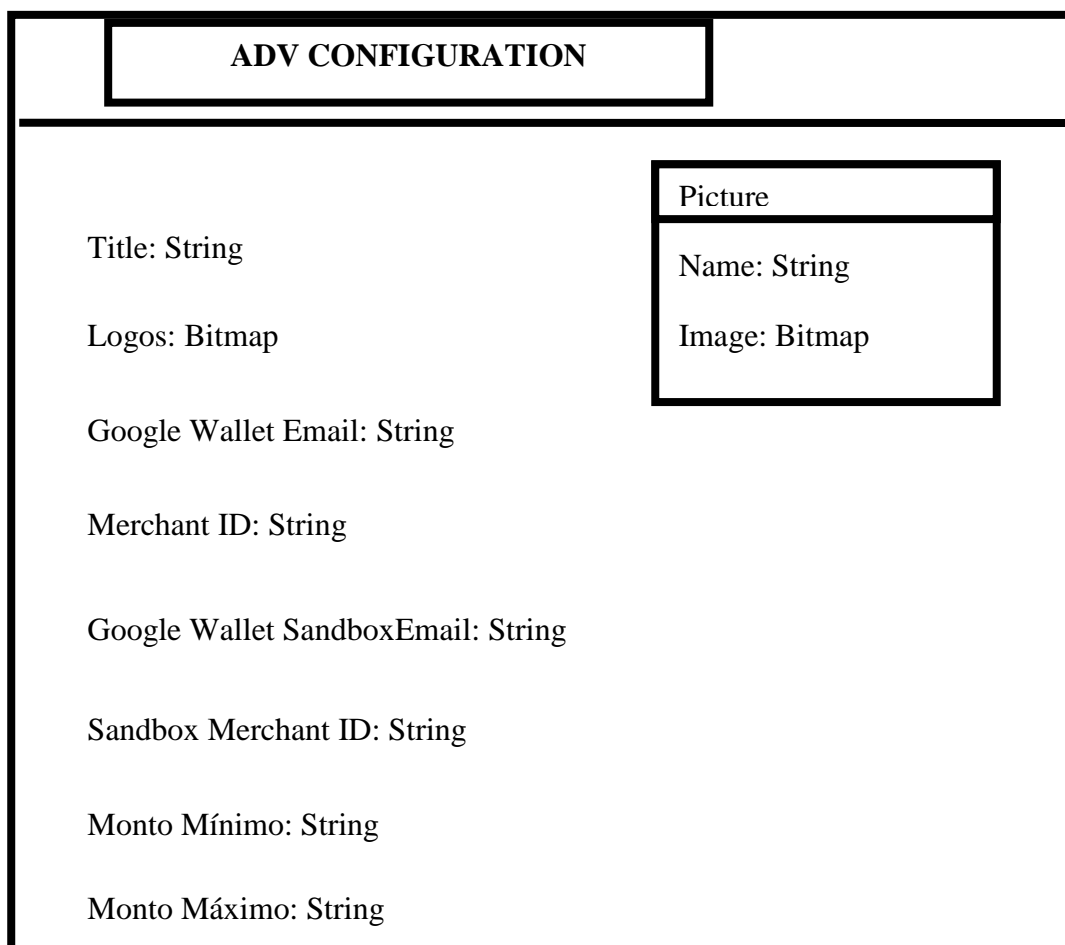


Figura 3.11 ADV Configuración Plugin

CAPÍTULO IV IMPLEMENTACIÓN Y PRUEBAS

4.1 Implementación

El plugin así como la solución e-commerce, se encuentran alojados actualmente en un ambiente de pruebas, dentro de un servidor que posee el CMS Joomla, dentro de una red LAN, dicho servidor cuenta con las herramientas necesarias para poder levantar tanto la solución e-commerce Virtuemart como el plugin que se encuentra desarrollado en php, tales como:

- Apache
- MySQL

Debido a que es un ambiente de pruebas se encuentra con el sistema Windows XP, dentro de un dominio privado. El manual de implementación se encuentra en el ANEXO A.

4.2 Pruebas

4.2.1 Plan de Pruebas

4.2.1.1 Introducción

Los sistemas informáticos al igual que cualquier producto desarrollado por el hombre no se encuentra exento de errores, por esta razón se hace necesario disponer de un adecuado plan para identificar y corregir los mismos.

4.2.1.2 Propósito

El presente plan de pruebas describe las pruebas, los actores y las herramientas que se usarán sobre el sistema.

4.2.1.3 Alcance

Todas las pruebas serán llevadas a cabo mediante técnicas y herramientas que se encontrarán bien definidas en los casos de prueba.

4.2.1.4 Definición de términos y acrónimos

- Google Wallet. Billetera Virtual.
- OOHDM. Metodología de Diseño Hipermedia Orientado a Objetos.
- Joomla. Sistema Gestor de Contenidos.
- Virtuemart. Solución e-commerce para uso conjunto con Joomla.
- Firebug. Plug-in del navegador Firefox, para monitoreo de actividades.

4.2.1.5 Visión General del Plan de Pruebas

En las secciones siguientes se describirán los recursos y herramientas a utilizarse en la aplicación de pruebas sobre el sistema desarrollado.

4.2.1.6 Ítems a Probar

4.2.1.6.1 Funciones

- ✓ Instalar Plugin Google Wallet
- ✓ Configurar Plugin Google Wallet
- ✓ Publicar Plugin Google Wallet
- ✓ Usar Forma de Pago Google Wallet

4.2.1.6.2 Módulos

- Virtuemart.
- Plugin de Pago Google Wallet.

4.2.1.6.3 Interfaces

- Interfaz del botón de pago Google Wallet.

4.2.1.7 Estrategia

Para evaluar el sistema se tendrá en cuenta los objetivos planteados de la tesis “Plugin de Google Wallet para pagos online”, en base a los requerimientos propuestos al inicio de la tesis se podrá verificar el producto.

4.2.1.8 Documentos Representativos

Para realizar el documento Plan de Pruebas se toma en consideración los siguientes documentos:

- ✓ Especificación de requerimientos
- ✓ Casos de uso

4.2.1.9 Recursos

Para realizar las pruebas se hará uso de un computador con las características siguientes:

- Procesador Intel Pentium I5.
- Memoria RAM 4GB.
- Navegador Mozilla Firefox 23 o superior, o Google Chrome versión 29 o superior.
- Sistema Operativo Windows 7.
- Conexión a internet.
- Servidor web.

4.2.1.10 Calendarización

A continuación se realiza la calendarización de las pruebas en la tabla

Tabla 4.1 Calendarización

| | Plan de Pruebas | Caso de Prueba 1 | Caso de Prueba 2 | Caso de Prueba 3 | Caso de Prueba 4 | Reporte de Errores |
|------------|-----------------|------------------|------------------|------------------|------------------|--------------------|
| 30/08/2013 | | | | | | |
| 05/09/2013 | | | | | | |

4.2.1.11 Responsables

A continuación, se muestra en la tabla los responsables de las pruebas

Tabla 4.2 Responsables Pruebas

| Responsable | Tareas o Actividades |
|------------------|--|
| Santiago Pazmiño | Probar el caso de prueba Instalar Plugin |
| | Probar el caso de prueba Configurar Plugin |
| | Probar el caso de prueba Publicar Plugin |
| | Probar el caso de prueba Usar Plugin |

4.2.2 Casos de Prueba

A continuación se muestra el caso de prueba CS-001 en la tabla

Tabla 4.3 Caso de Prueba CS-001

| Caso de Prueba [CS-001] | |
|--|---|
| Elaborado por | Santiago Pazmiño |
| Código de Identificación | PLU.INS.V01 |
| Nombre del Proyecto | Google Wallet |
| Descripción (Alcance y Objetivos) | Encontrar las posibles fallas cuando se instale el plugin |
| Requisitos asociados: | Saber la dirección de el panel de administración del CMS Joomla |
| Variables de Entrada (Inputs) | Es necesario seleccionar el plugin a instalar |
| Flujo normal del evento | <ol style="list-style-type: none"> 1. Ingreso al administrador de extensiones 2. Selecciono plugin a instalar 3. Presiono botón upload and install |
| Resultado esperado | “Plugin correctamente instalado” |
| Flujo alterno | En caso de tener archivo de instalación xml malformado |
| Resultado alternativo esperado | Aparece mensaje de error, archivo xml malformado |
| Evaluación de Prueba | |
| Fecha de Ejecución | 30/08/2013 |
| Ejecutado por | Ing. Fernando Galárraga |
| Lugar de ejecución | Universidad de las Fuerzas Armadas ESPE |
| Resultados Obtenidos | Satisfactorios, se produce el flujo normal y se instala el plugin |
| Observaciones | |
| Gravedad del Error | Ninguna |
| Notas del programador | |
| Estado | N/A |
| Acciones de corrección | N/A |
| Corregido por: | N/A |

A continuación se muestra el caso de prueba CS-002 en la tabla

Tabla 4.4 Caso de Prueba CS-002

| Caso de Prueba [CS-002] | |
|--|--|
| Elaborado por | Santiago Pazmiño |
| Código de Identificación | PLU.CONF.V01 |
| Nombre del Proyecto | Google Wallet |
| Descripción (Alcance y Objetivos) | Encontrar las posibles fallas cuando se configure el plugin |
| Requisitos asociados: | Saber acceder al panel de formas de pago de virtuemart |
| Variables de Entrada (Inputs) | Es necesario crear y configurar una nueva forma de pago, con el plugin Google Wallet |
| Flujo normal del evento | <ol style="list-style-type: none"> 1. Ingreso al administrador de virtuemart 2. Selecciono formas de pago 3. Creo forma de pago Google Wallet 4. Configuro forma de pago Google Wallet |
| Resultado esperado | “Plugin correctamente configurado” |
| Flujo alterno | En caso de configuración incorrecta |
| Resultado alternativo esperado | Aparece mensaje de error, forma de pago no configurada |
| Evaluación de Prueba | |
| Fecha de Ejecución | 30/08/2013 |
| Ejecutado por | Ing. Fernando Galárraga |
| Lugar de ejecución | Universidad de las Fuerzas Armadas ESPE |
| Resultados Obtenidos | Satisfactorios, se produce el flujo normal y se configura el plugin |
| Observaciones | |
| Gravedad del Error | Ninguna |
| Notas del programador | |
| Estado | N/A |
| Acciones de corrección | N/A |
| Corregido por: | N/A |

A continuación se muestra el caso de prueba CS-003 en la tabla 4.5

Tabla 4.5 Caso de Prueba CS-003

| Caso de Prueba [CS-003] | |
|--|--|
| Elaborado por | Santiago Pazmiño |
| Código de Identificación | PLU.PUB.V01 |
| Nombre del Proyecto | Google Wallet |
| Descripción (Alcance y Objetivos) | Encontrar las posibles fallas cuando se publique el plugin |
| Requisitos asociados: | Saber acceder al panel de formas de pago de virtuemart |
| Variables de Entrada (Inputs) | Es necesario seleccionar una forma de pago de Google Wallet |
| Flujo normal del evento | <ol style="list-style-type: none"> 1. Ingreso al administrador de virtuemart 2. Selecciono formas de pago 3. Selecciono forma de pago Google Wallet 4. Publico forma de pago Google Wallet |
| Resultado esperado | “Plugin correctamente publicado” |
| Flujo alterno | En caso de problemas al publicar |
| Resultado alternativo esperado | Aparece mensaje de error, forma de pago no publicada |
| Evaluación de Prueba | |
| Fecha de Ejecución | 05/09/2013 |
| Ejecutado por | Ing. Fernando Galárraga |
| Lugar de ejecución | Universidad de las Fuerzas Armadas ESPE |
| Resultados Obtenidos | Satisfactorios, se produce el flujo normal y se configura el plugin |
| Observaciones | |
| Gravedad del Error | Ninguna |
| Notas del programador | |
| Estado | N/A |
| Acciones de corrección | N/A |
| Corregido por: | N/A |

A continuación se muestra el caso de prueba CS-004 en la tabla 4.6

Tabla 4.6 Caso de Prueba CS-004

| Caso de Prueba [CS-004] | |
|--|---|
| Elaborado por | Santiago Pazmiño |
| Código de Identificación | PLU.USO.V01 |
| Nombre del Proyecto | Google Wallet |
| Descripción (Alcance y Objetivos) | Encontrar las posibles fallas cuando se use el plugin |
| Requisitos asociados: | Saber acceder a la tienda virtual |
| Variables de Entrada (Inputs) | Es necesario seleccionar al menos un producto y comenzar un proceso de checkout |
| Flujo normal del evento | <ol style="list-style-type: none"> 1. Ingreso a la tienda virtual 2. Selecciono productos a comprar 3. Inicio proceso de checkout 4. Selecciono forma de pago Google Wallet 5. Termino proceso de checkout 6. Pago en página de Google Wallet |
| Resultado esperado | “Forma de pago usada correctamente” |
| Flujo alterno | En caso de configuración incorrecta |
| Resultado alternativo esperado | Aparece mensaje de error, forma de pago no configurada |
| Evaluación de Prueba | |
| Fecha de Ejecución | 05/09/2013 |
| Ejecutado por | Ing. Fernando Galárraga |
| Lugar de ejecución | Universidad de las Fuerzas Armadas ESPE |
| Resultados Obtenidos | Satisfactorios, se produce el flujo normal y se usa la forma de pago |
| Observaciones | |
| Gravedad del Error | Ninguna |
| Notas del programador | |
| Estado | N/A |
| Acciones de corrección | N/A |
| Corregido por: | N/A |

4.2.3 Reporte de Errores e Inconsistencias

A continuación en la tabla se muestra el reporte de errores e inconsistencias.

Tabla 4.7 Reporte de Errores e Inconsistencias

| Reporte de Errores e Inconsistencias | | | |
|---|---------------------------------------|------------------------------|-------------------------------|
| Nombre del Proyecto | Plugin Google Wallet | | |
| Fecha de Pruebas | 05/09/2013 | | |
| Módulos | Plugin de pago | | |
| Analista | | | |
| Responsable | Santiago Pazmiño | | |
| Identificación de Caso de Prueba | Descripción de Prueba | Descripción del Error | Acciones de Corrección |
| CS-001 | Se ejecutaron las pruebas sin novedad | N/A | N/A |
| CS-002 | | | |
| CS-003 | | | |
| CS-004 | | | |

4.2.4 Control de Calidad

Se evaluará el plugin según el estándar ISO/IEC9126, donde se evalúa la calidad del software, sus características y sub características como se detalla a continuación.

Para este control se definen los siguientes actores:

- Evaluador: Santiago Pazmiño
- Evaluador de Calidad: Ing. Fernando Galárraga

4.2.4.1 Funcionalidad

4.2.4.1.1 Idoneidad

Se evaluó que las funciones desarrolladas son las adecuadas para cumplir las funciones especificadas en el documento de requerimientos

4.2.4.1.2 Exactitud

Se evaluó que la información generada y enviada a Google por el plugin con la información contenida en el carrito de compras es correcta.

4.2.4.1.3 Interoperabilidad

Se evaluó la correcta integración del plugin con el CMS Joomla, gracias a estar desarrollado siguiendo los lineamientos de creación de plugins de Joomla.

4.2.4.1.4 Seguridad

Se cuenta con una alta seguridad de acceso al plugin, esto gracias a la propia seguridad del CMS.

4.2.4.1.5 Conformidad de la funcionalidad

Se siguieron los estándares y lineamientos de desarrollo propuestos para el uso del HTML API de Google Checkout.

4.2.4.2 Usabilidad

4.2.4.2.1 Entendibilidad

El plugin al ser desarrollado siguiendo los lineamientos de Google y Joomla, es fácilmente entendible para usuarios con conocimientos del uso de plugins en el CMS, así como para usuarios con conocimientos de uso de Google Wallet.

4.2.4.2.2 Aprendibilidad

El uso del plugin es de fácil aprendizaje gracias al manual de usuario y de administrador, que se encuentra en los anexos del proyecto.

4.2.4.2.3 Operatibilidad

El plugin es fácil de instalar como una extensión más de Joomla, así como también es fácil de configurar la forma de pago dentro del administrador de la tienda virtual.

4.2.4.2.4 Conformidad de la usabilidad

El plugin sigue los lineamientos de Google para el uso del HTML API de Google Checkout.

4.2.4.3 Eficiencia

4.2.4.3.1 Comportamiento en el tiempo

El nivel de desempeño del plugin en tiempos de respuesta y preparación de la información, se ha medido bajo los parámetros de equipo especificados en la fase de pruebas, dando muy buenos resultados.

4.2.4.3.2 Utilización de recursos

El uso en memoria de la aplicación se encuentra ligado al uso de la misma por parte del servidor web, así como el uso del plugin en la transmisión de datos desde la tienda virtual a los servidores de Google.

4.2.4.4 Mantenibilidad

4.2.4.4.1 Analizabilidad

El plugin es fácilmente analizable por estar dividido en cuatro archivos esenciales, además de estar muy bien documentados las funciones internas y su uso.

4.2.4.4.2 Cambiabilidad

El plugin puede ser modificado de manera fácil ya que se entrega el código del mismo.

4.2.4.4.3 Estabilidad

El plugin mantendrá su estabilidad después de una posible modificación, solo si se mantienen los lineamientos de desarrollo de plugins de Joomla.

4.2.4.5 Portabilidad

4.2.4.5.1 Adaptabilidad

Al ser un plugin desarrollado en PHP, XML y HTML, es fácilmente adaptable a cualquier ambiente web.

4.2.4.5.2 Instalabilidad

El plugin es fácilmente instalable en cualquier servidor web, que cuente con el CMS Joomla 2.5 o superior y la solución e-commerce Virtuemart 2.0 o superior.

4.2.4.5.3 Coexistencia

El plugin podrá funcionar al mismo tiempo con otros plugins de pago que se encuentren operativos en el momento.

4.2.4.5.4 Remplazabilidad

El plugin podrá fácilmente reemplazar otras formas de pago existentes en la tienda virtual.

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se desarrolló e implementó un plugin de la billetera virtual Google Wallet para realizar pagos online, siguiendo los lineamientos de desarrollo de plugins propuestos por Joomla y Virtuemart, para una fácil integración del plugin con estos sistemas; se logró esto gracias al amplio soporte en línea existente para la integración de Google Wallet en cualquier sitio web a través de su HTML API, a la documentación existente sobre desarrollo de extensiones en Joomla y a los plugins existentes de formas de pago creados por los desarrolladores de Virtuemart.
- Se investigaron los lineamientos de desarrollo de plugins propuestos por los desarrolladores de Joomla y Virtuemart, ya que Virtuemart 2 adoptó Joomla framework para su estructura, el plugin así mismo está basado en este framework, por lo que su integración con el CMS es mas completa.
- Se desarrolló el plugin siguiendo los lineamientos especificados por Google para el uso del HTML API de Google Checkout, se eligió este API por su fácil integración a nivel de código con el lenguaje PHP usado por Joomla Framework para sus clases.
- Se implementó el plugin en una tienda virtual de prueba, misma que se encuentra alojada en un ambiente de pruebas; para una completa experiencia con la tienda virtual se adquirió un template profesional para el frontend de Virtuemart, además se instaló los datos de prueba del template.

- El desarrollo de plugins facilita la reutilización de código y al mismo tiempo genera nuevas funciones; para este proyecto el desarrollo del plugin para Virtuemart 2 supuso un cambio muy significativo con respecto a la forma en que se desarrollaban los plugin para la versión 1, esto por la migración de la solución e-commerce al uso de Joomla Framework para su desarrollo.

5.2 Recomendaciones

- Se recomienda el uso de software libre como el CMS Joomla, para el desarrollo de sitios web, por su gran comunidad de desarrolladores, abundante documentación en línea, e innumerables extensiones para agregar todo tipo de funciones al sitio desarrollado.
- Propender a un mayor desarrollo de aplicaciones orientadas a la web, esto por su fácil migración y gran capacidad de difusión mundial.
- Usar una metodología probada, como OOHDM, con enfoque al tipo de desarrollo de software que se desea para el proyecto, ya que brindan una mejor visión para desarrollos en la web.
- Al momento de desarrollar aplicaciones que funcionen dentro de otros sistemas, seguir los lineamientos de desarrollo propuestos por los mismos sistemas, esto para alcanzar una óptima interacción entre ambos.
- Difundir el uso de billeteras virtuales para las compras por internet, esto para mejorar la experiencia de compra y asegurar la información crediticia de los usuarios.

BIBLIOGRAFÍA

Marriot, J., & Waring, E. (2010). *The Official Joomla Book*. Nueva York: Addison Wesley.

Rossi, G., & Schwabe, D. (2001). *An Object Oriented Approach to Web-Based Application Design*. : Río de Janeiro: Dept. of Informatics, PUC-Rio.

Rossi, G., Schwabe, D., & Lyardet, F. (1999). *Patterns for Designing Navigable Information Spaces*. Río de Janeiro: Dept. of Informatics, PUC-Rio.

Schwabe, D., Moura, I., & Almeida, R. (2000). *OOHDM-Web: An Environment for Implementation of Hypermedia Applications in the WWW*. Río de Janeiro: Dept. of Informatics, PUC-Rio.

Schwabe, D., Vilain, P., & Sieckenius, C. (2002). *A Diagrammatic Tool for Representing User Interaction in UML*. Río de Janeiro: Dept. of Informatics, PUC-Rio.

Watson, K. (2012). *Virtuemart 2 User Manual*. : Chicago: ShowMe.

Trabajos citados

- [1] Google, «Acerca de Google Wallet,» [En línea]. Available: <http://support.google.com/wallet/bin/answer.py?hl=es&answer=105653>. [Último acceso: 24 Abril 2013].
- [2] Google, «Payment Methods,» [En línea]. Available: <https://support.google.com/checkout/sell/bin/answer.py?hl=es&answer=134420&topic=2450513&ctx=topic>. [Último acceso: 24 Abril 2013].
- [3] Google, «Fraud Protection Benefits,» [En línea]. Available: https://support.google.com/checkout/sell/topic/2445371?hl=en&ref_topic=2445409. [Último acceso: 15 Mayo 2013].
- [4] Google, «Settings,» [En línea]. Available: https://support.google.com/checkout/sell/topic/2450414?hl=en&ref_topic=2440279. [Último acceso: 15 Mayo 2013].
- [5] Google, «Merchant Transaction Fees,» [En línea]. Available: https://support.google.com/checkout/sell/answer/89800?hl=en&ref_topic=2445410. [Último acceso: 16 Mayo 2013].
- [6] Google, «Google Checkout Merchant Help,» [En línea]. Available: <https://support.google.com/checkout/sell/?hl=en#topic=2445411>. [Último acceso: 16 Mayo 2013].
- [7] Google, «Google Checkout API Overview,» [En línea]. Available: https://support.google.com/checkout/sell/answer/134461?hl=en&ref_topic=2450428. [Último acceso: 16 Mayo 2013].
- [8] Google, «Sandbox testing,» [En línea]. Available: https://support.google.com/checkout/sell/answer/134469?hl=en&ref_topic=2476197. [Último acceso: 16 Mayo 2013].
- [9] J. y. W. E. Marriot, de *THE OFFICIAL JOOMLA BOOK*, 2010, p. 4.
- [10] J. y. W. E. Marriot, de *THE OFFICIAL JOOMLA BOOK*, 2010, p. 10.
- [11] iStraxx, «What is Virtuemart,» [En línea]. Available:

- <http://virtuemart.net/features/what-is-virtuemart>. [Último acceso: 10 Mayo 2013].
- [12] K. Watson, de *SHOWME GUIDES VIRTUEMART 2 USER MANUAL*, 2012, p. 16.
- [13] K. Watson, de *SHOWME GUIDES VIRTUEMART 2 USER MANUAL*, 2012, p. 17.
- [14] T. P. Group, «Prefacio,» [En línea]. Available:
<http://www.php.net/manual/es/preface.php>. [Último acceso: 10 Mayo 2013].
- [15] W3C, «Extensible Markup Language (XML),» [En línea]. Available:
<http://www.w3.org/TR/2008/REC-xml-20081126/>. [Último acceso: 10 Mayo 2013].
- [16] G. P. Rossi, O. Schwabe y L. Olsina, *Web Engineering - Modelling and Implementing Web Applications*, Londres: Springer.
- [17] I. Object Management Group, «OMG UML,» [En línea]. Available:
http://www.omg.org/gettingstarted/what_is_uml.htm. [Último acceso: 12 Mayo 2013].