

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

Santiago Pazmiño Pinto¹, Ing. Fernando Galárraga², Ing. Germán Ñacato³

1 Universidad de las Fuerzas Armadas ESPE, Ecuador, santzero@hotmail.com

2 Universidad de las Fuerzas Armadas ESPE, Ecuador, jfgalarraga@espe.edu.ec

3 Universidad de las Fuerzas Armadas ESPE, Ecuador, gnacato@espe.edu.ec

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, OOHDM, Joomla, Virtuemart.

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, OOHDM, Joomla, Virtuemart.

1. INTRODUCCIÓN

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 ya que 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.

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

El presente proyecto busca integrar a través de un plugin, la billetera virtual Google Wallet y la solución e-commerce Virtuemart.

2. METODOLOGÍA

La metodología usada en el presente proyecto es OOHDM, la cual es complementada por el estándar IEEE 830 para la determinación de requerimientos.

OOHDM, Object Oriented Hypermedia Design Methodology, es una metodología para aplicaciones web e hipermedia que consiste en una serie de pasos susceptibles de repetición, similares a los de otras metodologías. Utiliza básicamente notación UML.

2.1 Fases OOHDM [1]

El desarrollo de una aplicación web basada en OOHDM consiste de las siguientes 5 fases:

- Determinación de Requerimientos.
- Diseño conceptual.
- Diseño navegacional.
- Diseño de interfaces abstractas.
- Implementación.

Las fases se muestran en la figura 1.

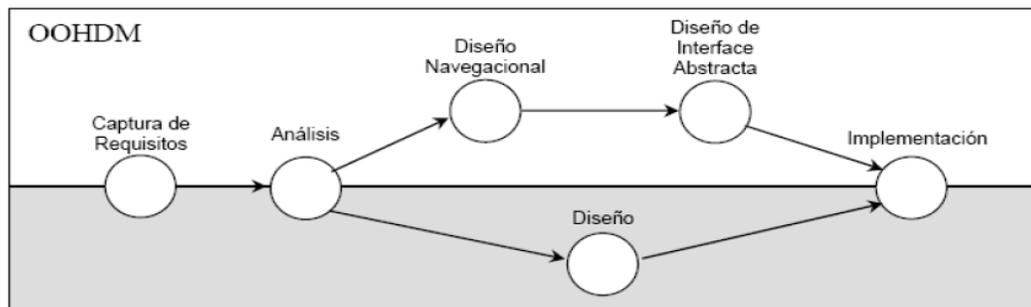


Figura 1: Fases de OOHDM

2.1.1 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.

2.1.2 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.

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.

2.1.3 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.

2.1.4 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.

2.1.5 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.

3. DESARROLLO DEL PLUGIN

3.1 IEEE 830 [2]

Se especifica los requerimientos basados en el estándar IEEE 830, en donde se delinearán el alcance del proyecto determinado como sigue:

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.

3.2 Google Wallet Checkout HTML API [3]

Permite intercambiar información con Google Wallet Checkout usando pares de nombre/valor en HTML forms. También se puede usar el HTML API para enviar pares de nombre=valor vía solicitud servidor a servidor HTTP POST. Órdenes en implementaciones HTML no pueden ser firmadas digitalmente, de manera que las integraciones con el HTML API que no emitan solicitudes servidor a servidor deberán revisar las órdenes manualmente. A continuación una muestra del HTML API.

```
<form method="POST"
  action="https://checkout.google.com/api/checkout/v2/checkoutForm/Merchant/1234567890"
  accept-charset="utf-8">
  <input type="hidden" name="item_name_1" value="Peanut Butter"/>
  <input type="hidden" name="item_description_1" value="Chunky peanut butter."/>
  <input type="hidden" name="item_quantity_1" value="1"/>
  <input type="hidden" name="item_price_1" value="3.99"/>
  <input type="hidden" name="item_currency_1" value="USD"/>
  <input type="image" name="Google Checkout" alt="Fast checkout through Google"
  src="http://checkout.google.com/buttons/checkout.gif?merchant_id=1234567890&w=180&h=46&style=white
  &variant=text&loc=en_US"
  height="46" width="180"/>
</form>
```

3.3 Lineamientos Desarrollo Plugins Joomla [4]

Para el desarrollo de cualquier plugin en Joomla, es necesario contar con dos archivos esenciales, un archivo XML que contendrá la información de instalación y configuración del plugin, y un archivo PHP que contendrá las funciones que se espera realice el plugin.

4. ENTREGABLES DEL PROYECTO

El software desarrollado, así como la documentación, están sujetos a cambios, incluso después de finalizar el proyecto. El uso de la metodología OOHDM implica un proceso iterativo del desarrollo de software. Los documentos generados en el desarrollo se listan a continuación.

4.1 Modelo de Casos de Uso

Se especificaron los diagramas de casos de uso basados en las actividades que generarán los actores externos, el administrador y el cliente, a continuación en la Figura 2 se muestra el C.U. Instalar Plugin.

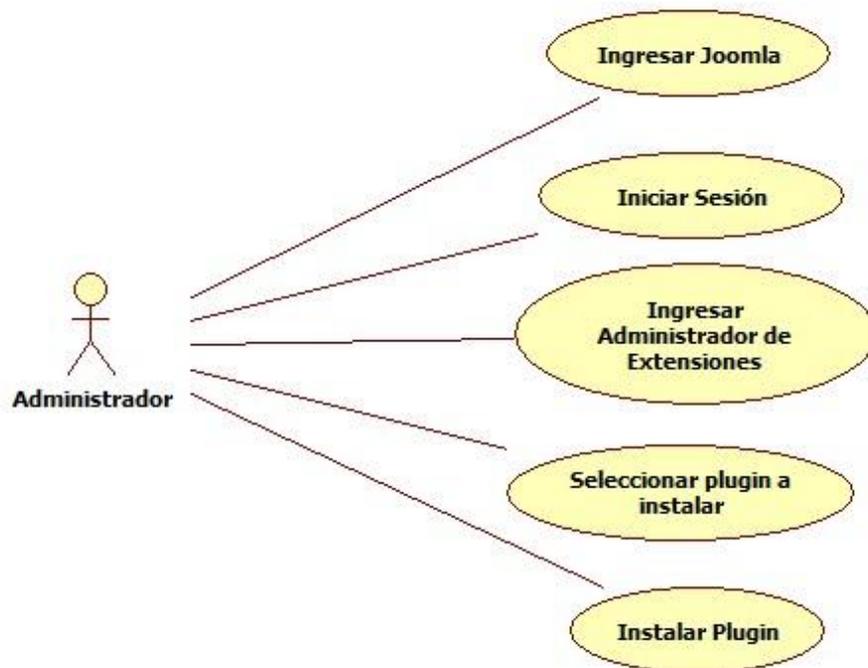


Figura 2: Caso de Uso Instalar Plugin

4.2 Especificación de Casos de Uso

Gracias a la creación de cada modelo se especificó mediante una plantilla el diccionario de los casos de uso. A continuación en la Tabla 1 se muestra la especificación del C.U. Instalar Plugin.

Tabla 1: Especificación C.U. 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. Ingres 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. Ingres a 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		

4.3 Especificación UIDs

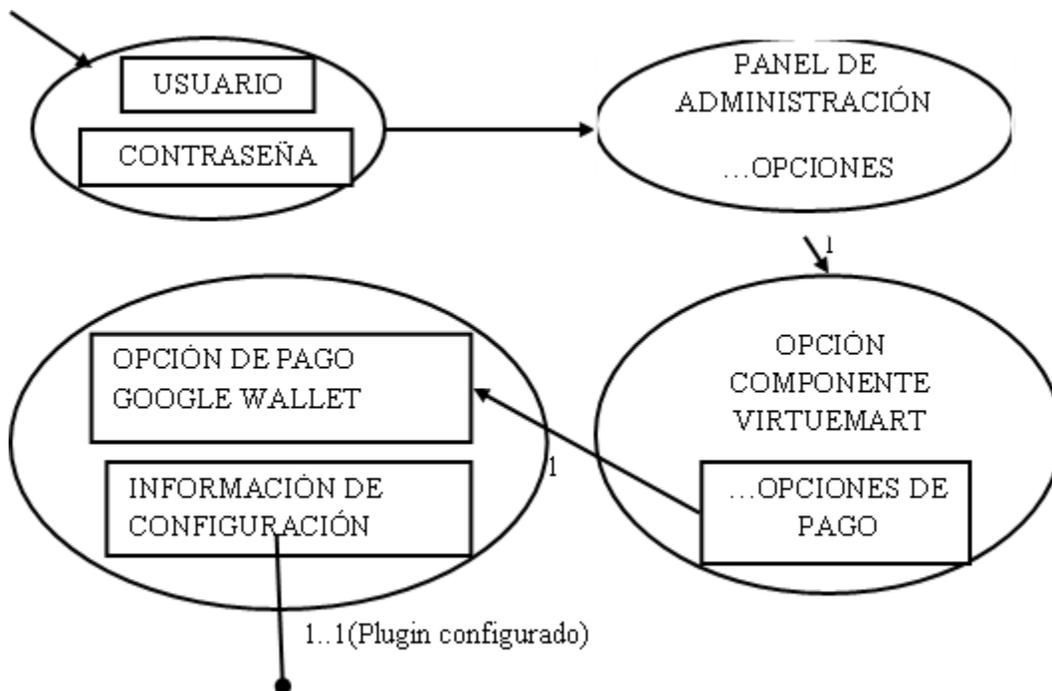


Figura 3: UID Configuración Plugin

4.4 Aplicación del Diseño Conceptual

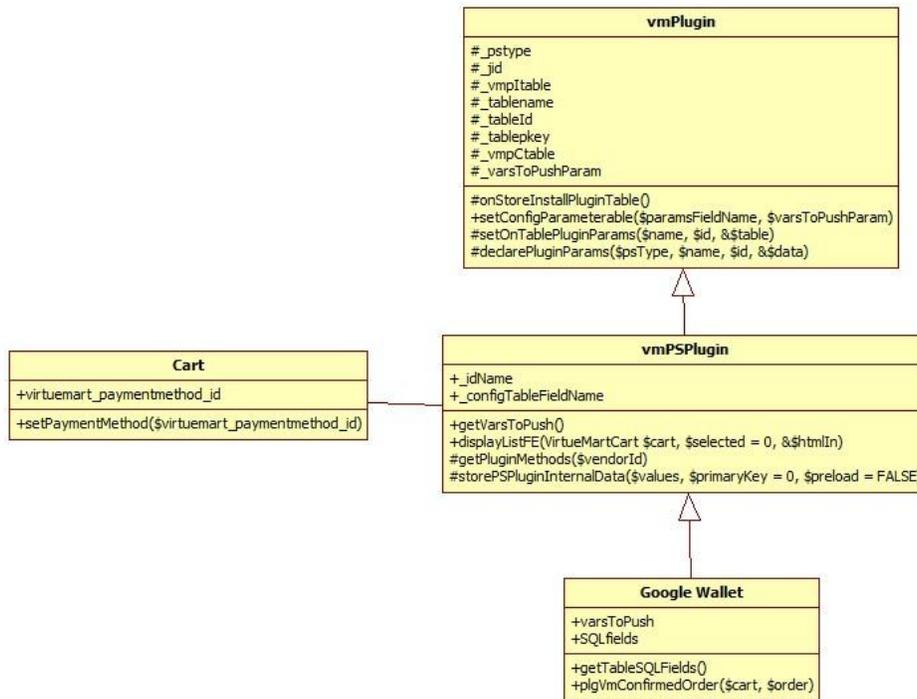


Figura 4: Diseño Conceptual

4.5 Aplicación del Diseño Navegacional

Al realizar una correcta integración del plugin con el CMS se espera que el plugin se desarrolle dentro del siguiente contexto navegacional que se muestra en la figura 5.

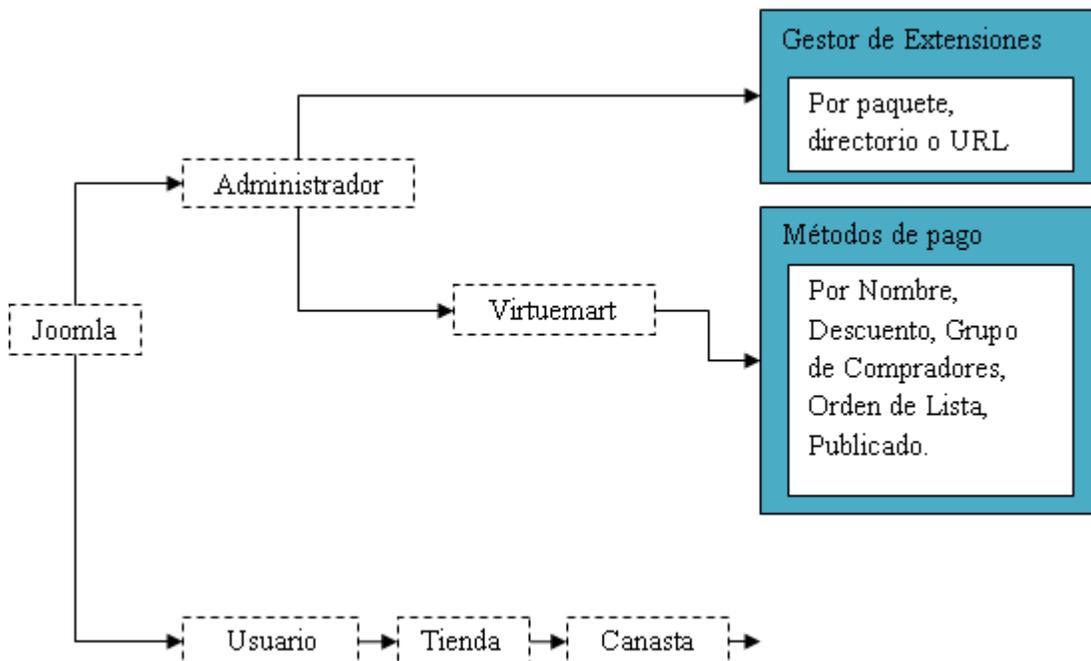


Figura 5: Contexto Navegacional

4.6 Aplicación del Diseño de Interfaz Abstracta

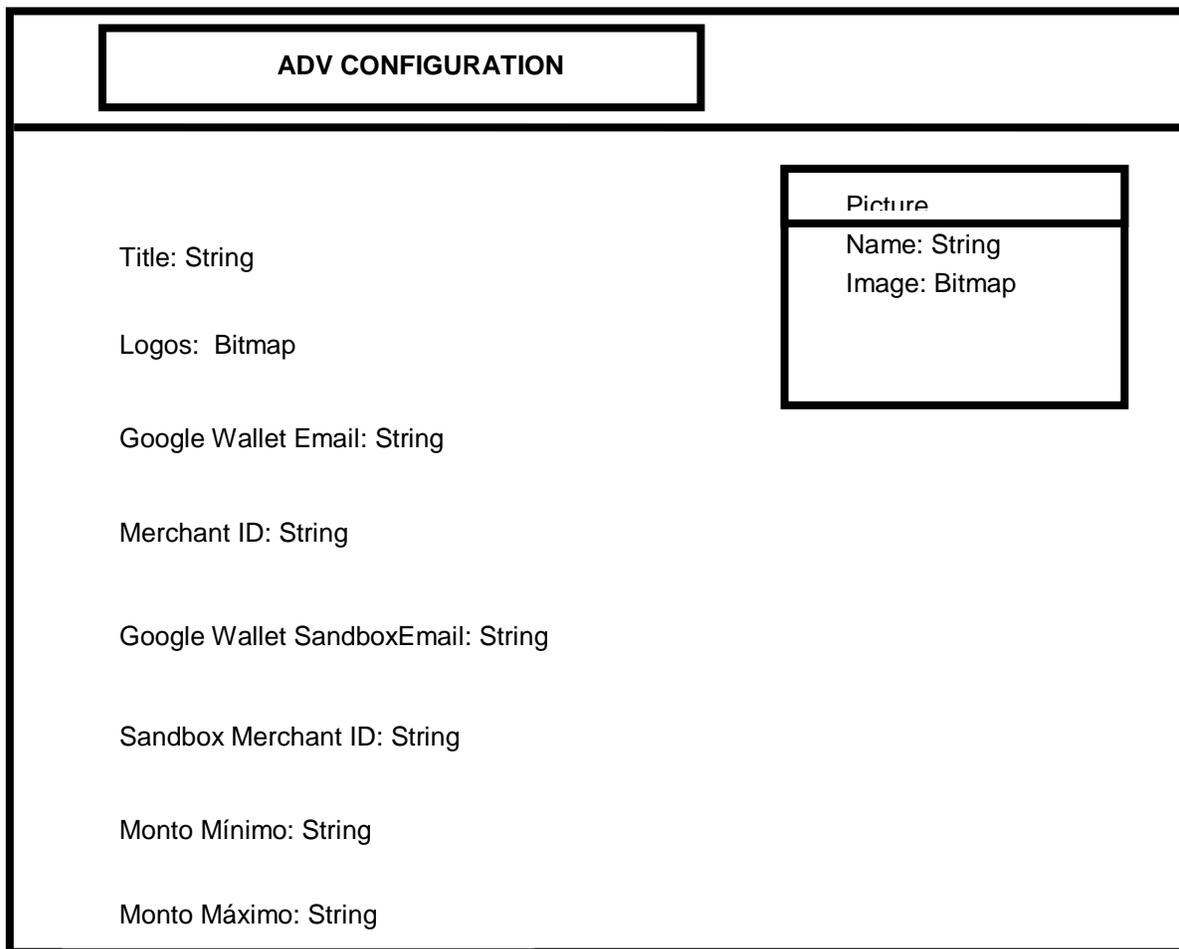


Figura 6: ADV Configuración

5. RESULTADOS

El plugin, junto con la tienda virtual de prueba, se encuentran alojados en un servidor de pruebas. Además se entregará en un CD el código fuente del plugin y los archivos que se requieren para levantar la tienda virtual de prueba.

5.1 Pruebas

Las pruebas del plugin desarrollado constan de tres partes:

1. El plan de pruebas
2. Los casos de pruebas
3. El reporte de errores

A continuación, en la Tabla 2, se muestra el caso de prueba Instalar Plugin (CS-001).

Tabla 2: 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 alternativo	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, en la figura 7, se muestra una captura de pantalla del caso de prueba Instalar Plugin (CS-001).



Figura 7: Caso de Prueba CS-001

6. TRABAJOS RELACIONADOS

Terán, Análisis, diseño e implementación de un portal web para el pueblo de Uyumbicho, <http://repositorio.espe.edu.ec/bitstream/21000/4699/1/T-ESPE-032841.pdf>

Boada, Guerra, Análisis, diseño e implementación del portal web de la dirección de sistemas de información y comunicaciones de la Fuerza Terrestre, <http://repositorio.espe.edu.ec/bitstream/21000/644/1/T-ESPE-021810.pdf>

Llerena, Vivero, Desarrollo de una plataforma informática para el soporte de una comunidad virtual Universitaria, <http://repositorio.espe.edu.ec/bitstream/21000/305/1/T-ESPE-027462.pdf>

Los trabajos presentados se relacionan con el aquí descrito por el uso del CMS Joomla para generar contenidos y la metodología OOHDM para el desarrollo, el agregado de este trabajo es que, no solo se uso el CMS para generar un contenido, si no que se agregaron funciones al mismo a través del plugin desarrollado.

7. 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 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.

8. AGRADECIMIENTOS

A los Ingenieros Fernando Galárraga y Germán Nacato por su colaboración en este proceso de investigación.

9. REFERENCIAS BIBLIOGRÁFICAS

- [1] G. P. Rossi, O. Schwabe y L. Olsina, Web Engineering - Modelling and Implementing Web Applications, Londres: Springer.
- [2] IEEE, Estándar IEEE 830, 1998.
- [3] Google, «Google Checkout API,» [En línea]. Available: https://support.google.com/checkout/sell/answer/134461?hl=en&ref_topic=2450428. [Último acceso: 13 Septiembre 2013].
- [4] Joomla. [En línea]. Available: http://docs.joomla.org/J2.5:Creating_a_Plugin_for_Joomla. [Último acceso: 13 Septiembre 2013].