



**Universidad de las Ciencias Informáticas**

**Facultad 6**

**Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.**

**Título:** Desarrollo de un módulo de pago para el acceso a los materiales audiovisuales publicados en la Plataforma VideoWeb 2.0.

**Autor:**

Yissel Herrera Santana

**Tutor:**

Ing. Sisley Sosa Vazquez.

La Habana, junio de 2013

“Año 55 de la Revolución”

*"Mi confianza en el triunfo final de lo que creo, es completa"*

*Ernesto Che Guevara*

## DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Firma del Autor

Yissel Herrera Santana

\_\_\_\_\_  
Firma del Tutor

Ing. Sisley Sosa Vázquez.

### DATOS DE CONTACTO

Tutor: Ing. Sisley Sosa Vázquez

Centro de Desarrollo Geoinformática y Señales Digitales (GEYSED)

Universidad de las Ciencias Informáticas (UCI)

Email: [ssosa@uci.cu](mailto:ssosa@uci.cu)

*Esta tesis va dedicada a las 3 personas más importantes para mí:*

*A mi abuelita Olga, mami por estar siempre que la he necesitado, en cada momento y etapa de mi vida, por darme consejos y aunque a veces no los siga, siempre los escucho porque yo no sé cómo ni porque pero no tiene el don de que no se le escapa una, porque para mí es la mejor abuela del mundo y a pesar de no ser su nieta preferida, nunca ha marcado ninguna diferencia ni en el cariño ni el trato, yo sé que ella me quiere tanto como yo a ella, siempre me lo ha demostrado y a ella le doy las gracias por todo lo que ha hecho por mí.*

*A la más especial de todas, mi mamá, a ti titi, va dedicada especialmente esta tesis porque en primera de no ser por ti no existiría, de no ser por la educación que siempre me has dado, no sería quien soy hoy, de no ser por ti, no estaría aquí hoy, a ti quiero darte las gracias por muchas cosas, son tantas que no podría ponerlas todas aquí y hasta te aburrirías de escucharme decírtelas, pero hay algunas que no pueden pasar por alto:*

*Zuiero darte las gracias por ser todo el tiempo mamá, papá y más que eso, mi mejor amiga, gracias por compartir la mayoría de tus secretos conmigo y darme la confianza de yo compartir los míos contigo, gracias por ayudarme a comprender la vida y guiarme por un buen camino, por apoyarme en mis decisiones, por comprenderlas, por ser mi ejemplo a seguir ya que siempre he visto que a pesar de que la vida te pone en momentos muy difíciles, no te derrumbas, sigues adelante y créeme, yo he intentado hacer lo mismo. Zuiero que sepas que aunque los años van pasando nunca voy a olvidar las cosas que has hecho por mí, como me peinabas y maquillabas de pequeña, como me ayudabas a montar las coreografías en la secundaria, como además nunca voy a olvidar que fuiste la madre que más se emocionó cuando vio que su hija cogió la UCA, nunca olvidare el grito que diste en el IPI. En fin gracias por existir y estar siempre ahí para mí.*

*A quien ha sido y será siempre para mí como otra madre, por preocuparse siempre por mí, por haberme dado mucho amor, porque sé que a pesar de no estar presente físicamente, está viéndome donde quiera que este, porque ella siempre quiso que llegara hasta donde hoy llegue, porque sé que como siempre lo estubo, hoy está orgullosa de mí, a mi tía Nana, a ti, porque quiero que sepas que no te voy a olvidar nunca que siempre vas a estar en mi corazón.*

*Le agradezco al tribunal Yordani, Cesar y Eridniel por sus criticas y sugerencias, y a pesar de todo especialmente a Suarez, porque desde que nos conocemos ha hecho mucho por mí, me ha ayudado y apoyado y tengo muchas cosas por las cuales agradecerle.*

*A mi tutor Sisley que mejor no lo podría tener y lo mismo puedo decir de mi oponente Yoendry.*

*A todas mis amistades que de una forma u otra han estado ahí en estos 5 años de Universidad, tanto para el estudio de la carrera, como para algún problema personal mío, para ir a fiestas, o simplemente por brindarme su amistad, como son:*

*A mis bichitos Edgar, Marchi y Falcón porque con ellos la carrera fue mucho más fácil ya que me obligaban a estudiar para las pruebas y con ellos siempre pude contar para lo que necesitara. A Luisma por alagar tanto mis comidas y en general a todos los de mi grupo. A las muchachitas del apto Yaima, Ela, Darci y sobre todo a Big mama 2, Yeilen por enseñarme tantas cosas de parejas. A Paul por ayudarme a darme cuenta de cómo es en realidad la vida y por enseñarme a bailar kisomba, a Leiser por soportarme en tantas ocasiones, a mis grandes amigos Javier y Camilo, Leo, Yacer, Julio. A David Tavares por acordarse siempre de mí aun estando lejos. A Yuni (Albrecht) por haberme dado su amistad. A mi gran amigo Rojas por enseñarme como es realmente la vida en la UCI. Mi gente de Camagüey, del IPI, Lisandra, Natali, Darian, kire, Alexander, Eider, Yaismel. A Héctor René por hacer que las clases de P3 fueran las mejores de todas las asignaturas. A Roci por hacer realidad uno de mis sueños, bailar tango, a los demás del grupo Eloy, Rene, Leo, Robin, Katia y Edu por continuar con la amistad que creamos ahí. Dayan por ser tan especial a la hora de hablar. A Tomas Dueñas por ayudarme de cierta forma con la tesis y por su linda amistad. A Dany, Karel, Aroldo, Damian. A Sandor, Raidel y Pedro por los maravillosos espaguetis que hemos hecho en casa de Sandor. A Isbel claro, por estar en cada fiesta y hacernos estar a mí y a Jessie. A mis ex suegros Puch y Ale por compartir tanto tiempo conmigo. A Yudiel más conocido como la figura, por ser un buen amigo y ayudarme en varias ocasiones. A Yulier por su amistad. A mis padres de la UCI, sobre todo a mi mamá Oricel por su cariño. A mi amiga querida Yeni por cumplir muy bien con su papel de amiga. A Jeem, Alleyne y Osmel por tantos ranchones y fiestas juntos y por apoyarme en todo momento, los quiero mucho.*

*A mis tíos Nadia y a Pimpi por portarse tan bien conmigo y ayudarme en muchas cosas. A mis tíos Yao y Guille por acogerme tan bien desde que nos conocimos, sobre todo en su casa, durante el tiempo que he pasado en la UCI.*

*A Tomas Dayron, titi, gracias por haberme dado la oportunidad de conocerte, por haberme soportado tanto tiempo, por haber sido tan comprensivo, por ayudarme en tantas cosas, a superar tantas cosas juntas que pasaban en mi vida, por cada detalle que tuviste conmigo, nunca voy a olvidar nada de eso. A Mercedes por cada comida los domingos, por los viajes, por su preocupación sobre mis cosas. A Roke por haberme hecho reír tanto con él.*

*A Arlhey, por ser mi más grande amigo en la UCI, mi pañuelo de lágrimas, mi consejero, a él gracias por estar conmigo sobre todo en momentos muy difíciles.*

*A mi esposa Jessie, que a pesar de ser un poco testaruda, ha sido una muy grande amiga desde que la conocí, mi mejor compañera de fiestas, a ella quiero darle las gracias por escucharme decir todo lo que he querido, bueno o malo, por aconsejarme sobre esas cosas, por estar para mí y cuidarme en todo momento, la quiero mucho.*

*A mi novio Tito Robert, por reaparecer en mi vida de la forma en que lo ha hecho, por ser el rayo de luz que me ha ayudado a salir de la oscuridad, por darme tanto cariño, por hacer que cada segundo que pase a su lado valga la pena, te quiero mucho. A mis suegros por recibirme con tanto cariño a pesar de las locas decisiones que hemos tomado tito y yo.*

*A toda mi familia de Camagüey, especialmente a mis hermanos Pedri y Alfre y a mi querido primo Yaicel.*

*A mi papá Pedro que a pesar de todas las cosas que hemos pasado lo quiero muchísimo y sé que está orgulloso de mí.*

*A mi papá Luis Alberto le doy las gracias por recibirme de la forma en que lo hizo desde que nos conocimos y quiero que sepa que se ganó desde el primer momento un lugar en mi corazón. Agradecerle a Mari, Yuliet y Sol por cumplir tan bien el papel de familia que les toco, por ser tan buenos conmigo y espero que sigan siéndolo.*

### RESUMEN

La actividad comercial que se realiza a través de las redes digitales sin la presencia de mediadores se denomina comercio electrónico. La posibilidad de realizar transacciones monetarias de manera automática, rápida y segura es una de las principales ventajas de este proceso comercial. Los sistemas que brinden productos o servicios por Internet y no cuenten con un mecanismo de pago electrónico se encuentran en clara desventaja, reduciendo drásticamente sus posibilidades de competencia. Tal es el caso de la plataforma VideoWeb que necesita un sistema automatizado de pago para ampliar sus posibilidades comerciales. El presente trabajo de diploma contiene la investigación y posibles resultados para obtener un módulo de pasarela de pago que permita automatizar el proceso de cobro por los servicios de reproducción y descarga en la plataforma VideoWeb 2.0 del centro de desarrollo Geoinformática y Señales Digitales. Con el fin de obtener los mejores resultados se trabaja bajo la guía de la metodología RUP, la aplicación a implementar es un módulo para el *Content Management System* (CMS) Drupal 7.x haciendo uso del lenguaje de programación PHP y el entorno de desarrollo integrado NetBeans. Se presentan los requisitos identificados y se presentará el cumplimiento de los mismos en la aplicación final, que será validada además por las pruebas a realizar, propuestas por la metodología de desarrollo utilizada. El impacto social de los resultados de esta investigación radicará en que el proyecto VideoWeb 2.0, podrá insertar su plataforma en entornos comerciales relacionados a los negocios electrónicos.

**Palabras claves:** comercio, pasarela, pago, VideoWeb.



## ÍNDICE GENERAL

INTRODUCCIÓN.....	1
CAPÍTULO 1: Introducción al estudio del Módulo de Pasarela de Pago. Fundamentación teórica.....	6
1.1 Introducción .....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.2.1 Comercio electrónico .....	6
1.2.2 Sistema de pago electrónico .....	6
1.2.3 Tarjeta de crédito.....	7
1.2.4 Tarjeta de débito.....	7
1.2.5 Monedero electrónico.....	7
1.3 Objeto de estudio .....	8
1.3.1 Descripción general.....	8
1.3.2 Proceso de compra en la web usando pasarela de pago .....	8
1.3.2.1 Proceso de compra en la Plataforma VideoWeb1.0:.....	9
1.3.3 Ventajas de las pasarelas de pago .....	10
1.4 Descripción de la situación problemática .....	10
1.5 Soluciones existentes.....	11
1.5.1 XtraLife.es .....	12

1.5.2	Tematika.com.....	12
1.6	Tecnologías para el desarrollo de la aplicación.....	14
1.6.1	Metodología de Desarrollo de Software .....	14
1.6.1.1	Metodología RUP.....	14
1.6.2	Lenguaje de modelado .....	15
1.6.3	Lenguajes de programación .....	16
1.6.3.1	PHP.....	16
1.6.3.2	JavaScript.....	17
1.6.4	Herramientas.....	17
1.6.4.1	NetBeans IDE 7.2.....	17
1.6.4.2	Herramienta CASE .....	18
1.6.4.3	Drupal 7.0.....	18
1.6.4.4	PostgreSQL 9.1 .....	19
1.6.4.5	PgAdmin III 1.10.0.....	19
1.7	Conclusiones.....	19
CAPÍTULO2: Análisis y diseño. ....		21
2.1	Introducción .....	21
2.2	Modelo de Dominio .....	21

2.2.1	Definición de las clases del modelo de dominio .....	21
2.2.2	Diagrama de clases del modelo de dominio .....	22
2.3	Especificación de requisitos .....	23
2.3.1	Requisitos funcionales.....	24
2.3.2	Requisitos no funcionales .....	27
2.4	Descripción del sistema propuesto .....	27
2.4.1	Descripción de actores .....	28
2.4.2	Diagramas de casos de uso.....	28
2.4.3	Descripción de los casos de uso del sistema .....	29
2.5	Modelo de análisis .....	36
2.5.1	Diagrama de clases del análisis .....	37
2.5.2	Diagramas de interacción .....	38
2.6	Modelo de diseño .....	41
2.6.1	Diagramas de Clases del Diseño .....	41
2.7	Patrones arquitectónicos .....	43
2.7.1	Modelo Vista Controlador .....	43
2.8	Patrones de diseño .....	44
2.9	Conclusiones.....	46

<b>CAPITULO 3: Implementación y prueba.....</b>	<b>47</b>
<b>3.1 Introducción .....</b>	<b>47</b>
<b>3.2 Diagrama de componentes.....</b>	<b>47</b>
<b>3.3 Diagrama de despliegue .....</b>	<b>48</b>
<b>3.4 Pruebas de software .....</b>	<b>49</b>
<b>3.4.1 Pruebas de caja negra.....</b>	<b>50</b>
<b>3.4.2 Resultados de las pruebas.....</b>	<b>58</b>
<b>3.5 Conclusiones.....</b>	<b>58</b>
<b>REFERENCIA BIBLIOGRÁFICA .....</b>	<b>62</b>
<b>GLOSARIO.....</b>	<b>66</b>

## ÍNDICE DE TABLAS

Tabla 1: Ejecutar pago.....	29
Tabla 2: Generar reporte .....	30
Tabla 3: Exportar reporte .....	31
Tabla 4: Gestionar paquetes de pago .....	31
Tabla 5: Cambiar nivel de acceso a media .....	34
Tabla 6: Cambiar estado de usuario .....	35
Tabla 7: Establecer límite de deuda .....	36
Tabla 8: SC_Gestionar Paquete de Pago.....	51
Tabla 9: Descripción de las variables .....	52
Tabla 10: SC1_ Adicionar Paquete de Pago .....	53
Tabla 11: SC3_Eliminar Paquete de Pago .....	54
Tabla 12: SC_Ejecutar pago.....	54
Tabla 13: Descripción de variables .....	55
Tabla 14: SC1_Ejecutar pago.....	55
Tabla 15: SC1_Cambiar estado de usuario. ....	56
Tabla 16: Descripción de variables .....	57
Tabla 17: SC1_Cambiar estado de usuario .....	57

### ÍNDICE DE FIGURAS

<b>Figura 1: Modelo del dominio .....</b>	<b>23</b>
<b>Figura 2: Diagrama de casos de uso del sistema .....</b>	<b>29</b>
<b>Figura 3: DCA Realizar pago y Generar reporte .....</b>	<b>37</b>
<b>Figura 4: DCA Cambiar Nivel de Archivo Multimedia .....</b>	<b>37</b>
<b>Figura 5: DCA Cambiar estado de usuario .....</b>	<b>37</b>
<b>Figura 6: DCA Gestionar paquetes de pago .....</b>	<b>38</b>
<b>Figura 7: DC Realizar pago y Generar reporte .....</b>	<b>39</b>
<b>Figura 8: DC Cambiar nivel de media.....</b>	<b>39</b>
<b>Figura 9: DC Cambiar estado de usuario.....</b>	<b>40</b>
<b>Figura 10: DS Realizar pago y Generar reporte .....</b>	<b>41</b>
<b>Figura 11: DCD Cambiar nivel de archivo multimedia.....</b>	<b>42</b>
<b>Figura 12: DCD Cambiar estado de usuario .....</b>	<b>42</b>
<b>Figura 13: DCD Realizar pago y generar reporte .....</b>	<b>43</b>
<b>Figura 14: Diagrama de componente .....</b>	<b>48</b>
<b>Figura 15: Diagrama de despliegue.....</b>	<b>49</b>

## INTRODUCCIÓN.

El comercio aparece desde que comienzan las relaciones humanas. Con un sentido amplio, es la actividad socioeconómica consistente en la compra y venta de bienes, sea para su uso, venta o transformación. Es el cambio o transacción de algo a cambio de otra cosa de igual valor. (Galazar, 2005).

La masividad de las redes mundiales de la información han transformado el mundo y han acercado a las personas a través de las innovaciones en las comunicaciones. Esto ha propiciado un cambio profundo en todos los ámbitos de las actividades humanas, por ejemplo: la búsqueda de empleos, educación y transacciones comerciales. (Gallimard, 2011)

Debido a los cambios tecnológicos, entendimiento y adopción de Internet, formada por más de cincuenta mil redes de datos, cerca de cuatro millones de sistemas informáticos y más de setenta millones de usuarios que navegan a través de él, ISOC (Internet Society, 2006), ha sido posible la creación de nuevas tecnologías que se abren a nuevos mercados, denominados "economía digital". (M.Sc. Milagros Pacheco, 2010)

La apertura de estos nuevos mercados es fundamental en el rápido crecimiento del uso de nuevos servicios y la asimilación de nuevas tecnologías. Esto implica que en la práctica las empresas utilizan Internet, como un nuevo canal de ventas, ya que pueden hacer pedidos electrónicos, sustituyendo así los correos o teléfonos y las visitas personales, debido a que gestionar un pedido por Internet, cuesta 5% menos que hacerlo por vías tradicionales. Nace entonces el comercio electrónico, como una alternativa de reducción de costos y una herramienta fundamental en el desempeño empresarial. (Mi propio negocio INC., 1992)

El comercio, haciendo uso de las redes de datos de computadoras, se ha estado desarrollando hasta alcanzar un papel importante y representar un futuro aún mejor, facilitando la reducción del tiempo de compra y adquisición de productos. Sin la aparición de ningún intermediario, cuando el producto llega a manos del comprador el ciclo de venta culmina. De esta forma el contacto personal es nulo además de que se elimina el costo del transporte y otros valores agregados que se evidencian en otro tipo de ventas.

El comercio electrónico se expande con mayor participación en los mercados minoristas en la misma medida que se logra mayor conectividad con Internet. Se basa en las fortalezas tecnológicas que beneficia a empresas y consumidores finales, significando ahorro de costo. (CEPEC, 2001)

En Cuba, debido a la infraestructura de conectividad y servicios informáticos, el comercio electrónico está reducido, por la falta del marco jurídico adecuado y personal capacitado, infraestructura de cobros y pagos electrónicos en línea, infraestructura de logística, infraestructura de certificación y registro. (Consejo de Ministros, 2006). Es por esto que una posible área de aplicación de esta forma de pago podría ser el turismo, donde existen mejores condiciones técnicas y los usuarios están más versados en estos temas por experiencias previas en sus respectivos países. De igual modo tienen mayor cultura sobre el pago por servicios de ocio como materiales audiovisuales y otros, que a diferencia, en Cuba se enfocan en la educación generalizada y la formación de valores.

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el centro de desarrollo Geoinformática y Señales Digitales (GEYSED) en la facultad 6, destinado al desarrollo de aplicaciones informáticas vinculadas al campo de las señales digitales (imagen, sonido y video), y en este, el Proyecto de Catalogación y Publicación de Medias. Su trabajo estuvo centrado en la creación de la Plataforma VideoWeb en su versión 1.0, desarrollado con Drupal 6, para la gestión y transmisión de contenidos audiovisuales en redes de área local o Internet, que permite la reproducción y descarga de estos de manera automática.

Actualmente, el proyecto está enmarcado en elevar el nivel operacional de la plataforma y migrar hacia Drupal 7, cumpliendo con las normativas de desarrollo, dando surgimiento de esta manera a la segunda versión de la plataforma. Entre las prestaciones de la versión 1.0 de VideoWeb se encuentra un mecanismo que gestiona el pago por concepto de consumo de los contenidos audiovisuales. Esta funcionalidad es de sumo interés para la nueva versión si se tiene en cuenta que se desea insertar en mercados potenciales como hoteles, cibercafés y aeropuertos, donde cobrar estos servicios de una manera segura y efectiva es primordial.

Se ha analizado que mantener dicha funcionalidad en Drupal 6, mientras el resto de la plataforma asciende a una versión superior de esta tecnología, no es lo más aconsejable, pues repercutiría negativamente en el despliegue del sistema, ya que en el servidor se tendrían que instalar dos versiones de drupal que estarían consumiendo doblemente sus recursos, ralentizando el tiempo de respuesta de la aplicación. Por otra parte una nueva entrega de drupal trae consigo mejoras importantes como es la velocidad del acceso a los datos y facilidades en el desarrollo de las aplicaciones, además garantizaría seguridad ante cualquier vulnerabilidad o fallo de seguridad.



También es real el hecho de que, pasado un tiempo, la comunidad de desarrollo dejará de darle soporte a Drupal 6 y pueden aparecer problemas de seguridad a los que no habrá forma de darle solución rápida y efectiva. (B Herrero Hurtado, 2010)

A partir de lo anteriormente expuesto se define el siguiente **problema a resolver**: ¿cómo lograr registrar el pago por concepto de consumo de contenidos en la Plataforma VideoWeb 2.0?

Para dar solución al problema científico se plantea como **objeto de estudio**: los procesos de transacciones en las pasarelas de pago en aplicaciones informáticas, delimitando como **campo de acción**: los procesos de transacciones en las pasarelas de pago para la plataforma VideoWeb 2.0.

Se presenta como **idea a defender**: con el desarrollo de una pasarela de pago en la Plataforma VideoWeb 2.0, se lograrán registrar las operaciones de cobro y pago electrónico por concepto de consumo de contenidos publicados.

Se define como **objetivo general**: desarrollar un módulo de pasarela de pago para la Plataforma VideoWeb 2.0.

Para dar solución al objetivo se han definido las siguientes **tareas**:

1. Caracterizar los procesos de transacciones que ocurren en las pasarelas de pago a nivel nacional e internacional.
2. Analizar los procesos de transacciones existentes en el módulo de pasarela de pago de la plataforma VideoWeb en su versión 1.0 y otros sistemas similares.
3. Seleccionar las técnicas y tecnologías necesarias para el desarrollo de la pasarela de pago para la plataforma VideoWeb 2.0.
4. Describir las funcionalidades que debe brindar el módulo de pasarela de pago de la plataforma VideoWeb 2.0.
5. Realizar el diseño de los componentes necesarios para el desarrollo de la pasarela de pago para la plataforma VideoWeb 2.0.
6. Implementar el módulo de pasarela de pago para la plataforma VideoWeb 2.0.

7. Integrar el módulo de pasarela de pago en la plataforma VideoWeb 2.0.
8. Realizar pruebas necesarias para la comprobación del correcto funcionamiento de la pasarela de pago para la plataforma VideoWeb 2.0.

Para llevar a cabo la realización de las tareas antes mencionadas se emplearon los siguientes métodos científicos de investigación:

## **Métodos Teóricos**

**Analítico-Sintético:** para la realización de un análisis sobre los mecanismos de pago electrónico existentes, detallando sus características y potenciales ventajas. Este método permitió entender el tema, la dinámica de sistemas similares, simplificando el estudio del mismo.

**Inductivo-Deductivo:** para la adquisición de conocimiento general sobre los mecanismos de pago electrónico, basándose en el estudio de aspectos específicos encontrados en la bibliografía asociada a la investigación del tema.

**Histórico-Lógico:** este método permitió el estudio de la situación actual del tema (estado del arte), analizando posibles soluciones existentes, además permitió seguir la evolución y desarrollo, incluyendo las tendencias actuales, de los mecanismos de pago electrónico para la solución del problema anterior.

**Modelación:** se utilizó este método en el modelado de la implementación del módulo, como forma de abstracción de la realidad.

## **Métodos Empíricos**

**Experimento:** como forma de validar la estrategia de recomendación se realizaron, mediante la experimentación, una serie de pruebas al módulo desarrollado, creando y adaptando las condiciones en un ambiente específico para la obtención de resultados que permitió verificar el cumplimiento del objetivo planteado.

El desarrollo de la investigación se organiza de la siguiente manera:

**Capítulo 1: Introducción al estudio del módulo pasarela de pago. Fundamentación Teórica;** se abordan conceptos asociados al dominio del problema. Además, se realiza un estudio del estado del arte y análisis de soluciones. De igual manera se hace una valoración de las herramientas y el lenguaje de programación, así como la metodología de desarrollo a utilizar.

**Capítulo 2: Análisis y Diseño.** Presentación de la Solución Propuesta, se desarrollan los artefactos correspondientes al análisis y diseño, se hace una modelación de la propuesta de solución, se identifican los requisitos funcionales y no funcionales, se diseña la arquitectura y los diagramas correspondientes.

**Capítulo 3: Implementación y prueba.** Se realiza la implementación del módulo pasarela de pago y se valida la solución propuesta realizando pruebas de aceptación al módulo, donde se muestra el resultado de las pruebas realizadas para demostrar que la solución es correcta. Finalmente se dan las conclusiones generales y un conjunto de recomendaciones con vista a trabajos futuros.

## CAPÍTULO 1: Introducción al estudio del Módulo de Pasarela de Pago. Fundamentación teórica.

### 1.1 Introducción

En el presente capítulo se hace un análisis de los principales conceptos asociados a la investigación para una mejor comprensión, dígase mecanismos de pago electrónico, describiendo sus especificidades y modo de funcionamiento. Se caracterizan las herramientas a utilizar, particularidades y motivos para su uso y se define la metodología de desarrollo de software a emplear, una descripción de la misma, sus fases, flujos de trabajo y artefactos que arroja. Se describe la situación que genera el problema a resolver y se realiza un análisis de las soluciones ya existentes, justificando la negativa a usar alguna de estas y la decisión de implementar una solución completamente nueva.

### 1.2 Conceptos asociados al dominio del problema

#### 1.2.1 Comercio electrónico

El Comercio Electrónico o *e-commerce* como también es conocido, es una metodología que da respuesta a varias necesidades de empresas y consumidores, como reducir costes, mejorar la calidad de productos y servicios, acortar el tiempo de entrega o mejorar la comunicación con el cliente. (Dans, 2007). Se define como cualquier forma de transacción o intercambio de información comercial basada en la transmisión de datos sobre redes de comunicación, tales como Internet y otras redes informáticas.

Es necesario tener en cuenta además que la mayor parte del comercio electrónico consiste en la compra y venta de productos o servicios entre personas y empresas, sin embargo un porcentaje considerable consiste en la adquisición de artículos virtuales, software y derivados en su mayoría, tales como el acceso a contenido "*premium*"<sup>1</sup> de un sitio web.

#### 1.2.2 Sistema de pago electrónico

Un sistema de pago electrónico es un sistema de pago que facilita la aceptación de pagos electrónicos para las transacciones en línea a través de Internet.

Los EPS o sistemas de pagos electrónicos, realizan la transferencia del dinero entre compradores y vendedores en una acción de compra-venta electrónica a través de una entidad financiera autorizada por

---

<sup>1</sup>Un producto *premium* es aquel que combina una excelente calidad, altos precios, distribución a través de canales de más calidad y una moderada inversión en publicidad.

ambos. Es, por ello, una pieza fundamental en el proceso de compra-venta dentro del comercio electrónico. (Josep Pegueroles Vallés, 2009)

### **1.2.3 Tarjeta de crédito**

Una tarjeta de crédito es un instrumento financiero de pago, emitido por una entidad, que puede ser una tarjeta plástica con una banda magnética, un microchip y un número en relieve. Esta tarjeta permite al titular acceder a un saldo superior al disponible en su cuenta corriente, ya que el dinero proviene de un crédito por parte de la entidad, que el titular deberá amortizar posteriormente conforme a los plazos y condiciones pactados con su banco o caja de ahorros. (Monetos, 2012)

Es muy importante si se tiene en cuenta que no se lleva el dinero encima, es más segura que el efectivo. Entre las más conocidas del mercado están: Visa, American Express, MasterCard, Diners Club, JCB, Discover y Cabal.

### **1.2.4 Tarjeta de débito**

La tarjeta de débito es un instrumento financiero de pago emitido por una entidad bancaria o financiera, que permite al titular acceder únicamente al saldo positivo que dispone en su cuenta corriente o de ahorro, en el momento en que realiza un movimiento. Un pago a través de una tarjeta de débito trae como consecuencia un cargo directo a la cuenta asociada a la misma. La tarjeta de débito también puede ser vista como un medio por el cual se utilizan los fondos que deposita el titular en una cuenta determinada. En este sentido, el portador de la tarjeta es al mismo tiempo el titular de la cuenta. (Monetos, 2012)

En este tipo de tarjeta el dinero que se usa es el que se toma a débito del que el titular dispone en su cuenta bancaria y no el que le presta el banco como ocurre con las tarjetas de crédito.

### **1.2.5 Monedero electrónico**

El monedero electrónico es un dispositivo en forma de tarjeta plástica dotada de un “chip”, con capacidad de transferencia y almacenamiento de un valor monetario y de lectura, escritura y almacenamiento de transacciones comerciales realizadas en moneda nacional. Dichos monederos deberán concentrar la información de las operaciones que se realicen por ese medio en un banco de datos que reúna elementos de seguridad e inviolabilidad, a través de una institución de crédito. (Artículo 29-A, 2012)

Esta tarjeta a diferencia de la tarjeta de crédito, no requiere que se verifique con el banco si existen fondos o no, el chip incorporado al monedero establece directamente esta información.

## 1.3 Objeto de estudio

### 1.3.1 Descripción general

Una pasarela de pago (también llamada "gateway de pago", "payment gateway" o "Terminal Punto de Venta (TPV) virtual") es un proveedor de servicios de aplicación de comercio electrónico que autoriza pagos a negocios electrónicos, minoristas electrónicos, ladrillo y clic<sup>2</sup>o el tradicional ladrillo y mortero<sup>3</sup>. Es el equivalente de una TPV física ubicada en la mayoría de los almacenes al detalle. Las pasarelas de pago cifran información sensible, tal como números de tarjetas de crédito, para garantizar que la información pasa en forma segura entre el cliente y el vendedor. (Lidera Network, 2009)

Una pasarela de pagos es la pieza de software que sirve para procesar los cobros y pagos de un sistema de comercio electrónico. Su misión principal es hacer que los cobros y pagos sean rápidos, sencillos y seguros, para que las transacciones de compraventa se puedan ejecutar de manera efectiva entre el comercio y los clientes, actuando la entidad procesadora de la operación como intermediario.

En el mercado existen multitud de pasarelas de pago en forma de software que distintas empresas ponen a disposición de los negocios de comercio *online*. Esto significa que no es necesario que cada uno de ellos desarrolle una específica para su tienda *online*, sino que basta con integrar una pasarela de pagos en su sistema. Por ejemplo, las empresas Sage, ofrecen a sus clientes su servicio de pasarela de pago *SagePay*<sup>4</sup>, la cual se puede integrar fácilmente con distintas plataformas de comercio online disponibles en el mercado. (Pablo Herrero, 2012)

### 1.3.2 Proceso de compra en la web usando pasarela de pago

Una pasarela de pago facilita la transferencia de información entre un portal de pago (como puede ser un sitio web o un servicio *Interactive Voice Response*, por sus siglas en inglés IVR) y el Procesador Interfaz o

---

<sup>2</sup>Los ladrillos y clics se utiliza para que las organizaciones sean capaces de vender a sus clientes de una manera diferente y ofrecerles ofertas en línea que pueden obtener dentro de la tienda. Ladrillos y clics también permite a las empresas que permitan a sus clientes productos de reserva para que puedan adquirirlos en una fecha posterior.

<sup>3</sup>Ladrillo y Mortero de negocios es un término usado principalmente en Internet para diferenciar entre las empresas que se basan exclusivamente en línea, y los que tienen una contrapartida en el mundo real. Un negocio del ladrillo y el mortero tiene una dirección comercial de ladrillo y el mortero, donde los clientes pueden realizar transacciones cara a cara. La compañía también podría tener una presencia en línea.

<sup>4</sup>SagePay es el proveedor de servicios de pago independiente con el crecimiento más rápido. Es una división de la empresa de software para empresas Sage.

banco adquirente de manera rápida y segura. Cuando un cliente ordena un producto de un vendedor que tiene habilitado una pasarela de pago, esta realiza una serie de tareas para procesar la transacción, de manera transparente para el comprador. Por ejemplo:

- Un cliente realiza un pedido en un sitio web presionando el botón de "emitir orden" o ingresa los detalles de su tarjeta de crédito a un servicio IVR.
- Si la orden es a través de un sitio web, el navegador web del cliente cifra la información que viaja hasta el servidor web del vendedor. Esto se hace normalmente mediante cifrado SSL (Secure Socket Layer).
- El vendedor reenvía los detalles de la transacción a su pasarela de pago, el cual contiene los detalles de las cuentas de sus vendedores. Normalmente, esta es otra conexión cifrada mediante SSL al servidor de pago, almacenada en la pasarela de pago.
- La pasarela de pago que recibe la información de la transacción del vendedor reenvía la información al banco adquirente del vendedor.
- El banco adquirente reenvía la información de la transacción al banco emisor (el banco que le emitió la tarjeta de crédito al cliente) para autorización.
- El banco emisor de la tarjeta recibe el pedido de autorización y envía una respuesta a la pasarela de pago (a través del banco adquirente) con un código de respuesta. Además de determinar el destino del pago (es decir, aprobado o rechazado), el código de respuesta se usa para definir la razón por la cual la transacción falló (como por ejemplo, por fondos insuficientes o enlace al banco no disponible).

### **1.3.2.1 Proceso de compra en la Plataforma VideoWeb 2.0**

El flujo de procesos está diseñado para que entre las opciones de pago que tenga el usuario, esté la existencia de un banco local donde se despliegue el sistema. De esta manera se tiene que la persona deposita un monto de dinero en una cuenta correspondiente a su usuario en VideoWeb 2.0 que le permitirá consumir posteriormente los diferentes materiales audiovisuales publicados en la plataforma, además puede establecer también un límite de deudas para cada usuario, lo que le permitiría seguir reproduciendo materiales de la plataforma aunque ya no tenga dinero en su cuenta, hasta llegar al tope del límite de la

deuda; por ejemplo, en el caso de los hoteles, después de haber llegado a ese límite se le entrega al usuario una factura donde se le indica cuánto ha gastado, o sea, cuanto tiene que pagar de deuda.

La causa de la existencia de un banco local, puede estar dada porque donde se despliegue el sistema no exista conectividad con bancos u otros medios de pago externos, o porque el usuario no tenga cuantas bancarias ni tarjetas de crédito.

### **1.3.3 Ventajas de las pasarelas de pago**

Las pasarelas de pago facilitan la gestión del cobro masivo de los clientes, al tiempo que le permite obtener la información de gestión para la conciliación automática del cobro. Es un servicio que se puede personalizar adaptándose a las necesidades de su empresa en base a sus obligaciones y necesidades de registro de datos e integración y brinda la posibilidad de inclusión en la web, lo que facilita y agiliza la gestión del cobro.

### **1.4 Descripción de la situación problemática**

El resultado positivo de un negocio electrónico está condicionado por la rentabilidad del mismo, ya sea por la venta de productos o por los servicios que brinda. La problemática de cómo cobrar por esto es a lo que se le debe prestar mayor atención, por lo que es necesario contar con una forma de pago que sea adecuada para el mismo y confiable. Las entidades que comercialicen con productos informáticos tienen como ventaja que para llevar a cabo sus transacciones económicas pueden usar, además de la vía tradicional, los beneficios del desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC por sus siglas). Contar con un mecanismo de pago electrónico de fácil uso, seguro, fiable, con un diseño adecuado y bien empleado puede llegar a incrementar de forma significativa los ingresos previstos.

La plataforma de transmisión, reproducción y descarga de materiales audiovisuales VideoWeb, a través de la red ofrece facilidades y ventajas a sus usuarios. Dicha plataforma se ha diseñado para que sus servicios sean configurables y se ajusten a las necesidades específicas de cada cliente, cuenta con un mecanismo de gestión de pagos por concepto de consumo de los contenidos audiovisuales de la plataforma, desarrollado con Drupal 6. Con la inevitable migración de VideoWeb 1.0 hacia VideoWeb 2.0, se hace necesaria la migración de dicho mecanismo de pago a Drupal 7, debido a que se ha analizado que mantener dicha funcionalidad en Drupal 6, mientras el resto de la plataforma asciende a una versión superior de esta tecnología, no es lo más aconsejable, pues repercutiría negativamente en el despliegue del sistema, ya que en el servidor donde se instale el sistema deberá contar con mejores recursos de



hardware, lo que representa un aumento del costo monetario de la tecnología empleada, ya que deberá funcionar de manera estable con dos sistemas ejecutándose simultáneamente sobre versiones distintas de Drupal o de lo contrario las respuestas ofrecidas a las estaciones clientes podrían verse afectadas por la demora.

Además, aunque la versión 6 tiene una estabilidad muy elevada y una gran cantidad de módulos, la comunidad de desarrollo está centrada en la versión 7, por lo que pronto dejarán de darle soporte. (B Herrero Hurtado, 2010)

De esta forma mientras que los módulos en la versión 7 de Drupal serán actualizados frecuentemente por la comunidad de desarrollo, mejorando sus características y rendimiento, en la versión 6 irán siendo olvidadas con el tiempo y quedando fuera del soporte de seguridad. Este es un tema muy importante para el desarrollo del módulo pasarela de pago, si se tiene en cuenta que trabaja con información muy sensible, que de filtrarse pudiera perjudicar económicamente tanto al usuario como a la empresa en la que se despliegue el sistema. Además en la versión 7 de Drupal, visto desde la óptica de los desarrolladores que emplean drupal, cuenta con menús de más fácil navegación y se trabaja más rápido, lo que permite que el coste de tiempo de desarrollo sea menor y su nivel operacional aumenta considerablemente.

### **1.5 Soluciones existentes**

La Plataforma VideoWeb 1.0 cuenta con un mecanismo de pago desarrollado con la finalidad de gestionar los pagos que se hacen por concepto de consumo de los servicios que la misma brinda (visualización y reproducción de los contenidos audiovisuales). La pasarela desarrollada es capaz de disminuir el costo que implica el pago de manera presencial, ya sea costo de tiempo o monetario, en el entorno donde se despliegue el sistema, además permite atender múltiples usuarios simultáneamente, aumentando las prestaciones de la plataforma. El mismo fue desarrollado basándose en las características específicas de la plataforma, además brinda la posibilidad de seleccionar entre un grupo abarcador de tipos de pagos: cheque electrónico, transferencia bancaria, tarjeta de crédito o débito; compañías de las cuales fuera posible aceptar pago: VISA, MasterCard, PayPal; datos necesarios para consumir el servicio: número de tarjeta, fecha de caducidad, *Personal Identificacion Number* (por sus siglas en inglés PIN) u otra información solicitada por la institución bancaria.

Teniendo en cuenta esta opción, se hizo un análisis para integrar este módulo a la nueva versión de la

plataforma, en el cual se llegó a la conclusión de que no es factible, debido al cambio en la versión de la tecnología de Drupal, además de que la plataforma VideoWeb 2.0 necesita funcionalidades nuevas que ese modulo no tiene, como es el ejemplo de los límites de deuda y el banco que va a tener integrado el nuevo módulo. No obstante se hizo una búsqueda de negocios online que tuvieran en funcionamiento mecanismos similares para la compra o descarga de materiales audiovisuales y observar las tendencias y patrones que usan.

### 1.5.1 XtraLife.es

<http://www.xtralife.es/>

Es un sitio web español para la venta de videojuegos. Tiene como particularidad que permite comprar por pedido recibiendo la adquisición en casa o descargar directamente el material a tu computadora, siendo esta opción muy atractiva al ahorrar el importe asociado al servicio de entrega y los materiales relacionados a la compra en sí. Cuenta con un carrito de compra accesible desde cualquier lugar con información asociada al pedido, la posibilidad de elegir diferentes métodos de pago y brinda una ayuda sobre los datos a introducir para la compra. Posee la respectiva seguridad usando protocolos seguros y la solicitud de información personal para la factura, además de información sobre cómo conducirse en caso de error en la transacción y los posibles motivos de este.

El principal atractivo de este ejemplo es la innovadora capacidad de efectuar ambas formas de venta: entrega y descarga; y las ventajas y facilidades que presenta esta última, especialmente en contenidos virtuales como videojuegos, películas o música, aprobando la descarga después de haber validado la compra, no pudiendo el usuario acceder al vínculo de descarga sin haberse efectuado la transacción. Esta solución representa un ejemplo muy útil en la implementación de una posible solución de mecanismo de pago para la plataforma VideoWeb 2.0.

### 1.5.2 Tematika.com

<http://www.tematika.com/>

Es un sitio web argentino destinado a la comercialización de productos de ocio (libros, música, películas). Este no permite la descarga de manera directa, sino enviando los pedidos al domicilio especificado. Omitiendo este detalle, el sitio presenta un ejemplo muy adecuado en cuanto al uso de mecanismos de

pago electrónico para la compra de materiales audiovisuales, y donde se pueden señalar algunos aspectos muy útiles en la posterior implementación de un módulo similar, por ejemplo: posee diferentes medios de pago, siendo el más sobresaliente el pago por tarjeta de crédito, permitiendo el uso de una gran variedad de estas; presentando además una interfaz intuitiva y amigable, para introducir la información que solicita para la transacción la entidad adquirente, en este caso el *Santander Rio Online Banking*.

También las facilidades del carrito de compra, visible desde cualquier lugar del sitio con información del monto al que asciende el pedido, pudiendo acceder a este para organizar la compra (eliminar/añadir unidades, entre otras) o ir directamente al sistema de compra donde se especificarán los detalles de la transacción. Todo esto sustentado por una elevada seguridad al efectuarse todas las operaciones a través de una conexión cifrada por el protocolo *Hypertext Transfer Protocol Secure* (por sus siglas en inglés HTTPS) y solicitar información personal sobre el comprador, como puede ser número de identidad, para evitar suplantaciones.

### **Comparación**

Se tiene que con la pasarela de VideoWeb es posible controlar las transacciones de cobro a los usuarios por concepto de reproducción y descarga de los materiales audiovisuales, mientras que en tematika la entrega de sus productos son enviados directamente al cliente, no siendo así en XtraLife, que permite descargar directamente el material a tu computadora. Además la pasarela cuenta con la existencia de un banco local donde se despliegue el sistema, facilitando a los usuarios que no tienen tarjetas de crédito o de débito, o a la institución que no tiene un sistema de comunicación eficiente para establecer las transacciones necesarias con los bancos que pueden ser nacionales o internacionales, que se pueda realizar el consumo y cobro por el mismo del contenido publicado. Al igual que Xtralife, la pasarela aprueba la descarga después de haber validado la compra, pero no ofrece, como tematika, un importe al realizar tu cuenta de 20 dólares para tu primera compra.

Para la nueva pasarela de pago se utilizan aspectos importantes que estas soluciones existentes presentan, como es el caso de que permite reproducir o descargar directamente el material a tu computadora. Cuenta con un carrito de compra accesible desde cualquier lugar con información asociada al pedido y la posibilidad de elegir diferentes métodos de pago. Aprueba la reproducción o descarga después de haber validado la compra, no pudiendo el usuario acceder a estos vínculos sin haberse

efectuado la transacción. Cuenta con la existencia de un banco local el cual fue implementado en el mismo módulo.

## **1.6 Tecnologías para el desarrollo de la aplicación**

### **1.6.1 Metodología de Desarrollo de Software**

Como su nombre indica las metodologías son un conjunto de métodos, reglas, que por una parte sirven de guía para realizar los trabajos que van dando forma a los desarrollos y que por otra obligan a la dirección del proyecto y a los componentes de los equipos a realizar ciertas comprobaciones sistemáticas de modo que el resultado final, al menos desde un punto de vista formal, no presente incoherencias y esté dirigido a un objetivo claro y prefijado. (Pressman, Roger. 2003)

En la selección de la metodología de software los criterios habitualmente usados son un tanto subjetivos, por tanto es necesario tener en cuenta una serie de competencias que atañen al equipo de desarrollo y son críticas a la hora de elegir la más adecuada, a través de un exhaustivo análisis que arrojará cuál es la de mayor idoneidad. (Pressman, Roger. 2003)

En este caso el módulo a desarrollar debe ser adaptado a una aplicación existente para la cual ya fue previamente seleccionada una metodología, en aras de mantener la uniformidad y compatibilidad entre los sistemas se necesita lograr un consenso en cuanto a esto. También el grado de conocimiento de la metodología en cuestión es un factor de indudable relevancia, donde la experiencia, entendimiento o preparación del equipo sobre la utilización de esta es fundamental para evitar errores o los consecuentes retrasos por la investigación sobre el tema y la capacitación requerida. Es por ello que la metodología seleccionada es RUP.

#### **1.6.1.1 Metodología RUP**

El Proceso Unificado de Desarrollo es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible.

RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica acceda a la misma base de datos de conocimiento. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar software. (Agil Unified Process, 2012)

## **RUP posee tres características fundamentales:**

**Dirigido por casos de uso:** El proceso de desarrollo está dirigido por casos de uso, sigue un hilo que avanza a través de una serie de flujos de trabajo que parten de los casos de uso.

**Centrado en la arquitectura:** La arquitectura es el elemento central a partir del cual el sistema se desarrolla y evoluciona, no solo en su desarrollo inicial, sino también a lo largo de las futuras iteraciones y generaciones, por lo que esta debe permitir el correcto desarrollo de todos los casos de uso requeridos tanto en el presente como en el futuro.

**Iterativo e incremental:** En RUP el trabajo se divide en partes más pequeñas o mini-proyectos llamados iteraciones, con la culminación de cada iteración se logra un incremento o crecimiento del producto. (Jacobson, y otros, 2000)

El Proceso Unificado usa el Lenguaje de Modelado Unificado (*Unified Modeling Language* por sus siglas en inglés UML) en la preparación de todos los diagramas del sistema. De hecho, UML es una parte integral del Proceso Unificado, fueron desarrollados a la par.

### **1.6.2 Lenguaje de modelado**

Lenguaje de Modelado Unificado (*UML* por sus siglas en inglés). Está respaldado por el *Object Management Group* (por sus siglas en inglés OMG). El UML prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. (Popkin Software and Systems, 2007)

UML ayuda a especificar, visualizar y documentar los modelos de sistemas de software, incluyendo su estructura y diseño, de manera que cumpla con todos estos requisitos. Se puede usar UML para el modelado de negocios. Si se utiliza cualquiera de la gran cantidad de diagramas UML, se pueden analizar los requisitos de su aplicación futura y diseñar una solución que les satisface. (OMG, 2012)

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado de Desarrollo o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas, diagramas que son necesarios para la realización del módulo por lo que se selecciona como lenguaje de modelado a utilizar. (Fowler y Sccott, 1999)

## **Tipos de diagramas:**

- Diagramas de Casos de Uso para modelar los procesos 'negocio'.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

## **1.6.3 Lenguajes de programación**

### **1.6.3.1 PHP**

Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) tomando el código en PHP como su entrada y creando páginas web como salida. (Torre, 2006)

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*) y está publicado bajo la *PHP License*, la *Free Software Foundation* considera esta licencia como software libre. (MiguelAngel, 2001)

Algunas de las más importantes capacidades de PHP son: compatibilidad con los gestores de bases de datos más comunes, como MySQL, PostgreSQL, Oracle, Informix, y ODBC. Incluye funciones para el envío de correo electrónico, *upload* de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales. (Miguel Angel, 2001)

La plataforma VideoWeb 2.0 está desarrollada con el CMS Drupal, el cual utiliza el lenguaje de programación PHP, siendo este un factor importante en la selección de este lenguaje para el desarrollo del módulo. Además, posibilita mantener una compatibilidad con los demás módulos que utiliza la plataforma.

### **1.6.3.2 JavaScript**

JavaScript es un lenguaje utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes orientados a objetos mucho más complejos. Con JavaScript se pueden crear diferentes efectos e interactuar con usuarios. (Maestros del Web, 1997)

Este lenguaje posee varias características, entre ellas se destaca que es un lenguaje basado en acciones que posee menos restricciones. Además, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. (Maestros del Web, 2010)

### **1.6.4 Herramientas**

#### **1.6.4.1 NetBeans IDE 7.2**

Entorno de desarrollo integrado (por sus siglas en inglés IDE). Para la implementación se utilizó el Netbeans IDE 7.2. Esta es una herramienta de código abierto bajo licencia pública (GPL) que permite el desarrollo de aplicaciones con características modulares. Entre sus funcionalidades permite escribir, depurar, compilar y ejecutar programas, además soporta otros lenguajes de programación; es un producto libre y gratuito sin restricciones de uso, se pueden encontrar varias extensiones para el mismo. Entre las ventajas se puede destacar el auto-completamiento de código, interfaz para el diseño de GUI<sup>5</sup>, la capacidad de importar clases, software multiplataforma y extensible. Entre los lenguajes de programación para los que brinda soporte se encuentra PHP, lenguajes en el cual ha sido desarrollada la plataforma VideoWeb 2.0, así como los módulos que utiliza. (NetBeans Community, 2000)

---

<sup>5</sup>La interfaz gráfica de usuario, conocida también como GUI (del inglés *graphica luser interface*) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

### 1.6.4.2 Herramienta CASE

Visual Paradigm es una herramienta CASE profesional que soporta la última versión de UML 2.1 así como el ciclo de vida completo del desarrollo de software, análisis y diseño orientados a objetos, construcción, pruebas y despliegue, exportación desde Rational Rose, exportación/importación XML, generación de informes y edición de figuras. (Ambysoft, 2012).

#### Ventajas:

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse con la herramienta Netbeans IDE.
- Disponibilidad en múltiples plataformas.
- Ingeniería Inversa tiene capacidad de soporte para PHP. .

### 1.6.4.3 Drupal 7.0

Drupal es un Sistema de gestión de contenidos (CMS<sup>6</sup>) que se distribuye como software libre bajo licencia GNU GPL (General Public License) versión 2 o superior. (Rodríguez, Fran Gil, 2012)

El software está desarrollado con el lenguaje de programación PHP. Está maquetoado con hojas de estilo CSS, con lo que es posible construir sitios web totalmente accesibles. (Rodríguez, Fran Gil, 2012)

Drupal 7 es el resultado de casi 3 años de desarrollo colaborativo, en el que han participado un gran número de miembros de la comunidad. Si bien es cierto que Drupal 7 se desarrolla a partir de Drupal 6, son muchos los cambios conceptuales y estructurales, lo que obliga a usuarios y desarrolladores a llevar a cabo un importante esfuerzo de actualización de conocimientos. (Rodríguez, Fran Gil, 2012)

---

<sup>6</sup>Un sistema de gestión de contenidos (Content Management System en inglés, abreviado CMS), es un software que permite crear una estructura base para la creación y administración de contenidos, principalmente de páginas web. Generalmente un CMS es una aplicación con una base de datos asociada en la que se almacenan los contenidos, separados de los estilos o diseño. El CMS controla también quién puede editar y visualizar los contenidos, convirtiéndose en una herramienta de gestión integral para la publicación de sitios web.



## 1.6.4.4 PostgreSQL 9.1

Para el desarrollo del módulo se utiliza PostgreSQL 9.1, nueva versión que incorpora avanzadas características en los temas de seguridad, soporte de aplicaciones, control y seguimiento, así como rendimiento y almacenamiento de datos especializados. (Rafael Martínez, 2012)

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (Rafael Martínez, 2012)

## 1.6.4.5 PgAdmin III 1.10.0

Es una aplicación gráfica para el trabajo con el gestor de bases de datos PostgreSQL, de código abierto. Es capaz de gestionar versiones a partir de la versión PostgreSQL 7.3. Está escrita en C++ usando la librería gráfica multiplataforma wx Widgets, lo que permite que se pueda usar en Linux, y Windows.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP<sup>7</sup> y puede encriptarse mediante SSL para mayor seguridad. (PgAdmin III, 2008)

## 1.7 Conclusiones

Con la culminación de este capítulo se puede concluir que:

Es necesaria la realización de un adecuado mecanismo de pago para VideoWeb 2.0 que sustituya al existente en la versión 1.0. El estudio del estado del arte evidencia la existencia de una aplicación para efectuar el pago de los servicios que ofrece la plataforma VideoWeb, desarrollada en Drupal 6, que deberá ser sustituida debido al proceso de ampliación operacional en el que ha entrado la plataforma que trae como consecuencia la migración de la misma a Drupal 7.

---

<sup>7</sup>Denominación común que se utiliza para designar al conjunto de protocolos de red en los que está basada Internet.

Tras delimitar las características de la presente investigación conllevan al uso de RUP como metodología de desarrollo, los lenguajes de programación PHP, porque Drupal está escrito en PHP, además de cualquier modificación (medianamente profunda) que se quiera hacer, deberá hacerse con códigos en PHP, Java Script para realizar las validaciones de la aplicación, como herramientas Drupal 7, CMS en el cual se realizó el módulo, PostgreSQL por ser un sistema gestor de bases de datos objeto-relacional que Drupal utiliza, NetBeans IDE por ser compatible con los lenguajes de programación utilizados, Visual Paradigm como herramienta CASE para modelar el sistema y PgAdmin III debido a que es una aplicación gráfica que trabaja con el gestor de base de datos utilizado.

### CAPÍTULO 2: Análisis y diseño.

#### 2.1 Introducción

En el presente capítulo se conceptualiza el entorno mediante un modelo de dominio en el cual se analizan cada una de las entidades y conceptos presentes en el contexto donde trabajaría el módulo para la plataforma, además de las relaciones existentes entre cada uno de estos, se especifican los requisitos funcionales y no funcionales que deberá cumplir la solución propuesta, se presentan los diagramas de casos de uso del sistema, se describen los actores y se detallan los casos de uso del sistema. Así como demás diagramas de comunicación para describir la interacción entre las entidades y diagramas de clases para una visión general del funcionamiento de la aplicación.

#### 2.2 Modelo de Dominio

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades. (Craig Larman, 2003)

El modelo del dominio muestra (a los modeladores) clases conceptuales significativas en un dominio de problema; es un artefacto de la disciplina de análisis construido durante la fase de inicio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. (Craig Larman, 2003)

Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.

##### 2.2.1 Definición de las clases del modelo de dominio

**Plataforma VideoWeb 2.0:** Plataforma que permite la reproducción y descarga de los archivos multimedia publicados que contiene de manera automática y a través de la red.

**Archivos Multimedia gratis:** Materiales audiovisuales publicados en la plataforma de forma gratuita que pueden ser de interés a los usuarios, ya sean películas, documentales o videos musicales.

**Archivos Multimedia con costo:** Materiales audiovisuales publicados en la plataforma que pueden ser de interés a los usuarios, ya sean películas, documentales o videos musicales que deben pagar para reproducir.

**Usuario:** Usuario no registrado que puede convertirse en miembro pleno tras finalizar su registro en la plataforma y que pueden ver los archivos multimedia publicados que son gratis.

**Usuario registrado:** Usuarios autenticados en la plataforma con permisos para la reproducción de archivos multimedia tanto gratis como con costo una vez pagados.

**Administrador:** Máxima jerarquía dentro de los niveles de usuario, con dominio total sobre los permisos de los usuarios y de los archivos multimedia, puede cobrar a los usuarios para permitirles utilizar los servicios de la plataforma.

**Pasarela de pago:** Módulo que permite automatizar el proceso de cobro por los servicios de reproducción y descarga en la plataforma VideoWeb 1.0.

### 2.2.2 Diagrama de clases del modelo de dominio

El modelado de dominio tiene como objetivo describir y comprender las clases y objetos más significativos dentro del contexto del problema, lo cual ayuda a definir los procesos y roles más relevantes para el sistema a desarrollar. Esto ayuda a los usuarios, desarrolladores y clientes a manejar un vocabulario común, que permita compartir el conocimiento manteniendo un lenguaje único y consistente que evite confusiones y posibilite el entendimiento de todas las partes interesadas.

Se realizó un modelo conceptual que permitiese de manera visual, mostrar los principales conceptos que se manejan en el dominio del sistema. Donde se pudo definir qué:

La plataforma VideoWeb 2.0 publica archivos multimedias gratis y con costo. Los usuarios que no están autenticados pueden reproducir solo los archivos gratis y los usuarios que están autenticados, además de los gratis los archivos con costo, cumpliendo con la condición de que tiene que pagar para poder reproducirlos. Además el administrador es quien gestiona todos los permisos en la plataforma y en la pasarela de pago que esta contiene.

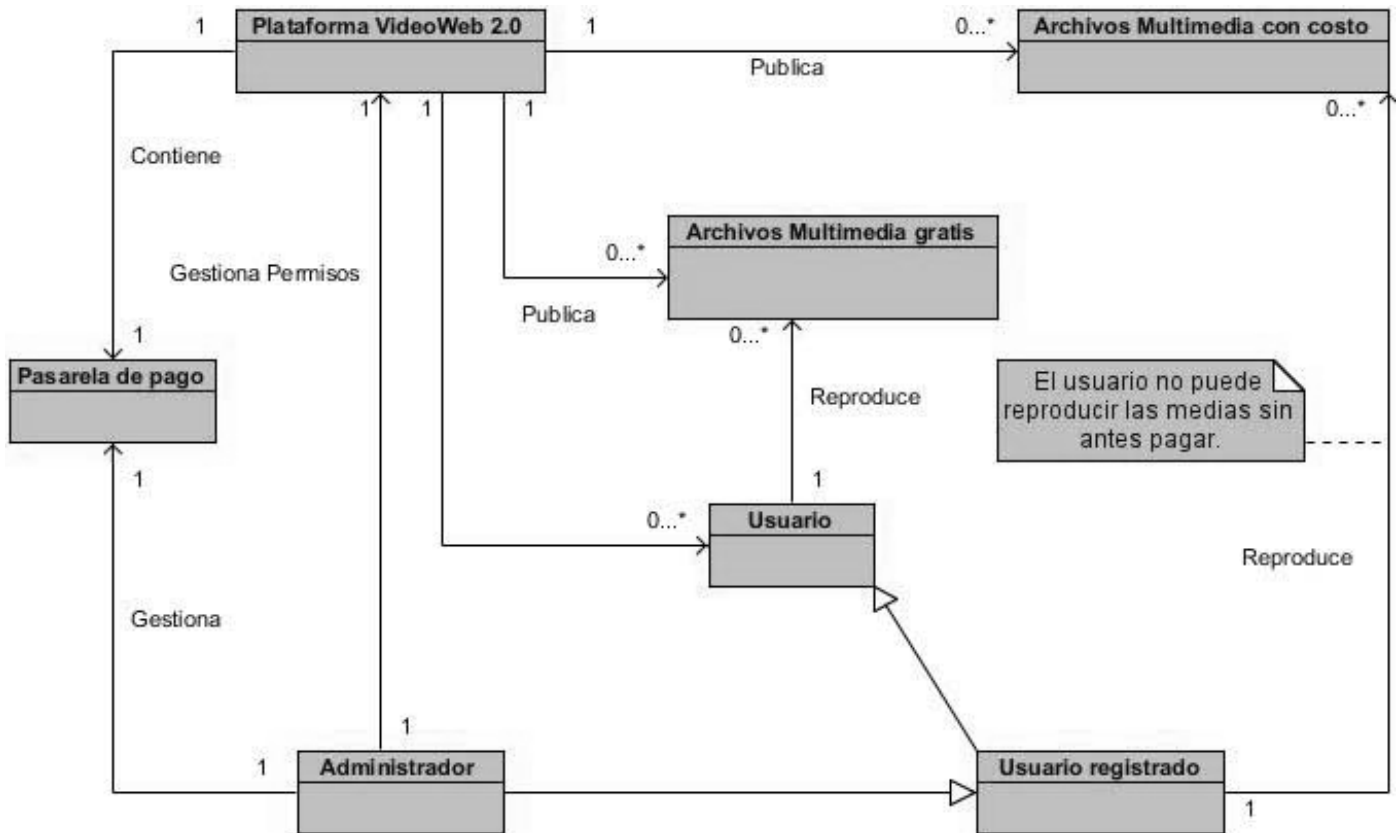


Figura 1: Modelo del dominio

### 2.3 Especificación de requisitos

La Especificación de Requisitos de Software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Además de los casos de uso, la ERS también contiene requisitos no funcionales (o complementarios). Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación. (Raymond Turner, 2005)

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre qué debe y qué no debe hacer el sistema. (Adrián Botta, 2010)

La captura de requisitos es la actividad en la que un grupo especializado extrae, de cualquier fuente de información disponible (documentos, aplicaciones existentes, entrevistas, etc.), las necesidades de

cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, debido a esto existen un conjunto de técnicas que permiten hacer este proceso de una forma más eficiente y precisa, obteniéndose necesidades y modelos del sistema. (Adelaida Ramírez, 2012)

### 2.3.1 Requisitos funcionales

**RF1: Ejecutar pago:** El módulo debe permitir enviar los datos cifrados al servidor para realizar la transacción y esperar un código respuesta.

Entrada:

- Datos de los formularios asociados al mecanismo de pago utilizado.

Salida:

- Reporte generado del destino de la transacción de acuerdo al código respuesta recibido del servidor.

**RF2: Seleccionar modalidad de pago:** El módulo debe permitir al usuario elegir el mecanismo de pago a utilizar.

Entrada:

- Selección del mecanismo de pago.

Salida:

- Muestra la interfaz correspondiente al mecanismo de pago seleccionado.

**RF3: Generar reporte:** El módulo debe permitir generar un reporte de acuerdo al código de respuesta recibido del servidor.

Entrada:

- Código de respuesta recibido desde el servidor con el destino de la transacción.

Salida:

- Reporte generado con la información relacionada al pago, datos si satisfactorio o razones del error.

**RF4: Exportar reporte:** El módulo debe permitir exportar el reporte generado en formato PDF.

Entrada:

- Reporte generado a partir del código respuesta recibido del servidor.

Salida:

- Impresión del reporte usando el hardware correspondiente.

**RF5: Adicionar paquete de pago:** El módulo debe permitir al administrador adicionar un paquete que el usuario pueda elegir para pagar.

Entrada:

- Código de respuesta recibido desde el servidor con el destino de la transacción.

Salida:

- Reporte generado con la información relacionada al pago, datos si satisfactorio o razones del error.

**RF6: Modificar paquete de pago:** El módulo debe permitir al administrador modificar un paquete que el usuario pueda elegir para pagar.

Entrada:

- Código de respuesta recibido desde el servidor con el destino de la transacción.

Salida:

- Reporte generado con la información relacionada al pago, datos si satisfactorio o razones del error.

**RF7: Eliminar paquete de pago:** El módulo debe permitir al administrador eliminar un paquete que el usuario no desee.

Entrada:

- Código de respuesta recibido desde el servidor con el destino de la transacción.

Salida:

- Reporte generado con la información relacionada al pago, datos si satisfactorio o razones del error.

**RF8: Cambiar nivel de media:** El módulo debe permitir cambiar el nivel de acceso a las medias publicadas.

Entrada:

- Nuevo nivel seleccionado para la media: Se selecciona el nuevo nivel de acceso para la media. No hay oportunidad de error en la entrada.

Salida:

- Mensaje de notificación: Muestra un mensaje de notificación y además la interfaz para seguir realizando operaciones relacionadas a la gestión del nivel de medias.

**RF9: Cambiar estado de usuario:** El módulo debe permitir cambiar los privilegios de un usuario en la aplicación.

Entrada:

- Nuevo estado seleccionado para el usuario: Se selecciona el nuevo estado de privilegios para el usuario. No hay oportunidad de error en la entrada.

Salida:

- Mensaje de notificación: Muestra un mensaje de notificación y además la interfaz para seguir realizando operaciones relacionadas a la gestión del estado de los usuarios.

**RF10: Establecer límite de deuda:** El módulo debe permitir establecer un límite de deuda para el usuario registrado.

Entrada:

- Límite de deuda establecido para el usuario: Se establece el límite de deuda para el usuario. No hay oportunidad de error en la entrada.

Salida:

- Mensaje de notificación: Muestra un mensaje de notificación y además la interfaz para seguir realizando operaciones relacionadas a la gestión del límite de deuda de los usuarios.



### 2.3.2 Requisitos no funcionales

#### Apariencia o Interfaz Externa

- El módulo poseerá una interfaz web sencilla, amigable, lo más atractiva y clara posible para el usuario, además su funcionamiento debe ser de fácil comprensión.

#### Seguridad

- El módulo debe garantizar que la información viaje de manera segura durante todo el proceso.

#### Usabilidad

- El módulo debe garantizar una fácil interacción entre cliente y PC, de tal forma que no haya conflictos de usabilidad entre ambos.
- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

#### Restricciones en el diseño y la implementación

- El componente a desarrollar es un módulo de Drupal por lo que debe ser implementado en PHP.

#### Legales

- Las herramientas utilizadas para el desarrollo del módulo deben estar basadas en software libre.

#### Requisitos de hardware

- [1] Ver Especificación de requisitos v2.0, proyecto VideoWeb 2.0.

### 2.4 Descripción del sistema propuesto

El sistema a desarrollar es un módulo desarrollado sobre el CMS Drupal 7 para el proyecto VideoWeb 2.0, que permitirá el pago online de los usuarios por los servicios brindados en la plataforma, además de gestionar lo referente al acceso a los materiales audiovisuales restringidos y al estado de los usuarios para acceder a estos. Cuenta con un carrito de compra accesible desde cualquier lugar con información asociada al pedido y la posibilidad de elegir diferentes métodos de pago. Aprueba la reproducción o descarga después de haber validado la compra, no pudiendo el usuario acceder a estos vínculos sin haberse efectuado la transacción. Cuenta con la existencia de un banco local el cual fue implementado

en el mismo módulo. También se encarga de la seguridad necesaria para realizar la transacción del pago y facilidades al usuario para conocer el destino de la operación con constancia de la misma.

### 2.4.1 Descripción de actores

Un Actor especifica un rol jugado por un usuario o cualquier otro sistema que interactúa con el sujeto. Los actores pueden representar roles jugados por usuarios humanos, hardware externo, u otros sujetos. Un actor no necesariamente representa una entidad física específica, sino simplemente una faceta particular (es decir, un "rol") de alguna actividad que es relevante a la especificación de sus casos de uso asociados. Así, una única instancia física puede jugar el rol de muchos actores diferentes y, asimismo, un actor dado puede ser interpretado por múltiples instancias diferentes. □ (OMG, 2012)

**Usuario:** Persona que interactúa con la pasarela de pago, permitiendo ejecutar las funcionalidades y principal responsable de estas.

**Administrador:** Usuario registrado con dominio total sobre los estados tanto de los usuarios como de los archivos multimedia. Él propiamente no está vinculado a la realización del pago.

### 2.4.2 Diagramas de casos de uso

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. (Jonsson y otros, 2006)

Los casos de uso pretenden ser herramientas simples para describir el comportamiento del software o de los sistemas. Un caso de uso contiene una descripción textual de todas las maneras que los actores previstos podrían trabajar con el software o el sistema. Los casos de uso no describen ninguna funcionalidad interna del sistema, ni explican cómo se implementará. Simplemente muestran los pasos que el actor sigue para realizar una operación. (Jonsson y otros, 2006)

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Los diagramas de casos de uso

se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

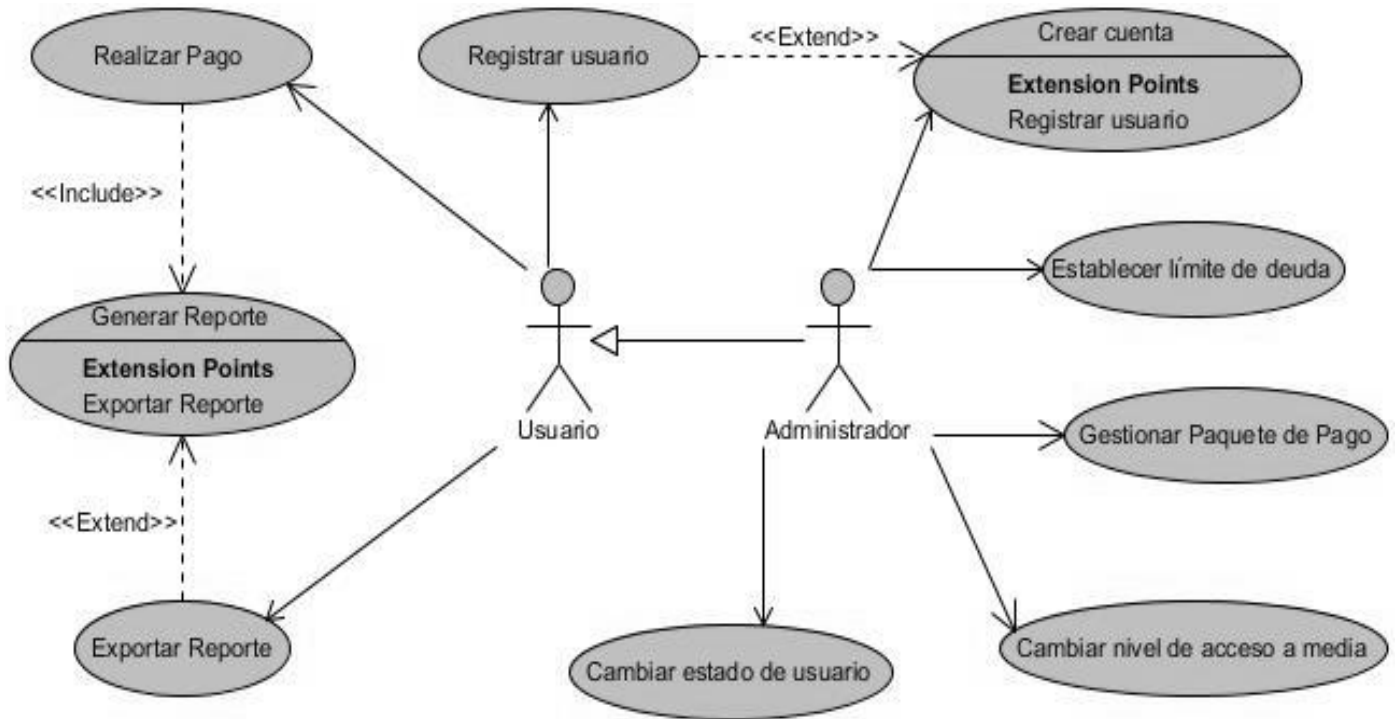


Figura 2: Diagrama de casos de uso del sistema

### 2.4.3 Descripción de los casos de uso del sistema

Tabla 1: Ejecutar pago

<b>Caso de Uso:</b>	Ejecutar pago
<b>Actores:</b>	Usuario
<b>Resumen:</b>	Este caso de uso se inicia cuando el usuario desde la opción Realizar pago, en el menú, pueda llevar a cabo la transacción que le permitiría convertirse en miembro pleno de la aplicación. El caso de uso termina cuando el sistema muestra un informe al usuario con el destino del pago efectuado.
<b>Precondiciones:</b>	Que el usuario esté autenticado.
<b>Referencias</b>	RF1, RF2
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

<p>1. El usuario presiona el vínculo “Realizar pago” de su interfaz de trabajo.</p>	<p>2. El sistema muestra un formulario con un listado de campos relacionados al pago a efectuar, así como un menú desplegable que permite cambiar la modalidad de pago.</p>
<p>3. El usuario llena los campos referentes a su transacción y presiona el botón “Enviar”</p>	<p>4. El sistema valida la información.</p> <p>5. El sistema envía la información al banco y espera por el código de respuesta.</p> <p>6. Se ejecuta el caso de uso “Generar reporte”.</p>
<b>Flujos Alternos</b>	
	<p>4.1 Si existen campos en blanco o incorrectos el sistema muestra un mensaje de error.</p>
	<p>5.1 Si el código de respuesta recibido es diferente de pago exitoso el sistema muestra un mensaje de error.</p> <p>5.2 Se ejecuta el caso de uso “Generar reporte”.</p>
<b>Prototipo de Interfaz</b>	
<b>Poscondiciones</b>	El usuario ejecuta el pago.

**Tabla 2: Generar reporte**

<b>Caso de Uso:</b>	Generar reporte
<b>Actores:</b>	Usuario (No inicializa)
<b>Resumen:</b>	El caso de uso inicia una vez el usuario haya ejecutado el caso de uso “Ejecutar pago” y se reciba un código de respuesta desde el servidor con la información referente al destino del pago. Y finaliza cuando interpreta el código y muestra un reporte al usuario como constancia de la transacción.
<b>Precondiciones:</b>	Que el usuario haya ejecutado el caso de uso “Ejecutar pago”.

<b>Referencias</b>	RF3
<b>Prioridad</b>	Secundario
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema recibe un código de respuesta desde el servidor con información de la solicitud de pago realizada con anterioridad.
	2. El sistema confecciona y muestra un reporte en una página HTML partiendo de la interpretación del código recibido.
<b>Poscondiciones</b>	Se crea un reporte con los datos de la transacción.

**Tabla 3: Exportar reporte**

<b>Caso de Uso:</b>	Exportar reporte
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El caso de uso inicia cuando el usuario acciona el botón "Generar" de la página que contiene el reporte. El caso de uso termina con la generación del documento PDF con la información del reporte.
<b>Precondiciones:</b>	Que se haya ejecutado el caso de uso "Generar reporte"
<b>Referencias</b>	RF4
<b>Prioridad</b>	Secundario
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario acciona el botón "Exportar" del reporte mostrado.	2. El sistema genera un archivo PDF con la información del reporte.
<b>Poscondiciones</b>	Se genera un archivo PDF con la información de la transacción.

**Tabla 4: Gestionar paquetes de pago**

<b>Caso de Uso:</b>	Gestionar paquetes de pago
<b>Actores:</b>	Administrador
<b>Resumen:</b>	Este caso de uso inicia cuando el administrador desde el menú accede al vínculo gestionar paquetes y selecciona alguna de las opciones disponibles,

	finalizando cuando se adiciona, modifica o elimina un paquete, o cuando se cancela cualquiera de las opciones anteriores.
<b>Precondiciones:</b>	Que el administrador se haya autenticado
<b>Referencias</b>	RF5, RF6, RF7
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador accede a la funcionalidad Gestionar Paquetes del sistema en su área de trabajo.	2. El sistema muestra un formulario con un botón para adicionar un nuevo paquete, así como un listado con todos los paquetes existentes y las opciones “Editar” y “Eliminar” para cada uno.
3. El administrador selecciona la opción: <ul style="list-style-type: none"> <li>• Adicionar paquete. Ver Sección “Adicionar paquete”.</li> <li>• Editar. Ver Sección “Editar paquete”.</li> <li>• Eliminar. Ver Sección “Eliminar paquete”.</li> </ul>	
<b>Adicionar paquete</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4. El sistema muestra un formulario con los campos a llenar y las opciones de “Guardar” y “Cancelar”.
5. El administrador llena los campos y presiona el botón “Guardar”.	6. El sistema valida los datos entrados por el administrador.
	7. El sistema almacena los datos del paquete, muestra un mensaje indicando que se ha creado un nuevo paquete y regresa al paso 2 del flujo normal de eventos.
<b>Flujos Alternos</b>	

<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
4.1 El administrador presiona el botón "Cancelar".	4.2 El sistema regresa al paso 2 del flujo normal de eventos.
	6.1 El sistema muestra un mensaje indicando que los datos no son válidos y pasa a la acción 4 del flujo normal de los eventos.
<b>Editar paquete</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4. El sistema muestra los datos del paquete y las opciones de "Guardar" y "Cancelar".
5. El administrador modifica los datos del paquete y presiona el botón "Guardar".	6. El sistema valida los datos entrados por el administrador.
	7. El sistema modifica los datos del paquete, muestra un mensaje indicando que la operación fue realizada y regresa al paso 2 del flujo normal de eventos.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
4.1 El administrador presiona el botón "Cancelar".	4.2 El sistema regresa al paso 2 del flujo normal de eventos.
	6.1 El sistema muestra un mensaje indicando que los datos no son válidos y pasa a la acción 4 del flujo normal de los eventos.
<b>Eliminar paquete</b>	

Acción del Actor		Respuesta del Sistema
		4. El sistema muestra un formulario con un mensaje de confirmación y las opciones de “Eliminar” y “Cancelar”.
5. El administrador selecciona la opción “Eliminar”.		6. El sistema elimina el paquete seleccionado y se muestra un mensaje indicando que se eliminó correctamente, se regresa al paso 2 del flujo normal de eventos.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
4.1 El administrador selecciona la opción “Cancelar”.		4.2 El sistema regresa al paso 2 del flujo normal de eventos.
<b>Poscondiciones</b>	Quedan registrados los datos del paquete.	

**Tabla 5: Cambiar nivel de acceso a media**

<b>Caso de Uso:</b>	Cambiar nivel de acceso a media	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador accede desde el menú a Cambiar Nivel de Media. Finaliza cuando se hallan hechos los cambios y se presione el botón “Guardar” de las características de las medias.	
<b>Precondiciones:</b>	Que el administrador se haya autenticado y que existan archivos multimedia.	
<b>Referencias</b>	RF8	
<b>Prioridad</b>	Crítico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. El administrador accede a la funcionalidad Cambiar nivel de acceso a media en su área de trabajo dentro del sistema.		2. El sistema muestra un listado con todos los archivos multimedia y la opción de cambiar el nivel de acceso al archivo multimedia, además de las opciones “Guardar” y “Cancelar”.



3. El administrador cambia el nivel de acceso al archivo multimedia y presiona el botón “Guardar”.	4. El sistema guarda los cambios y actualiza la base de datos.
<b>Flujos Alternos</b>	
2.1 El administrador no hace ningún cambio y presiona el botón “Cancelar”.	2.2 El sistema vuelve al paso 2 del flujo normal de eventos.
<b>Prototipo de Interfaz</b>	
<b>Poscondiciones</b>	El archivo multimedia cambia su nivel de acceso.

**Tabla 6: Cambiar estado de usuario**

<b>Caso de Uso:</b>	Cambiar estado de usuario	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El caso de uso inicia cuando el administrador accede desde la opción “Cambiar estado de usuario” en el menú de trabajo. El caso de uso termina con el cambio de estado del usuario seleccionado o la cancelación de dicha operación.	
<b>Precondiciones:</b>	Que el administrador se haya autenticado y que existan usuarios.	
<b>Referencias</b>	RF9	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador accede a la funcionalidad Cambiar estado de usuario en su área de trabajo.	2. El sistema muestra un listado con todos los usuarios registrados en la plataforma y la posibilidad de cambiarle el estado, además de las opciones “Guardar” y “Cancelar”.	
3. El administrador cambia el estado del usuario y presiona el botón “Guardar”.	4. El sistema guarda los cambios y actualiza la base de datos.	
<b>Flujos Alternos</b>		
2.1 El administrador no hace ningún cambio y presiona el botón “Cancelar”.	2.2 El sistema vuelve al paso 2 del flujo normal de eventos.	
<b>Prototipo de Interfaz</b>		

<b>Poscondiciones</b>	Quedan registrados los datos del estado de usuario.
-----------------------	---

**Tabla 7: Establecer límite de deuda**

<b>Caso de Uso:</b>	Establecer límite de deuda	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El caso de uso inicia cuando el administrador accede desde la opción “Establecer límite de deuda” en el menú de trabajo. El caso de uso termina cuando el administrador establece el límite de deuda.	
<b>Precondiciones:</b>	Que el administrador se haya autenticado.	
<b>Referencias</b>	RF10	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador accede a la funcionalidad Establecer límite de deuda en su área de trabajo.	2. El sistema muestra un listado con todos los usuarios registrados en la plataforma y la posibilidad de establecerle un límite de deuda, además de las opciones “Guardar” y “Cancelar”.	
3. El administrador establece el límite de deuda del usuario y presiona el botón “Guardar”.	4. El sistema guarda los cambios y actualiza la base de datos.  5. El sistema muestra al usuario el límite de deuda establecido.	
<b>Flujos Alternos</b>		
2.1 El administrador no establece el límite de deuda y presiona el botón “Cancelar”.	2.2 El sistema vuelve al paso 2 del flujo normal de eventos.	
<b>Prototipo de Interfaz</b>		
<b>Poscondiciones</b>	Quedan registrados los límites de deudas.	

## 2.5 Modelo de análisis

Es un modelo que describe la realización de casos de uso, clases de análisis y paquetes de análisis y sirve como una abstracción. También es conocido como el modelo conceptual.

A continuación se reflejan los diagramas de clases de análisis correspondiente a los casos de uso descritos anteriormente.

2.5.1 Diagrama de clases del análisis

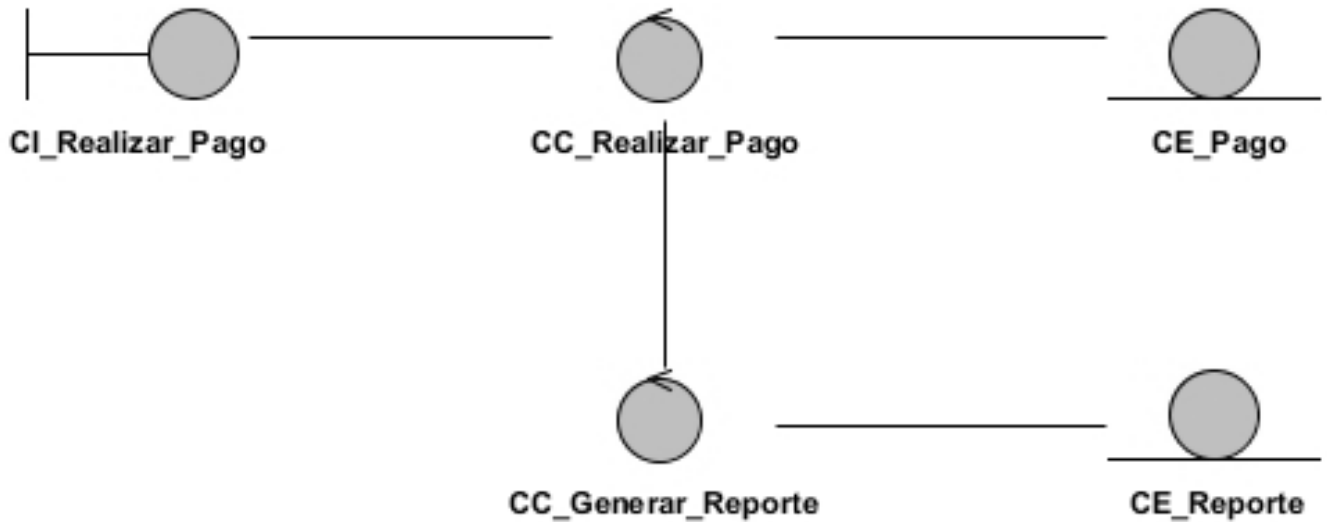


Figura 3: DCA Realizar pago y Generar reporte



Figura 4: DCA Cambiar Nivel de Archivo Multimedia

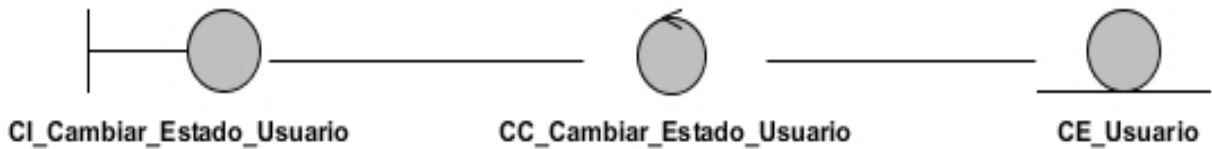
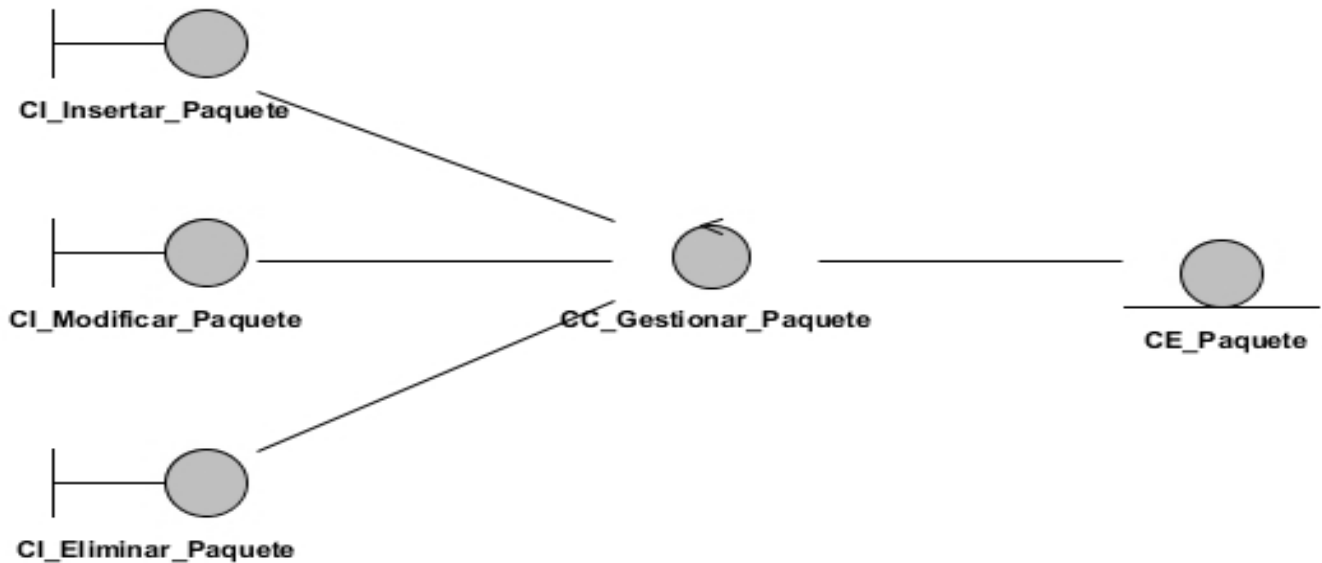


Figura 5: DCA Cambiar estado de usuario



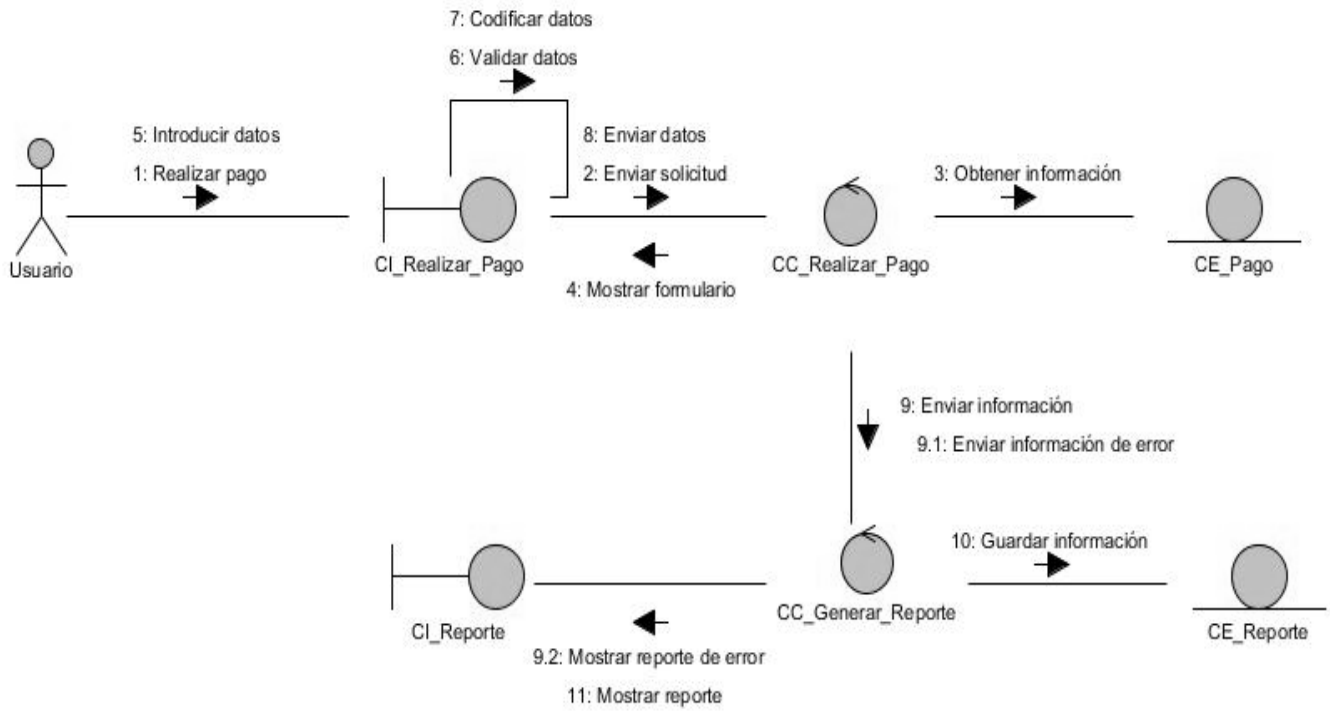
*Figura 6: DCA Gestionar paquetes de pago*

## 2.5.2 Diagramas de interacción

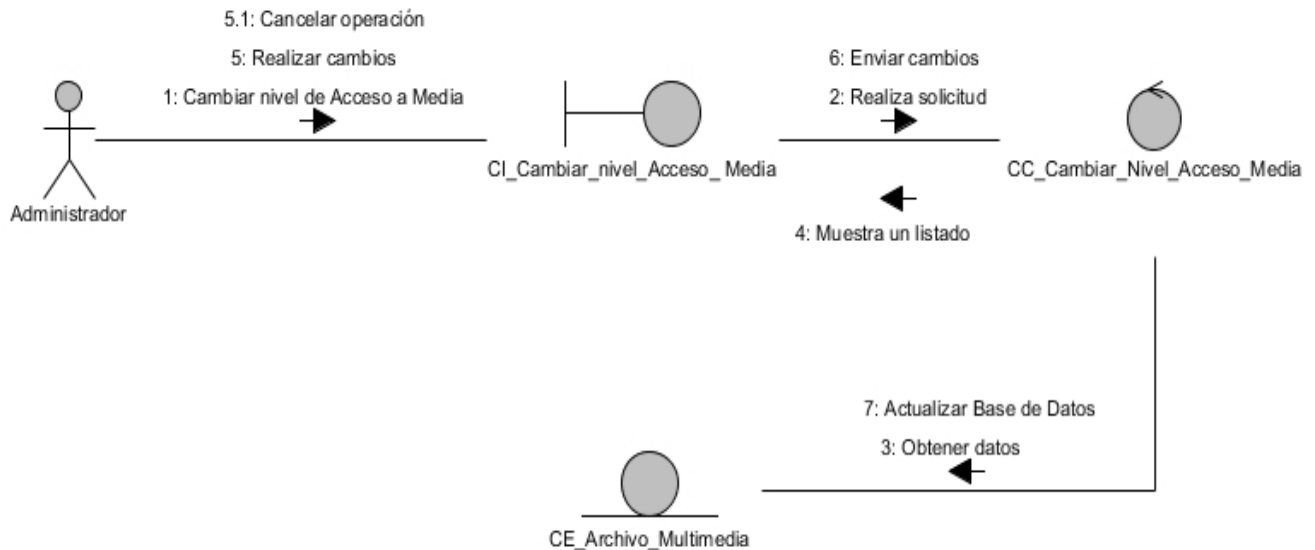
Los Diagramas de interacción son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento. Se deberán usar diagramas de interacción si se quiere analizar el comportamiento de un grupo de objetos en un mismo caso de uso. Los diagramas de interacción muestran cierto número de ejemplos de objetos y los mensajes que se pasan entre estos objetos dentro del caso de uso. Hay dos tipos de diagramas de interacción: diagramas de colaboración y diagramas de secuencia. Los diagramas de secuencia dan una visión del orden temporal (qué pasa primero y qué pasa después), mientras que los diagramas de colaboración sólo proporcionan una visión de las interrelaciones (Jose.L Rodríguez, 2009)

### Diagramas de colaboración

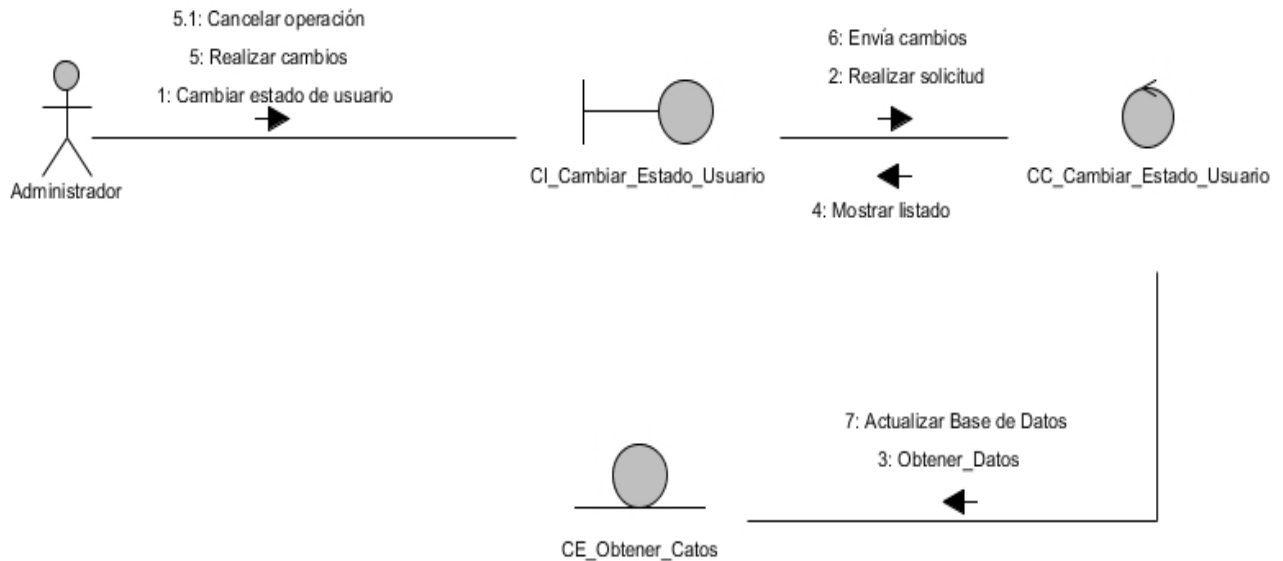
Un diagrama de colaboración muestra cómo las instancias específicas de las clases trabajan juntas para alcanzar un objetivo común. En cierta forma, en un diagrama de colaboración se detallan las asociaciones que se muestran en un diagrama de clases, describiendo el intercambio de mensajes entre objetos y las relaciones entre los objetos, sin tomar en cuenta la oportunidad o la dimensión temporal de dichas relaciones. (Jose.L Rodríguez, 2009)



**Figura 7: DC Realizar pago y Generar reporte**



**Figura 8: DC Cambiar nivel de media**



*Figura 9: DC Cambiar estado de usuario*

### Diagramas de secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes (llamadas a métodos) que intercambian, ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical llamada línea de vida. Los mensajes se representan mediante flechas horizontales entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren. (Jose.L Rodríguez, 2009)

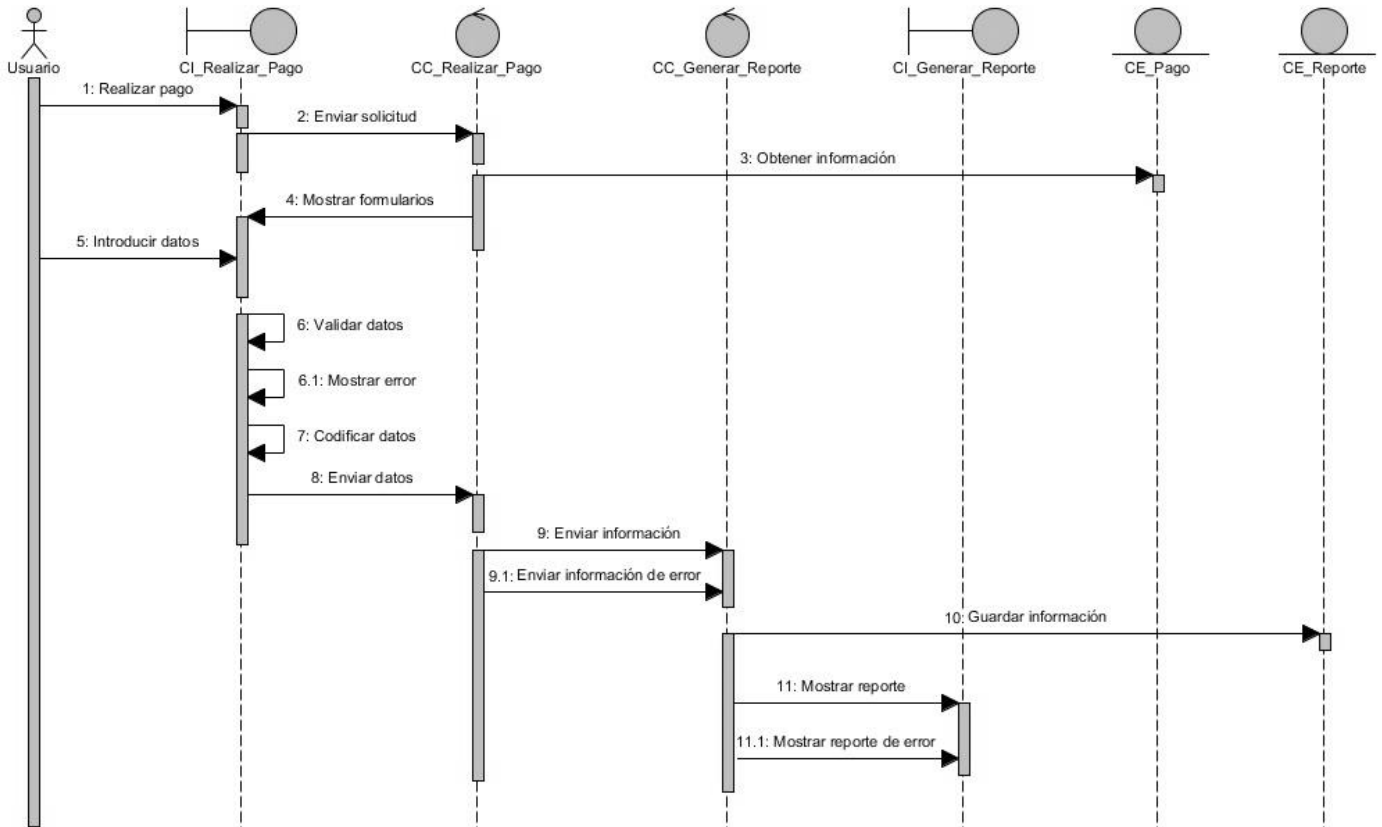


Figura 10: DS Realizar pago y Generar reporte

## 2.6 Modelo de diseño

El modelo de diseño debe definir una solución software que satisfaga de modo efectivo y eficiente los requisitos especificados en el análisis, y al hacer esto el modelo incorporará nuevos artefactos (nuevas clases, nuevos atributos y operaciones para las clases, etc.) y tendrá en cuenta la plataforma tecnológica concreta sobre la que debe construirse el sistema software. (Génova y otros, 2006)

### 2.6.1 Diagramas de Clases del Diseño

El Diagrama de Clases de Diseño describe gráficamente las especificaciones de las Clases de Software y las Interfaces en una aplicación. Describe la estructura de un sistema mostrando: Clases, asociaciones y atributos. Interfaces, con sus operaciones y constantes. Información sobre los tipos de atributos. Navegabilidad. Dependencia. (Juan Pablo Bustos, 2011)

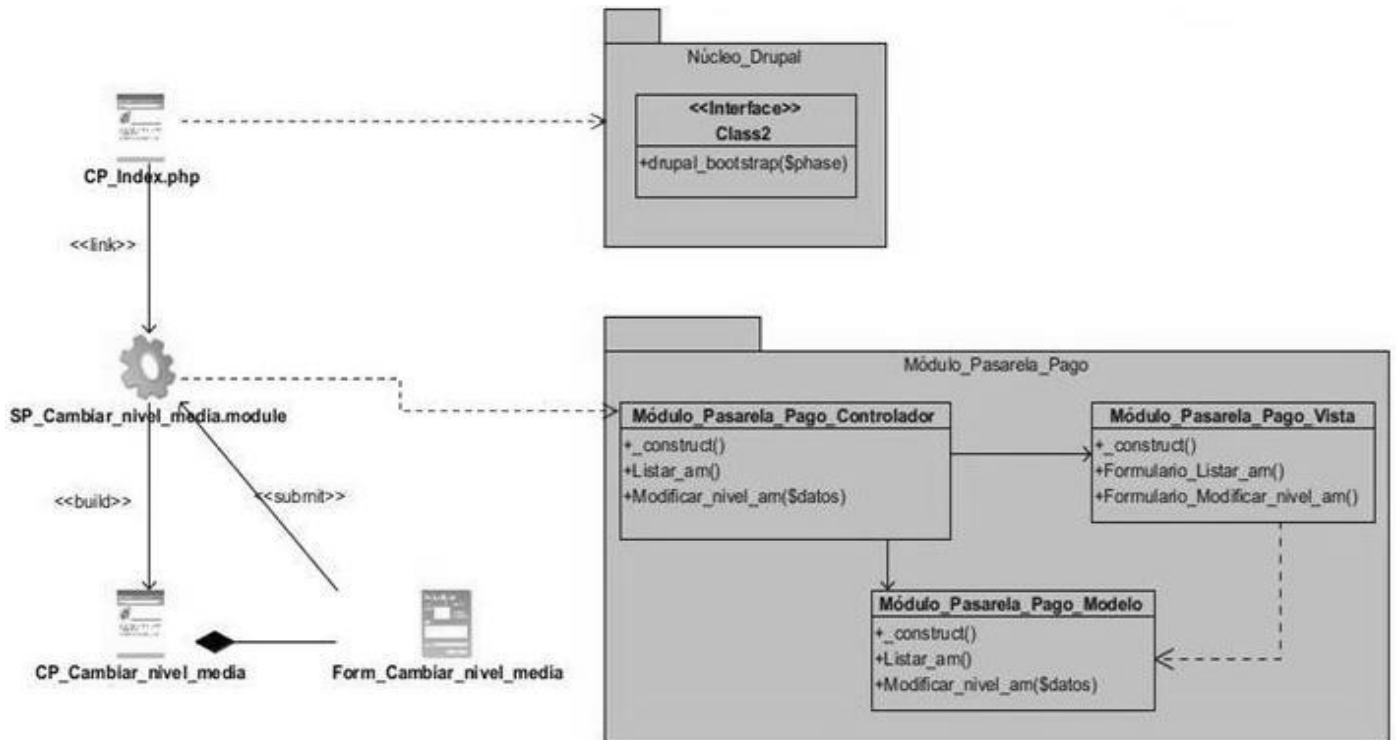


Figura 11: DCD Cambiar nivel de archivo multimedia

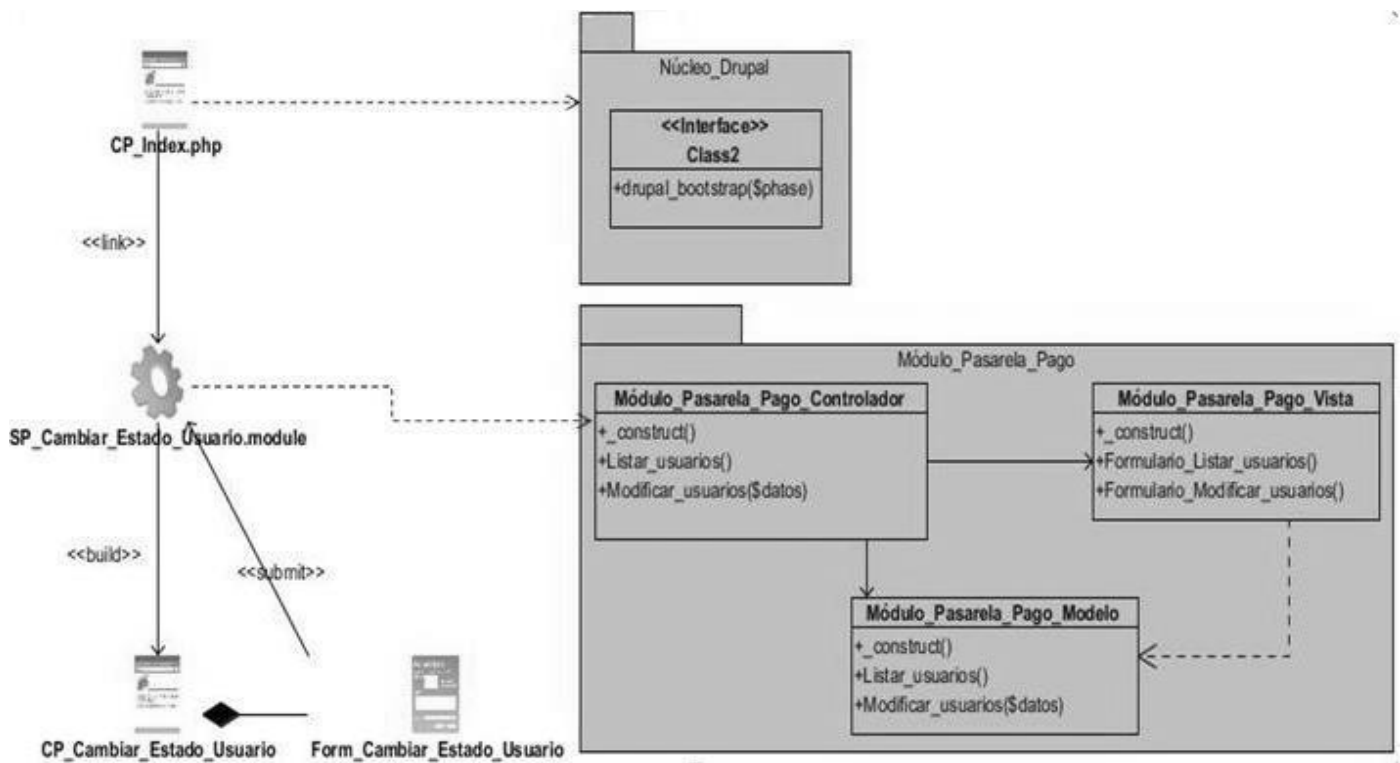


Figura 12: DCD Cambiar estado de usuario



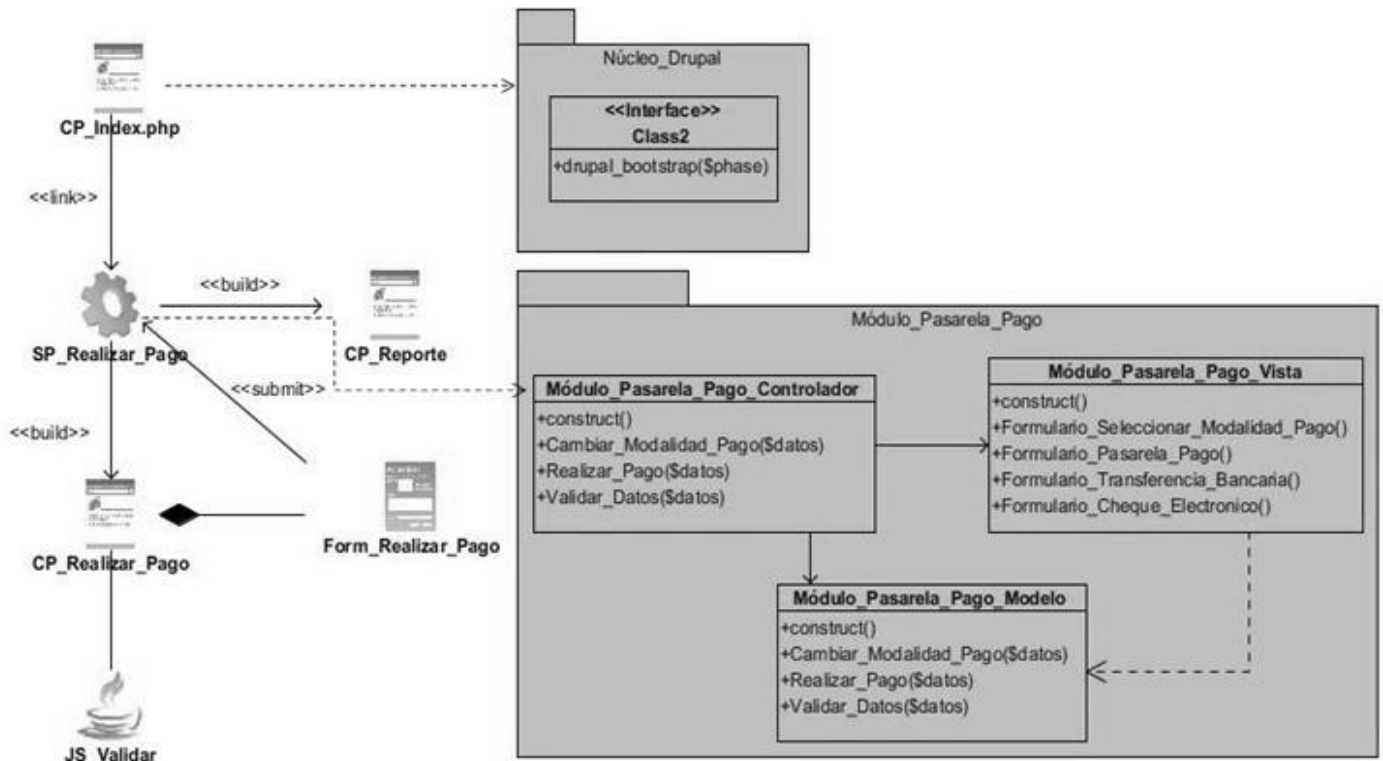


Figura 13: DCD Realizar pago y generar reporte

## 2.7 Patrones arquitectónicos

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. (Pressman, 2011)

La estructura principal del módulo está basada en el estilo arquitectónico de llamada y retorno, dentro de este estilo se estará haciendo uso del patrón arquitectónico MVC.

### 2.7.1 Modelo Vista Controlador

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de

datos dinámicos a la página, el modelo es el sistema de gestión de base de datos y la lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista. (Sebastián, 2010)

### **Modelo:**

Esta es la representación específica de la información con la cual el sistema opera.

Define las reglas de negocio, la funcionalidad del sistema.

Lleva un registro de las vistas y controladores del sistema.

### **Controlador:**

Recibe los eventos de entrada, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

### **Vista:**

Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

## **2.8 Patrones de diseño**

El CMS Drupal, en su versión 7, utiliza recursos de la programación orientada a objetos como el manejo de objetos para datos, herencia, encapsulamiento, abstracción y polimorfismo. Las características estructurales que lo definen evidencian la aplicación de patrones de diseño.

Seguidamente se describen ejemplos de patrones GOF (dadas sus siglas en inglés) presentes en el sistema y como se evidencian.

### **Singleton (Instancia única)**

Se evidencia si se asocian los módulos y los temas como objetos, cada uno de ellos en sí no almacena datos, sino que representan un conjunto de funciones que extienden el núcleo para añadirle funcionalidades por separado. Esta asociación se puede considerar un módulo como una clase con una instancia única. Los módulos que se utilizan para el desarrollo, tal es el caso del Node, Field API y el Views son un ejemplo de como ellos extienden el núcleo de manera independiente cada uno utilizando las funcionalidades o hooks (ganchos) que definen e implementan.

### **Decorator (Decorador)**

El sistema de hooks de Drupal 7, representa la evidencia de aplicación de este patrón. De esta forma se brinda la flexibilidad de que nuevos módulos puedan modificar el comportamiento del núcleo en cuanto al tratamiento de los datos y en cada uno de los eventos del sistema. Este comportamiento podría ser imitado por el uso de decoradores, por ejemplo con el hook\_node\_load definido por sistema de nodos de Drupal, un nuevo módulo puede implementar ese hook para insertar o modificar los datos de los nodos a la hora de ser cargados de la base de datos. Un caso de evidencia de este patrón en la solución está dado por la implementación de los hooks de validación de datos o campos por el módulo Field Validation.

### **Observer (Observador)**

Este patrón se evidencia con el mismo comportamiento descrito anteriormente para la aplicación del patrón decorator, ya que por ejemplo, los módulos que implementan un hook determinado por evento de inserción o actualización de un determinado nodo, son declarados como observadores de los mismos. Este patrón se ilustra en la implementación del módulo Field API que es invocado en el momento de realizar cualquier operación de almacenamiento o recuperación de información sobre las entidades a las cuales son adjuntados, tal es el caso de los productos de tipo Paquete.

### **Bridge (Puente)**

La implementación de la capa de abstracción de datos de Drupal sigue el patrón bridge. Cada módulo es programado de manera que sea independiente del motor de base de datos que utiliza el sistema. Esto se logra gracias a la capa de abstracción de base de datos, sobre la que se puede desarrollar otras nuevas siguiendo la API definida, para añadir soporte a nuevos sistemas gestores de base de datos sin necesidad de modificar el código de los módulos implementados. La capa de abstracción de base de datos para PostgreSQL que utiliza el sistema para la comunicación con la base de datos es un caso de evidencia de este patrón.

### **Chain of responsibility (Cadena de responsabilidades)**

La evidencia de este patrón lo representa el sistema de menús de Drupal. En cada petición de una ruta determinada, este determina si hay algún módulo para responder, o si el usuario tiene acceso al recurso solicitado y qué función se debe llamar para procesar la misma. En este proceso se transmite el mensaje de la petición por cada uno de los componentes que se encuentran inmersos. De esta forma se continúa

la cadena hasta que un módulo atienda la petición, deniegue el acceso o la cadena se agote. Los enlaces de menú definidos por las vistas de reportes y de gestión creadas en el sistema evidencian el uso de este patrón, ya que definen al módulo Views como controlador de estas peticiones, para los casos de denegación de accesos indebidos y para los casos en que se le muestra la información a los usuarios que lo requieran y posean los permisos específicos.

### **2.9 Conclusiones**

Con la culminación de este capítulo se puede concluir que:

El Modelo de Dominio realizado a partir de los procesos identificados permitió conocer todos los términos y conceptos presentes en el entorno, los cuales fueron descritos y especificados en un glosario de términos y representados en un diagrama de clases, el cual permitió comprender mejor el funcionamiento e interrelación de los mismos.

Mediante la realización de la disciplina Levantamiento de Requisitos, se consiguió identificar las funcionalidades que el módulo de pago para la Plataforma VideoWeb 2.0 debe brindar y las restricciones sobre las que va a operar.

La conformación del Diagrama de Casos de Uso del Sistema permitió traducir los requisitos funcionales en interacciones de los actores con el sistema, lo que permitió obtener una visión de cómo sería la interacción de los roles identificados con las funcionalidades que brindará el sistema.

La descripción detallada del flujo de eventos de cada caso de uso en un lenguaje formal y entendible permitió comprender en detalle los procesos incluidos en el cumplimiento de los objetivos trazados para cada caso de uso.

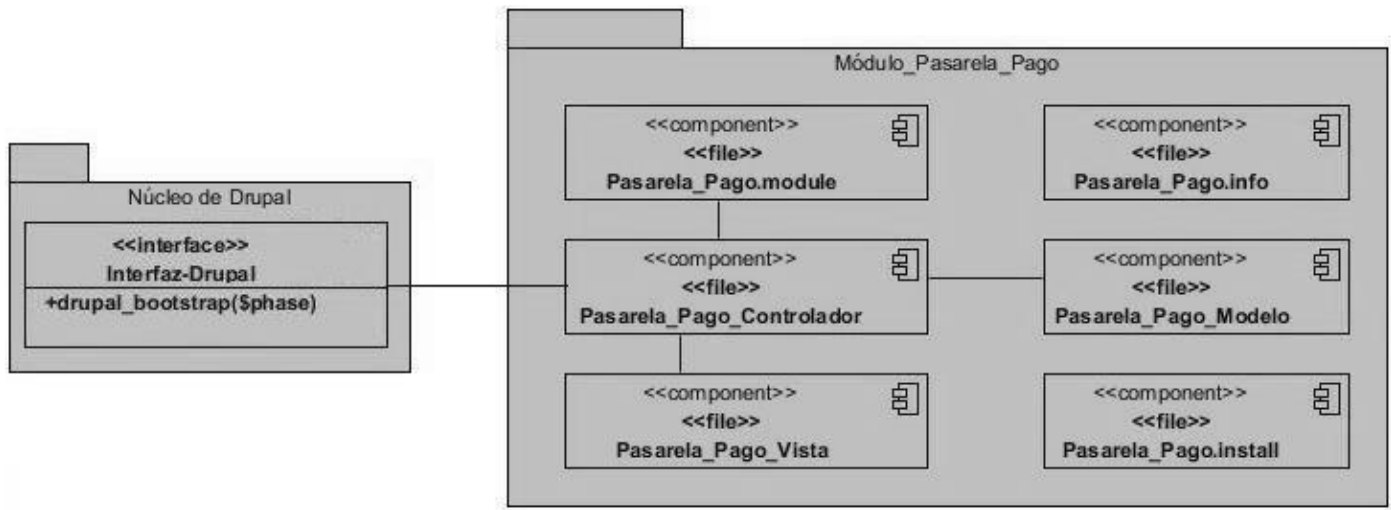
### **CAPITULO 3: Implementación y prueba.**

#### **3.1 Introducción**

En este capítulo se muestra el flujo de trabajo de implementación, el cual forma parte de la fase de construcción, durante este flujo se describe como los elementos del modelo de diseño se implementan en términos de componentes y cómo estos se organizan en el modelo de despliegue. Además se realizan pruebas de caja negra al módulo desarrollado para comprobar el funcionamiento del mismo y detectar y corregir las posibles no conformidades que se detecten.

#### **3.2 Diagrama de componentes**

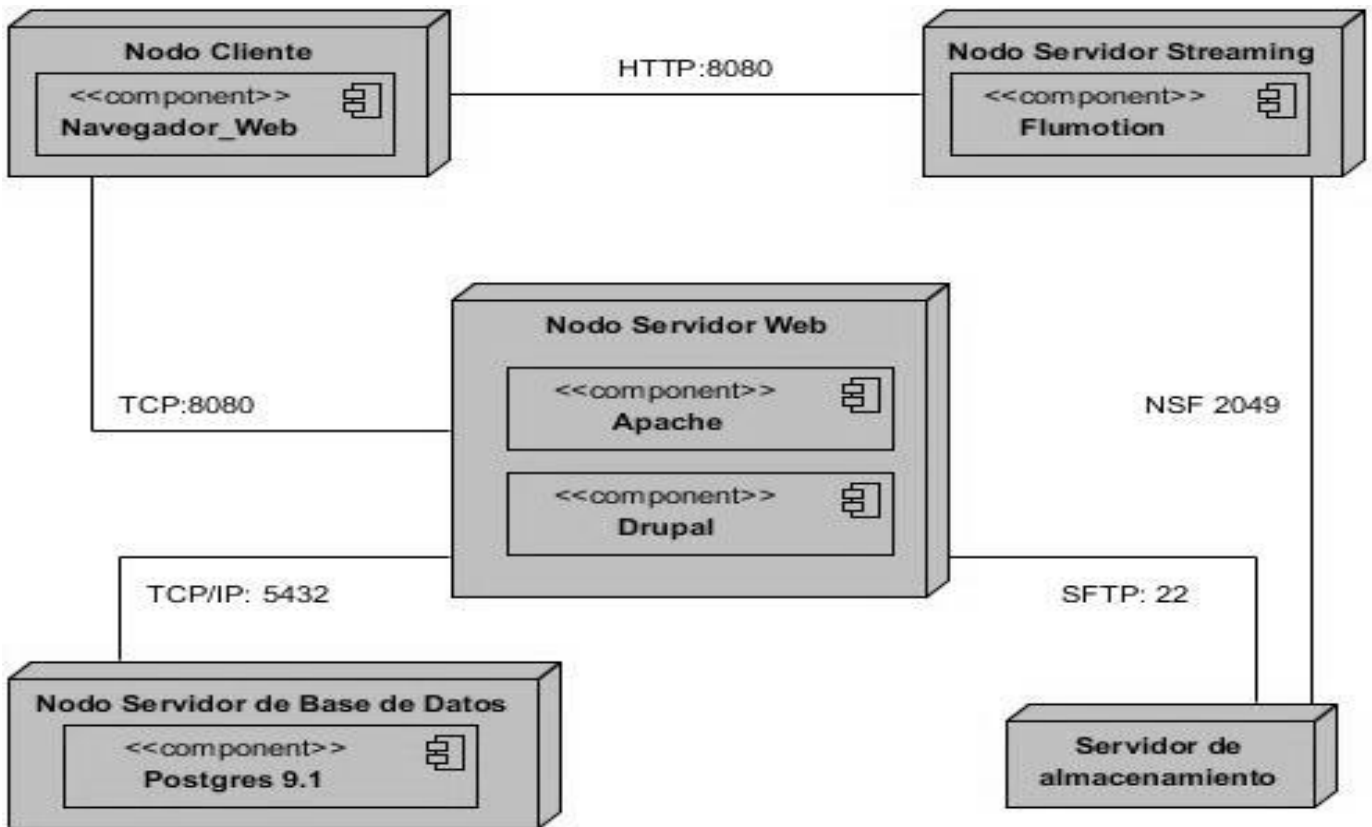
Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables, ilustran las piezas del software, controladores embebidos, etc. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema, es decir para describir la vista de implementación estática de un sistema. Los diagramas de componentes se relacionan con los diagramas de clases, ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones pero un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema. (Elisa Arizaca Ramirez, 2009)



**Figura 14: Diagrama de componente**

### 3.3 Diagrama de despliegue

Un diagrama de despliegue muestra la configuración de nodos que participan en la ejecución y de los componentes que residen en ellos. Describen la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos y componentes de software. Describen la topología del sistema: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Es un objeto físico en tiempo de ejecución que representa un recurso computacional que generalmente tiene memoria y capacidad de procesamiento. Los nodos pueden contener objetos, instancias, instancias del componente. Un nodo representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes. Los nodos se conectan mediante asociaciones de comunicación. Estas asociaciones indican algún tipo de ruta de comunicación entre los nodos. El tipo de comunicación se identifica con un estereotipo que indica el protocolo de comunicación o la red. Los nodos intercambian objetos o envían mensajes a través de esta ruta. (Tutorial UML 2. 2007)



*Figura 15: Diagrama de despliegue*

### 3.4 Pruebas de software

Durante el flujo de trabajo de pruebas se verifica el resultado de la implementación probando la estructura, tanto en la construcción interna como intermedia, así como las versiones finales del sistema a ser entregado (Jacobson, y otros, 2000). Concretamente los objetivos de las pruebas son:

- Planificar las pruebas
- Diseñar e implementar las pruebas
- Aplicar las pruebas y manejar los resultados

Este flujo es uno de los más importantes propuestos por la metodología de desarrollo de software utilizada (RUP), pues verifica y valida el trabajo durante todos los flujos anteriores. Además las pruebas se hacen con el objetivo de encontrar deficiencias en el software para entregar el sistema al usuario final sin errores. Las pruebas son actividades en las cuales un sistema o componente es ejecutado bajo

condiciones o requerimientos especificados permitiendo que los resultados sean observados y registrados. Estas son un elemento crítico para la garantía de la calidad del software. En el ámbito de la Ingeniería de Software existe un método fundamental de pruebas: las pruebas de caja negra.

### 3.4.1 Pruebas de caja negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Después se derivan conjuntos de condiciones de entrada que utilicen todos los requisitos funcionales de un programa.

Las pruebas de caja negra pretenden encontrar estos tipos de errores:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

### Casos de prueba

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de pruebas se pueden derivar de los casos de uso del sistema o de la realización de estos en el modelo de diseño, permitiendo así validar los requerimientos funcionales del sistema. Para el desarrollo de las pruebas al módulo se desarrollaron cuatro casos de



pruebas, uno por cada caso de uso del sistema. Los casos de pruebas desarrollados se muestran a continuación:

### Caso de Prueba: Gestionar paquetes de pago

Descripción: Este caso se inicia cuando el administrador desde el menú accede al vínculo gestionar paquetes y selecciona alguna de las opciones disponibles, finalizando cuando se adiciona, modifica o elimina un paquete, o cuando se cancela cualquiera de las opciones anteriores.

Condiciones de ejecución: Que el administrador se haya autenticado.

**Tabla 8: SC\_Gestionar Paquete de Pago.**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Adicionar Paquete de Pago.	EC1.1: Adicionar Paquete satisfactoriamente.	El sistema muestra un formulario con los campos a llenar, el Administrador llena los campos y los envía al sistema que los valida y almacena los datos del paquete, muestra un mensaje indicando que se ha creado un nuevo paquete satisfactoriamente.
	EC1.2: Adicionar Paquete falla.	El sistema muestra un formulario con los campos a llenar, el Administrador llena los campos y los envía al sistema que los valida y muestra un mensaje de alerta indicando que estos no son correctos.
SC2: Editar Paquete de Pago.	EC2.1: Editar Paquete satisfactoriamente.	El sistema muestra los datos del paquete, el Administrador modifica los datos del paquete y los envía al sistema que valida los datos entrados y modifica la información del paquete, mostrando un mensaje indicando

		que la operación fue realizada satisfactoriamente.
	EC2.2: Editar Paquete falla.	El sistema muestra los datos del paquete, el Administrador modifica los datos del paquete y los envía al sistema que valida los datos entrados y muestra un mensaje de alerta indicando que estos no son correctos.
SC3: Eliminar Paquete de Pago.	EC3.1: Eliminar Paquete satisfactoriamente.	El sistema muestra un formulario con un mensaje de confirmación, el Administrador selecciona la opción "Eliminar" y el sistema elimina el paquete seleccionado mostrando un mensaje indicando que se eliminó correctamente.
	EC3.2: Eliminar Paquete falla.	El sistema muestra un formulario con un mensaje de confirmación, el Administrador selecciona la opción "Eliminar" y el sistema muestra un mensaje de alerta indicando que ocurrió un error durante el proceso.

**Tabla 9: Descripción de las variables**

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	nombre_paquete	Campo de texto	No	Campo de texto que contiene el nombre, puedo incluir números.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

2	descripción	Campo de texto	Si	Breve descripción del paquete, es opcional.
3	cant_dias	Campo de texto	No	Número entero, contiene los días Premium a obtener.
4	precio	Campo de texto	No	Costo total de la transacción.

### Secciones a revisar:

**Tabla 10: SC1\_ Adicionar Paquete de Pago**

Id escenario	nombre_paquete	descripción	cant_dias	precio	Respuesta del sistema	Resultado de la prueba
EC1.1: Adicionar Paquete satisfactoriamente	V "3x2, 5x3, paq1"	N/A	V "Número"	V "Número"	El sistema almacena los datos del paquete y muestra un mensaje indicando que ha ocurrido satisfactoriamente.	Satisfactoria
EC1.2: Adicionar Paquete falla.	I	N/A	I	V	El sistema muestra un mensaje de alerta indicando que los datos no son correctos.	Satisfactoria
	V	N/A	V	I		
	I	N/A	V	V		
	V	N/A	I	I		
	I	N/A	I	I		

**Tabla 11: SC3\_Eliminar Paquete de Pago**

Id escenario	Id_paquete	Respuesta del sistema	Resultado de la prueba
EC3.1: Eliminar Paquete satisfactoriamente.	V	El sistema elimina el paquete seleccionado mostrando un mensaje indicando que lo hizo correctamente.	Satisfactoria
EC3.2: Eliminar Paquete falla.	I	El sistema muestra un mensaje de alerta indicando que ocurrió un error durante el proceso.	Satisfactoria

### Caso de Prueba: Ejecutar pago

Descripción: Este caso de uso se inicia cuando el usuario desde la opción Realizar pago, en el menú, pueda llevar a cabo la transacción que le permitiría convertirse en miembro pleno de la aplicación. El caso de uso termina cuando el sistema muestra un informe al usuario con el destino del pago efectuado.

Condiciones de ejecución: Que el usuario esté autenticado.

**Tabla 12: SC\_Ejecutar pago**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Ejecutar pago.	EC1.1: Ejecutar pago satisfactoriamente.	El sistema muestra un formulario con un listado de campos relacionados al pago a efectuar, así como un menú desplegable que permite cambiar la modalidad de pago, el usuario llena los campos referentes a su transacción y el sistema valida la información y la codifica para que viaje de forma segura hasta el servidor, envía la información encriptada al banco y espera por el código de respuesta.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

	EC1.2: Ejecutar pago falla.	El sistema muestra un formulario con un listado de campos relacionados al pago a efectuar, así como un menú desplegable que permite cambiar la modalidad de pago, el usuario llena los campos referentes a su transacción y el sistema valida la información, si existen campos en blanco o incorrectos el sistema muestra un mensaje de error.
--	-----------------------------	---

**Tabla 13: Descripción de variables**

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	tarjeta_crédito	Campo de texto	No	Número de 12 dígitos asociado a la tarjeta de crédito a utilizar.
2	código_seguridad	Campo de texto	No	Número de 4 dígitos que aparece en dicha tarjeta.
3	fecha_caducidad	Selector de fecha	No	Fecha en que expira la tarjeta de crédito, aparece en ella.

**Secciones a revisar:**

**Tabla 14: SC1\_Ejecutar pago**

Id escenario	tarjeta_crédito	código_seguridad	fecha_caducidad	Respuesta del sistema	Resultado de la prueba
--------------	-----------------	------------------	-----------------	-----------------------	------------------------

EC1.1: Ejecutar pago satisfactoriamente .	V " Número, 12 dígitos"	V " Número, 4 dígitos"	V "Fecha"	El sistema codifica la información para qué viaje de forma segura hasta el servidor, la envía al banco y espera por el código de respuesta.	Satisfactoria
EC1.2: Ejecutar pago falla.	I	V	V	El sistema muestra un mensaje de error indicando que hay campos en blanco o incorrectos.	Satisfactoria
	V	I	V		
	V	V	I		
	I	I	I		

### Caso de Prueba: Cambiar estado de usuario

Descripción: El caso de uso se inicia cuando el administrador accede desde la opción "Cambiar estado de usuario" en el menú de trabajo. El caso de uso termina con el cambio de estado del usuario seleccionado o la cancelación de dicha operación.

Condiciones de ejecución: Que el administrador se haya autenticado y que existan usuarios.

**Tabla 15: SC1\_ Cambiar estado de usuario.**

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
----------------------	--------------------------	---------------------------------

SC1: Cambiar estado de usuario.	EC1.1: Ejecutar pago satisfactoriamente.	El sistema muestra un listado con todos los usuarios registrados en la plataforma, el administrador selecciona el usuario que desea modificar y se muestra las propiedades del usuario seleccionado, incluyendo su estado. El administrador cambia el estado del usuario y presiona el botón "Guardar", el sistema guarda los cambios y actualiza la base de datos.
	EC1.2: Cancelar cambio estado de usuario.	El administrador no hace ningún cambio y presiona el botón "Cancelar".

**Tabla 16: Descripción de variables**

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	estado_usuario	checkbox	N/A	Selector que determina si el usuario es Premium o no.
2	fecha_fin	Selector de fecha	No	Fecha hasta la cual el usuario será considerado Premium.

**Secciones a revisar:**

**Tabla 17: SC1\_Cambiar estado de usuario**

Id escenario	estado_usuario	fecha_fin	Respuesta del sistema	Resultado de la prueba
--------------	----------------	-----------	-----------------------	------------------------

<p>EC1.1: Cambiar estado de usuario satisfactoriamente.</p>	<p>N/A</p>	<p>N/A</p>	<p>El sistema guarda los cambios, actualiza la base de datos y muestra un mensaje satisfactorio.</p>	<p>Satisfactoria</p>
<p>EC1.2: Cancelar cambio estado de usuario.</p>	<p>N/A</p>	<p>N/A</p>	<p>El sistema vuelve al estado anterior del flujo normal de eventos.</p>	<p>Satisfactoria</p>

### 3.4.2 Resultados de las pruebas

Fueron probados tres casos de uso que tienen entrada de datos que pueden ser evaluadas. Para cada requisito se tuvo en cuenta los diferentes escenarios que pudieran existir, los cuales implican cada una de las posibles interacciones del usuario o combinaciones de situaciones posibles con el fin de evaluar el comportamiento del sistema ante cada funcionalidad. El resultado de cada prueba coincide con la especificación de requisito lo que demuestra la correcta implementación de los mismos.

### 3.5 Conclusiones

Como resultado de este capítulo se obtuvo el diagrama de componentes el cual muestra la estructura física que tiene el componente de software. Se logró de esta manera obtener el código del módulo pasarela de pago, basándose en el análisis y diseño realizado en el capítulo anterior, además de los diagramas de componentes. Además para validar la completitud de respuesta a los requerimientos identificados se realizaron pruebas de caja negra al módulo, quedando elaborado de esta forma el Modelo de Pruebas, conformado por tres casos de prueba. Los resultados de las pruebas fueron satisfactorios, lo que permitió



evaluar el comportamiento del sistema y su respuesta ante las solicitudes del usuario, confirmando la validez de la solución propuesta.

### CONCLUSIONES GENERALES

La realización de la investigación arrojó los siguientes resultados que dan cumplimiento a los objetivos propuestos:

- Se realizó un estudio de los principales conceptos relacionados a los mecanismos de pago electrónicos existentes, demostrando la necesidad de implementar un módulo para la plataforma VideoWeb 2.0 que permita reducir el esfuerzo que representa cobrar por los servicios de reproducción y descarga en la misma con la automatización del proceso.
- Para desarrollar el sistema se caracterizaron las herramientas y tecnologías lo que permitió sentar las bases para el posterior trabajo con las mismas.
- Se definieron los requisitos funcionales, los no funcionales y los casos de uso con sus descripciones permitiendo definir las características del sistema y sirviendo de guía para los flujos y fases posteriores.
- Se realizó el análisis y diseño de la solución describiendo el funcionamiento de la misma además del flujo de eventos e intercambio de información entre los componentes, elaborando también los diagramas necesarios en los que apoyarse para la posterior implementación del módulo.
- Para comprobar la calidad y correcto funcionamiento del sistema, se diseñaron y ejecutaron casos de prueba, los que arrojaron resultados satisfactorios, demostrando el cumplimiento de los requerimientos funcionales establecidos en la fase inicial del proceso de desarrollo del componente.
- El componente implementado como resultado de la investigación realizada será de gran utilidad en el proceso de automatización del cobro de los servicios brindados en la plataforma VideoWeb 2.0 y le permitirá insertarse en otras áreas relacionadas con los negocios electrónicos reduciendo el esfuerzo que este representa, traducido a menos personal necesario, ahorro en los costos y en el tiempo de cobro.

### RECOMENDACIONES

Una vez cumplidos los objetivos del presente trabajo y en correspondencia con los resultados obtenidos, los cuales se expusieron en el propio documento, se recomienda como trabajos futuros:

- ❖ Vincular el módulo desarrollado a un sistema de seguridad que permita denegar acceso a usuarios no *premium* a contenido de pago.
- ❖ Probar el funcionamiento del módulo con un banco real que proporcione los servicios web adecuados para utilizar todos los mecanismos de pagos implementados e incluir otros nuevos.
- ❖ Aplicar nuevas formas de cobro de servicios que representen un aumento en las prestaciones a los usuarios como pueden ser: promociones, por puntos, tarjetas de recarga, etc.

### REFERENCIA BIBLIOGRÁFICA

**Agil Unified Process. 2012** ambyssoft.com. [En línea] [Citado el: 14 de 12 de 2012.] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.

**Agustín Dondo. 2011** programacion.com. [En línea] [Citado el: 12 de 12 de 2012.] <http://www.programacion.com/%20php/articulo/porquephp>.

**Ambyssoft. 2012** [En línea] Agil Unified Process. [Citado el: 12 de 12 de 2012.] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.

**Artículo29-A. 2012** aportacionesfiscales.com. Para los efectos del artículo 29-A, último párrafo del CFF, 1.2.10.18. [En línea] [Citado el: 29 de 11 de 2012.] <http://www.aportacionesfiscales.com/index.php/forum/2-fiscal/61779-concepto-de-moneder-electronico>.

**bbva. 2012** bbva.es. [En línea] [Citado el: 20 de 11 de 2012.] <https://www.bbva.es/TLBS/tlbs/esp/segmento/empresas/necesidades/otrservi/pasapag/index.jsp>.

**Borudreau, Tim, Tulach, Jaroslav y Wielenga, Geertjan. 2007.** Rich Client Programming Plugging into the NetBeans Platform. Stoughton, Massachusetts: Sun Microsystems, 2007. 0-13-235480-2.

**CEPEC. 2001.** Habana, Cuba: Centro del Comercio Exterior y la Inversión Extranjera de Cuba. , 2001.

**Dans, Enrique. 2007** Comercio electrónico. S.I.: Instituto de Empresa.

**Miguel Angel Alvarez. 09/05/2001** DesarrolloWeb.com. [En línea] [Citado el: 12 de 12 de 2012.] <http://www.desarrolloweb.com/articulos/392.php>...

**EcuRed. 2010.** [En línea] Enciclopedia colaborativa, 14 de 12 de 2010. [Citado el: 12 de 12 de 2012.] <http://www.ecured.cu/index.php/PHP>.

**Fowler, Martin and Rice, David. 2003.** Addison-Wesley: Patterns of Enterprise Application Architecture. S.I, 2003. 0321127420.

**Jacobson, Ivar G. B. and Rumbaugh, James. 2000** .Proceso Unificado de Desarrollo De Software. Madrid: s.n, 2000.

**Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000.** El Proceso Unificado de Desarrollo de Software. Madrid: Pearson Educación. S.A, 2000. 0201571692.

**JUAN, PABLO GOMEZ GALLEGO, ING JORGE GALVES. 2007.** [En línea] UNIVERSIDAD TECNOLÓGICA DE PEREIRA, 16 de 09 de 2007. [Citado el: 11 de 12 de 2012.] <http://es.scribd.com/doc/297224/RUP>.

**Lidera Network. 2009.** lidera.net. [En línea] Lidera Network, S.L, 1999. [Citado el: 28 de 11 de 2012.] <http://www.lidera.net/>.

- Maestros del Web. 2010.** maestrosdelweb.com. [En línea] 1997. [Citado el: 13 de 12 de 2012.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/...>
- Ministros, Consejo de. 26/12/2006.** Lineamientos para el desarrollo en Cuba del Comercio Electrónico. Habana, Cuba: Consejo de Ministros, 26/12/2006.
- Monetos. 2012.** monetos.es. [En línea] [Citado el: 29 de 11 de 2012.] [http://www.monetos.es/financiacion/tarjetas-credito/definicion-formato/.](http://www.monetos.es/financiacion/tarjetas-credito/definicion-formato/)
- Monetos. 2012.** monetos.es. [En línea] [Citado el: 29 de 11 de 2012.] [http://www.monetos.es/inversiones/tarjetas-debito/definicion-formato/.](http://www.monetos.es/inversiones/tarjetas-debito/definicion-formato/)
- NetBeans Community. 2000.** [En línea] NetBeans Community, 2000. [Citado el: 12 de 12 de 2012.] <http://netbeans.org/>
- OMG. 2012.** [En línea] Unified Modeling Language™ (UML®), 12 de 04 de 2012. [Citado el: 11 de 12 de 2012.] [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm).
- Popkin Software and Systems. 19/06/2007** tldp.org. [En línea] [Citado el: 11 de 12 de 2012.] <http://es.tldp.org/Tutoriales/doc...UML/doc-modelado-sistemas-uml.pdf>.
- Pressman, Roger. 2003.** Ingeniería de Software, un enfoque práctico. New York: s.n, 2003.
- Torre, Aníbal de la. 2006.** [En línea] 2006. [Citado el: 12 de 12 de 2012.] [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html).
- Rodríguez, Fran Gil. 2012.** Experto en Drupal 7. Nivel Inicial. Gran Canaria: Forcontu S.L, 9/2/2012. ISBN-13.
- Rafael Martinez. 2012.** postgresql. [En línea] Creative Commons, 02 de 10 de 2012. [Citado el: 15 de 01 de 2013.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
- Pablo Herrero. 2012.** [En línea] Innovación y tecnología , 14 de 11 de 2012. [Citado el: 11 de 02 de 2013.] [http://blog.sage.es/innovacion-tecnologia/la-pasarela-de-pago-es-el-motor-de-los-negocios-online/.](http://blog.sage.es/innovacion-tecnologia/la-pasarela-de-pago-es-el-motor-de-los-negocios-online/)
- PgAdmin III.2008.** [En línea] 10 de 03 de 2008. [Citado el: 15 de 12 de 2012.] [http://www.guia-ubuntu.org/index.php?title=PgAdmin\\_III..](http://www.guia-ubuntu.org/index.php?title=PgAdmin_III..)
- Larman, Craig. 2003.** UML y Patrones. s.l. : Prentice Hall, 2003. 2a Edición.
- \*Garcerant, Iván. 2008.** Tecnología y Synergix. [En línea] 10 de 07 de 2008. [Citado el: 12 de 02 de 2013.] <http://synergix.wordpress.com/?s=modelo+del+dominio>.
- Turner, Raymond. 2005.** The Foundations of Specification. s.l. : Journal of Logic and Computation, Vol. 15, No. 5. Octubre 2005. pp. 623–663.

- Botta, Adrián. 2010.** Diseño de Sistemas. scribd. [En línea] 2010. [Citado el: 12 de 02 de 2013.] <http://www.scribd.com/doc/85235594/9/FLUJO-DE-TRABAJO-DE-CAPTURA-DE-REQUISITOS>.
- Fernández, Adelaida Ramírez. Abril 2012.** CLASIFICACIONES DE TIPOS DE REQUISITOS. Madrid : Universidad Carlos 111, Abril 2012.
- Jacobson, I., P. Jonsson, M. Christerson and G. Overgaard. 2006.** Ingeniería de Software Orientada a Objetos - Un acercamiento a través de los casos de uso. Upper Saddle River, N.J. : Addison Wesley Longman, 2006.
- oocities.2009.** [En línea] 27 de 10 de 2009. [Citado el: 20 de 02 de 2013.] [www.oocities.org/gusdelact/ingsoft/tema3/temaIII.html](http://www.oocities.org/gusdelact/ingsoft/tema3/temaIII.html).
- Villanueva, Jose Luis Rodriguez. 28/10/2009.** Diagramas De Interaccion. Costa Rica : s.n., 28/10/2009.
- Gonzalo Génova, María C. Valiente, Jaime Nubiola. 2006.** Modelos en UML: un enfoque semiótico. Madrid, España : Departamento de Informática, Universidad Carlos III de Madrid, 2006.
- Thames, Juan Pablo Bustos. 09/08/ 2011.** Diagramas de clases. Argentina : Facultad Regional Tucumán de la Universidad Tecnológica Nacional (U.T.N. - F.R.T.), 09/08/ 2011.
- Pressman, Roger S. 10/11/2011.** Software Engineering, a practitioner's approach. (7th edición), McGraw-Hill, pp. 895, ISBN 9780071267823
- Sebastián, Ing. Juan. 13/11/2010.** Modelo Vista Controlador. Definicion y características. Colombia : Universidad de Manizales , 13/11/2010.
- Tutorial UML 2. 2007.** sparxsystems. [Citado el: 20 de 02 de 2013.] 2007. [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_deploymentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html).
- OMG, 2012,** Unified Modeling Language (OMG UML), Superstructure, V2.1.2 p.586-588.
- Josep Peguerols Vallés . 2009.** Sistemas de pago electrónicos. [En línea] 2009. <http://upcommons.upc.edu>.
- Hurtado, B Herrero. 2010.** DESARROLLO WEB CON DRUPAL. [En línea] 2010. [Citado el: 12 de 12 de 2012.] [eciencia.urjc.es](http://eciencia.urjc.es).
- Elisa Arizaca Ramirez. 2009.** Analisis y diseno de sistema 2. [Online] 06 9, 2009. <http://virtual.usalesiana.edu.bo/web/practica/archiv/compon.doc>.

### BIBLIOGRAFÍA

**Byron, Angela, y otros. 2009.** Using Drupal. S.l.: Anaya Multimedia/O'Reilly, 2009. 978-84-415-2612-9.

**Corbalán Sánchez, Luis María y Gómez Cáceres, Diego. 2001.** Mercados Electrónicos. Nuevos sistemas de pago. 1. s.l.: ESIC, 2001. pág. 318. 8473562755.

**García del Poyo, Rafael, Gil Rabadán, Juan y Ant, Juan. 2010.** El libro del Comercio Electrónico. S.l.: ESIC, 2010.

**Gutmans, Andi, Bakken, Stig y Rethans, Derick. 2004.** PHP 5 Power Programming. Indianapolis: Prentice-Hall, 2004. 0-131-47149-X.

**Larman, Craig.** UML y patrones. Introducción al análisis y diseño orientado a objetos. 2. Montevideo. Uruguay: Prentice-Hall. 84-205-3438-2.

**Lurig, Mario.** PHP Reference: Beginner to Intermediate PHP5.

**Montoro, Julio Fraile.** Sistemas de pago en el comercio electrónico.

**Pérez, Javier Eguíluz. 2009.** Introducción a JavaScript. S.l.: Autoedición, 2009. pág. 140.

**Santillán, Jesús Roberto López. 2011.** Pruebas de Caja Negra. Chihuahua: s.n., 2011.

**drupalcommerce.2013.**<http://www.drupalcommerce.org/>

### GLOSARIO DE TÉRMINOS

**Terminal Punto de Venta:** Dispositivo y tecnologías que ayudan en la tareas de gestión de un establecimiento comercial de venta al público. Ayudan en las tareas de gestión del punto de venta mediante una interfaz accesible para los vendedores que se denomina TPV.

**IVR:** *Interactive Voice Response* o respuesta de voz interactiva. También se utiliza el término VRU (*Voice Response Unit*) o unidad de respuesta de voz. Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el humano a través de grabaciones de voz y el reconocimiento de respuestas simples, como "sí", "no" u otras. Es un sistema automatizado de respuesta interactiva, orientado a entregar y/o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras operaciones.

**IDE:** Del inglés *Integrated Development Environment*, en español Entorno de Desarrollo Integrado, es un programa que ofrece una serie de herramientas que facilitan el trabajo de los desarrolladores de software para programar sus programas.

**Herramienta CASE:** es un software que permite a los desarrolladores modelar parte o todos los componentes de una aplicación. Generalmente a través de diagramas. Las siglas CASE provienen de *Computer Aided Software Engineering*, que en castellano significan Ingeniería de Software asistida por computadora.

**Comercio electrónico:** Compra y venta de productos o de servicios a través de medios digitales, tales como Internet y otras redes informáticas, usando como forma de pago medios electrónicos, como las tarjetas de crédito.

**Pasarela de pago:** Proveedor de servicios de aplicación de comercio electrónico que autoriza pagos a negocios electrónicos y minoristas electrónicos. Es el equivalente de una TPV (Terminal Punto de Venta) física ubicada en la mayoría de los almacenes al detalle.