

# Universidad de las Ciencias Informáticas

## Facultad 6



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

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

**Autor:**

Lieter Manuel Rodríguez Reyes

**Tutor:**

Ing. Ángel Dayán Marín Abreu

**Co-Tutor:**

Ing. Frank Torres Rodríguez

La Habana, junio de 2012

“Año 54 de la Revolución”

## Declaración de autoría

---

---

Declaro ser autor del presente trabajo “**Desarrollo de un módulo de pago para el acceso a los materiales audiovisuales publicados en la Plataforma VideoWeb**” y reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2012.

---

Lieter M. Rodríguez Reyes  
Autor

---

Ing. Angel Dayán Marín Abreu  
Tutor

**Tutor:** Ing. Angel Dayán Marín Abreu

**Especialidad de graduación:** Ingeniería en Ciencias Informáticas

**Categoría docente:** Sin categoría

**Categoría científica:** Sin categoría

**Años de experiencia en el tema:** 1

**Años de graduado:** 2

**Correo electrónico:** [admarin@uci.cu](mailto:admarin@uci.cu)

**Co-tutor:** Frank Torres Rodríguez

**Especialidad de graduación:** Ingeniería en Ciencias Informáticas

**Categoría docente:** Sin categoría

**Categoría científica:** Sin categoría

**Años de experiencia en el tema:** 1

**Años de graduado:** 2

**Correo electrónico:** [ftrodriguez@uci.cu](mailto:ftrodriguez@uci.cu)

## 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 el desarrollo realizado 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 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 implementada es un módulo para el *Content Management System* (CMS) Drupal haciendo uso del lenguaje de programación PHP y el entorno de desarrollo integrado NetBeans. Se presentan los requisitos identificados y el cumplimiento de los mismos en la aplicación final, que es validada además por las pruebas realizadas, propuestas por la metodología de desarrollo utilizada. El impacto social de los resultados de esta investigación radica en que el proyecto VideoWeb podrá insertar su plataforma en entornos comerciales relacionados a los negocios electrónicos.

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

# Índice de Figuras

---

---

Figura 1: Fases e hitos de RUP. ....	16
Figura 2: Los nueve flujos de trabajo de RUP. ....	17
Figura 3: Modelo de dominio.....	23
Figura 4: Diagrama de casos de uso del sistema.....	27
Figura 5: DCA Gestionar paquetes de pago.....	36
Figura 6: DCA Cambiar estado de usuario.....	36
Figura 7: DCA Realizar pago y Generar reporte .....	37
Figura 8: DCA Exportar reporte.....	37
Figura 9: DCA Cambiar Nivel de Archivo Multimedia .....	37
Figura 10: DC Cambiar nivel de media .....	38
Figura 11: DC Cambiar estado de usuario .....	39
Figura 12: DC Exportar reporte .....	40
Figura 13: DC Realizar pago y Generar reporte.....	40
Figura 14: DS Cambiar nivel de media.....	41
Figura 15: DS Cambiar estado de usuario .....	42
Figura 16: DS Exportar reporte .....	42
Figura 17: DS Realizar pago y Generar reporte .....	43
Figura 18: DCD Cambiar nivel de archivo multimedia .....	45
Figura 19: DCD Cambiar estado de usuario.....	46
Figura 20: DCD Gestionar pago.....	47
Figura 21: DC Gestionar Paquetes .....	50

# Índice de Figuras

---

---

Figura 22: DC Ejecutar Pago .....	51
Figura 23: DC Exportar a PDF .....	51
Figura 24: Diagrama de despliegue .....	52

---

---

<b>Introducción .....</b>	<b>1</b>
<b>Capítulo 1: Fundamentación Teórica.....</b>	<b>6</b>
1.1 Conceptos asociados al dominio del problema .....	6
1.1.1 Comercio electrónico .....	6
1.1.2 Sistema de pago electrónico.....	7
1.2 Proceso de compra en la web usando pasarela de pago.....	9
1.3 Descripción de la situación problemática .....	10
1.4 Soluciones existentes .....	11
1.4.1 Tematika.com .....	11
1.4.2 XtraLife.es .....	12
1.4.3 Módulo de comercio electrónico de Drupal .....	12
1.5 Tecnologías para el desarrollo de la aplicación.....	13
1.5.1 Metodología de Desarrollo de Software .....	13
1.5.2 Lenguaje de modelado .....	18
1.5.3 Lenguajes de programación.....	18
1.5.4 Entorno de desarrollo integrado (por sus siglas en inglés IDE) .....	20
1.5.5 Herramienta CASE .....	20
1.6 Conclusiones .....	21
<b>Capítulo 2: Análisis y Diseño .....</b>	<b>22</b>
2.1 Modelo de Dominio.....	22
2.1.1 Definición de las clases del modelo de dominio.....	22
2.1.2 Diagrama de clases del modelo de dominio.....	23
2.2 Especificación de requisitos.....	24

---

---

2.3	Descripción del sistema propuesto .....	26
2.3.1	Descripción de actores .....	26
2.3.2	Diagramas de casos de uso .....	26
2.3.3	Descripción de los casos de uso del sistema.....	27
2.4	Modelo de análisis .....	35
2.4.1	Diagrama de clases del análisis.....	35
2.4.2	Diagramas de interacción .....	37
2.5	Modelo de diseño .....	43
2.5.1	Diagramas de Clases del Diseño.....	44
2.6	Patrones arquitectónicos .....	47
2.6.1	Modelo Vista Controlador .....	48
2.7	Conclusiones .....	48
	<b>Capítulo 3: Implementación y pruebas .....</b>	<b>50</b>
3.1	Diagrama de componentes .....	50
3.2	Diagrama de despliegue .....	51
3.3	Pruebas de software .....	52
3.3.1	Pruebas de caja negra.....	53
3.3.2	Resultados de las pruebas .....	60
3.4	Conclusiones .....	61
	<b>Conclusiones generales .....</b>	<b>62</b>
	<b>Recomendaciones.....</b>	<b>63</b>
	<b>Referencias.....</b>	<b>64</b>
	<b>Bibliografía .....</b>	<b>66</b>



# Índice General

---

---

Anexos .....	67
Glosario.....	68

## Introducción

El comercio es una actividad ancestral del ser humano, desde su comienzo ha evolucionado de diversas maneras; pero su significado y su fin es siempre el mismo. El comercio es "el proceso y los mecanismos utilizados, necesarios para colocar las mercancías, que son elaboradas en las unidades de producción, en los centros de consumo donde se aprovisionan los consumidores, último eslabón de la cadena de comercialización. Es comunicación y trato" (Galazar, 2005). En líneas generales, y con un sentido amplio, el comercio implica la investigación de mercado con el fin de interpretar los deseos del consumidor. Incluye además la publicidad que anuncia la existencia del producto, la posibilidad de adquirirlo, y en qué lugar, a la vez que se utilizan los métodos de persuasión, la venta al por menor y finalmente, la adquisición por parte del público.

La masividad de las redes mundiales de la información ha transformado el mundo y ha 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; mejorando notablemente la calidad de vida de las naciones. Con estas nuevas tecnologías la distancia y el tiempo ya no son un problema, pueden llegar a millones de personas por igual, propiciando la competitividad entre empresas y el libre intercambio de conocimientos. En el nuevo siglo de las comunicaciones el intercambio no conoce fronteras.

El rápido desarrollo y difusión de Internet, que ya cuenta con más de 8 millones de servidores encargados de los servicios y sobrepasa los 250 millones de usuarios de más de 100 países, ha permitido la creación de nuevas tecnologías que se abren a nuevos mercados, denominados "economía digital". Los productores, proveedores de bienes y servicios, además de los usuarios logran tener acceso y expansión de una forma sencilla y económica, ya sean con fines comerciales o sociales. La apertura de estos nuevos mercados es fundamental en el rápido crecimiento del uso de nuevos servicios y la asimilación de tecnologías nuevas. En la práctica, las empresas están comenzando a usar Internet como un nuevo canal de ventas, sustituyendo las visitas personales, correo y teléfono por pedidos electrónicos, ya 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)

Dentro de estas actividades comerciales por la red el sector del software y el ocio, entiéndase música, películas y videojuegos, ha alcanzado un papel importante y presenta aún un futuro prometedor al posibilitar la adquisición del producto comprado en muy poco tiempo, sin el molesto inconveniente de esperar la entrega como ocurre con cualquier otro producto tangible. El ciclo de la venta se culmina con la llegada del producto a manos del comprador. Por esta razón es denominado “comercio electrónico real”, sin la aparición de ningún intermediario, el contacto personal es nulo; además los costos de transporte y otros valores agregados que se evidencian en otro tipo de ventas son reducidos drásticamente, permitiendo adquirir estos productos a precios muy cercanos a los de realización. En la web han ido apareciendo múltiples sitios y negocios destinados a estas transacciones, evidenciando su rentabilidad además de la aceptación generalizada, por permitir el completo anonimato del cliente.

En Cuba los avances y facilidades de este tipo de comercialización de productos no tienen casi representación, principalmente por la inexistencia de soporte tecnológico. Aunque se han visto pequeñas incursiones en este campo, como el pago automático de los servicios de electricidad y de agua para los usuarios de La Habana que cuenten con tarjetas del Banco Metropolitano. De todas maneras existe muy poca experiencia en este tema, principalmente por el desconocimiento general del uso de tarjetas magnéticas y otros sistemas de pago electrónico. 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 de Cuba donde se enfocan en la educación generalizada y la formación de valores.

En la Universidad de las Ciencias Informáticas (UCI) se crean productos destinados al mercado internacional o a la informatización de la sociedad cubana de crearse la infraestructura necesaria. Es política de esta que sus estudiantes participen de manera activa en proyectos productivos donde los conocimientos adquiridos en el plan de estudio sean aplicados en situaciones prácticas y vean su utilidad en la vida real. Con este fin cada facultad tiene diferentes sectores dentro de las ciencias de la información en las que trabajar, donde se agrupan los diferentes proyectos según su línea de investigación. En el caso de la facultad 6, el proceso productivo está dividido en dos centros de

desarrollo, el centro de Tecnologías de Gestión de Datos (DATEC) y el centro de Geoinformática y Señales Digitales (GEYSED).

Dentro del centro GEYSED, destinado al desarrollo de aplicaciones informáticas vinculadas al campo de las señales digitales (imagen, sonido y video), se encuentra el proyecto VideoWeb. Este ha creado una plataforma 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. Cuenta además con diversas facilidades para los usuarios de la misma, los cuales desde sus respectivas sesiones pueden almacenar y organizar los contenidos, listas de reproducción y otros según sus preferencias.

Como deficiencia se puede señalar que la ausencia de un mecanismo de pago adecuado imposibilita su inserción en potenciales mercados como hoteles, cibercafés, aeropuertos, etc., donde cobrar estos servicios de una manera segura y efectiva es primordial, quedando limitada entonces a instituciones educativas o culturales. De añadirse un mecanismo de pago electrónico adecuado la plataforma podría insertarse al mercado y competir en igualdad de condiciones con cualquier otra empresa, aumentando sus prestaciones actuales y aumentando el valor añadido a la misma.

Ante la problemática planteada se deriva el siguiente problema a resolver: **¿Cómo disminuir el esfuerzo que implica cobrar la reproducción y descarga de los materiales audiovisuales publicados en la Plataforma VideoWeb?**

Para dar solución al problema científico se plantea como objeto de estudio: **Técnicas para el desarrollo de mecanismos de pago electrónico**, delimitando como campo de acción: **Pasarela de pago para la Plataforma VideoWeb.**

Defendiendo la idea: **Con el desarrollo de un módulo de pago para la Plataforma VideoWeb se logrará disminuir el esfuerzo que implica cobrar la reproducción y descarga de los materiales audiovisuales publicados en la misma.**

Como solución al problema planteado se define como objetivo general: **Desarrollar un módulo de pasarela de pago para la Plataforma VideoWeb.**

Las tareas de investigación a cumplir son:

1. **Elaborar el marco teórico sobre mecanismos de pago electrónico.**
2. **Caracterizar las técnicas de comercio electrónico existentes para el desarrollo de pagos en línea.**
3. **Seleccionar las herramientas y tecnologías a utilizar en el desarrollo del módulo.**
  - 3.1 **Definir y argumentar la metodología de desarrollo de software a usar en el proceso de desarrollo.**
4. **Identificar las funcionalidades que debe brindar el módulo.**
5. **Realizar el modelado de los artefactos necesarios para la implementación del módulo.**
6. **Realizar las modificaciones necesarias a la base de datos de la Plataforma VideoWeb.**
7. **Implementar el módulo de pago para el acceso a los materiales audiovisuales publicados en la Plataforma VideoWeb.**
8. **Realizar pruebas al módulo desarrollado.**

Para el cumplimiento de las tareas antes mencionadas se emplean los siguientes métodos científicos de investigación, teóricos y empíricos:

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 permite entender el tema, simplificando el estudio del mismo, además de propiciar el arribo a conclusiones.

**Inductivo-Deductivo:** para la adquisición de un 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 permite el estudio de la situación actual del tema (estado del arte), analizando posibles soluciones existentes, además permite seguir la evolución y desarrollo, incluyendo las tendencias actuales, de los mecanismos de pago electrónico para la solución del problema anterior.

## Métodos Empíricos

**Entrevistas**, método aplicado a profesores y directivos pertenecientes al proyecto con el objetivo de obtener información sobre la investigación en curso y en vista de obtener los posibles requisitos.

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

Capítulo 1: Fundamentación Teórica, se realiza la elaboración del marco teórico sobre mecanismos de pago electrónicos. Además, se realizará un estudio del estado del arte, análisis de soluciones y definición de conceptos asociados al dominio del problema. De igual manera se definirán las herramientas y el lenguaje de programación a utilizar, así como la metodología de desarrollo más recomendable.

Capítulo 2: Presentación de la Solución Propuesta, se desarrollan los artefactos correspondientes al análisis y diseño, se hará una modelación de la propuesta de solución, se identificarán los requisitos funcionales y no funcionales, se diseñará la arquitectura y los diagramas correspondientes.

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

## Capítulo 1: Fundamentación Teórica

En el presente capítulo se abordan los principales conceptos asociados al dominio del problema, permitiendo llegar a una mayor comprensión sobre el mismo; se refiere a los mecanismos de pago electrónico, describiendo sus especificidades y modo de funcionamiento. Se analizan las diferentes tecnologías y herramientas a usar en el desarrollo del módulo, particularidades y motivos para su uso. Se representa el proceso de selección de una metodología de desarrollo de software, las razones que permitieron arribar a tal decisión y una descripción de la misma, sus fases, flujos de trabajo y artefactos que arroja. Se hace una descripción de la situación que genera el problema a resolver, un análisis de las posibles soluciones aplicables al mismo y las ya existentes, justificando la negativa a usar alguna de estas y la decisión de implementar una solución completamente nueva.

### 1.1 Conceptos asociados al dominio del problema

#### 1.1.1 Comercio electrónico

El comercio electrónico, también conocido como *e-commerce* (*electronic commerce* en inglés), consiste en la compra y venta de productos o de servicios a través de medios electrónicos, tales como Internet y otras redes informáticas. Originalmente el término se aplicaba a la realización de transacciones mediante medios electrónicos tales como: el intercambio electrónico de datos. Con el advenimiento de la Internet y la *World Wide Web* a mediados de los años 90 comenzó a referirse principalmente a la venta de bienes y servicios a través de Internet, usando como forma de pago medios electrónicos, tales como las tarjetas de crédito. (Laudon, 2009)

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

## 1.1.2 Sistema de pago electrónico

### Tarjeta de crédito

La tarjeta de crédito es un instrumento material de identificación del usuario, que puede ser una tarjeta plástica con una banda magnética, un microchip y un número en relieve. Es emitida por un banco o entidad financiera que autoriza a la persona a cuyo favor es emitida, utilizarla como medio de pago en los negocios adheridos al sistema, mediante su firma y la exhibición de la tarjeta. Es otra modalidad de financiación, por lo tanto, el usuario supone asumir la obligación de devolver el importe dispuesto y de pagar los intereses, comisiones bancarias y gastos pactados. Entre las más conocidas del mercado están: Visa, *American Express*, *MasterCard*, *Diners Club*, *JCB*, *Discover*, *Cabal*, entre otras.

Las grandes tiendas y almacenes del mundo también emiten tarjetas de crédito para sus clientes. La mayor ventaja es la flexibilidad que le da al usuario, quien puede pagar sus saldos por completo en su fecha límite mensual o puede pagar una parte. La tarjeta establece el pago mínimo y determina los cargos de financiamiento para el saldo pendiente. Las tarjetas de crédito también se pueden usar en los cajeros automáticos o en un banco para servirse de un adelanto de efectivo, aunque a diferencia de las tarjetas de débito, se cobra un interés por la disposición, comisión y, en algunos países, un impuesto porque se trata de un préstamo. Se suele cobrar una cuota anual por el uso de la tarjeta. Las compras con tarjeta de crédito pueden tener diversos seguros sobre el saldo financiado (Asociación de Bancos de México, A.C., 2003).

### Tarjeta de débito

La tarjeta de débito es una tarjeta bancaria con una banda magnética en el reverso que guarda información sobre los datos de acceso y el saldo de dinero que dispone el titular en su cuenta, usada para poder efectuar con ella operaciones financieras activas (incrementar el saldo), pasivas (disminuir el saldo) o neutrales (no incrementan ni disminuyen el saldo disponible).

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. Algunos



# Capítulo 1: Fundamentación Teórica

---

---

bancos realizan acuerdos con sus clientes para permitirles extraer dinero en descubierto, generando un préstamo con sus respectivos intereses.

Su cuota anual es más barata que la de crédito o incluso resulta gratuita y es uno de los instrumentos financieros más utilizados en el mundo civilizado (Asociación de Bancos de México, A.C., 2003).

## Monedero electrónico

La tarjeta monedero o llamado también monedero electrónico, ha sido el sistema de micropagos multipropósito más eficiente. Ha servido para aquellas transacciones de bajo monto y alto volumen que requieren gran velocidad y seguridad. El sistema ha permitido a los usuarios pagar más rápido que con efectivo y las transacciones se llevan en menos tiempo. Algunas ventajas:

- Se evitan las filas y la congestión en los puntos de pagos.
- Se gana más tiempo.
- Es más fácil, rápido y práctico.
- Da mayor seguridad y control de los pequeños gastos de todos los días.
- Se elimina los problemas del cambio y la falta de monedas.

La tarjeta monedero contiene un chip electrónico que se recarga con dinero y permite pagar productos y servicios deslizando la tarjeta en el lector cuyo importe exacto se descargará o acreditará del chip. Se puede recargar en cualquier establecimiento que tenga una terminal para recibir el pago (tiendas abonando un mínimo establecido). También se puede recargar con tarjeta de débito o crédito. Para no quedarte sin saldo es recomendable vincular tu tarjeta a una tarjeta de crédito o débito, y de esta manera tu monedero se recargará automáticamente (IDSOLUTIONS, A.C., 2004).

## Pasarela de pago

Una pasarela de pago, *payment gateway*, *gateway* de pago o Terminal Punto de Venta (TPV) virtual consiste en un proveedor de servicios de aplicación de comercio electrónico que autoriza pagos a negocios electrónicos y a minoristas electrónicos. Cumple en Internet la misma función que los sistemas tradicionales de cobro mediante tarjeta de crédito (TPV físico ubicado en la mayoría de los

# Capítulo 1: Fundamentación Teórica

---

---

almacenes), pues permite que los clientes puedan pagar una compra utilizando una tarjeta de crédito, ya que cifran información sensible, tales como números de tarjetas, para garantizar que la información pasa en forma segura entre el cliente y el vendedor. Gracias a las pasarelas de pago, las empresas de compra-venta online pueden realizar de una manera rápida y segura las transacciones entre compradores y vendedores (Lidera Network, S.L., 2011).

## 1.2 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 su siglas en inglés IVR) y el Procesador Interfaz o banco adquirente de manera rápida y segura. Cuando un cliente ordena un producto de un vendedor que tiene habilitado una pasarela de pago, ésta 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 similar 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, ésta 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).

# Capítulo 1: Fundamentación Teórica

---

---

- La pasarela de pago recibe la respuesta y la reenvía al sitio web (o cualquier otra interfaz que haya sido usada para procesar el pago), donde se interpreta y se revela una respuesta al cliente.
- El proceso completo demanda unos 3-4 segundos.

Al final del día bancario (o período de liquidación), el banco adquirente deposita el total de los fondos aprobados en la cuenta nominada del vendedor. Esta cuenta puede encontrarse en el mismo banco adquirente si el vendedor realiza sus operaciones con el mismo banco o una cuenta con otro banco (Lidera Network, S.L., 2011).

## 1.3 Descripción de la situación problemática

El éxito de cualquier negocio electrónico está dado por la rentabilidad que represente, ya sea por la venta de productos o los servicios que brinde. La interrogante de cómo cobrar por esto es la principal atención, es necesario contar con una forma de pago adecuada y confiable. En el caso de las empresas que comercialicen materiales virtuales, entiéndase todos aquellos productos fruto de las tecnologías en forma de datos o información que tienen equivalente en el mundo real, tienen como ventaja que pueden utilizar, además de la vía tradicional, los beneficios de la red de comunicaciones para efectuar la transacción comercial. Un mecanismo de pago electrónico bien diseñado y empleado, seguro, fiable y fácil de usar, puede incrementar de manera dramática los índices de ingresos previstos.

La plataforma de transmisión, reproducción y descarga de materiales audiovisuales a través de la red (VideoWeb) brinda diversas facilidades y ventajas a los usuarios de la misma, diseñada sobre una amplia gama de servicios configurables según las necesidades de clientes específicos. Aunque tiene como principal inconveniente las dificultades a las que se enfrenta a la hora de cobrar los servicios consumidos por los usuarios de la misma. El pago debe efectuarse de manera presencial, ya sea por cuotas o cada servicio independiente, lo que dificulta la agilidad del proceso y hace imposible aprovechar al máximo las ganancias que puede reportar la aplicación. De existir algún mecanismo de pago electrónico que automatizara este proceso se lograría disminuir el costo que implica el mismo, además de permitir atender múltiples usuarios simultáneamente, aumentando las prestaciones de la plataforma.

## 1.4 Soluciones existentes

Un módulo pasarela de pago o TPV virtual debe ser implementado especialmente para cada aplicación, teniendo en cuenta las especificaciones de esta además de las características del banco adquirente al cual está vinculado el negocio: tipos de pago que acepta, cheque electrónico, transferencia bancaria, tarjeta de crédito o débito, entre otros; compañías de las cuales acepta el pago: VISA, MasterCard, PayPal o cualquier otro; datos que solicita para consumir el servicio: número de tarjeta, fecha de caducidad, *Personal Identification Number* (por sus siglas en inglés PIN) u otra información solicitada por la institución bancaria; además del mecanismo de encriptación que use para proteger los datos del usuario que debe ser implementado en el módulo: *Secure Socket Layer* (por sus siglas en inglés SSL) o *Transport Layer Security* (por sus siglas en inglés TLS).

Teniendo en cuenta estas cuestiones es imposible encontrar alguna solución pre-existente que pueda ser aplicada a la plataforma VideoWeb, 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.

### 1.4.1 Tematika.com

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

Tematika es un negocio electrónico argentino destinado a la comercialización de productos de ocio (libros, música, películas y más), aunque no de manera directa a través descarga, sino enviando los pedidos al domicilio especificado. Obviando 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 que es 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

## Capítulo 1: Fundamentación Teórica

---

---

especificarán los detalles de la transacción; todo esto sustentado por una elevada seguridad al efectuarse todas las operaciones a través de una conexión cifrada a través del 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. De manera general Tematika.com posee un mecanismo de pago muy bien implementado en el cuál se ha tenido en cuenta todos los detalles imprescindibles para el correcto funcionamiento de este.

### 1.4.2 XtraLife.es

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

Un sitio 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. También cuenta con las respectivas facilidades de compra habituales: carrito de compra accesible desde cualquier lugar con información asociada al pedido, la posibilidad de elegir diferentes métodos de pago, ayuda sobre los datos a introducir para la compra, 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.

### 1.4.3 Módulo de comercio electrónico de Drupal

Aunque no es una solución que sea utilizada en ningún lugar y se encuentra en una etapa muy básica en cuanto a facilidades y manejo, el módulo de Drupal para comercio electrónico posee algunas funcionalidades muy llamativas que pueden servir de ejemplo, e incluso más, ser incluidas

# Capítulo 1: Fundamentación Teórica

---

---

en la aplicación a desarrollar y facilitar considerablemente el trabajo. El estudio del funcionamiento y visión que tiene el *Content Management System* (por sus siglas en inglés CMS) Drupal sobre los mecanismos de pago electrónico es de fundamental importancia pues la plataforma VideoWeb está desarrollada sobre él y la comprensión que se alcance sobre el tema podría agilizar el proceso de desarrollo en la reutilización de componentes existentes.

Dentro de las funciones más sobresalientes se encuentra que permite el uso de una gran variedad de monedas en las transacciones, incluso el Peso Cubano (CUP) y el Peso Convertible (CUC), permitiendo añadir o quitar a antojo, aunque el cambio hay que especificarlo manualmente. También la posibilidad de tratar los contenidos como productos, asignándoles algunos campos propios de estos como precio, identificador, entre otros; la posibilidad de utilizar diferentes métodos de pago, aunque hay que implementar los que se desee utilizar. Además la creación de una factura donde se contemple todo lo relacionado a la compra a realizar, incluido el cálculo automático del importe de los artículos existentes en el carrito.

Como inconveniente fundamental que estos módulos están solamente disponibles para la versión 6.x, dificultando la futura migración de la plataforma a la versión 7.x, con el módulo incluido. Se puede concluir que la utilización del módulo de comercio electrónico de Drupal en el desarrollo de una solución para el proyecto VideoWeb es solamente demostrativa. Aunque permitiendo observar cómo funcionan estos mecanismos con la vinculación con el CMS y quizás la utilización de algún elemento muy puntual en la implementación de la solución propuesta.

## 1.5 Tecnologías para el desarrollo de la aplicación

### 1.5.1 Metodología de Desarrollo de Software

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar software, es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Además obliga 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.

# Capítulo 1: Fundamentación Teórica

---

---

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. Aunque a primera vista podría suponerse que para un equipo de trabajo tan reducido y para el desarrollo de una aplicación que puede presentar numerosos cambios durante su creación lo ideal sería una metodología ágil que permitiera capturar estas modificaciones y adaptarlas en la solución sin gran esfuerzo o aportar entregables con frecuencia para evaluar el estado del trabajo no es del todo correcto, es necesario analizar otras competencias de gran importancia que pueden contravenir esta selección.

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, si no el de mayor, 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.

Se tienen en cuenta otros factores que pueden afectar en esta decisión por ejemplo: que arroje documentación abundante y detallada que facilite trabajos posteriores; que cuente con soporte y documentación adecuada sobre la cual basar la investigación y encontrar ayuda; que permita el desarrollo sobre cualquier tecnología, particularmente tecnologías libres en este caso; y cualquier otra consideración que el equipo decida tener en cuenta porque pueda representar algún inconveniente para la finalización exitosa del proceso.

Utilizando el método de selección antes descrito y analizando las características de las metodologías previamente escogidas a cada competencia se asignó una puntuación según criterio del autor, siguiendo la guía del método; teniendo como resultado que la de mayor idoneidad según las competencias del equipo de desarrollo es la metodología Proceso Unificado de Desarrollo (por sus siglas en inglés RUP).

## Metodología RUP

“El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos” (Jacobson, y otros, 2000).

**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).

**La metodología RUP divide en 4 fases el desarrollo del software:**

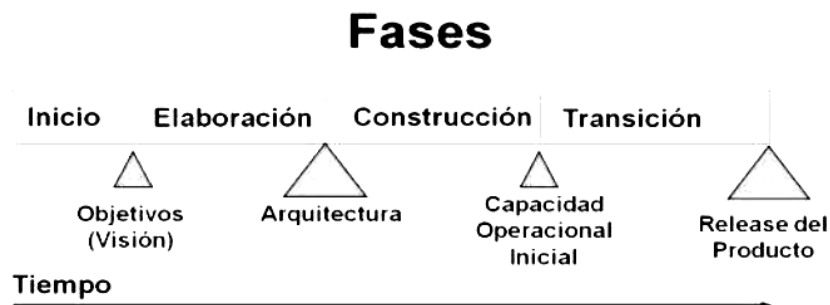
**Inicio:** El objetivo en esta etapa es determinar la visión del proyecto y alcanzar un acuerdo entre todos los interesados respecto a los objetivos del ciclo vital para el mismo.

**Elaboración:** En esta etapa el objetivo es determinar la arquitectura del sistema óptima y proporcionar una base estable para el diseño y el esfuerzo de implementación en la fase siguiente.

**Construcción:** En esta etapa el objetivo es llegar a completar el desarrollo del sistema, basándose en la arquitectura seleccionada.

**Transición:** El objetivo es llegar a obtener la liberación o *release* del proyecto, garantizando que el software esté listo para ser entregado a los clientes.





*Figura 1: Fases e hitos de RUP.*

(Jacobson, y otros, 2000)

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. El ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas o flujos de trabajo:

## Flujos de Trabajo de Ingeniería

**Modelación del negocio:** Donde se logra un entendimiento sobre las necesidades del negocio.

**Requisitos:** Donde se traducen las necesidades del negocio a un sistema automatizado.

**Análisis y Diseño:** Donde se trasladan los requisitos a la arquitectura de software seleccionada.

**Implementación:** Donde se crea el software, ajustándose a la arquitectura y se logra que tenga el comportamiento deseado.

**Pruebas:** Donde se asegura que el comportamiento implementado es el correcto y que todo lo solicitado está presente.

**Despliegue:** Donde se hace todo lo necesario para la salida del proyecto y su posterior puesta en funcionamiento.

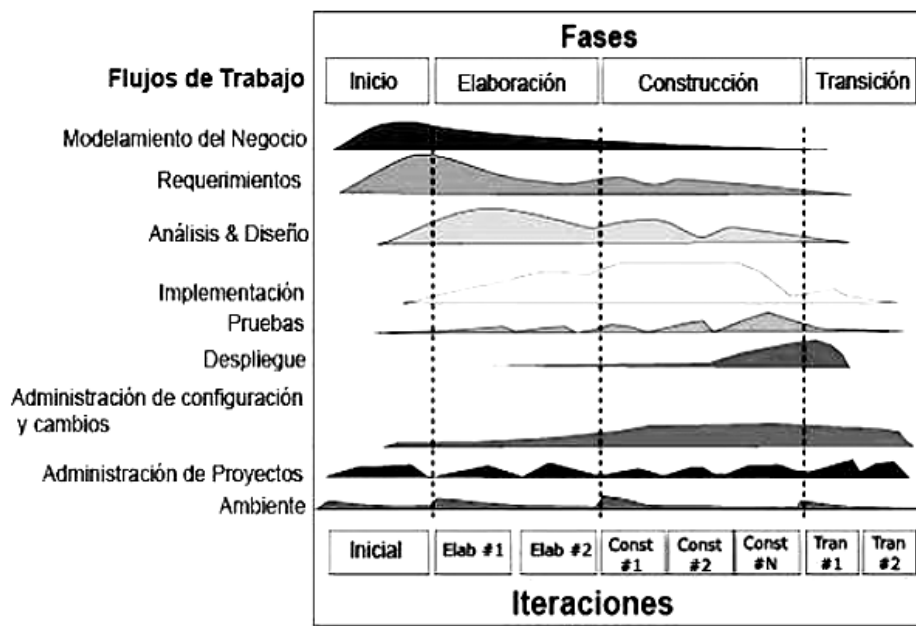
# Capítulo 1: Fundamentación Teórica

**Flujos de Trabajo de Soporte.**

**Administración de configuración y cambios:** Donde se guarda todas las versiones del proyecto.

**Administración de proyectos:** Donde se administran los horarios y recursos del proyecto.

**Ambiente:** Donde se administra el ambiente de desarrollo y el marco de trabajo del equipo.



**Figura 2: Los nueve flujos de trabajo de RUP.**

(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.

## 1.5.2 Lenguaje de modelado

UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el *Object Management Group* (por sus siglas en inglés OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

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 Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requisito. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos. UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas (Fowler, y otros, 1999).

## 1.5.3 Lenguajes de programación

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*), es un lenguaje scripting de propósito general ampliamente utilizado para el desarrollo web y puede ser embebido en páginas HTML, diseñado originalmente para la creación de páginas dinámicas. Es un lenguaje script del lado del servidor. Publicado bajo la *PHP License*, la *Free Software Foundation* considera esta licencia como software libre. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de

# Capítulo 1: Fundamentación Teórica

---

---

sitios web y en un millón de servidores, aunque el número de sitios en PHP ha compartido algo de su preponderante posición con otros nuevos.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos. Permite también la conexión a diferentes tipos de servidores de bases de datos tales como *MySQL*, *PostgreSQL*, *Oracle*, *ODBC*, *DB2*, *Microsoft SQL Server*, *Firebird* y *SQLite*.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y otros basados en él, como *Linux* o *Mac OS X*) y *Microsoft Windows*, y puede interactuar con los servidores web más populares ya que existe en versión *Common Gateway Interface* (por sus siglas en inglés CGI), módulo para Apache, e *Internet Services Application Programming Interface* (por sus siglas en inglés ISAPI) (The PHP Group, 2001).

JavaScript es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (*Server-side JavaScript* o *SSJS* por sus siglas en inglés). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente *widgets*) es también significativo.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del *Document Object Model* (por sus siglas en inglés DOM). Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML (Flanagan, 2001).

## 1.5.4 Entorno de desarrollo integrado (por sus siglas en inglés IDE)

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto de código abierto y gratuito sin restricciones de uso. Es también un proyecto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. *Sun Microsystems* fundó el proyecto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

El proyecto de NetBeans ofrece una versión del IDE a medida para el desarrollo de aplicaciones web en PHP que abarca una gran variedad de secuencias de comandos y lenguajes de marcado. El editor de PHP es dinámico e integrado con HTML, JavaScript y CSS, con funciones de edición. El IDE NetBeans es totalmente compatible con el desarrollo iterativo, por lo que las pruebas de proyectos PHP siguen los patrones clásicos familiares para los desarrolladores web (NetBeans Community, 2000).

## 1.5.5 Herramienta CASE

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Proporciona además abundantes tutoriales, demostraciones interactivas y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones.

Visual Paradigm proporciona un conjunto de galardonados productos, que facilitan diseñar visualmente y esquemáticamente, integrar y desplegar las aplicaciones de misión crítica y sus

# Capítulo 1: Fundamentación Teórica

---

---

bases de datos subyacentes. Ayuda al equipo de desarrollo de software en la creación de todo el modelo y posterior construcción e implementación en el proceso de desarrollo de software, maximiza y acelera tanto al equipo como a las contribuciones individuales, elimina además la complejidad, mejorara la productividad, y comprime los plazos de desarrollo del software (Visual Paradigm International, 2004).

## 1.6 Conclusiones

Durante este capítulo se trataron los aspectos significativos de la investigación que serán el soporte teórico de la futura solución y demuestran la necesidad de la realización del presente trabajo. Se demuestra la importancia de un adecuado mecanismo de pago en los negocios electrónicos.

El estudio del estado arte evidencia la inexistencia de aplicaciones o sistemas que solucionen el problema científico enmarcado en el entorno donde se aplica.

Las condiciones de trabajo sobre la que está respaldada la presente investigación conllevan al uso de RUP como metodología de desarrollo. Al tener en cuenta las particularidades de la solución propuesta se decide utilizar el IDE NetBeans y el lenguaje de programación PHP y JavaScript para realizar las validaciones de la aplicación, además de Visual Paradigm como herramienta CASE.

### Capítulo 2: Análisis y Diseño

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.

#### 2.1 Modelo de Dominio

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema (Jacobson, y otros, 2000).

Un modelo de dominio 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 (Fowler, y otros, 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.

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

**Plataforma VideoWeb:** 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:** Materiales audiovisuales publicados en la plataforma que pueden ser de interés a los usuarios, ya sean películas, documentales o videos musicales.

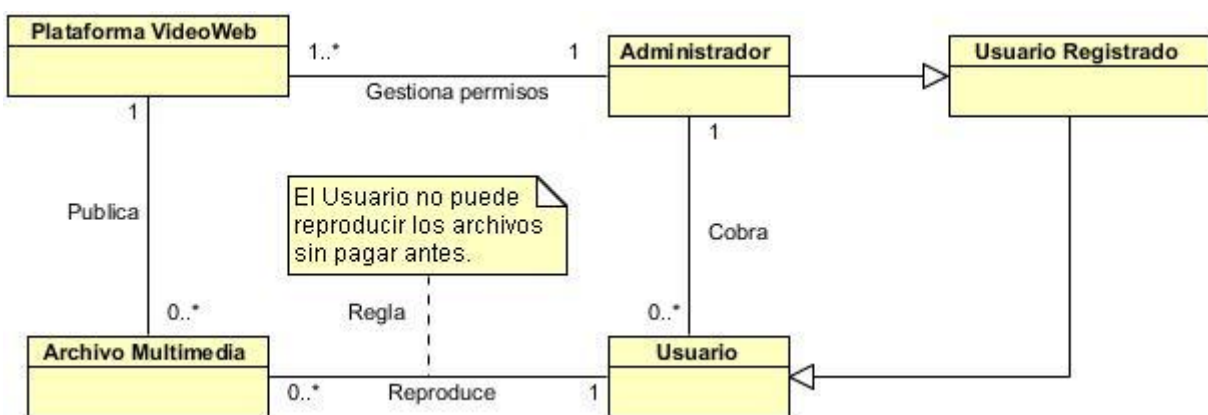
**Usuario:** Usuarios no registrado que puede convertirse en miembro pleno tras finalizar su registro en la plataforma.

**Usuario Registrado:** Usuarios autenticados en la plataforma con permisos para la reproducción de archivos multimedia.

**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.

### 2.1.2 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.



**Figura 3: Modelo de dominio**



### 2.2 Especificación de requisitos

Según Pressman en su libro “Ingeniería de Software. Un enfoque práctico”: *“La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan dificultosa como establecer los requisitos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada en forma errónea. Ninguna otra parte es tan dificultosa de rectificar posteriormente”* (Pressman, 2003).

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 (es decir, las condiciones o capacidades que el sistema debe cumplir) suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente (incluyendo a los usuarios) y los desarrolladores sobre que debe y que no debe hacer el sistema.

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, por lo que se recomienda trazar una estrategia a seguir que permita llevar a cabo este proceso correctamente obteniendo los requisitos reales que debe cumplir el sistema a implementar. La Ingeniería de Requisitos establece modelos de procesos que incluyen técnicas, métodos y plantillas que permiten realizar la captura de requisitos de una manera más eficiente y segura.

#### 2.2.1 Requisitos funcionales

**RF1:** Cambiar modalidad de pago: El módulo debe permitir al usuario elegir el mecanismo de pago a utilizar.

**RF2:** Gestionar paquetes de pago. El módulo debe permitir al administrador crear, modificar y eliminar los paquetes que el usuario puede elegir para pagar.

**RF3:** Realizar pago: El módulo debe permitir enviar los datos cifrados al servidor para realizar la transacción y esperar un código respuesta.

**RF4:** Generar reporte: El módulo debe permitir generar un reporte de acuerdo al código de respuesta recibido del servidor.

**RF5:** Exportar reporte: El módulo debe permitir exportar el reporte generado en formato PDF.

**RF6:** Cambiar nivel de media: El módulo debe permitir cambiar el nivel de acceso a las medias publicadas.

**RF7:** Cambiar estado de usuario: El módulo debe permitir cambiar los privilegios de un usuario en la aplicación.

### 2.2.2 Requisitos no funcionales

#### Requisitos de seguridad

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

#### Requisitos de usabilidad

- El módulo debe contar con una interfaz amigable e intuitiva para la gestión de las funcionalidades.

#### Restricciones en el diseño y la implementación

- El componente a desarrollar es un módulo de Drupal por lo que debe ser implementado en PHP.
- El patrón arquitectónico que se debe emplear en el desarrollo es el modelo-vista-controlador.

#### Restricciones 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.

### 2.3 Descripción del sistema propuesto

El sistema a desarrollar es un módulo basado en el CMS Drupal para el proyecto VideoWeb que permitirá el pago on-line 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. 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.

#### 2.3.1 Descripción de actores

Un actor del sistema es un rol que un usuario juega con respecto al sistema. Es importante destacar que un actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema, es decir que un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información.

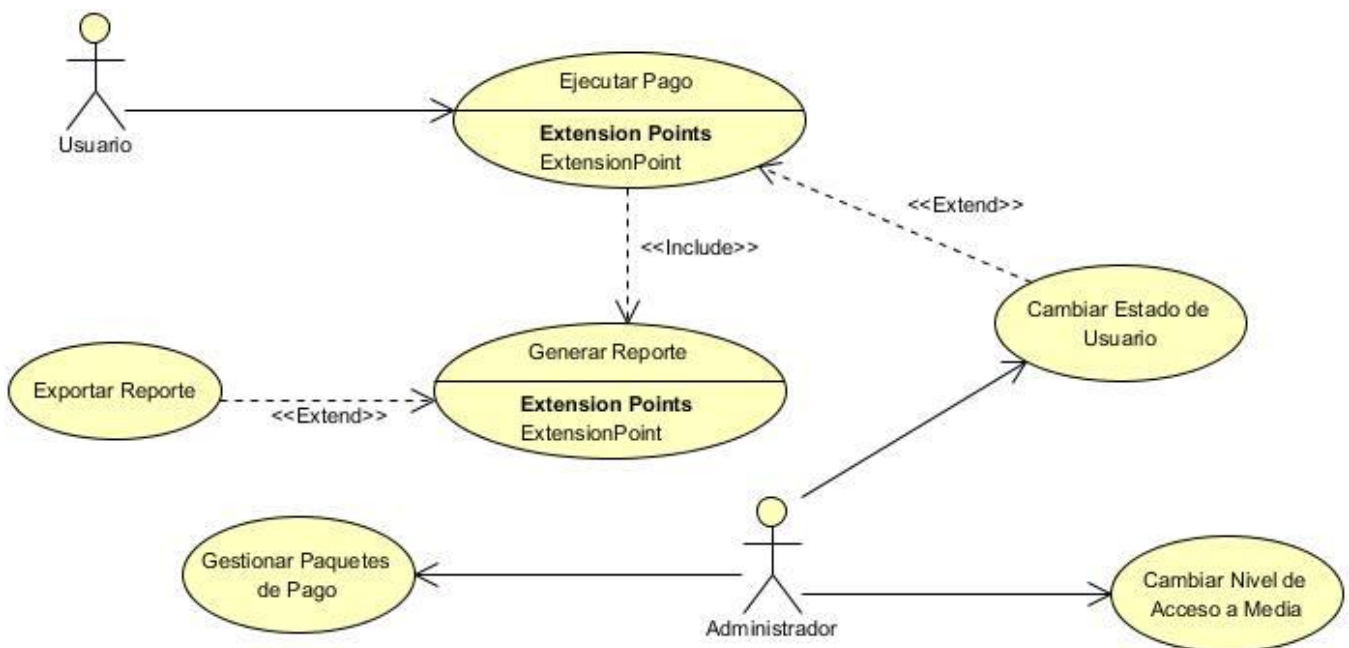
**Usuario:** Persona que interactúa con la pasarela de pago, permitiendo ejecutar las funcionalidades y principal responsable de estas.

**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.

#### 2.3.2 Diagramas de casos de uso

Se define un caso de uso de manera precisa como sigue: Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto (Jacobson, y otros, 2000).

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.



**Figura 4: Diagrama de casos de uso del sistema**

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

#### 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

## Capítulo 2: Análisis y Diseño

<b>Referencias</b>	RF2
<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

## Capítulo 2: Análisis y Diseño

	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>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	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.

### 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, RF3
<b>Prioridad</b>	Crítico

### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1 El usuario presiona el vínculo "Realizar pago" de su interfaz de trabajo.	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.
3 El usuario llena los campos referentes a su transacción y presiona el botón "Enviar"	4 El sistema valida la información y la codifica para que viaje de forma segura hasta el servidor. 5 El sistema envía la información encriptada al banco y espera por el código de respuesta. 6 Se ejecuta el caso de uso "Cambiar estado de usuario".

## Capítulo 2: Análisis y Diseño

	7 Se ejecuta el caso de uso “Generar reporte”.
<b>Flujos Alternos</b>	
	4.1 Si existen campos en blanco o incorrectos el sistema muestra un mensaje de error.
	6.1 Si el código de respuesta recibido es diferente de pago exitoso no se ejecuta el caso de uso “Cambiar estado de usuario”. 6.2 Se ejecuta el caso de uso “Generar reporte”.
<b>Prototipo de Interfaz</b>	
<b>Poscondiciones</b>	El usuario es registrado como <i>Premium</i> .

### 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>	RF4
<b>Prioridad</b>	Secundario
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1 El usuario ejecuta el caso de uso “Ejecutar pago”.	2 El sistema recibe un código de respuesta desde el servidor con información de la solicitud de pago realizada con anterioridad.
	3 El sistema confecciona y muestra un



## Capítulo 2: Análisis y Diseño

---

---

	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.

### 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>	RF5
<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 "Generar" del reporte mostrado.	2. El sistema genera un archivo PDF con la información del reporte.
<b>Poscondiciones</b>	Queda un archivo PDF con la información de la transacción.

### 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>	RF6
<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

## Capítulo 2: Análisis y Diseño

	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.

### 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>	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 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	2.2 El sistema vuelve al paso 2 del flujo

---

---

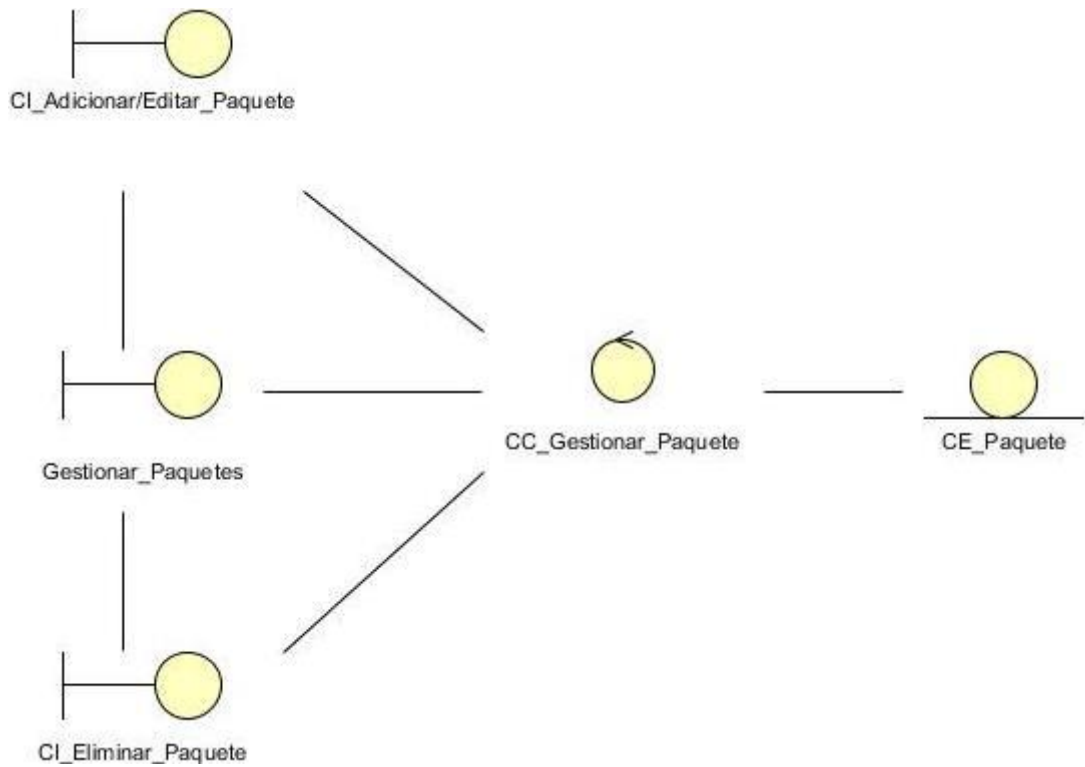
presiona el botón "Cancelar".	normal de eventos.
<b>Prototipo de Interfaz</b>	
<b>Poscondiciones</b>	El archivo multimedia cambia su nivel de acceso.

### 2.4 Modelo de análisis

El modelo de análisis es una aproximación al modelo del diseño. En este modelo hay un refinamiento de los requisitos, sin embargo no se tiene en cuenta el lenguaje de programación que se va a utilizar en la construcción de la aplicación, debido a que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución (Universidad Politécnica de Valencia, 2001).

A continuación se refleja el diagrama de clases de análisis correspondiente a los casos de uso descritos anteriormente.

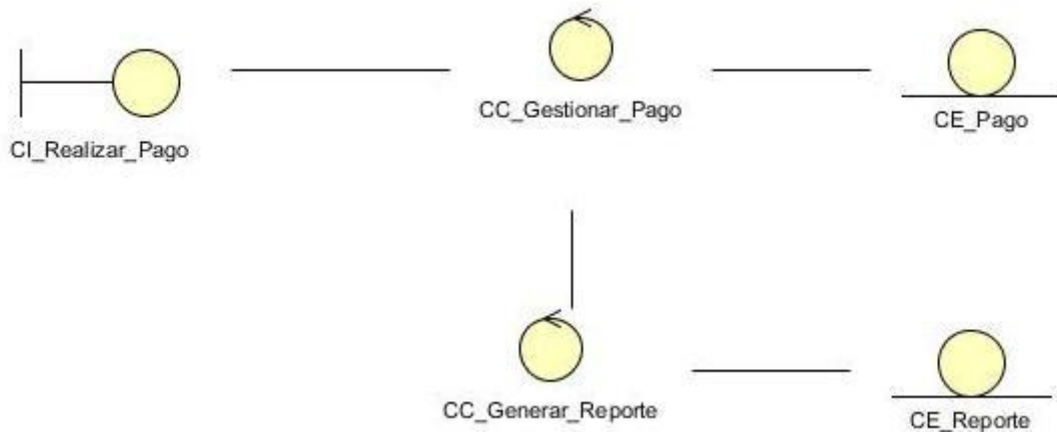
#### 2.4.1 Diagrama de clases del análisis



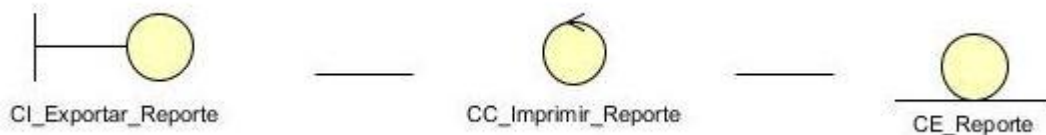
**Figura 5: DCA Gestionar paquetes de pago**



**Figura 6: DCA Cambiar estado de usuario**



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



**Figura 8: DCA Exportar reporte**



**Figura 9: DCA Cambiar Nivel de Archivo Multimedia**

### 2.4.2 Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva a modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos. Todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases se describe la forma en que grupos de objetos colaboran para proveer un comportamiento.

Mientras que un diagrama de casos de uso presenta una visión externa del sistema, las funcionalidades de dichos casos de uso se recogen como un flujo de eventos y estas se utilizan para lograr representar las interacciones entre objetos (Universidad de Chile, 2005).

### Diagramas de colaboración

Un diagrama de colaboración es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración, también llamados diagramas de comunicación, muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de comunicación no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes.

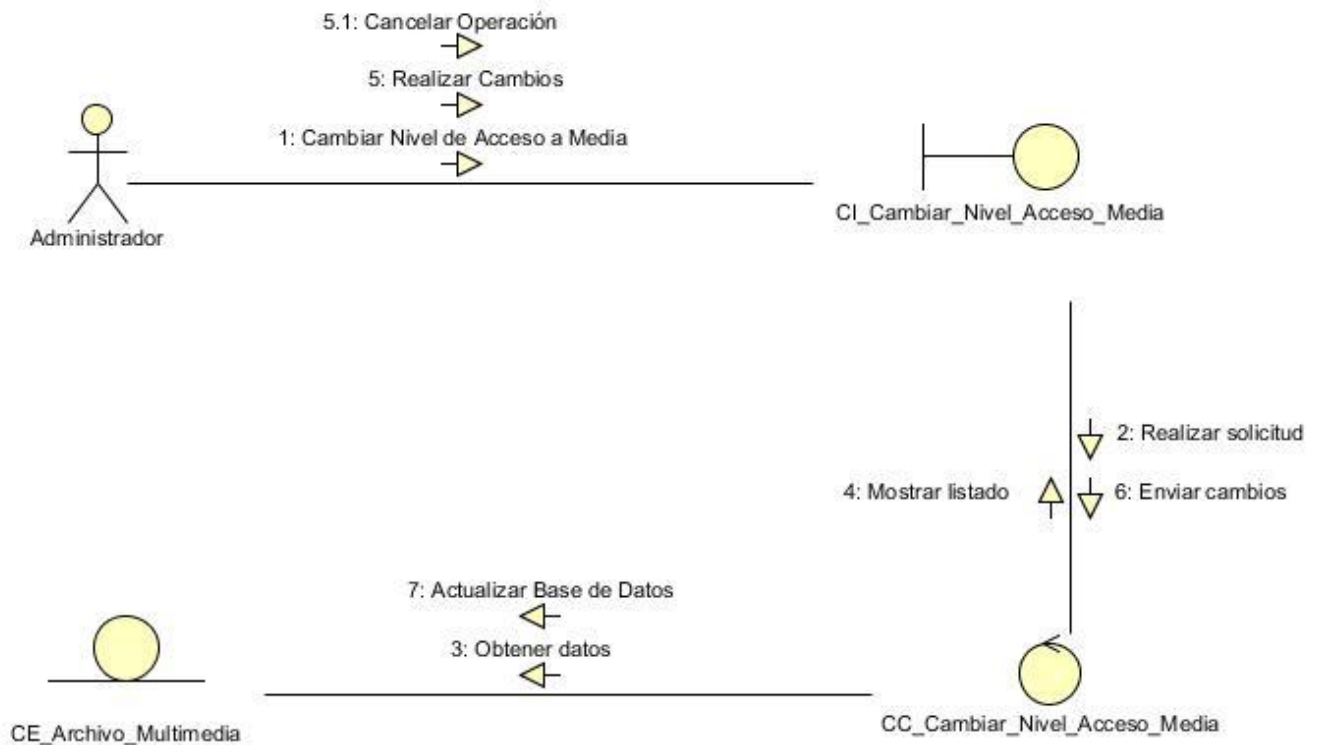
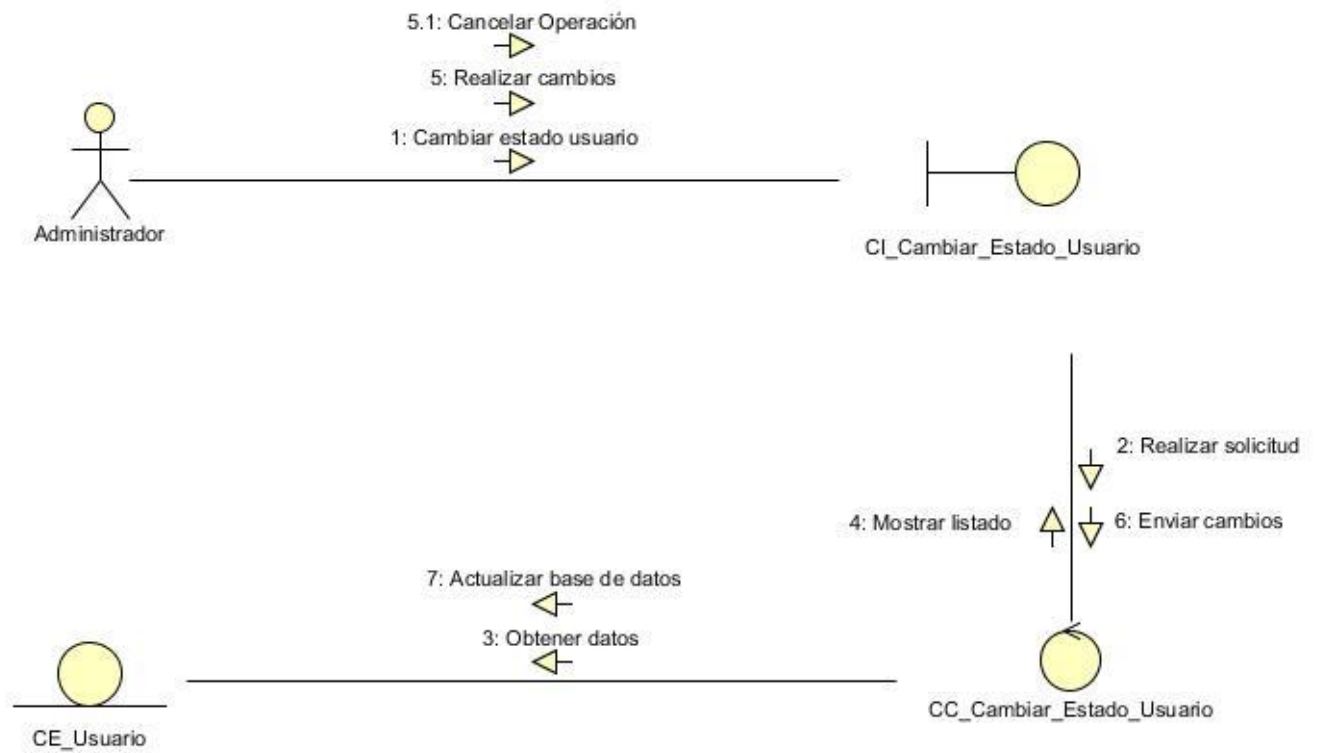
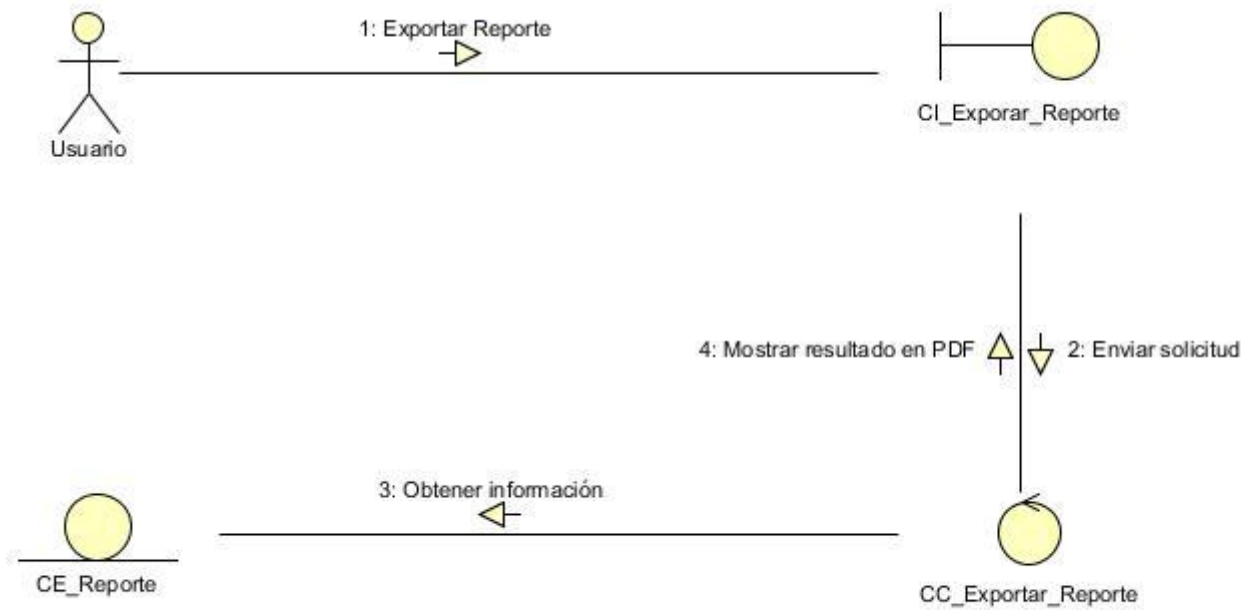


Figura 10: DC Cambiar nivel de media

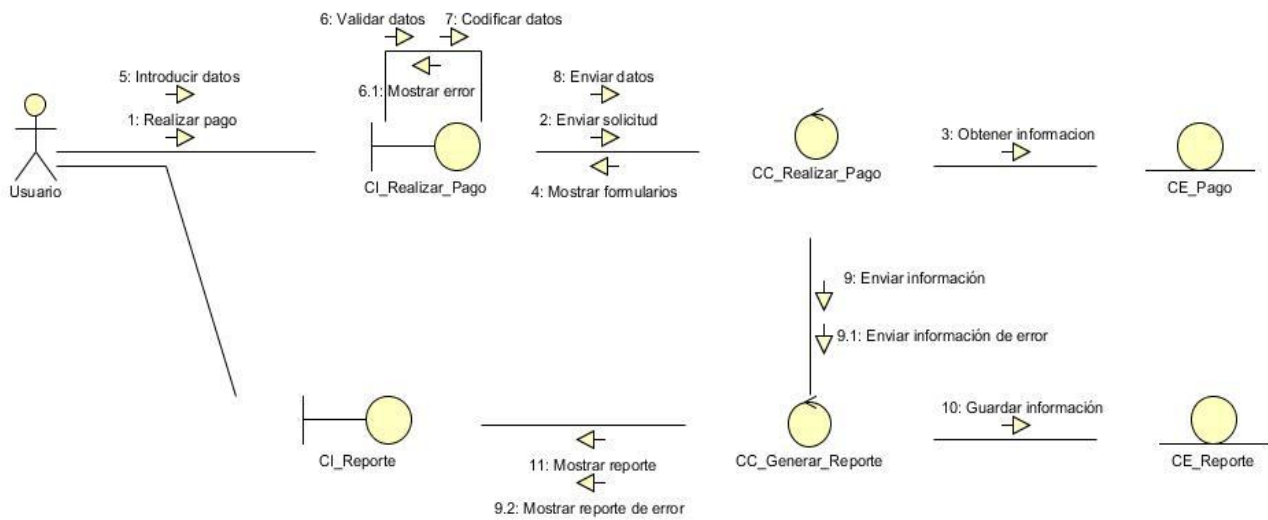


**Figura 11: DC Cambiar estado de usuario**





**Figura 12: DC Exportar reporte**



**Figura 13: DC Realizar pago y Generar reporte**

### Diagramas de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso, contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

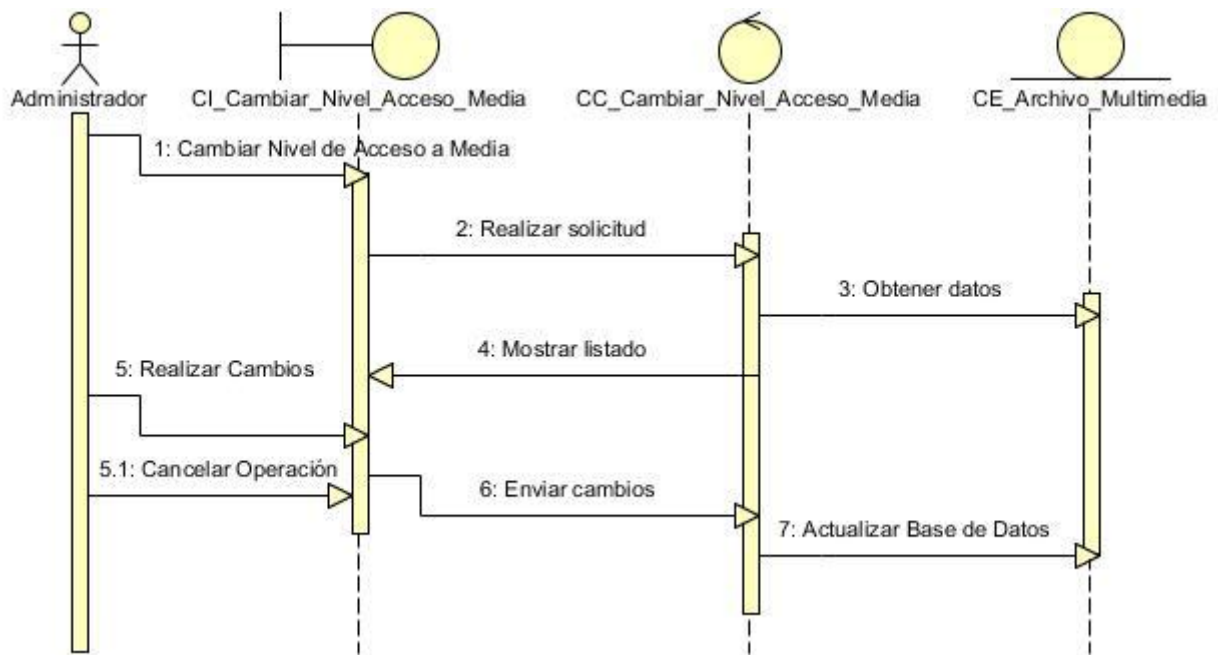
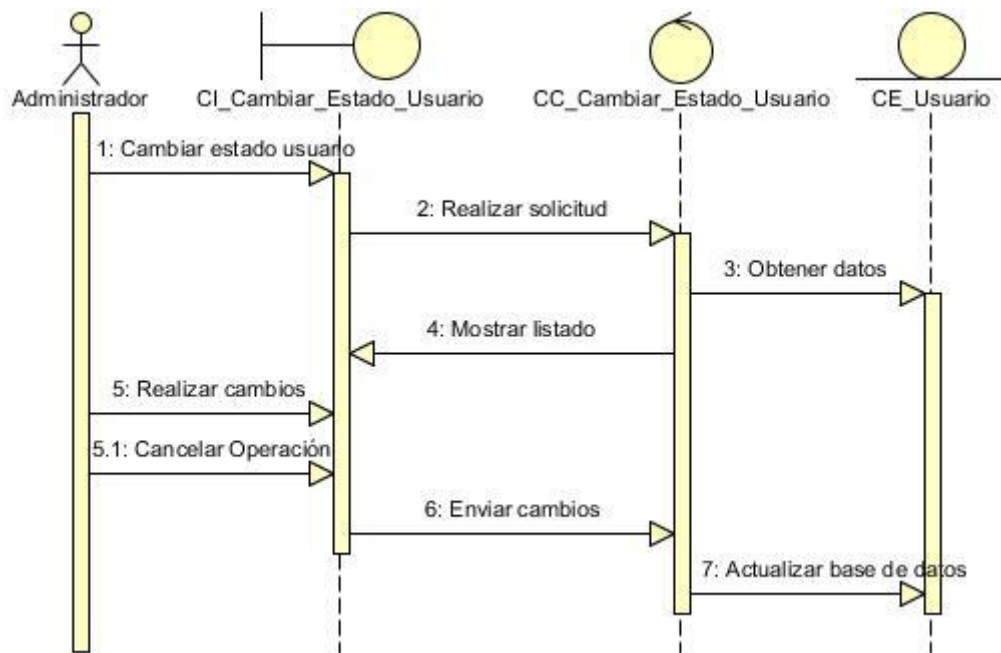
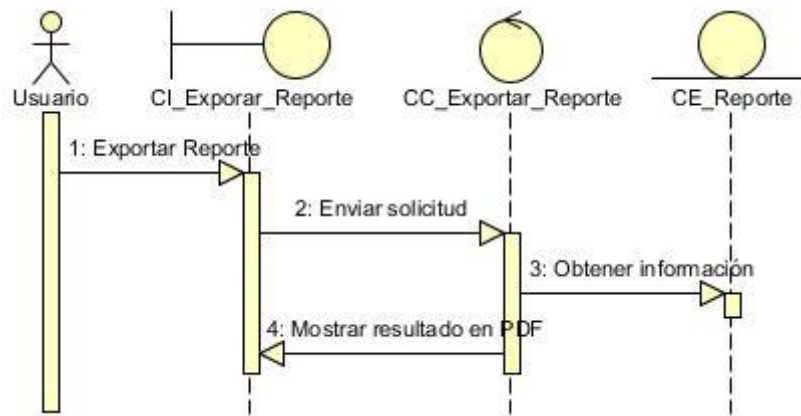


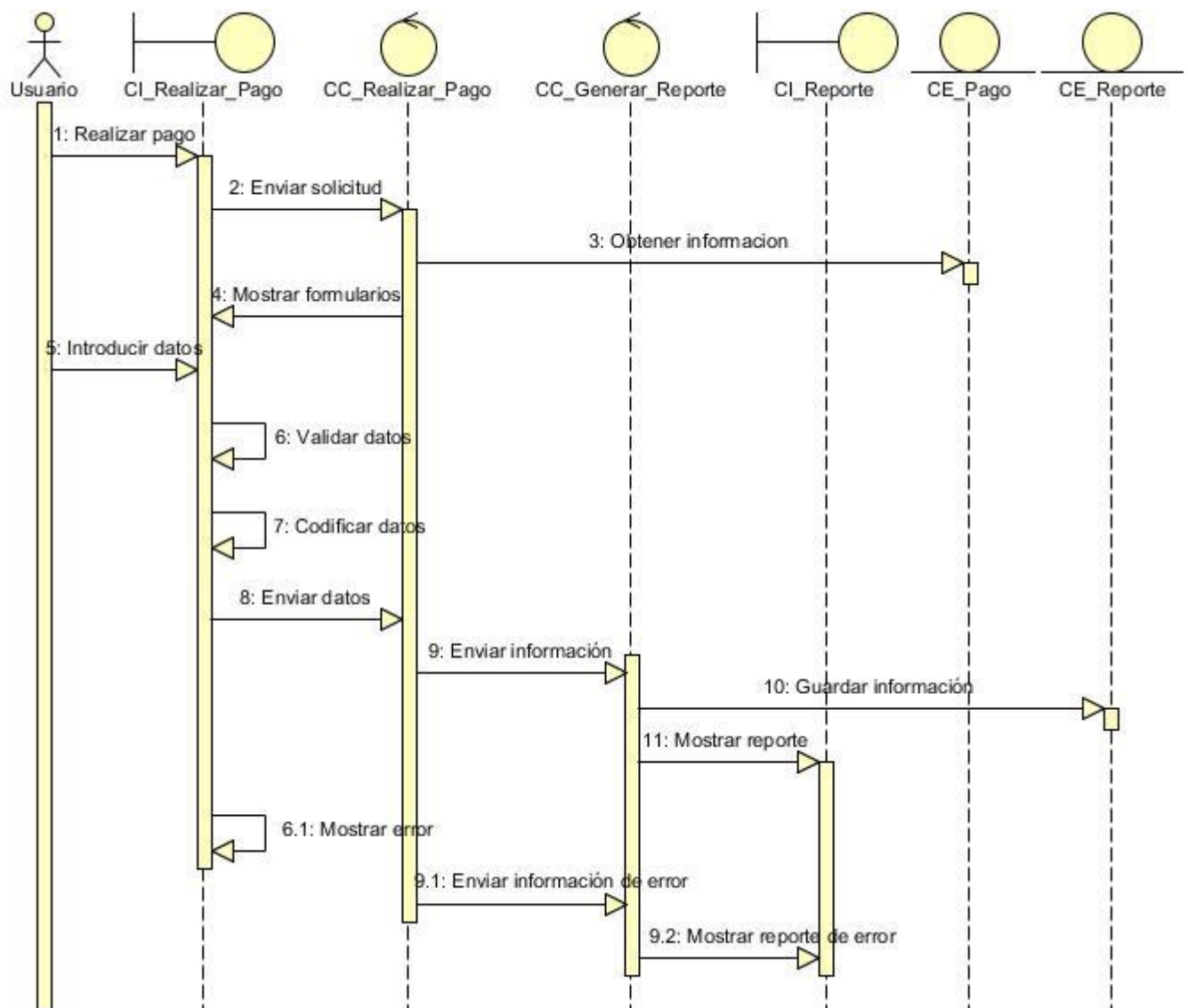
Figura 14: DS Cambiar nivel de media



**Figura 15: DS Cambiar estado de usuario**



**Figura 16: DS Exportar reporte**



**Figura 17: DS Realizar pago y Generar reporte**

### 2.5 Modelo de diseño

En el diseño se modela el sistema y se encuentra su forma, incluida la arquitectura, para que soporte los requisitos, incluyendo los requisitos no funcionales y otras restricciones que se le suponen. Los propósitos del diseño en concreto son:

- Adquirir una comprensión en profundidad de los aspectos relacionados a los requisitos, funcionales y no, además de otras restricciones relacionadas a los lenguajes de programación, sistema operativo, tecnologías, entre otros.
- Crear una entrada y punto de partida para las actividades de implementación siguientes, capturando los requisitos o subsistemas individuales, interfaces y clases.
- Ser capaces de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo en diferentes momentos, teniendo en cuenta la concurrencia (Jacobson, y otros, 2000).

### 2.5.1 Diagramas de Clases del Diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. La definición de clase incluye definiciones para atributos y operaciones (Universidad de Chile, 2005).

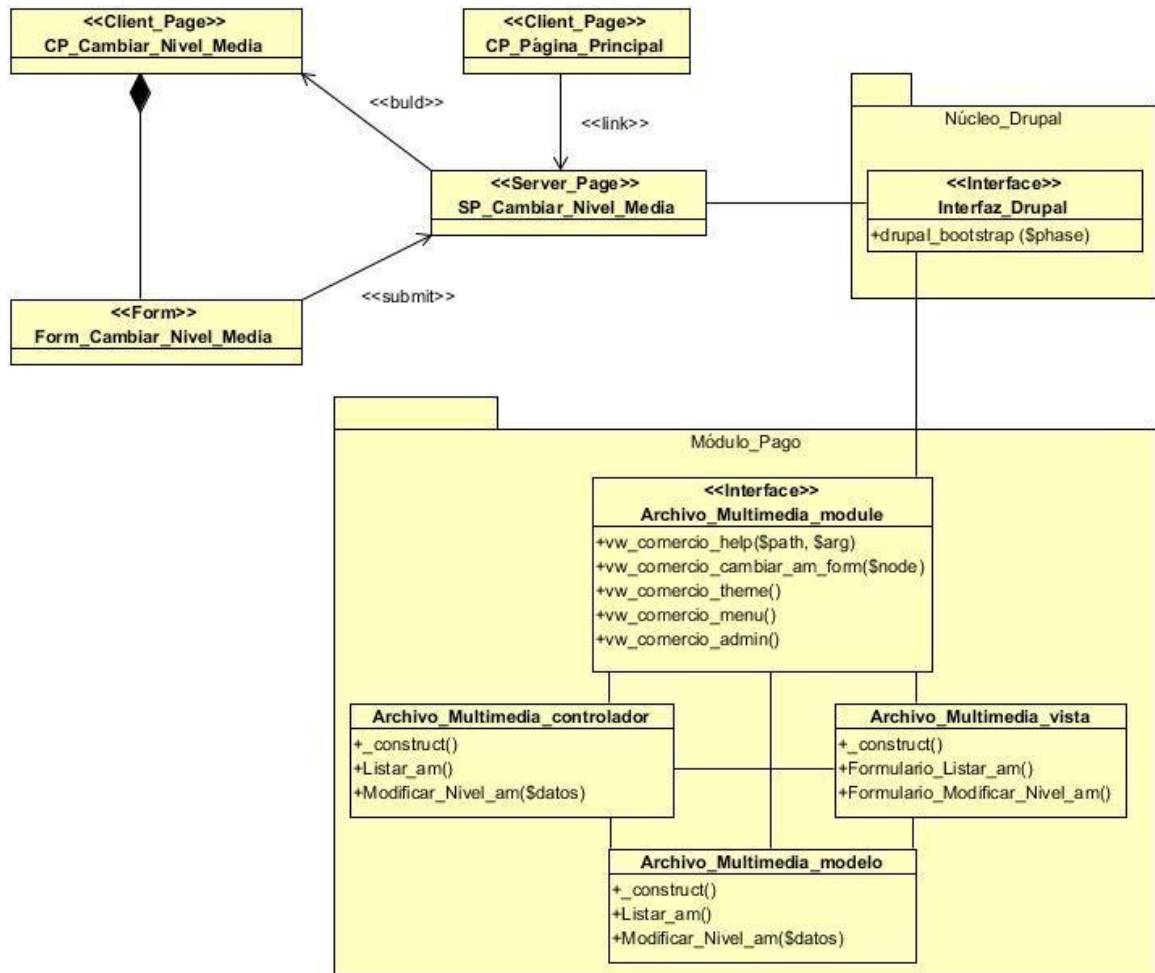


Figura 18: DCD Cambiar nivel de archivo multimedia

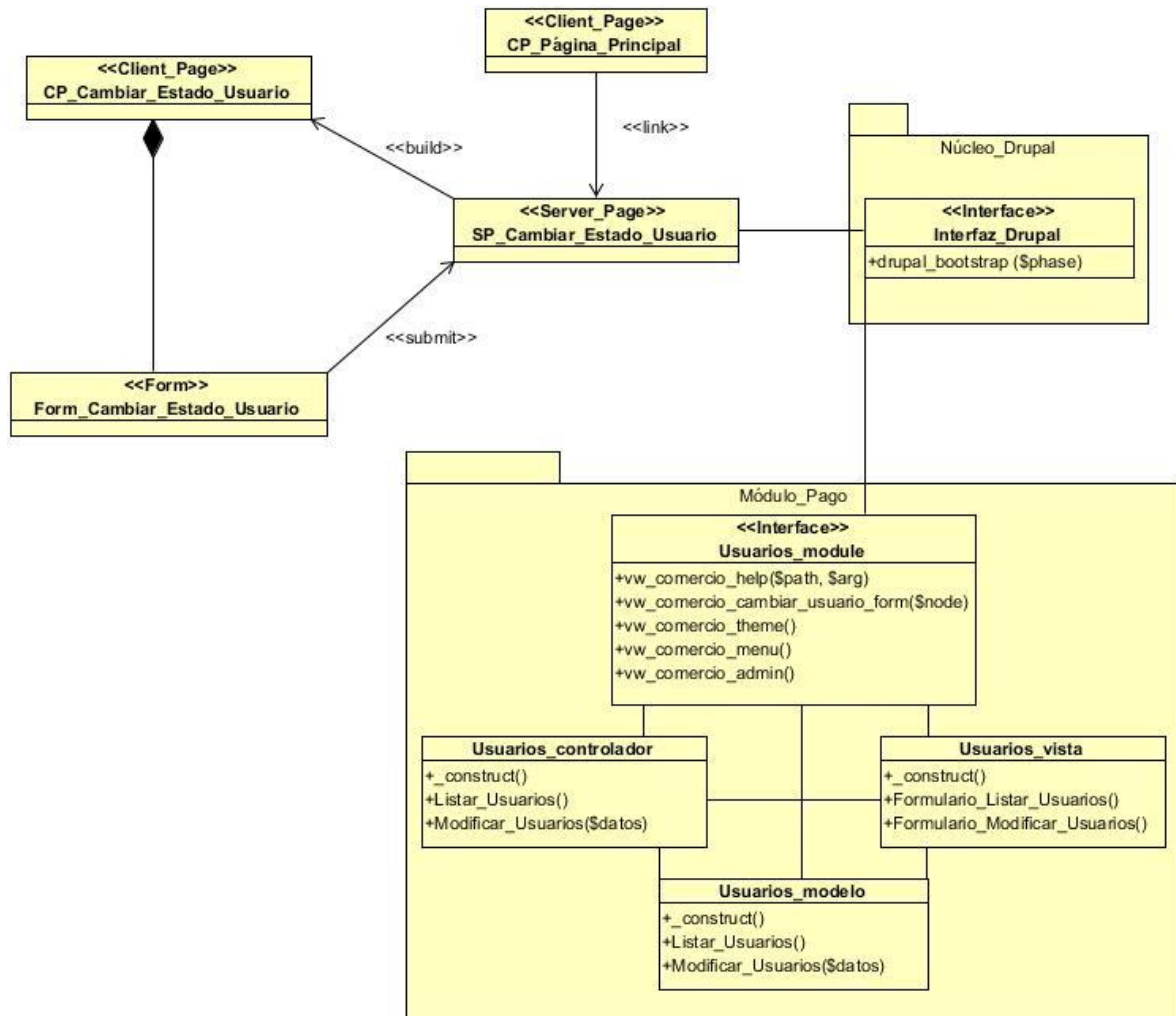


Figura 19: DCD Cambiar estado de usuario

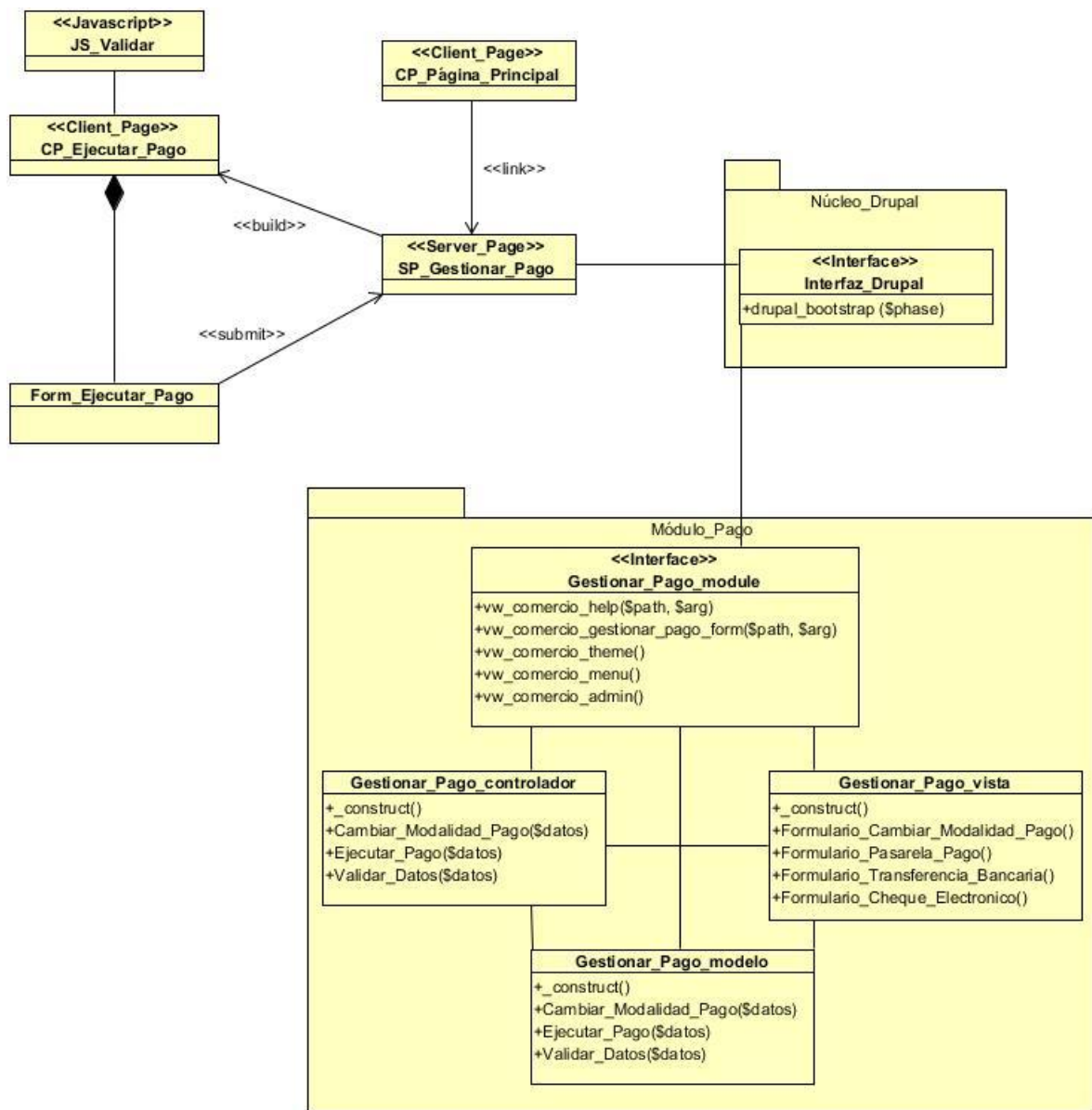


Figura 20: DCD Gestionar pago

### 2.6 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 (Bass, y otros, 2005).

### 2.6.1 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 negocio en tres componentes distintos. El patrón de llamada y retorno 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.

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

### 2.7 Conclusiones

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 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 posibilitó obtener una visión de cómo sería la interacción de los roles identificados con las funcionalidades que brindará el sistema.

## Capítulo 2: Análisis y Diseño

---

---

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.

## Capítulo 3: Implementación y pruebas

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.

### 3.1 Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software y muestran la organización y las dependencias entre un conjunto de componentes, sean éstos componentes fuentes, binarios o ejecutables. Los componentes software tienen tipo, que indica si son útiles en tiempo de compilación, enlace o ejecución.

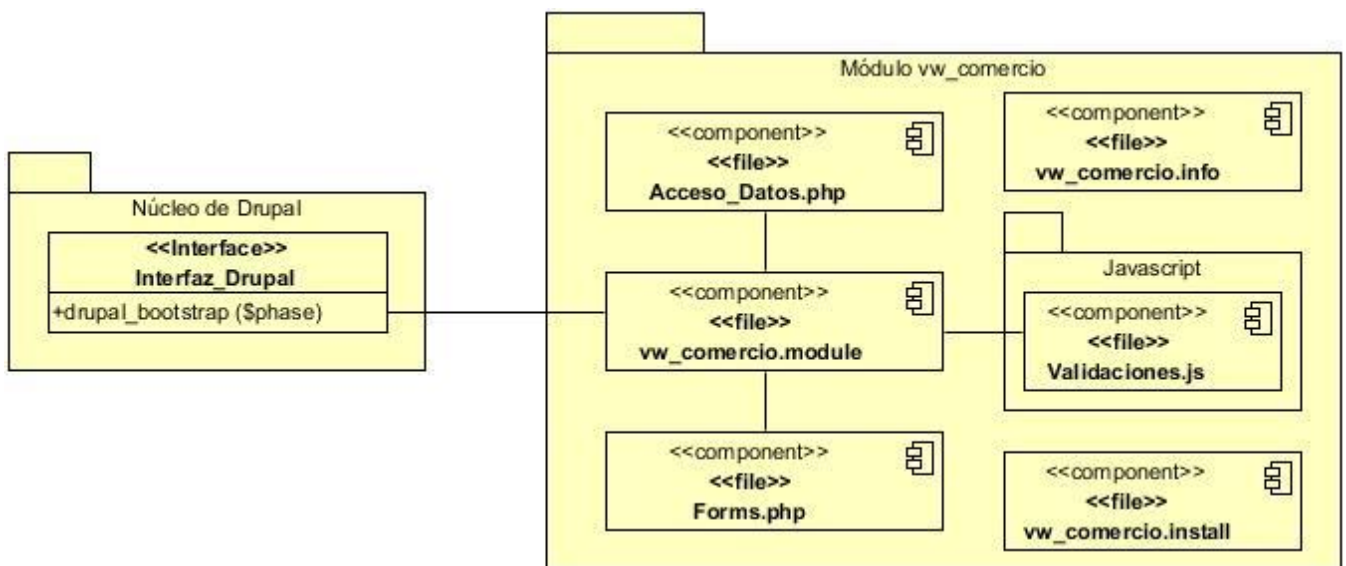


Figura 21: DC Gestionar Paquetes

## Capítulo 3: Implementación y pruebas

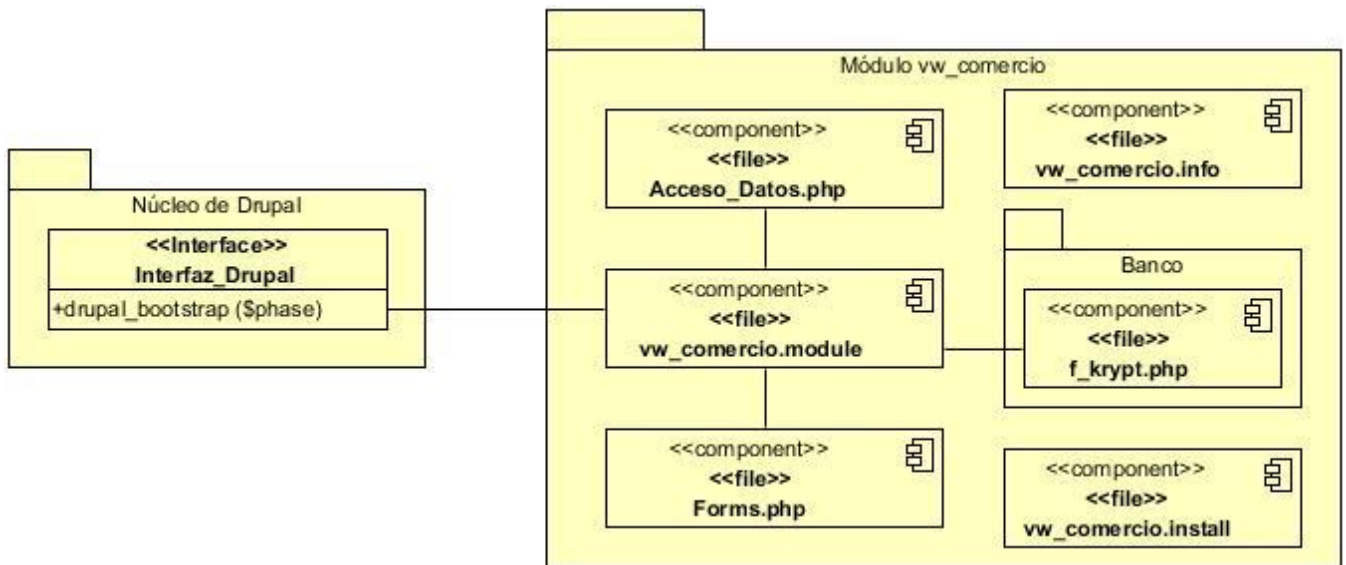


Figura 22: DC Ejecutar Pago

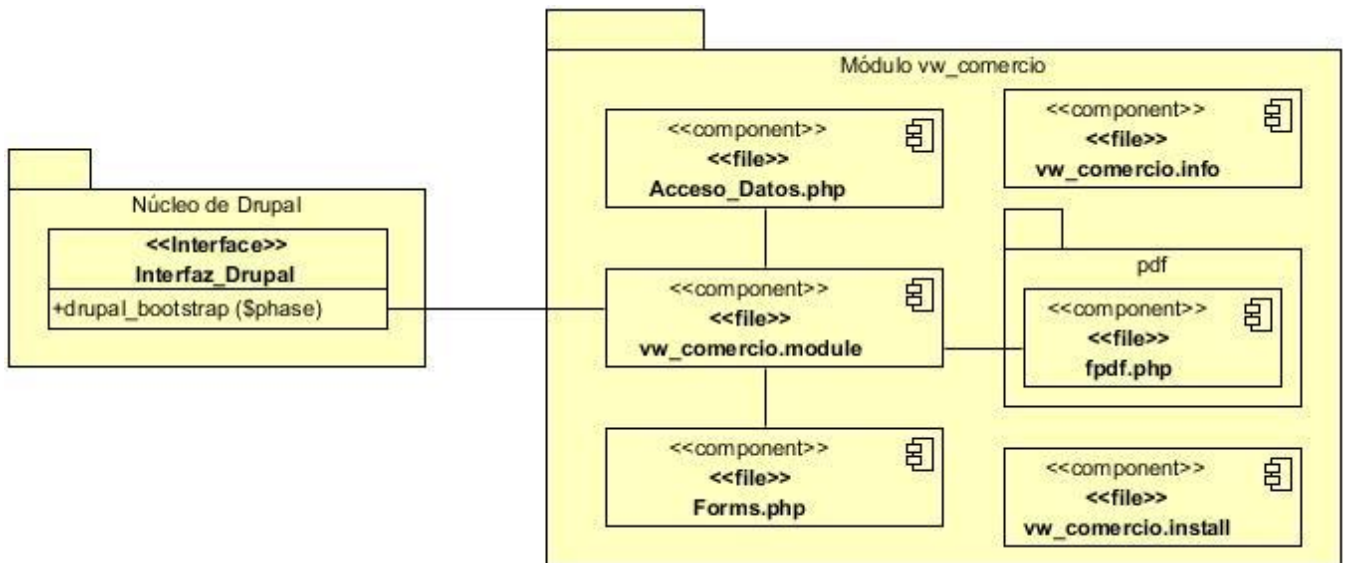


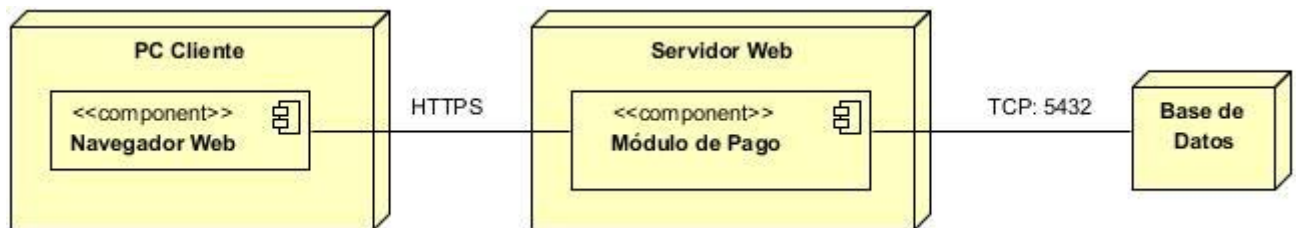
Figura 23: DC Exportar a PDF

### 3.2 Diagrama de despliegue

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema la estructura de los

## Capítulo 3: Implementación y pruebas

elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP.



**Figura 24: Diagrama de despliegue**

### 3.3 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.

## Capítulo 3: Implementación y pruebas

### 3.3.1 Pruebas de caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Para validar las principales funcionalidades del componente se realizaron pruebas de caja negra para las cuales fue necesario crear cuatro casos de prueba.

#### 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 componente 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 usuario 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.

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

## Capítulo 3: Implementación y pruebas

		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

## Capítulo 3: Implementación y pruebas

		un error durante el proceso.
--	--	------------------------------

### 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.
2	descripcion	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:

#### SC1: Adicionar Paquete de Pago

Id escenario	nombre_paquete	descripción	cant_dias	precio	Respuesta del sistema	Resultado de la prueba
EC1.1: Adicionar Paquete satisfacto	V "3x2, 5x3, paq1"	N/A	V "Número"	V "Número"	El sistema almacena los datos del paquete y	Satisfactoria



## Capítulo 3: Implementación y pruebas

riamente.					muestra un mensaje indicando que ha ocurrido satisfactoriam ente.	
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		

### 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

## Capítulo 3: Implementación y pruebas

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.

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.
	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.

Descripción de variables

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
-----	------------------	---------------	------------	-------------

## Capítulo 3: Implementación y pruebas

1	tarjeta_credito	Campo de texto	No	Número de 12 dígitos asociado a la tarjeta de crédito a utilizar.
2	codigo_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:

SC1: Ejecutar pago

Id escenario	tarjeta_credito	codigo_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	Satisfactoria
	V	I	V		
	V	V	I		
	I	I	I		

## Capítulo 3: Implementación y pruebas

				que hay campos en blanco o incorrectos.	
--	--	--	--	---	--

### 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.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Cambiar estado de usuario.	EC1.1: Cambiar estado de usuario 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”.

Descripción de variables

No.	Nombre del	Clasificación	Valor	Descripción
-----	------------	---------------	-------	-------------

## Capítulo 3: Implementación y pruebas

---

---

	campo		Nulo	
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:

SC1: Cambiar estado de usuario

Id escenario	estado_usuario	fecha_fin	Respuesta del sistema	Resultado de la prueba
EC1.1: Cambiar estado de usuario satisfactoriamente.	N/A	N/A	El sistema guarda los cambios, actualiza la base de datos y muestra un mensaje satisfactorio.	Satisfactoria
EC1.2: Cancelar cambio estado de usuario.	N/A	N/A	El sistema vuelve al estado anterior del flujo normal de eventos.	Satisfactoria

### 3.3.2 Resultados de las pruebas

Fueron probados tres casos de uso, solo aquellos 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.

### 3.4 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 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 component4es, 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 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 premiun 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.



### Referencias

Código	Título
[1]	1. ingeniería\1.1 requisitos\Especificación de requisitos v2.0.doc

**Asociación de Bancos de México, A.C. 2003.** ABM Asociación de Bancos de México. *ABM Asociación de Bancos de México*. [En línea] 2003. [Citado el: 22 de Noviembre de 2011.] <http://www.abm.org.mx/>.

**Flanagan, David. 2001.** *JavaScript: The Definitive Guide*. s.l. : Fourth Edition, 2001. 0596000480.

**Fowler, Martin y Sccott, Kendall. 1999.** *UML Gota a Gota*. s.l. : Pearson Prentice Hall, 1999. 9684443641.

**IDSOLUTIONS, A.C. 2004.** IDSOLUTIONS: Ingeniería y Desarrollo. *IDSOLUTIONS: Ingeniería y Desarrollo*. [En línea] 2004. [Citado el: 22 de Noviembre de 2011.] <http://www.idsolutions.com.mx/>.

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

**Laudon, Kenneth C. 2009.** *E-commerce: Negocios, tecnología y sociedad*. Mexico DF. : Pearson Prentice Hall, 2009. 9786074420494.

**Lidera Network, S.L. 1999.** Lidera.net. *Lidera.net*. [En línea] Lidera Network, 1999. [Citado el: 22 de Noviembre de 2011.] <http://www.lidera.net/>.

**MI PROPIO NEGOCIO INC. 1992.** Mi Propio Negocio. *Mi Propio Negocio*. [En línea] 1992. <http://www.myownbusiness.org/espanol/s9/index.html>.

**NetBeans Community. 2000.** NetBeans. *NetBeans*. [En línea] NetBeans Community, 2000. [Citado el: 30 de Noviembre de 2011.] <http://netbeans.org/>.

## Referencias bibliográficas

---

---

**The PHP Group. 2001.** PHP: Hypertext Preprocessor. *PHP: Hypertext Preprocessor*. [En línea] The PHP Group, 2001. [Citado el: 25 de Noviembre de 2011.] <http://www.php.net/>.

**Visual Paradigm International. 2004.** Visual Paradigm. *Visual Paradigm*. [En línea] 2004. [Citado el: 25 de Noviembre de 2011.] <http://www.visual-paradigm.com/>.

**Bass, Len, Clements, Paul and Kazman, Rick. 2005.** *Arquitectura de Software en la práctica*. s.l. : Addison-Wesley, 2005. 0321154959.

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

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

**Pressman, Roger. 2003.** *Ingeniería de Software, un enfoque práctico*. New York : s.n., 2003.

**Universidad de Chile. 2005.** Departamento de Ciencias de la Computación. [Online] Universidad de Chile, 2005. <http://www.dcc.uchile.cl/>.

**Universidad Politécnica de Valencia. 2001.** Departamento de Sistemas Informáticos y Computación. [Online] Universidad Politécnica de Valencia, 2001. <http://www.dsic.upv.es/>.

## 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.

---

---

**Anexos**

## Glosario

**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.

**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.

**Metodología de desarrollo de software:** a grandes rasgos una metodología de desarrollo de software define en un proyecto **quién** debe hacer **qué** cosa, **cuándo** se le dará cumplimiento y **cómo** lo va a hacer.

**UML:** Del inglés *Unified Modeling Language*, en castellano Lenguaje Unificado de Modelado es una de las herramientas más importantes en el mundo actual del desarrollo de sistemas. Es un lenguaje visual para especificar, construir y documentar los artefactos de los sistemas de software y es aplicado en sistemas orientados a objetos.

**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.

**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.