Universidad de las Ciencias Informáticas

Facultad 6

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

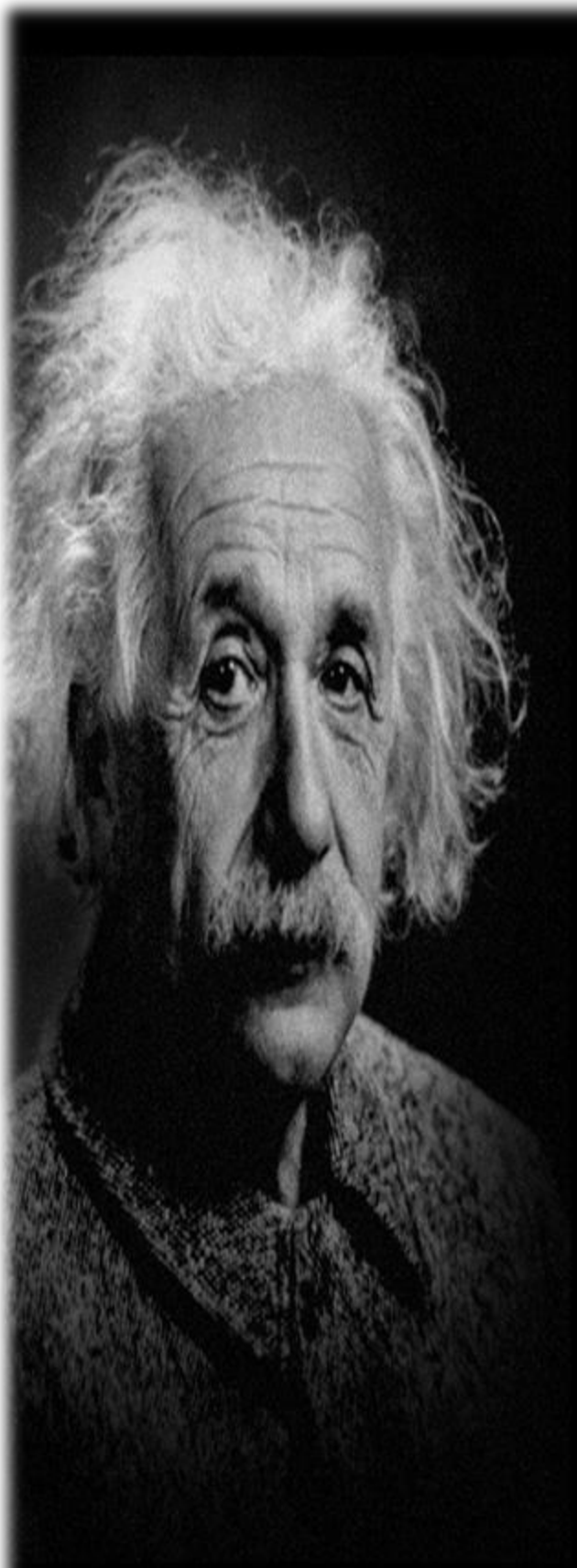
Título: “Módulo de pasarela de pago para la plataforma AGORAV”.

Autor: Raciel Correa Barbán.

Tutor: Ing.Pavel Mena Nodal.

La Habana, 2015

“Año 57 de la Revolución”



El genio se hace con
1% de talento,
y un 99% de trabajo.

Albert Einstein

Declaración de autoría

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

Para que así conste firmo la presente a losdías del mes dedel año 2015.

Firma del Autor	Firma del Tutor
Raciel Correa Barbán	Ing. Pavel Mena Nodal

Datos de contacto

Tutor: Ing. Pavel Mena Nodal.

Centro de trabajo: Universidad de las Ciencias Informáticas.

Cargo: Arquitecto y desarrollador

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas.

Año de graduación: 2014.

Institución en la que se graduó: Universidad de las Ciencias Informáticas.

Correo electrónico: pmena@uci.cu

Agradecimientos

A las personas más importantes en mi vida, mis padres, que siempre me han guiado y me han hecho ser una mejor persona.

A mi mamá Argelia Barbán Galafet y a mi papá Santo Correa Vásquez por el sacrificio y el esfuerzo que siempre hicieron para que pudiera llegar hasta aquí.

A mis abuelos por haberme dado siempre tanto cariño y preocuparse por mí, siendo mis segundos padres.

A mi abuelo Argelio Barbán Plat por ser siempre un ejemplo como hombre y como persona, por su cariño y su respeto.

A mi abuela Elvia Galafet García por siempre educarme y darme ese cariño inmenso que siempre me dio.

A Charo por ser como una madre para mí y siempre impulsarme en los estudios, por la preocupación y el cariño, por verme como un hijo para ella.

A todos mis primos por ser los hermanos que siempre quise, por siempre estar a mi lado.

A mis tíos por el afecto que siempre me han dado.

A mi novia por estar siempre a mi lado y ayudarme a que hoy me encuentre aquí, por todo el cariño que me brindó en todos estos años.

A todos mis amigos, los del barrio que vienen conmigo desde la infancia y a los que conocí en estos maravillosos 5 años.

A la familia de mi novia por el apoyo que siempre me han brindado y acogerme como uno más de su casa.

A mi tutor Pavel Mena Nodal por ser un gran tutor y amigo, por enseñarme tanto en tan poco tiempo. Por siempre estar ahí brindándome una solución a cada problema y ayudarme tanto.

A todas las personas que de una forma u otra han hecho posible que hoy sea un profesional y una mejor persona en la vida. A todos muchas gracias.

Dedicatoria

En especial a mis padres que siempre quisieron que fuera una persona de bien y que triunfara en la vida. Hoy me siento orgulloso de recompensarlos de esta manera por todo lo que han hecho para que este día llegara.

Resumen

En la actualidad, el comercio electrónico se ha convertido en la principal herramienta para las empresas llevar a cabo la comercialización de sus productos a través de Internet. En la Universidad de las Ciencias Informáticas (UCI), el proyecto de Catalogación y Publicación de Medias (CPM)desarrolla la plataforma AGORAV para la transmisión de medias a través de la web. El sistema cuenta actualmente con un módulo que se encarga de realizar la acción de pago por los servicios de visualización y descarga de las medias solicitadas por los usuarios. Sin embargo, la realización de dicho proceso de forma local, limita el campo de comercialización de la plataforma, el uso de bancos online y la utilización de diferentes modos de pago. El objetivo de la presente investigación esel desarrollo de un módulo de pasarela de pago, que permita a los usuarios la selección de diferentes modos de pagos y que esté orientado a bancos externos a la entidad.Para el desarrollo del módulo se utilizó el Sistema Gestor de Contenido Drupal y como metodología de desarrollo AUP, que rige todo el proceso de ingeniería de software. El módulo desarrollado posee funcionalidades como: la gestión del precio de los archivos multimedia, la administración del carrito de compra, la selección del modo de pago por los usuarios, además de generar un historial de transacciones luego de haber efectuado el pago.El desarrollo de este módulo contribuye a incrementar la comercialización de los productos ofrecidos por la plataforma AGORAV.

Palabras claves: medias, pasarela de pago, AGORAV.

Abstract

Nowaday, e-commerce has become the main tool for companies to carry out the marketing of its products over the Internet. At the University of Informatic Sciences (UIS), the project of cataloging and publishing of medias (CPM) develops AGORAV platform for the transmission of medias through the web. The system now has a module that is responsible for conducting the action of payment for the services of viewing and downloading of the medias requested by users. However, performing this process locally limits the field of marketing platform using online banks and different payment methods. The objective of this research is the development of a payment gateway module that allows users to select different modes of payments and is oriented to external banks to the entity. For module development DrupalContent Management System was used as development methodology AUP, which governs the whole process of software engineering. The module developed has features such as: managing the cost of media, administration cart, selecting the mode of payment by users, and generate transaction history after having paid. The development of this module contributes to increas the marketing of the products offered by the AGORAV platform.

Keywords: AGORAV, medias, payment gateway.

Índice

Índice de Figuras

Índice de Tablas

Introducción

Internet es una red mundial formada por millones de ordenadores de todo tipo y plataformas, conectados entre sí por diversos medios de comunicación, cuya función principal es la de localizar, seleccionar e intercambiar información desde el lugar en donde se encuentra hasta aquel donde haya sido solicitada o enviada(Internet y la sociedad Red, 2001) .

A través de los años, el crecimiento que tuvo internet y la aceptación dentro de la sociedad hizo inevitable que en esta se introdujera la economía y formara parte fundamental para las empresas, lo que conllevó al crecimiento de las mismas a través del comercio electrónico. El comercio electrónico a través de internet es otra forma de ofrecer productos y servicios; permite informar, publicar, vender, cobrar y dar servicios posventa(Dans, 2007).

Cuba no se encuentra ajena a esta tendencia de expandir el comercio electrónico en la medida que el desarrollo tecnológico y la infraestructura lo permite. Es por ello que la Universidad de las Ciencias Informáticas (UCI) prevé el desarrollo de sistemas que integren estas funcionalidades para facilitar el acceso a productos diversos de acuerdo a las condiciones y solicitudes de los clientes. Particularmente el Centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6 desarrolla como uno de sus productos una plataforma web para la transmisión de videosque ha sido implantada en la propia Universidad y es del interés de varios clientes externos.

La Plataforma AGORAV tiene una arquitectura modular que permite el desarrollo creciente de funcionalidades y está respaldada por un equipo; que basado en los procesos del negocio y otros sistemas de su tipo han identificado que tiene carencias para la comercialización online de sus recursos audiovisuales, y de mecanismos para garantizar el pago electrónico de los servicios de visualización y descarga de dichos recursos. Actualmente los usuarios de la plataforma para poder hacer uso de los recursos antes mencionados se ven obligados a depositar personal y físicamente su dinero en una cuenta local de AGORAV. Sin embargo, no se les brinda a dichos usuarios la posibilidad de realizar pagos online, lo cual limita las potencialidades del comercio electrónico actual. Por otra parte, el sistema carece de mecanismos para la agrupación de varios productos (carro de compras), esto reduce en cierta medida la adquisición de materiales audiovisuales.

Lo expuesto previamente limita a la plataforma AGORAV a expandir su campo de comercialización y de aceptación entre los usuarios. Por lo que se identifica como **problema a resolver:** ¿Cómo lograr el

cobroonline por concepto de visualización y descargas de los productos audiovisuales publicados en la Plataforma AGORAV?

Para darle solución al problema planteado se define como **objeto de estudio:** El proceso de cobro online de productos audiovisuales, delimitando como **campo de acción:** Cobro online de productos audiovisuales en la plataforma AGORAV a través de pasarela de pago.

Se plantea como **objetivo general:** Desarrollar un módulo de pasarela de pago para la plataforma AGORAV que garantice el cobro online por concepto de visualización y descarga de los productos audiovisuales.

Para darle solución al objetivo se definen las siguientes **tareas de la investigación:**

- Caracterización del proceso de cobro online a través de pasarelas de pago a nivel nacional e internacional.
- Identificación de los procesos que se realizan para efectuar el pago por los servicios que brinda la plataforma AGORAV.
- Descripción de las herramientas y tecnologías que se utilizarán para la implementación del módulo de pasarela de pago.
- Realización de la documentación correspondiente a todo el proceso de desarrollo.
- Implementación del módulo de pasarela de pago para la plataforma AGORAV.
- Realización de las pruebas correspondientes al módulo implementado para garantizar su correcto funcionamiento.

Se definen para guiar el proceso de investigación las siguientes **preguntas científicas:**

¿Qué variante tecnológica seleccionar para la implementación del módulo de cobro online para la plataforma AGORAV?

¿Cuáles son las características que debe cumplir el módulo para realizar un cobro seguro a través de la web?

¿El módulo desarrollado satisface la necesidad existente en la plataforma AGORAV?

Métodos de la Investigación Científica.

Métodos Teóricos: Permiten revelar las relaciones esenciales del objeto de investigación, no observables directamente(Zayas, 1995).

Analítico sintético: El análisis de un objeto se realiza a partir de la relación que existe entre los elementos que conforman dicho objeto como un todo; y a su vez, la síntesis se produce sobre la base de los resultados previos del análisis(Zayas, 1995). Este método ayudó a sintetizar el análisis realizado a toda la documentación consultada para el desarrollo de la investigación.

Método histórico-lógico: Está vinculado al conocimiento de las distintas etapas de los objetos en su sucesión cronológica; para conocer la evolución y desarrollo del objeto o fenómeno de investigación se hace necesario revelar su historia. Los métodos lógicos se basan en el estudio histórico, poniendo de manifiesto la lógica interna de desarrollo, de su teoría y hallan el conocimiento más profundo de esta, de su esencia(Zayas, 1995). Este método se utilizó para realizar un estudio de los sistemas desarrollados anteriormente en el ámbito nacional e internacional, para efectuar pagos a través de la web, sirviendo de base en la investigación e implementación del módulo de pasarela de pago para la plataforma AGORAV.

Método de modelación: La modelación opera en forma práctica o teórica con un objeto, este es justamente el método mediante el cual se crean abstracciones con vistas a explicar la realidad(Zayas, 1995). Este método se utilizó en el modelo de dominio, para comprender y describir los procesos principales de la plataforma AGORAV.

Método empírico: Revela y explica las características fenomenológicas del objeto(Zayas, 1995).

Entrevista: La entrevista es una técnica de recopilación de información mediante una conversación profesional, con la que además de adquirirse información acerca de lo que se investiga, tiene importancia educativa y depende en gran medida del nivel de comunicación entre el investigador y los participantes en la misma(Zayas, 1995). Este método se utilizó para identificar los requisitos y funcionalidades que debe cumplir el módulo a implementar, aplicándosele al líder del proyecto CPM y uno de sus desarrolladores. Ver anexo 1.

El trabajo se encuentra distribuido en tres capítulos. En el Capítulo 1 se realiza una fundamentación de la teoría de la investigación. Se describen términos técnicos de importancia y se realiza un estudio a

sistemas con características similares a las requeridas. Se concluye este capítulo con las herramientas y tecnologías a utilizar para el desarrollo del módulo. Se inicia el Capítulo 2 definiendo las características de la solución propuesta. Se muestra el diagrama de modelo del dominio, se identifican los requisitos funcionales y no funcionales con los que debe cumplir el módulo, se describen los casos de usos y actores del sistema. Se define la arquitectura del módulo y se muestran los resultados obtenidos en el desarrollo del proceso del diseño. Por último en el Capítulo 3 se abordan los temas referentes a la implementación y prueba del módulo. Se muestran los diagramas de despliegue y de componentes, así como la descripción de las pruebas realizadas al sistema y los resultados obtenidos.

Capítulo 1: Fundamentación teórica.

1.1 Introducción

En el presente capítulo se hace un análisis de los principales elementos teóricos relacionados con la investigación, permitiendo llegar a una mejor comprensión. Se hace un análisis de las herramientas, lenguajes y tecnologías que se utilizarán en el desarrollo del módulo, particularidades y motivos para su uso. Se define la metodología de desarrollo de software que se va a utilizar, las razones que permitieron arribar a tal decisión y una descripción de la misma. Se presenta un resumen de algunas soluciones ya existentes, relacionadas al problema planteado.

1.2 Conceptos del dominio

Es necesario describir algunos conceptos que se utilizan en la investigación para obtener una mejor comprensión de la situación problemática planteada.

1.2.1 Comercio electrónico

Comercio electrónico o e-commerce es una moderna metodología que da respuesta a varias necesidades de empresas y consumidores, como reducir costes, mejorar la calidad de productos y servicios, acortar el tiempo de entrega o mejorar la comunicación con el cliente. Más típicamente se suele aplicar a la compra y venta de información, productos y servicios a través de redes de ordenadores(Dans, 2007). En la presente investigación, se pone de manifiesto el comercio electrónico en la comercialización de productos audiovisuales.

1.2.2 Tarjeta de crédito

Una tarjeta de crédito es un medio de pago y a su vez una forma de financiación emitida por una entidad. Las tarjetas de crédito dejan al usuario realizar pagos u obtener dinero, hasta un límite establecido, sin necesidad de tener fondos en la cuenta bancaria en ese preciso momento. El límite del crédito disponible debe estar especificado en el contrato de la tarjeta, pero puede variar a lo largo del tiempo, con el consentimiento, tanto del titular de la tarjeta como de la entidad financiera(Monetos, 2012). En la presente investigación la tarjeta de crédito es una forma muy fácil y práctica de utilizar un crédito concedido por una entidad bancaria para que los usuarios efectúen la compra de los materiales audiovisuales. Entre las más conocidas se encuentran MasterCard, American Express, Visa y Discover.

1.2.3 Tarjeta de débito

Una tarjeta de débito constituye un medio de pago emitida por una entidad bancaria. Permiten realizar extracciones de dinero además de efectuar pagos, pero al realizar una operación, se genera un cargo directo por el importe de la compra en la cuenta corriente del cliente, por lo que es necesario que existan fondos suficientes para hacer frente al pago o a la retirada de efectivo. Los gastos que se abonan con la tarjetas se descuentan al instante del saldo en cuenta(Monetos, 2012).Este tipo de tarjeta realiza un descuento directo a la cuenta del titular al hacer uso de la misma, por ello es necesario contar con fondos suficientes para cubrir el gasto generado.

1.2.4 Pasarela de pago

Una pasarela de pago es un sistema de pago electrónico que permite la realización de pagos y transferencias entre tiendas electrónicas y entidades bancarias de manera segura. Se encarga de cifrar la información confidencial que se requiere para ejecutar transacciones bancarias por las redes y así garantizar la seguridad(Pasarela de pago para la seguridad de transacciones bancarias en líneas, 2013).Es la pieza de software que sirve para procesar los cobros y pagos de un sistema de comercio electrónico. Su misión principal es hacer que los cobros y pagos sean rápidos, sencillos y seguros, para que las transacciones de compraventa se puedan ejecutar de manera efectiva(Herrero, 2012).

1.3 Proceso de pago online de productos usando pasarela de pago

Dentro del proceso de compra en una tienda virtual la parte del software que entra en acción cuando un cliente realiza un pedido es la pasarela de pago, ésta realiza una serie de tareas para procesar la transacción de manera transparente para el comprador(Imprenta, 2015).

- 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 (Interactive Voice Response).
- Si la orden es a través de un sitio web, el navegador web del cliente cifra la información que viaja hasta el servidor web del vendedor. Esto se hace normalmente mediante cifrado SSL (Secure Socket Layer).
- El vendedor reenvía los detalles de la transacción a su pasarela de pago, el cual contiene los detalles de las cuentas de sus vendedores. Normalmente, esta es otra conexión cifrada mediante SSL al servidor de pago, almacenada en la pasarela de pago.

- La pasarela de pago que recibe la información de la transacción del vendedor la reenvíaal banco adquiriente del vendedor.
- El banco adquiriente 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 adquiriente) con un código de respuesta. Además, de determinar el destino de 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).
- La pasarela de pago recibe la respuesta y la reenvía al sitio web, donde se interpreta y se muestra una respuesta al cliente.
- El proceso completo demanda unos 3-4 segundos.

1.3.1 Proceso de pago en la plataforma AGORAV.

El proceso de pago con que cuenta actualmente la plataforma AGORAV, mediante el cual los usuarios realizan el pago por el consumo de los servicios solicitados, se realizan a través de un banco local. De esta manera se tiene que la persona deposita un monto de dinero correspondiente a su cuenta de usuario en la plataforma AGORAV, la cual le permite consumir los archivos multimedia publicados. En este modo de pago se puede definir un límite de deuda para el usuario, permitiéndole poder consumir los servicios que brinda la plataforma sin tener fondo suficiente en su cuenta.La ventaja que puede brindar este modo de pago es que permite realizar el proceso de pago en lugares donde sea desplegado el sistema y no exista conectividad con bancos u otros medios de pagos externos.

1.4 Análisis de soluciones existentes

La plataforma AGORAVcuenta con un mecanismo de cobro por los servicios ofrecidos a los usuarios, pero dada las limitaciones de no contar con soporte para diferentes modos de pago y la carencia de comercialización online de los productos que brinda a través de conexión con bancos y métodos de pago externos al sistema; se decidió realizar un módulo que corrigiera estas deficiencias, permitiéndole a los usuarios efectuar el pago por los servicios de visualización y descarga de los materiales audiovisuales publicados en la plataforma. Para ello se hizo una búsqueda de negocios online que tuvieran

implementados mecanismos similares, para que los usuarios realizaran el pago por los servicios consumidos entorno a materiales audiovisuales y así observar patrones y tendencias usadas.

1.4.1 Netflix

https://www.netflix.com

Es un sitio web estadounidense destinado al entretenimiento, el cual cuenta con un servicio de pago a través del que se pueden tener acceso a los contenidos audiovisuales.El sitio no permite descargar de manera directa sus productos audiovisuales, pero tiene la opción de realizar el envío de los pedidos a domicilio. Omitiendo este detalle, el sitio cuenta con un mecanismo de cobro por la visualización de los archivos audiovisuales a través de cuotas mensuales en la fecha de suscripción.Este mecanismo se realiza de la siguiente manera: El usuario accede al sistema, al estar dentro busca el espacio de suscripción gratis por un mes. Llena los datos solicitados y se le brinda la opción de escoger entre los modos de pagos existentes. Luego, será trasladado a un apartado en el cual se le informa del cargo mensual que se realizaráa su cuenta por el acceso a los contenidos audiovisuales. Para realizar este mecanismo de cobro el sitio presenta el uso de tarjetas de crédito internacional, ya sea Visa o Mastercard, además de cuentas PayPal,escogiendo el usuario entre uno de ellos. El mismo presenta un entorno amigable e intuitivo para introducir la información que necesita para realizar la transacción.Cuenta con una elevada seguridad al efectuartodas las operaciones a través de una conexión cifrada por el protocolo HTTPS (Hypertext Transfer Protocol Secure).

1.4.2 Nubeox

http://www.nubeox.com

Es un sitio web español que se dedica a la venta y alquiler de audiovisuales.Permite al seleccionar el producto, escoger la forma de consumo que másle convenga al cliente (alquiler y/o compra). Si el cliente decide comprar tiene la opción de descargar el producto, pero al hacer un alquiler solo permite visualizarla en un tiempo determinado. El sitio presenta un ejemplo muy adecuado en cuanto al uso de mecanismos de pago electrónico para la compra de materiales audiovisuales, donde se pueden señalar algunos aspectos muy útiles en la posterior implementación de un módulo similar, por ejemplo: posee diferentesmodos de pago, siendo el más usadoel pago por tarjetas de crédito(Visa, MasterCard o American Express) y cuentas PayPal, brindando la opción de escoger entre uno de ellos.Además, cuenta con el modo de pago Nubeox PIN, siendo este seguro y trasparente, permitiendo realizar compras de

formas muy sencillas, al introducir los datos de la tarjeta de crédito se activa este servicio y crea un número PIN que podrás utilizar para realizar otras compras sin necesidad de volver a introducir los datos de la tarjeta.Como medida de seguridad utiliza el CVV (Card Verification Value) de la tarjeta de crédito para toda transacción y para garantizar la confidencialidad en la transferencia de datos se utiliza el protocolo de seguridad SSL(Secure Socket Layer).

Tabla 1: Comparación Netflix_Nubeox

	Medida de seguridad al efectuar operaciones	Modos de pago que utiliza
Netflix	Cuenta con una elevada seguridad al efectuar todas las operaciones a través de una conexión cifrada por el protocolo HTTPS.	Cuenta con un mecanismo de cobro a través de tarifas mensuales a usuarios Premium, realizar este mecanismo por tarjeta de crédito.
Nubeox	Como medida de seguridad utiliza el CVV de la tarjeta de crédito para toda transacción, además de sistemas de seguridad SSL.	⬍ Pago a través de tarjetas de crédito (Visa, MasterCard o American Express). ⬍ PayPal. ⬍ Cuenta con el innovador modo de pago Nubeox PIN.

Los sistemas analizados cuentan con aspectos importantes dentro de los que se encuentran los diferentes modos de pagos que brindan la mayoría de ellos. Estos sistemasson de gran importancia ya que aportan funcionalidades que pueden ser incluidas en el desarrollo del módulo, contribuyendo de esta forma a aumentar la usabilidad del proceso de pago en la plataforma AGORAV.Algunas de estas funciones son el caso de la implementación de un carrito de compras disponible desde cualquier lugar para logar la agrupación de varios productos y que brinde la información necesaria sobre estos. Así como la selección de diferentes modos de pagos brindando una interfaz amigable para el cliente y aprobando la reproducción y descarga una vez efectuado el pago por los archivos multimediaseleccionados.

1.5 Herramientas y tecnologías para el desarrollo del módulo.

Para el desarrollo del módulo se caracterizan en el presente epígrafe las herramientas y tecnologías a utilizar. Para lograr una fácil integración con la plataforma AGORAV, se utilizarán las herramientas y tecnologías que se definen en el proyecto CPM. Dentro de estas se encuentra la metodología de desarrollo, un elemento fundamental para guiar, organizar, controlar y planear el proceso de desarrollo del software.

1.5.1 Metodología de desarrollo AUP (Agile Unified Process)

La metodología empleada para el desarrollo de esta investigación se basó en el método de desarrollo ágil AUP. El Proceso Unificado Ágil de AUP es una versión simplificada de RUP (Proceso Unificado de Software). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas agiles y conceptos que aún se mantienen válidos en RUP.En la UCI se decide hacer una variación de esta metodología, de forma tal que se adapte al ciclo de vida definido para la actividad productiva en la universidad,logrando estandarizar el proceso de desarrollo de software. De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola titulada Ejecución y se agrega la fase de Cierre (Sánchez, 2015). Esta metodología se utilizó por las características que presenta y por ser la utilizada para guiar el proceso de desarrollo del software en el proyecto, permitiendo una mejor compatibilidad en la documentación. A continuación se muestra la descripción de las fases y algunas técnicas tenidas en cuenta durante el desarrollo de la investigación.

Tabla 2: Descripción de las fases de AUP

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	En esta fase se realiza un estudio inicial de la organización cliente que permite obtener una información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Elaboración Construcción Transición	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
	Cierre	En esta fase se analizaron tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

AUP aplica técnicas ágiles como:

- Desarrollo Dirigido por Pruebas.
- Modelado ágil.
- Gestión de Cambios ágil.

1.5.2 UML

Lenguaje de modelado

El lenguaje de modelado es la notación, principalmentegráfica, que usan los métodos para expresar un diseño. En el proceso se indican los pasos que se deben seguir para llegar a dicho diseño. El modelado permite especificar el comportamiento deseado en la construcción de un sistema, comprender mejor lo que se construye y se descubren oportunidades de simplificación y reutilización (Booch, 2004).

UML

UML es un lenguaje estándar de modelado visual que se usa para especificar, construir, documentar y visualizar artefactos de un sistema de software. Permite a los desarrolladores visualizar el resultado de su trabajo en esquemas o diagramas estandarizados. Este lenguaje proporciona un vocabulario que incluye tres categorías: elementos, relaciones y diagramas, conteniendo aspectos conceptuales tales como

procesos de negocios, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables(Booch, y otros, 2000).

1.5.3 Lenguajes de programación

1.5.3.1 PHP5.5.12(Hypertext Preprocessor)

PHP es un lenguaje de programación interpretado que se utiliza para la generación de páginas web de forma dinámica. Es un lenguaje de código abierto, gratuito y multiplataforma lo que quiere decir que puede ser ejecutado en la mayoría de los sistemas operativos tales como Unix y Windows. Este lenguaje es usado principalmente en interpretación del lado del servidor, pero actualmente puede ser usado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas de QT y GTK+. (Álvarez, 2001). Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Permite la conexión a diferentes tipos de servidores de base de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. Este lenguaje ofrece la integración con las varias bibliotecas externas.La plataforma AGORAV está desarrollada con el CMSDrupal el cual utiliza en gran medida el lenguaje de programación PHP en la implementación de sus módulos, siendo esto un factor importante para la compatibilidad de los mismos, es por ello de gran importancia la selección de este lenguaje para el desarrollo del módulo pasarela de pago.

1.5.3.2 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como un lenguaje orientado a objetos, basado en prototipos, imperativo y dinámico. Es utilizado principalmente del lado del cliente permitiendo crear efectos atractivos y dinámicos en las páginas web. Este lenguaje posee varias características tales como:es un lenguaje simple, no hace falta tener conocimientos avanzados de programación para poder hacer un programa en él. Está basado en acciones que posee menos restricciones. Gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros (Valdés, 2007). Además, permite hacer validaciones del lado del cliente permitiendo estoun menor tiempo de respuesta y evita efectuar llamadas incorrectas al servidor.

1.5.4 IDE (Entorno de Desarrollo Integrado)

NetBeans IDE 7.3

NetBeans IDE es una herramienta de código abierto bajo la licencia pública (GPL). Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles. Da soporte a varias tecnologías como Java, PHP, Groovy, C/C++, HTML5. Entre sus funcionalidades permite escribir, depurar, compilar y ejecutar programas. Además, se destacan sus ventajas de auto completamiento de código, interfaz para el diseño de GUI y la capacidad para importar clases(NetBeans, 2015). NetBeans por sus características es el IDE de desarrollo que se va a utilizar para la programación web durante el proceso del desarrollo del módulo.

1.5.5 SVN (Subversion)

El control de versiones es el arte del manejo de los cambios en la información.SVN es un sistema de control de versiones libre y de código fuente abierto desarrollado bajo la licencia Apache/BSD. Estos sistemas permiten el regreso a un estado anterior del código fuente que almacena en caso de la ocurrencia de un estado no esperado. SVN puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. El uso de SVN fomenta la colaboración y agiliza el trabajo ya que no se precisa de un único flujo de modificaciones (Collins-Sussaman, y otros, 2004).La utilización de Subversion está dada por su condición de software libre y por ser usada en el proyecto CPM. Además, de las ventajas que esta permite a la hora de salvar las versiones de la información y regresar a versiones correctas en caso de errores. Otra característica importante que tiene este sistema es el desarrollo paralelo de varios equipos de trabajo que permite.

1.5.6PostgreSQL9.2

PostgreSQL es un sistema gestor de base de datos relacional, robusto, con escalabilidad, su código fuente está disponible libremente, publicado bajo la licencia BSD (Berkeley Software Distribution). Está disponible para una amplia variedad de plataformas tales como: Linux, Windows, Solaris, Mac OS X y otros. Soporta la realización de transacciones seguras, vistas, uniones, claves extranjeras, procedimientos almacenados, triggers, tipos de datos definidos por el usuario y otras operaciones que se realizan en estos sistemas. El almacenamiento es confiable y consistente. La manipulación es potente, flexible y eficiente. Está considerado como el sistema gestor de base de datos de código abierto más avanzada del mundo(Guerrero, 2010).Se decidió utilizar este sistema gestor de base de datos para trabajar en el

desarrollo del módulo por las características con las que cuenta este sistema, además es el sistema gestor de base de datos utilizado por la plataforma AGORAV.

1.5.7 Herramienta CASE

Las herramientas CASE son un conjunto de herramientas y métodos asociados que proporcionan asistencia automatizada en el proceso de desarrollo del software a lo largo de su ciclo de vida. Mediante el uso de estas herramientas se reduce el tiempo y costo del desarrollo y mantenimiento del software, así como mejorar su calidad(Pressman, 2002).

Visual Paradigm 8.0

Visual Paradigm es una herramienta CASE, que propicia un conjunto de ayudas para el desarrollo de programas informáticos dando soporte al modelado visual con UML desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Es soportada por varios sistemas operativos tales como: Ubuntu, Windows y otros. Esta herramienta ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de varios tipos de diagramas. Fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos(International, 2008).

Visual Paradigm ofrece:

- Entorno de creación de modelos conformes a UML.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales Entornos de Desarrollo Integrado (IDE).
- Disponibilidad en múltiples plataformas.
- Extensible mediante desarrollo de nuevos Módulos (plugins).

Visual Paradigm es la herramienta seleccionada para el modelado UML por las múltiples ventajas que presenta. Además, de ser la herramienta utilizada por el equipo de trabajo del proyecto CPM, permite el entendimiento del equipo de desarrollo así como la uniformidad en la documentación asociada a la Plataforma AGORAV.

1.5.8CMS (Content Management System)

Un CMS, es un software que permite crear una estructura base para la creación y administración de contenidos, principalmente de páginas web. Además, admite a un editor crear, clasificar y publicar cualquier tipo de información en una página web. Generalmente un CMS es una aplicación con una base de datos asociada en la que se almacenan los contenidos, separados de los estilos o diseño. El CMS controla también quién puede editar y visualizar los contenidos, convirtiéndose en una herramienta de gestión integral para la publicación de sitios web. Algunos de los CMS usados en la actualidad son Drupal, WordPress y Joomla.(Rodríguez, 2012).

Drupal 7.18

Drupal es un sistema de gestión de contenido modular y muy configurable. Es un CMS de código abierto, que se distribuye como software libre bajo la licencia GNU GPL (General Public License). Drupal puede ser modificado y distribuido libremente, pero siempre se debe hacer bajo la misma licencia. El software está desarrollado con el lenguaje de programación PHP y tiene soporte para base de datos MySQL, PostgreSQL, SQLite, Oracle, entre otros. Está maquetado con hojas de estilo CSS, con lo que es posible construir sitios web totalmente accesibles. Cuenta con una amplia comunidad de usuarios y desarrolladores a nivel mundial(Rodríguez, 2012).Para el desarrollo del módulo se escogió este CMS ya que es el sistema en el cual está desarrollada la Plataforma AGORAV.

1.6 Conclusiones del capítulo

En el presente capitulo se realizó el análisis de los principales conceptos de interés como parte de la fundamentación teórica de la investigación, lo que permitió una mejor comprensión del problema planteado. Se llevó a cabo un estudiosobre sistemas similares a la plataforma AGORAV que implementan diversos modos de pagospara comprender patrones y tendencias existentes.Estos sistemas a pesar de ser privativos, aportan funcionalidades para incluir en el módulo a desarrollar como, el uso de diferentes modos de pagos orientados a cuentas externas a la entidad y la utilización de un carrito de compras para

la agrupación de varios productos.Además, se realizó una descripción detallada de los lenguajes de programación, herramientas y tecnologías más apropiadas para el desarrollo de la solución.

Capítulo 2: Análisis y diseño.

2.1 Introducción

En este capítulo se realiza una descripción detallada por medio del modelo de dominio, en el cual de manera gráfica se analiza 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 debe cumplir la solución propuesta.Además, se agrupan los requisitos funcionales en casos de uso y a partir de estos se realiza el diagrama de casos de usos. También se muestran los diagramas de clases para lograr una visión generaldel funcionamiento de la aplicación.

2.2 Modelo del dominio

Un modelo del dominio se utiliza con frecuencia como fuente de inspiración para el diseño de los objetos del software. Es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés; es elartefacto más importante que se crea durante el análisis orientado a objetos (Larman, 2003).

El objetivo del modelado del dominio es comprender y describir las clases dentro del contexto del sistema y capturar los tipos más importantes de objetos en el mismo contexto. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Sedescribe mediante diagramas de UML, especialmente mediante diagramas de clases.Estos diagramas muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan una con otras mediante asociaciones(Booch, y otros, 2000). El modelo del domino de la plataforma AGORAV que se describe a continuación es el entorno donde se desarrolla el módulo de pasarela de pago.

2.2.1 Descripción del modelo del dominio

El modelo del dominio en la plataforma AGORAV se describe de la siguiente manera, un ***autor*** adiciona un ***archivo multimedia***el cual pasa a través del ***servidor web*** y este lo almacena en el ***servidor de media***,el ***documentalista*** crea una ***tipología*** que es relacionada al archivo multimedia.El ***revisor*** realiza una ***publicación de archivo multimedia***y para que esta pueda ser visualizada por el ***usuario*** el servidor web almacena el archivo multimedia a publicar en el ***servidor streaming***.

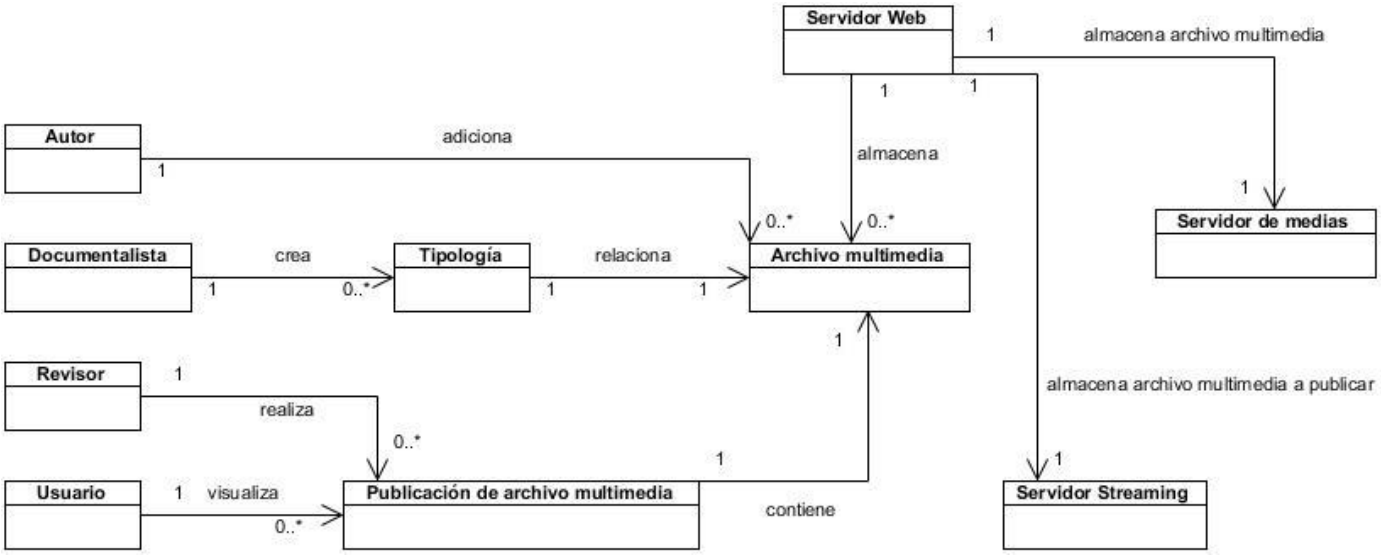


Fig. 1: Diagrama de clases del modelo de dominio

Descripción de las clases del modelo de dominio

Autor: Persona con permiso en la plataforma para adicionar archivos multimedia.

Documentalista: Persona con permiso en la plataforma para catalogar los archivos multimedia.

Revisor: Persona con permiso en la plataforma para realizar la publicación de los archivos multimedia.

Usuario: Persona con permiso en la plataforma solo para visualizar los archivos multimedia publicados.

Archivo multimedia: Archivo de audio y video almacenado en el servidor de media.

Tipología: Mecanismo para la clasificación de archivos multimedia.

Publicación de archivo multimedia: Archivos multimedia publicados en la plataforma.

Servidor web: Selecciona el archivo multimedia del servidor de medias y lo almacena en el servidor streaming para ser publicado.

Servidor de medias: Servidor donde se almacenan los archivos multimedia.

Servidor streaming: Servidor donde se almacenan los archivos multimedia para ser publicados.

2.3 Especificación de los requisitos del sistema

Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. Estos pueden clasificarse en requisitos funcionales y no funcionales. Los requisitos funcionales definen qué debe hacer un sistema y los requisitos no funcionales definen cómo debe ser el sistema(Sommerville, 2005).

2.3.1 Requisitos funcionales(RF) del sistema

RF1: Seleccionar modo de pago: El módulo debe permitir al usuario elegir el modo en que desea realizar su pago.

RF2: Efectuar Pago: El módulo debe permitir efectuar el pago usando el modo de pago seleccionado por el usuario.

RF3: Guardar datos en el historial de transacciones: El módulo debe ser capaz de almacenar los datos de cada transacción que se realice en el sistema.

RF4: Visualizar historial de transacciones: El módulo debe permitir al administrador visualizar el historial de transacciones.

RF5: Adicionar precio a contenidos multimedia: El módulo debe permitir al administrador poner precio al contenido multimedia que desee cobrar por su visualización y descarga.

RF6: Editar precio de contenidos multimedia: El módulo debe permitir al administrador editar el precio de los archivos multimedia.

RF7: Eliminar precio de contenidos multimedia: El módulo debe permitir al administrador eliminar el precio de los archivos multimedia en caso de que esta se desee ofertar gratis a los usuarios.

RF8: Listar contenido multimedia de pago: El sistema debe permitir al administrador listar los archivos multimedia que tengan asignado un precio.

RF9: Adicionar contenido multimedia al carrito de compras: El módulo debe permitir al usuario adicionar productos a su carrito de compras.

RF10: Visualizar contenidos multimedia del carrito de compras: El módulo debe permitir al usuario visualizar los productos que ha adicionado a su carrito de compras.

RF11: Eliminar contenido multimedia del carrito de compras: El módulo debe permitir al usuario eliminar productos de su carrito de compras.

2.3.2 Requisitos no funcionales(RNF) del sistema

RNF1 Hardware

Según los RNF de hardware definidos para la plataforma AGORAV, para garantizar su correcto funcionamiento se identifica que se debe contar con los siguientes elementos mínimos:

Máquinas clientes:

- Procesador Dual Core 1.6 GHz.
- Memoria RAM DDR3 de 1GB.

Máquinas servidor web:

- Procesador Intel Core i3 2.0 GHz.
- Memoria RAM DDR3 de 4 GB.
- Disco duro de 160 GB o superior.

RNF2 Software

Los requisitos de software imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe tener.

- Debe estar instalado en la computadora cliente un navegador web, se recomienda que sea Mozilla Firefox en su versión 20.
- El servidor de base de datos PostgreSQL versión 9.2 debe estar instalado en el servidor de la aplicación.

RNF3 Restricciones deimplementación

Restricciones que deben ser cumplidas a la hora de realizar la aplicación.

- Para la implementación se utilizaráel CMS Drupal 7.18, como lenguaje de programación del lado del servidor PHP y del lado del lado del cliente JavaScript y como gestor de base de datos se va a utilizar PostgreSQL 9.2.

RNF4 Seguridad

El módulo debe garantizar que la información viaje de manera segura mediante el mecanismo SSLdurante todo el proceso y debe quedar protegida de los usuarios que no estén autorizados a acceder a la aplicación.

2.4 Modelo de casos de uso

El modelo de casos de uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. La mayoría de los sistemas tienen muchos tipos de usuarios y cada tipo de usuario se representa mediante un actor. Los actores utilizan el sistema al interactuar con los casos de uso. Por lo tanto, se puede definir que todos los actores y casos de uso del sistema forman un modelo de casos de uso(Booch, y otros, 2000).

2.4.1 Definición de los actores del sistema

Un actor del sistema es un rol que un usuario asume con respecto al sistema. También pueden definirse como personas, sistemas o máquinas externas que interactúan con el sistema que se va a desarrollar(Booch, y otros, 2000).

Tabla 3: Actores del sistema

Autor	Descripción
Usuario	Usuario autenticado que puede realizar la compra de archivos multimedia.
Administrador	Usuario autenticado con permiso para administrar los precios de los archivos multimedia, así como visualizar el historial de los reportes de las transacciones efectuadas.

2.4.2 Diagrama de caso de uso de sistema

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente(Tello, 2012).

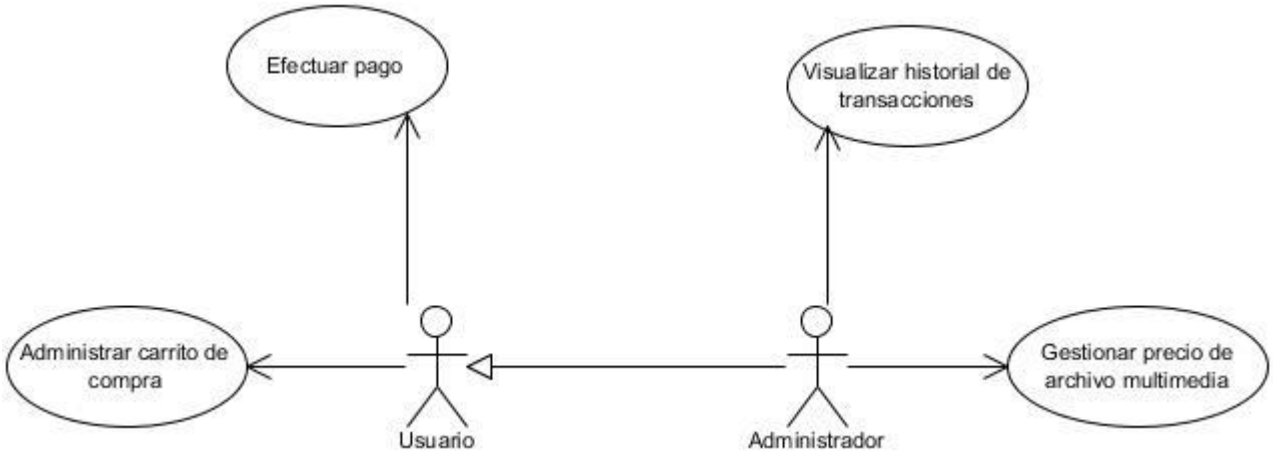


Fig. 2: Caso de uso del sistema

2.4.3 Descripción de Casos de Uso

A continuación se muestra la descripción de uno de los casos de uso perteneciente al módulo pasarela de pago.

Tabla 4: Gestionar precio de archivos multimedia

Objetivo	Adicionar, editar, eliminar y listarcontenido multimedia de pago.
Actores	Administrador.
Resumen	El caso de uso inicia al seleccionar la opción "*Gestionar precio de archivo multimedia*", muestra un listado de todos los archivos multimedia de pago con la opción de eliminar o editar. Además se muestra la opción de adicionar un archivo multimedia de pago.El caso de uso culmina al terminar de realizar cualquiera de estas operaciones.

Complejidad	Media
Prioridad	Crítico
Precondiciones	Que el administrador esté autenticado
Postcondiciones	Se listó el contenido multimedia de pago. Se eliminó un archivo multimedia de pago. Se editó el precio de un archivo multimedia de pago. Se adicionó un archivo multimedia de pago.
Referencias	RF5, RF6, RF7, RF8

Flujo de eventos

Flujo básico: Gestionar precio de archivos multimedia.

	Actor	**Sistema**
1.	Selecciona la opción "*Gestionar precio de archivo multimedia*".	
2.		Muestra una interfaz brindando un listado de los archivos multimedia de pago almacenados, con las siguientes opciones: ◆ Adicionar archivo multimedia de pago. Ver sección 1: "*Adicionar precio*". ◆ Editar. Ver sección 2: "*Editar precio de la media*". ◆ Eliminar archivo multimedia de

		pago: Ver sección 3: "*Eliminar precio de archivo multimedia*".
Sección 1: "*Adicionar archivo multimedia de pago*"		
Flujo básico: Adicionar archivo multimedia de pago		
	Actor	**Sistema**
1.	Selecciona la opción "*Adicionar media*".	
2.		Muestra una interfaz brindando un listado de los archivos multimedia libres de pago almacenados, con la opción "*Adicionar*".
3.	Selecciona la opción "*Adicionar*".	
4.		Muestra una interfaz con los datos requeridos para adicionar la media.
5.	Inserta los datos teniendo en cuenta los siguientes parámetros. • Precio de media (obligatorio). Luego de llenar los datos selecciona la opción "*Aceptar*".	
6.		Valida que no existan campos obligatorios vacíos.
7.		Inserta los datos en la base de datos.

8.		Muestra un mensaje de confirmación. Termina el caso de uso.
Flujos alternos		
Nº Evento: Selecciona la opción "*Cancelar*".		
	Actor	**Sistema**
1.	Selecciona la opción "Cancelar".	
2.		Regresa al paso 2 del Flujo Básico: Gestionar precio de archivos multimedia.
Flujos alternos		
Nº Evento: Campo obligatorio vacío.		
	Actor	**Sistema**
1.		Muestra un mensaje de error indicando que hay campos vacíos. Regresa al paso 5 del Flujo Básico: Adicionar media.
Sección 2: "*Editar precio de la media*"		
Flujo Básico: "Editar precio de la media"		
	Actor	**Sistema**
1.	Selecciona la media a la cual desea editar su precio.	

2.		Muestra una interfaz con el precio de la media seleccionada permitiendo su edición.
3.	Cambia el atributo: ◆ Precio de la media(obligatorio). Luego selecciona la opción “Guardar”.	
4.		Valida que no existan campos obligatorios vacíos.
5.		Busca en la base de datos la media seleccionada según su identificador y modifica su precio.
6.		Muestra un mensaje de notificación. Termina el caso de uso.
Flujos alternos		
Nº Evento: Selecciona la opción “*Cancelar*”.		
	Actor	**Sistema**
1.	Selecciona la opción “*Cancelar*”.	
2.		Regresa al paso 2 del Flujo Básico: Gestionar precio de archivos multimedia. Termina el caso de uso.
Flujos alternos		

Nº Evento: Campo obligatoriovacío.		
	Actor	**Sistema**
1.		Muestra un mensaje de error indicando que hay campos vacíos. Regresa al paso 3 del Flujo Básico: Editar precio de la media.
Sección 3: "Eliminar archivo multimedia de pago"		
Flujo básico: Eliminar archivo multimedia de pago.		
	Actor	**Sistema**
1.	Selecciona la media que desea eliminar.	
2.		Muestra un mensaje de notificación para saber si el usuario desea eliminar o no la media.
3.	Selecciona la opción "Aceptar".	
4.		Elimina de la base de datos de archivos multimedia de pago la media. Termina el caso de uso.
Flujos alternos		
Nº Evento: Selecciona la opción "Cancelar".		
	Actor	**Sistema**

1.	Selecciona la opción “Cancelar”.	
2.		Regresa al paso 2 del Flujo Básico: Gestionar precio de archivos multimedia. Termina el caso de uso.
Relaciones	**CU Incluidos**	Ninguno.
	CU Extendidos	Ninguno
Requisitos no funcionales	RNF3, RNF4	

2.5 Patrón Arquitectónico

Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. Para el desarrollo del módulo se decide hacer uso del patrón arquitectónico Modelo Vista Controlador (MVC).

Elpatrón arquitectónico MVC separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes que son el modelo, la vista y el controlador. El patrón se basa en las ideas de reutilización de código y la separación de conceptos. Las modificaciones realizadas a la vista no afectan en absoluto a los otros módulos de la aplicación. Este patrón utiliza el estilo de llamada y retorno ya que este estilo enfatiza la modificabilidad y la escalabilidad.(Pressman, 2011).Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual.

Modelo: El modelo administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formularios desde la vista) y a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Maneja la visualización de la información.

Controlador: Interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado. Solo se comunica con el modelo y responde a través de vistas.

2.6 Patrones de diseño

Los patrones de diseño expresan esquemas para definir estructuras de diseño o sus relaciones con las que construir sistemas de software. Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Con el uso de los patrones se facilitan la reusabilidad, extensibilidad y mantenimiento de los sistemas(Pressman, 2011). Entre los patrones de diseño más utilizados se encuentran los patrones GRASP (General Responsability Assignment Pattens)que representan los principios básicos de la asignación de responsabilidades a objetos y los patrones GOF (Gang of Four) surgen como una forma indispensable de enfrentarse a la programación y estos se clasifican en creacionales, estructurales y decomportamiento(Larman, 2003).

2.6.1 Patrones GRASP

Creador: Asignarle a una clase la responsabilidad de crear una instancia de otra clase(Larman, 2003). Este patrón se aprecia en la clase pasarela_de_pago_controlador, la cual contiene funciones que crean instancias de las clases pasarela_de_pago_modelo ($pasarela_de_pago_modelo = new pasarela_de_pago_modelo();) y pasarela_de_pago_vista ($pasarela_de_pago_vista = new pasarela_de_pago_vista();).

Experto: Asignar una responsabilidad al experto en información, es decir la clase que cuenta con la información necesaria para cumplir la responsabilidad(Larman, 2003). Este patrón se pone de manifiesto en el módulo ya que se hace uso para su implementación del patrón arquitectónico MVC, cada clase tiene responsabilidades específicas como acceder a la base de datos, controlar las operaciones que hace el módulo o mostrar la información.

Controlador: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase(Larman, 2003). Ejemplo de la utilización de esté patrón es la existencia de la clase pasarela_de_pago_controlador que se encarga de la lógica del negocio, esta utiliza información de la clase pasarela_de_pago_modelo y envía datos a la clase pasarela_de_pago_vista.

Bajo acoplamiento: Asignar una responsabilidad para mantener bajo acoplamiento, es decir que exista baja dependencia entre las clases(Larman, 2003). Se ve reflejado el uso de este patrón ya que en cada

clase solo existen instancias necesarias de otras clases. Esto facilita que si se modifica alguna clase las demás se vean afectadas lo menos posible.

Alta cohesión: Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme o excesivo(Larman, 2003). La utilización de este patrón se encuentra presente en la definición de tareas y responsabilidades de cada clase en el módulo. La clase controladora es la encargada de la lógica del negocio, el modelo del acceso a los datos y de la representación de la información en las vistas.

2.6.2 Patrones GOF

Fábrica abstracta (Abstract Factory): Este patrón permite a un sistema determinar la subclase a partir de la cual se va a instanciar un objeto en tiempo de ejecución. A menudo esta subclase es desconocida durante el desarrollo (Patrones de diseño). Se propone un módulo que implementa un modo de pago, con una arquitectura que le permite integrarse a la pasarela de pago,ignorándose el tipo de pago que se utilice y la forma en que se realice el mismo. Para la inclusión de otros modos de pago es necesario la implementación del módulo correspondiente, basándose en la estructura que se propone.

Acción(Command):Este patrón permite solicitar una operación a un objeto sin conocer realmente el contenido de esta operación, ni el receptor real de la misma. Presenta una forma sencilla y versátil de implementar un sistema basado en comandos facilitando su uso y aplicación (Mühlrad, 2008). Se puede apreciar la utilización de este patrón en la implementación de los ganchos (hook) necesarios para la comunicación del módulo con el núcleo de Drupal.

2.7 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es de ese modo utilizado como una entrada fundamental de las actividades de implementación(Booch, y otros, 2000).

2.7.1 Diagrama de clases del diseño

Un diagrama de clases del diseño (DCD) representa las especificaciones de las clases e interfaces de software en una aplicación. A diferencia de las clases conceptuales del modelo del dominio, las clases de

diseño de los DCD muestran las definiciones de las clases software en lugar de los conceptos del mundo real(Larman, 2003). Según (Conallen, 1999) se hace uso de las facilidades de extensión brindadas por el UML para basado en este lenguaje modelar aplicaciones Web. Esta extensión presenta como elementos más significativos 3 clases de UML estereotipadas con los siguientes estereotipos Server Page, Client Page y Form, empleados para el código servidor, código cliente y formularios respectivamente. A continuación se muestra el diagrama de clases del diseño del CU Gestionar precio de archivo multimedia. En este diagrama se encuentra el Server Page Index.php que es la encargada de atender todas las peticiones, para lo cual establece la comunicación con el núcleo de Drupal y pasarela_de_pago. Además, se encuentran las clases Client Page que tienen asociados los formularios que envían los datos introducidos por el usuario a la Server Page pasarela_de_pago, la cual se relaciona con las demás clases que intervienen en el funcionamiento del módulo.

Descripción de las clases:

pasarela_de_pago: Es la encargada de los ganchos (*hooks*) predefinidos por Drupal para ser ejecutadas durante una petición dirigida a este módulo.

pasarela_de_pago_controlador: En la clase encargada de desarrollar la lógica del negocio.

pasarela_de_pago_modelo: Es la clase donde se encuentran las funciones de acceso a los datos del sistema.

pasarela_de_pago_vista: Es la encargada de construir las interfaces para interactuar con el usuario.

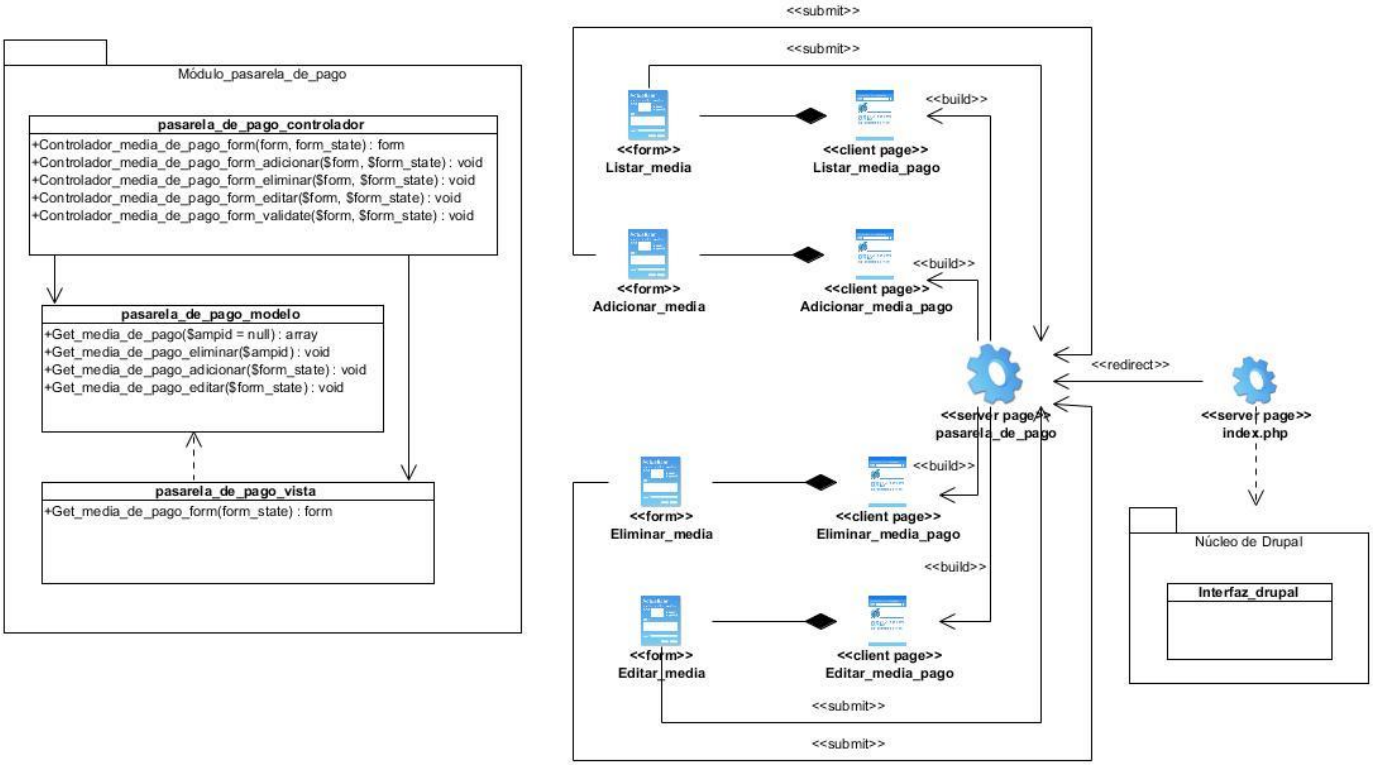


Fig. 3: Diagrama de clase del diseño

2.7.2 Diagrama de secuencia del diseño

En contraste con los diagramas de clases y con los diagramas de implementación, que muestran la estructura estática de un componente de software, un diagrama de secuencia se usa para mostrar las comunicaciones dinámicas entre objetos durante la ejecución de una tarea. Este tipo de diagramas muestra el orden temporal en el que los mensajes se envían entre los objetos para lograr dicha tarea. Puede usarse un diagrama de secuencia para mostrar las interacciones en un caso de uso o en un escenario de un sistema de software(Pressman, 2011).

Se muestra para una mejor comprensión el diagrama de secuencia del diseño del CU: Gestionar archivo multimedia de pago. El diagrama se presenta en tres escenarios por la cantidad de flujo de eventos que genera. En los eventos se muestra al administrador que puede efectuar las opciones de adicionar archivo multimedia de pago, editar archivo multimedia de pago y eliminar archivo multimedia de pago.

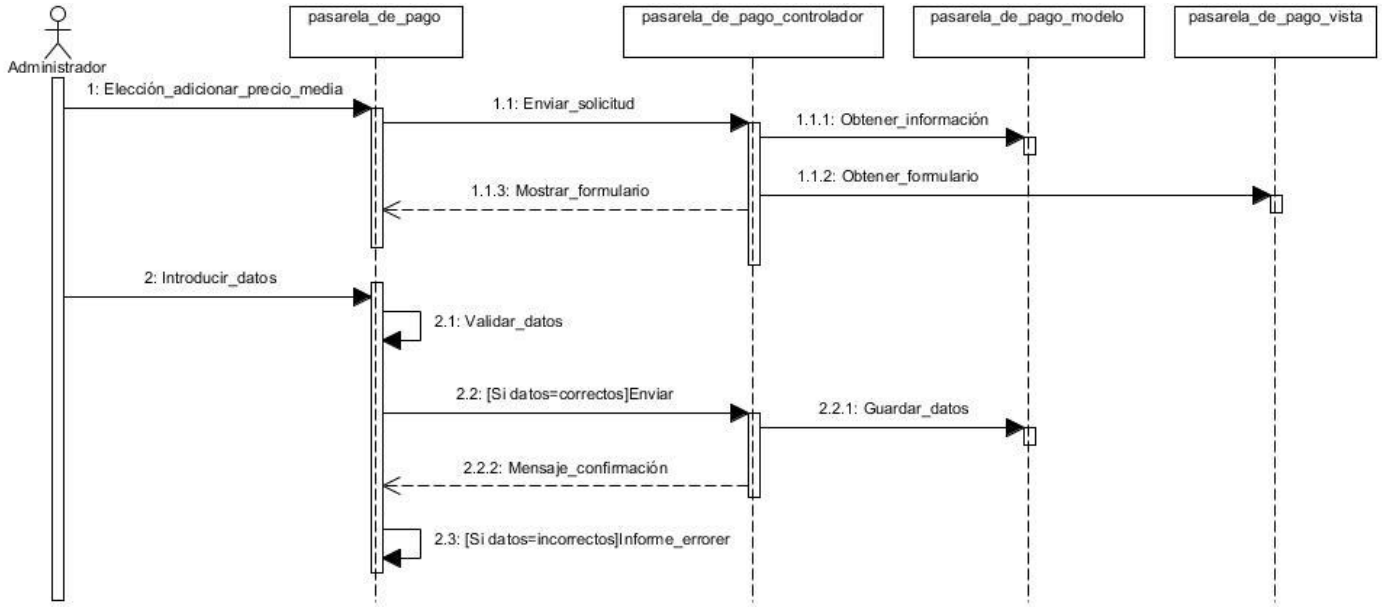


Fig. 4: Diagrama de secuencia del CU Gestionar archivo multimedia de pago: Escenario adicionar archivo multimedia de pago.

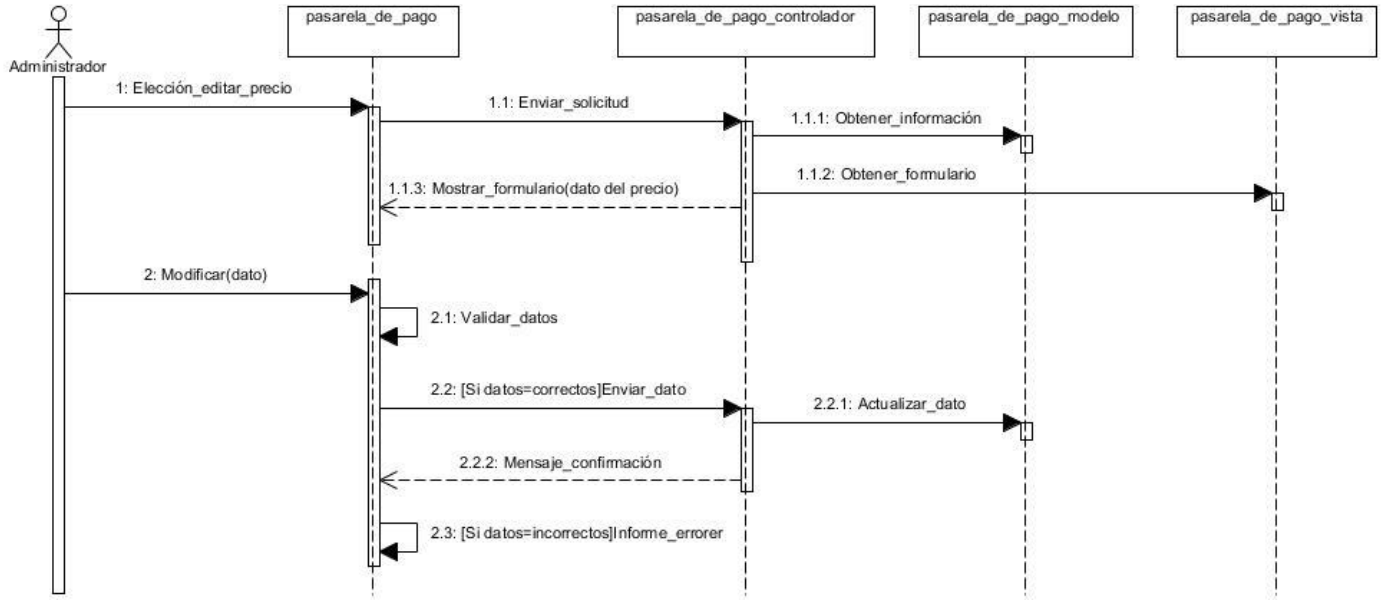


Fig. 5: Diagrama de secuencia del CU Gestionar archivo multimedia de pago: Escenario editar archivo multimedia de pago.

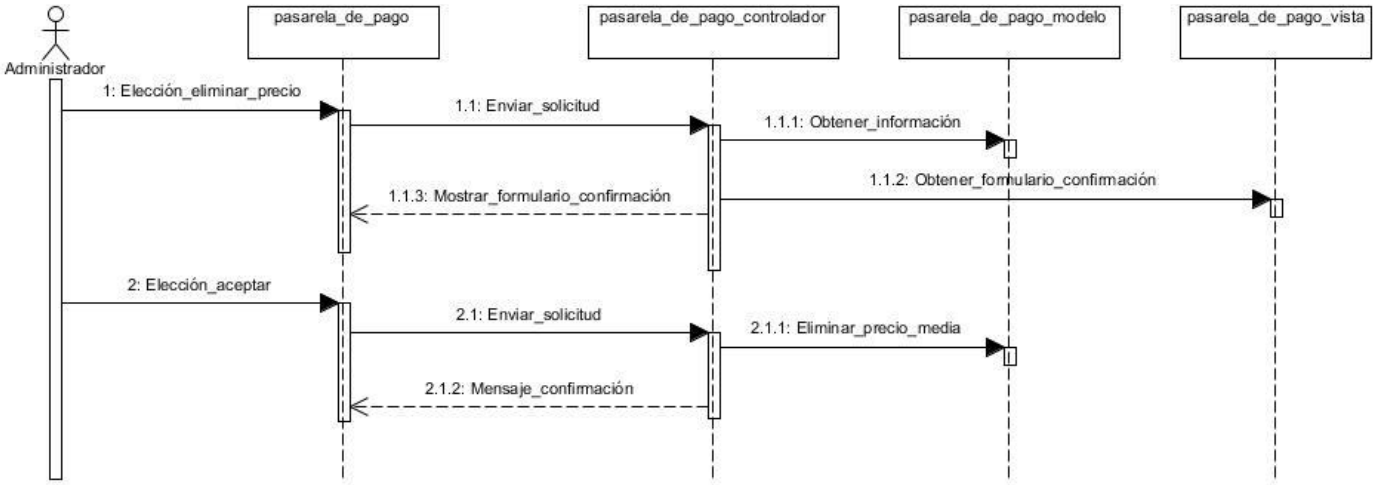


Fig. 6: Diagrama de secuencia del CU Gestionar archivo multimedia de pago: Escenario eliminar archivo multimedia de pago.

2.8 Propuesta de solución

CECA (Confederación Española de Cajas de Ahorro) es una entidad de crédito que brinda una serie de servicios web para permitir la venta de productos online. Se implementa como modo de pago en la propuesta de solución un módulo CECA, el cual permite el pago online mediante la entidad antes mencionada.Para el desarrollode este módulo se siguió el estándar de comunicación que presenta esta entidad. CECA permite realizar un pago seguro a través del uso de protocolo SSL. Para hacer uso de este protocolo basta con que el cliente cuente con un navegador web que soporte SSL 3.0 con claves de cifrado para 128 bits, prácticamente todas las versiones de los navegadores actuales cumplen con este requisito. Por otra parte, se garantiza la integridad de los datos que se envían por este medio seguro mediante firma electrónica. Dicha firma es calculada mediante el algoritmo SHA-1 y enviada junto a los datos a la entidad bancaria online.

El modo de pago CECA según el estándar provisto por la entidad debe enviar al banco los siguientes parámetros obligatorios: MerchantID el cual identifica al comercio, AcquirerBIN el cual identifica la caja, TerminalID el cual identifica la terminal, importe de la operación, código ISO-4217 que corresponde a la moneda en que se efectúa el pago, firma electrónica, cifrado el cual es el valor fijo SHA1. El módulo CECA

implementado establece el patrón a seguir para la creación de nuevos mecanismos de pago para la futura integración de estos con la pasarela. Lo antes mencionado garantiza la flexibilidad y el crecimiento modular del proceso de pago en AGORAV.

2.9 Conclusiones del capítulo

En el presente capítulo la realización del modelo de dominio permitió comprender la estructura del entorno donde se va a desarrollar el módulo. La especificación de los requisitosdel sistema posibilita describir las características con las que debe cumplir el módulo para su correcto funcionamiento. A partirde los requerimientos funcionales se determinaron los distintos casos de uso necesarios para definir las funcionalidades básicas. Los artefactos generados durante el capítulo permiten un mejor entendimiento del módulo a desarrollar.

Capítulo 3: Implementación y prueba.

3.1 Introducción

En este capítulo se muestra el flujo de trabajo de implementación, el cual forma parte de la fase de construcción. Como resultado de este flujo se muestrael diagrama de componente para visualizar con mayor facilidad la estructura general del módulo y el diagrama de despliegue para modelar la arquitectura del sistema. Además, se definen las pruebas necesarias que se le aplicarán al módulo y se muestran los resultados obtenidos.

3.2 Diagrama de componente

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. En estos diagramas se muestran los elementos de diseño de un sistema de software. Un componente representa una parte de un sistema modular, desplegable, reemplazable, que encapsula la implementación y expone un conjunto de clases (Larman, 2003). A continuación en la Fig. 7se muestra el diagrama de componentes del módulo pasarela de pago para la plataforma AGORAV. Este diagrama muestra los ficheros **.php**que son las clases pertenecientes al módulo, encargadas de su funcionamiento. El fichero **.module**que contiene las funciones o ganchos (*hooks*) predefinidos por Drupal para ser utilizadas en una petición. Además, se muestra el fichero **.install** donde se definen las acciones de las operaciones necesarias para el funcionamiento inicial del módulo. También se muestra el fichero **.info**que es el que contiene toda la información general referente al módulo. Por último el fichero **.js**posee las funciones JavaScript asociadas al módulo.

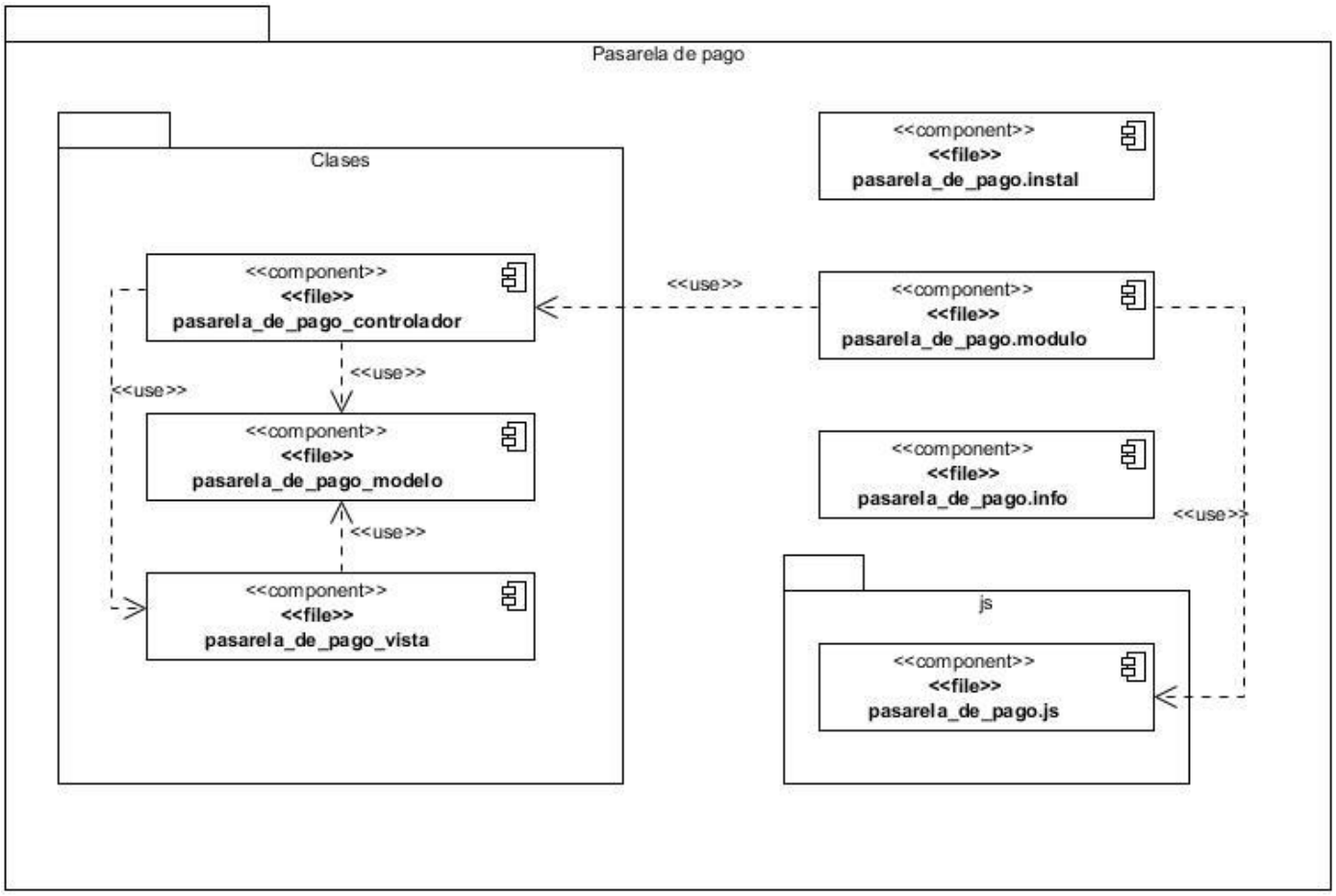


Fig. 7: Diagrama de componente.

3.3 Diagrama de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Este muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. Las relaciones entre los nodos constituyen medios de comunicación(UML, 2007). A continuación en la Fig. 8 se muestra el diagrama de despliegue correspondiente al módulo implementado. Se representa un nodo **PC Cliente** el cual tiene como componente el navegador web Mozilla Firefox 30.0, este nodo se comunica con el **Servidor Web**mediante el protocolo HTTPS por el puerto 443. El Servidor Web debe contener el servidor Apache 2.0 y la plataforma AGORAV la cual tiene contenido el módulo pasarela_de_pago, estese comunica con el

nodo**Banco Online**medianteel protocolo HTTPS por el puerto 443. Además, el nodo Servidor Web se comunica con otros nodos como son el **Servidor BD**a través del protocolo TCP por el puerto 5432, el cual debe contener el gestor PostgreSQL 9.2; y el **Servidor de Media**mediante el protocolo NFS por el puerto 2049. El **Servidor Streaming**cuenta con el componente Flumotion 0.10 y recibe las medias a publicar del Servidor de Medias a través del protocolo NFS por el puerto 2049. Por último la PC Cliente puede interactuar con las publicaciones en el sistema mediante el protocolo HTTPS por el puerto 443.

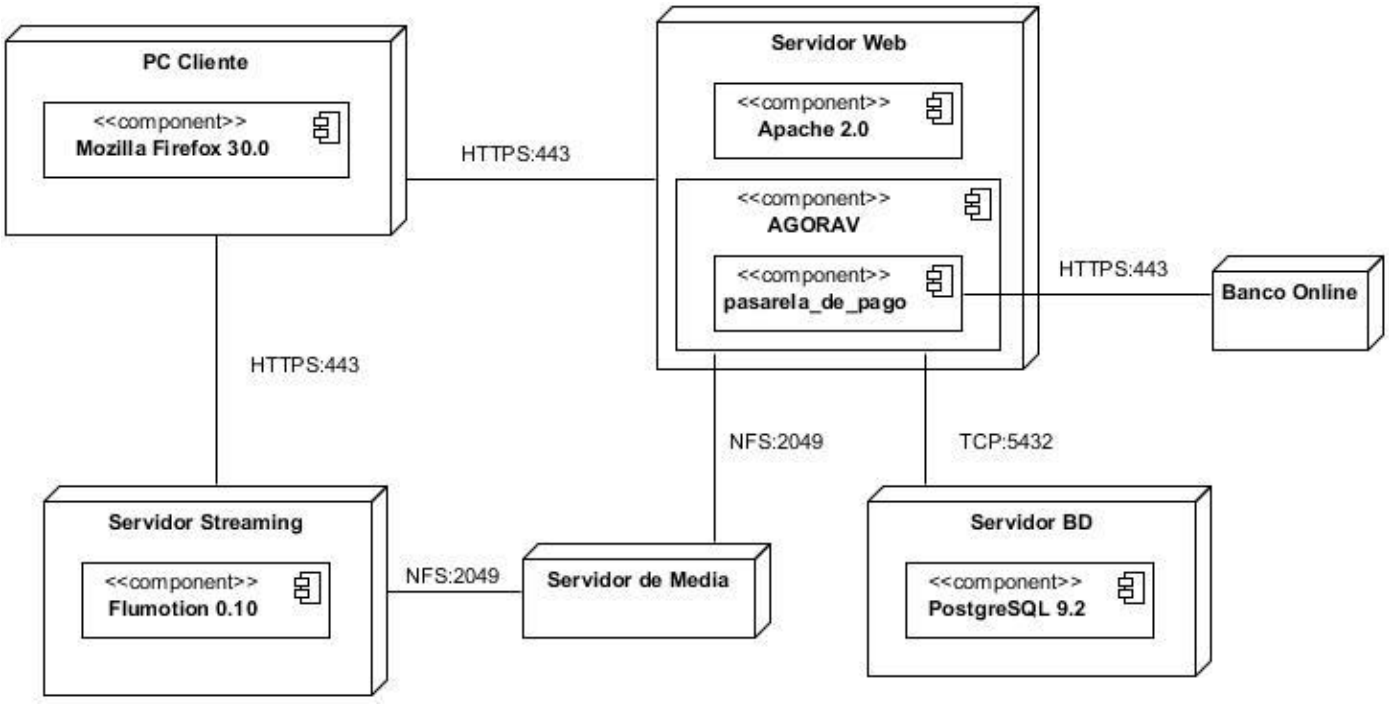


Fig. 8: Diagrama de despliegue.

3.4 Modelo de pruebas

Durante el flujo de trabajo de pruebas se verifica el resultado final de la implementación probando la estructura, tanto en la construcción interna como intermedia, así como las versiones finales del sistema a ser entregado. Uno de los artefactos generados en esta etapa es el modelo de pruebas que se describe como una colección de casos de pruebas, procedimientos de prueba y componentes de prueba.Las pruebas son actividades en las cuales un sistema o componente es ejecutable bajo condiciones o requerimientos específicos permitiendo que los resultados sean observados y registrados. Estas pruebas

se realizan con el objetivo de encontrar deficiencias existentes en el software. Existen dos métodos fundamentales que utilizan diseños de casos de pruebas que se le aplican a productos de software, estas son: las pruebas de caja negra y pruebas de caja blanca(Booch, y otros, 2000).

3.4.1 Pruebas de caja negra

Se seleccionó en esta etapa para aplicarle al módulo desarrollado las pruebas de caja negra. Según (Pressman, 2002)estas pruebas se centran en los requisitos funcionales del software. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca, más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores de las siguientes categorías:

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

Para esta prueba se utilizó la técnica partición equivalente, la cual divide el dominio de entrada en clases de datos de los que se pueden derivar casos de prueba (Pressman, 2002). A continuación se muestran el caso de prueba para el CU Gestionar archivo multimedia de pago.

Casos de pruebas

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 requisitos funcionales del sistema(Pressman, 2002). Para el desarrollo de las pruebas al módulo se desarrolló un caso de prueba. El caso de prueba desarrollado se muestra a continuación:

Caso de Prueba: Gestionar precio de archivos multimedia

Descripción: Este caso de uso inicia cuando el administrador accede a la opción "*Gestionar precio de archivo multimedia*" y selecciona alguna de las opciones disponibles, finalizando cuando se adiciona, modifica o elimina un precio, o cuando se cancela cualquiera de las operaciones anteriores.

Condición de ejecución: Que el administrador se haya autenticado.

Tabla 5: Gestionar precio de archivos multimedia

Caso de prueba: Gestionar archivo multimedia de pago.				
Nombre de la sección	**Escenarios de la sección**	**Descripción de la funcionalidad**	**Resultados Esperados**	**Resultados Obtenidos**
SC 1: Listar los archivos multimedia de pago.	EC 1.1: Listar los archivos multimedia de pago exitosamente.	El usuario selecciona la opción “Gestionar archivo multimedia de pago” y el sistema lista todos los archivos multimedia de pago con la opción de editar y eliminar precio.	El sistema lista todos los archivos multimedia de pago con la opción de editar y eliminar precio.	El sistema lista todos los archivos multimedia de pago con la opción de editar y eliminar precio.
	EC 1.2: Listar los archivos multimedia de pago fallido.	El usuario selecciona la opción “Gestionar archivos multimedia de pago” y el sistema no tiene ningún archivo multimedia de pago almacenado.	El sistema no tiene ningún archivo multimedia de pago almacenado por lo que muestra la lista vacía.	El sistema no tiene ningún archivo multimedia de pago almacenado por lo que muestra la lista vacía.

SC 2: Editar precio.	EC 2.1: Editar precio exitosamente.	El sistema muestra un formulario con el valor actual del precio del archivo multimedia seleccionado, el usuario presiona el botón “Guardar”. El sistema verifica que los datos modificados sean correctos, los guarda y muestra un mensaje confirmando que la acción se realizó correctamente.	El sistema modifica el precio del archivo multimedia, muestra un mensaje indicando que el precio se modificó satisfactoriamente.	El sistema modifica el precio del archivo multimedia.
	EC 2.2: Editar precio fallido.	El usuario presiona el botón “Guardar” antes de introducir el nuevo precio en el campo obligatorio y el sistema muestra un mensaje de error indicando que el campo es obligatorio y debe introducir un valor.	El sistema muestra un mensaje de error indicando que el campo es obligatorio y debe introducir un valor.	El sistema muestra un mensaje de error indicando que el campo es obligatorio y debe introducir un valor.

SC 3: Eliminar precio.	EC 3.1: Eliminar precio exitosamente.	El usuario presiona el botón “Eliminar precio” del archivo multimedia seleccionado, el sistema muestra un mensaje de confirmación de la acción y si el usuario acepta se elimina el precio del archivo multimedia y se muestra un mensaje indicando que la acción se realizó satisfactoriamente.	El sistema muestra un mensaje de confirmación, se elimina el precio del archivo multimedia satisfactoriamente y se actualiza la lista de archivos multimedia de pago.	El sistema muestra un mensaje de confirmación, se elimina el precio del archivo multimedia satisfactoriamente y se actualiza la lista de archivos multimedia de pago.
	EC 3.2: Eliminar precio fallido.	El usuario presiona el botón “Eliminar precio” del archivo multimedia seleccionado, el sistema muestra un mensaje de confirmación de la acción y si el usuario cancela no se realiza ningún cambio en los datos del archivo multimedia seleccionado.	El sistema no elimina y muestra el listado de los archivos multimedia de pago.	El sistema no elimina y muestra el listado de los archivos multimedia de pago.

SC 4: Adicionar archivo multimedia de pago.	EC 4.1: Adicionar archivo multimedia de pago exitosamente.	El usuario presiona el botón “Adicionar” y el sistema muestra una lista de todos los archivos multimedia libres de costo con la opción de adicionar precio.	El sistema lista todos los archivos multimedia libres de costo con la opción de adicionar precio.	El sistema lista todos los archivos multimedia libres de costo con la opción de adicionar precio.
	EC 4.2: Adicionar archivo multimedia de pago fallido.	El usuario presiona la opción “Adicionar” y el sistema no tiene ningún archivo multimedia libre de costo almacenado.	El sistema no tiene ningún archivo multimedia libre de costo por lo que muestra una lista vacía.	El sistema no tiene ningún archivo multimedia libre de costo por lo que muestra una lista vacía.
SC 5: Adicionar precio a archivo multimedia libre de costo.	EC 5.1: Adicionar precio a archivo multimedia libre de costo exitosamente.	El sistema muestra un formulario para adicionar el precio al archivo multimedia libre de pago seleccionado. El usuario presiona el botón “Guardar”. El sistema verifica que los datos introducidos sean correctos y muestra un mensaje informando que la acción se realizó correctamente.	El sistema adiciona el precio al archivo multimedia seleccionado, muestra un mensaje indicando que se realizó la acción y se actualiza la lista de archivos multimedia de pagos.	El sistema adiciona el precio al archivo multimedia seleccionado, muestra un mensaje indicando que se realizó la acción y se actualiza la lista de archivos multimedia de pagos.

	EC 5.2: Adicionar precio a archivo multimedia libre de costo fallido.	El usuario presiona la opción “Guardar” antes de introducir los datos obligatorios y el sistema muestra un mensaje de error indicando que existen campos obligatorios por llenar.	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.	El sistema muestra un mensaje indicando que existen campos obligatorios por llenar.

Tabla 6: Descripción de variables

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Precio	Campo de texto	No	Campo para entrar el precio del archivo multimedia, es obligatorio.

La siguiente tabla muestra aspectos como V (Válido), I (Inválido), N/A (No Aplica) que indican el valor del dato.

Tabla 7: Matriz de datos para el escenario Adicionar precio a archivo multimedia libre de costo

Escenario	Precio	Respuesta del sistema
Adicionar precio a archivo multimedia libre de costo exitosamente.	V: 123	Se adiciona el precio al archivo multimedia y se muestra un mensaje de confirmación.
Adicionar precio a archivo multimedia libre de costo fallido.	I: Campo vacío	El sistema muestra un mensaje de error indicando que se deben llenar los campos obligatorios.

Resultados de las pruebas:

Las pruebas de caja negra utilizando la técnica de partición equivalente se le realizó a cada uno de los casos de usos, generando un diseño de caso de prueba con sus respectivos escenarios por cada uno. En los resultados obtenidos durante la primera iteración se identificaron 21 no conformidades, la mayoría de estas relacionadas a faltas de ortografías en los mensajes mostrados al usuario y otras atadas directamente a errores en la comunicación con identidades externas al sistema, las cuales fueron corregidas. En una segunda iteración las pruebas arrojan un total de 6 no conformidades, la mayoría de estas asociadas a validaciones incorrectas en los formularios, corrigiéndose las mismas y obteniendo una tercera iteración libre de no conformidades. A continuación en la figura 9 se muestran los resultados de las iteraciones realizadas.

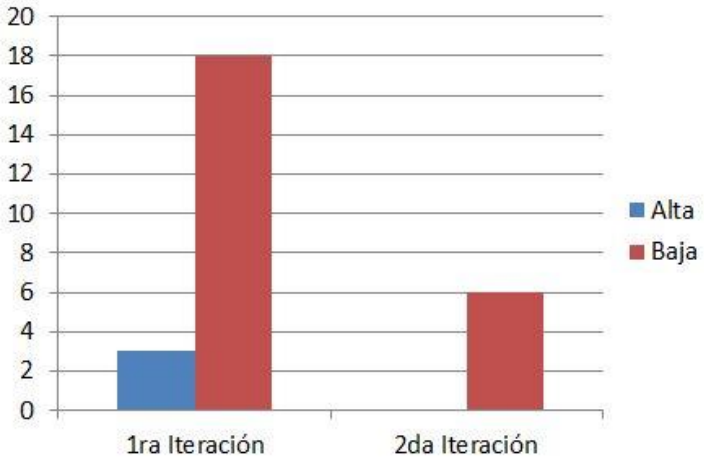


Fig. 9: Prueba de caja negra 1ra y 2da iteración.

3.5 Conclusiones del capítulo

Como resultado de este capítulo se obtuvo el diagrama de componentes el cual muestra la estructura física que tiene el componente de software y el diagrama de despliegue que permitió modelar la arquitectura del sistema. Con la realización de las pruebas de caja negra aplicando la técnica de partición equivalente se determinaron 21 no conformidades en su primera iteración a las cuales se le dieron solución. Con la realización de esta prueba el Modelo de Pruebas quedó conformado por el caso de pruebaGestionar archivo multimedia de pago.Los resultados de las prueba 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

La realización de la investigación arroja como principal resultado, el desarrollo de un módulo de pasarela de pago para la plataforma AGORAV, a partir de lo cual se le da cumplimiento a los objetivos propuestos:

- La elaboración del marco teórico de la investigación permitió realizar un estudio de los principales conceptos relacionados a los mecanismos de pago.Así como analizar diferentes sistemas que brindan aspectos relevantes y reutilizables en la implementación de un módulo para la plataforma AGORAV que permite realizar el pago por los servicios de visualización y descarga de archivos multimedia.
- La caracterización de las herramientas y tecnologías definidas en el proyecto CPM, utilizadas en el desarrollo del módulo permitieron sentar las bases de la implementación y facilitar la integración con la plataforma AGORAV.
- La utilización del patrón arquitectónico MVC permite la reutilización e incrementa la flexibilidad del módulo implementado, ya que propone la separación de la interfaz de usuario con la lógica del negocio.
- Se definieron requisitos funcionales y no funcionales, además de los diferentes casos de uso con sus respectivas descripciones permitiendo definir las características que debería cumplir del módulo.
- Los artefactos generados durante el desarrollo del módulo permiten el mantenimiento y futuro seguimiento de la solución propuesta.
- El módulo desarrollado como resultado de la investigación realizada, permite a los usuarios efectuar el pago online por los servicios de visualización y descarga de la plataforma AGORAV, aumentando así el campo de comercialización de los productos publicados en la misma, y permitiendo a esta plataforma insertarse en otras áreas de comercialización.

Recomendaciones

Luego de cumplidos los objetivos planteados en el presente trabajo se muestra a continuación algunas de las ideas recomendadas para trabajos futuros:

- Incorporar a la pasarela de pago otros modos de pago apoyándose para su creación en la arquitectura que se propone en el módulo CECA.
- Integrar a las reglas del negocio la posibilidad de pagos mensuales por consumo de todos los contenidos audiovisuales durante ese periodo de tiempo, lo que aumentaría la preferencia de los usuarios por el sistema.
- Incorporar a la lógica del negocio el almacenamiento de dinero en el sistema por parte de los usuarios, proporcionando a los mismos un modo de pago en el que no deban enviar los datos de su tarjeta en cada compra.

Bibliografía

Álvarez, Miguel Angel. 2001. Desarrollo web. *Desarrollo web.* [En línea] 9 de mayo de 2001. http://www.desarrolloweb.com/php/.

—. 2001. Desarrollo web. *Desarrollo web.* [En línea] 9 de mayo de 2001. http://www.desarrolloweb.com/php/.

—. 2008. Desarrollo Web. *Desarrollo Web.* [En línea] 11 de noviembre de 2008. http://www.desarrolloweb.com/articulos/que-es-un-cms.html.

Booch, Grady. 2004.*Object-Oriented Analysis and Design with Applications.* 2004.

Booch, Grady, Jacobson, Ivar y Rumbaugh, James. 2000.*El Proceso Unificado de Desarrollo de Software.* 2000.

Booch, Grady, Rumbaugh, James y Jacobson, Ivar. 2000.*El lenguaje unificado de modelado.* 2000.

Carlos Reynoso, Nicolás Kiccillof. 2004.*Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* 2004.

Cases, Eduard Fúmas. 2014. Ibrugor. *Ibrugor.* [En línea] 21 de octubre de 2014. http://www.ibrugor.com/blog/que-es-php-para-que-sirve/.

Collins-Sussaman, Ben, Fitzpatrick, Brian W y Pilato, C Michael. 2004. Control de versiones con Subversion. [En línea] 2004. http://svnbook.red-bean.com/nightly/es/svn-ch-1-sect-1.htm.

Conallen, Jim. 1999.*UML Extension for Web Applications.* 1999.

Dans, Enrique. 2007.*Comercio Electrónico.* 2007.

E.A. Bertino, L.A. Martino. 1995.*Sistemas de bases de datos orientadas a objetos. Ediciones Díaz de Santos.* 1995.

El comercio electrónico en internet. **Ferreiro, Teresa. 1997.** 35, s.l. : Distribución y consumo, 1997.

Geschwinde, E. y Schöenig, HJ. 2002.*Postgresql: Developer's Handbook.* 2002.

Gil, Fran. 2012.*Experto en Drupal 7 Nivel Inicial.* 2012.

Gómez, José Jorge Márquez.*Arquitectura MVC. Visión General.*

Gonzalo Génova, María C. Valiente, Jaime Nubiola. 2006.*Modelos en UML: un enfoque semiótico.* 2006.

Guerrero, Rafael Martinez. 2010. PostgreSQL-es. *PostgreSQL-es.* [En línea] 2 de 10 de 2010. [Citado el: 11 de 3 de 2015.] http://www.postgresql.org.es/sobre_postgresql.

Henst, Christian Van Der. 2001. Maestros del web. *Maestros del web.* [En línea] 23 de mayo de 2001. http://www.maestrosdelweb.com/.

Hernández León, Rolando Alfredo y Coello González, Sayda. 2011.*El proceso de investigación .* s.l. : La Habana : Editorial Universitaria, 2011.

Hernandez, Camelo. 2014. Slideshare. *Slideshare.* [En línea] 20 de septiembre de 2014. http://es.slideshare.net/carmeloh2/metodologa-open-up-39321348.

Hernández, Camelo. 2014. Slideshare. *Slideshare.* [En línea] 20 de septiembre de 2014. http://es.slideshare.net/carmeloh2/metodologa-open-up-39321348.

Hernández, S. R. 2008.*Metodología de la investigación. La Habana: Félix Varela.* 2008.

Herrero, Pablo. 2012. Blog Sage Experience. *Blog Sage Experience.* [En línea] 14 de 11 de 2012. [Citado el: 11 de 3 de 2015.] http://blog.sage.es/innovacion-tecnologia/la-pasarela-de-pago-es-el-motor-de-los-negocios-online/.

Hípola, Pedro. 1994.*World Wide Web: toda la Internet en un solo documento.* 1994.

Hispano, Drupal. 11. Comunidad de usuarios de Drupal. *Comunidad de usuarios de Drupal.* [En línea] 2005 de abril de 11. [Citado el: 25 de mayo de 2015.] http://drupal.org.es/drupa.

Hispano., Drupal. 2005. Comunidad de usuarios de Drupal. [En línea] 11 de abril de 2005. [Citado el: 26 de mayo de 2015.] http://drupal.org.es/drupa.

IEEE. 1990.*Standard Glossary of Software Engineering Terminology. New York.* 1990.

Imprenta, La. 2015. La Imprenta. Tienda de Impresión OnLine. *La Imprenta. Tienda de Impresión OnLine.* [En línea] 2015. [Citado el: 22 de 3 de 2015.] https://www.laimprentacg.com/imprentaonline/content/14-pago-seguro-con-ssl.

International, Visual Paradigm. 2008. Visual Paradigm. *Visual Paradigm.* [En línea] 2008. [Citado el: 11 de 3 de 2015.] http://www.visual-paradigm.com/product/vpuml/features/.

Internet y la Sociedad. **Castells, Manuel. 2000.** Cataluña : s.n., 2000.

Internet y la sociedad Red. **Castell, Manuel. 2001.** Catalunya : s.n., 2001.

Kenneth Kendall, Julie Kendall. 1998.*Análisis y Diseño de Sistemas.3ra Edición. México: Prentice Hall.* 1998.

Larman, Craig. 2003.*UML y Patrones. 2da Edicion.* 2003. 2.

Luke Welling, Laura Thomson. 2005.*Desarrollo Web con PHP y MySQL.* 2005.

Marc Gibert Ginestá, Oscar Pérez Mora.*Bases de datos en PostgreSQL.*

Mercer, David. 2010.*Drupal 7. Packt Publishing.* 2010.

Meza, Mirna. 2011. Herramientas CASE. *Herramientas CASE.* [En línea] 2 de abril de 2011. http://fds-herramientascase.blogspot.com/.

Monetos. 2012. monetos.es. *monetos.es.* [En línea] 2012. [Citado el: 11 de 3 de 2015.] http://www.monetos.es/financiacion/tarjetas-credito/definicion-formato/.

—. 2012. monetos.es. *monetos.es.* [En línea] 2012. [Citado el: 11 de 3 de 2015.] http://www.monetos.es/inversiones/tarjetas-debito/definicion-formato/.

Mora, Ridel Oscar García. 2014. Revista Cubana de Ciencias Informáticas. [En línea] 6 de marzo de 2014. http://scielo.sld.cu/scielo.php?pid=S2227-18992014000200008&script=sci_arttext.

—. 2014. Revista Cubana de Ciencias Informáticas. *Revista Cubana de Ciencias Informáticas.* [En línea] 6 de marzo de 2014. http://scielo.sld.cu/scielo.php?pid=S2227-18992014000200008&script=sci_arttext.

Mühlrad, Daniel. 2008.*Patrones de diseño.* 2008.

NetBeans. 2013. [En línea] 2013. [Citado el: 25 de mayo de 2015.] https://netbeans.org/features/index.html.

—. 2015. [En línea] 2015. [Citado el: 15 de 1 de 2015.] https://www.netbeans.org/features/index.html.

Pasarela de pago para la seguridad de transacciones bancarias en lineas. **Fonseca, Damaris Solis, Roque Peres, Wilfredo y Morilla Faurés, María Lourdes. 2013.** s.l. : Área de Innovación y Desarrollo, S.L., 2013.

Pasarela de pago para la seguridad de transacciones bancarias en líneas. **Fonseca, Damaris Solis, Roque Peres, Wilfredo y Morilla Faurés, María Lourdes. 2013.** s.l. : Área de Innovación y Desarrollo, S.L., 2013.

Pasarela de pagos para la seguridad de transacciones bancarias en línea. **Damaris Solis Fonseca, Wilfredo Roque Pérez, María Lourdes Morilla Faurés. 2013.** 2013.

Patrones de diseño. **Rojas, Juan Carlos Olivares.** Mexico : s.n.

Pressman, Roger. 2005.*Ingeniería de Software: Un Enfoque Práctico. 6ta Edición.* 2005.

Pressman, Roger S. 2002.*Ingeniería de Software, un enfoque práctico.* 2002.

—. 2002.*Ingeniería de Software: Un enfoque práctico. 5ta edicion.* s.l. : Mc Granw Hill, 2002.

Pressman, Roger. 2011.*Software Engineering, a practitioner's approach. (7ª edición).* 2011. ISBN 9780071267823.

Rodríguez, Fran Gil. 2012.*Curso de creación y gestión de portales web con Drupal 7.* s.l. : Forcontu S.L., 2012.

—. 2012.*Experto en Drupal 7.* s.l. : Forcontu, 2012.

Rojas, M.C. Juan Carlos Olivares.*Patrones de Diseño.*

Sánchez, Tamara Rodríguez. 2015.*Metodología de desarrollo para la Actividad productiva de la UCI. Versión 1.2.* La Habana : s.n., 2015.

Schmuller, Joseph. 2001.*Aprendiendo UML en 24 horas.Prentice Hall.* 2001.

Sommerville, Ian. 2005.*Ingeniería de Software, séptima edición.* Madrid : s.n., 2005.

—. 2007.*Software Engineering. 8va Edición.* 2007.

Studio, Visual. 2015. microsoft. *microsoft.* [En línea] 2015. [Citado el: 10 de mayo de 2015.] https://msdn.microsoft.com/es-es/library/dd409390.aspx.

Tello, Jesús Cáceres. 2012. Diagramas de Caso de Uso. *Diagramas de Caso de Uso.* [En línea] marzo de 2012. http://www.buenastareas.com/ensayos/Diagrama-De-Casos-De-Uso/3728817.html.

—. 2012. Diagramas de casos de uso. *Diagramas de casos de uso.* [En línea] marzo de 2012. http://www.buenastareas.com/ensayos/Diagrama-De-Casos-De-Uso/3728817.html/.

Trevor, James. 2010.*Drupal Web Services. Packt Publishing.* 2010.

UML, Tutorial. 2007. sparxsystems. *sparxsystems.* [En línea] 2007. [Citado el: 10 de mayo de 2015.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

Valdés, Damián Pérez. 2007. Maestros del web. *Maestros del web.* [En línea] 3 de julio de 2007. http://www.maestrosdelweb.com.

Zayas, Carlos Alvarez de. 1995.*Metodología de la Investigación Científica.* Santiago de Cuba : s.n., 1995.

Anexos

I. Entrevista realizada.

Fecha: Diciembre 2014.

Entrevistador: Raciel Correa Barbán.

Entrevistados: Ing. Pavel Mena Nodal (Arquitecto y desarrollador), Ing. Yarisel Rojas Castellanos (Líder del proyecto CPM).

Preguntas:

1. ¿Qué beneficio aportaría para la plataforma AGORAV una aplicación para efectuar el pago online por los servicios de visualización y descarga de los archivos multimedia publicados?
2. ¿Cuáles son los requisitos fundamentales que debería cumplir la aplicación a desarrollar?

Respuestas a la entrevista:

Pregunta 1: En la respuesta obtenida los entrevistados coincidieron que el principal beneficio que traería para la plataforma el desarrollo de este módulo sería la facilidad que le brinda a los usuarios efectuar el pago a través de sus cuentas externas. Este mecanismo de pago aumentaría la preferencia de los usuarios por el sistema.

Pregunta 2: Algunos requisitos en los cuales existió coincidencia en las respuestas de los entrevistados son:

- Contar con un carrito de compras accesible desde cualquier parte de la aplicación.
- Selección de varios modos de pago.
- Gestión de precio de los archivos multimedia.
- Generar un historial de transacciones.

II. Especificaciones de CU.

Tabla 8: Efectuar Pago

Objetivo	Efectuar pago por los archivos multimedia.
Actores	Usuario.
Resumen	El caso de uso inicia al seleccionar la opción "*Comprar*", muestra un listado de todos los archivos multimedia seleccionados para comprar y la opción de seleccionar el modo de pago. El caso de uso culmina al realizar el pago y registrarlo en el historial de transacciones.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Que el usuario esté autenticado
Postcondiciones	Se efectuó el pago.
Referencias	RF1, RF2, RF3

Flujo de eventos

Flujo básico: Efectuar pago.

	Actor	**Sistema**
1.	Selecciona la opción "*Comprar*".	
2.		Muestra una interfaz brindando un listado de los archivos multimedia seleccionados para pagar. Además de la opción de seleccionar un

		modo de pago. (Ver sección 1)
3.	Selecciona la opción “Pagar”.	Valida la completitud de los datos.
4.		Registra el pago realizado en el historial de transacciones.
5.		Envía los datos a la entidad bancaria realizando la transacción.
6.		Muestra un mensaje de notificación(“”) y actualiza el historial de transacciones. Termina el caso de uso.
Sección 1: “*Seleccionar modo de pago*”		
Flujo básico: Seleccionar modo de pago		
	Actor	**Sistema**
1.	Selecciona un modo de pago.	
2.		Muestra una interfaz para introducir los datos correspondientes al modo de pago seleccionado.
3.	Introduce los datos correspondiente al modo de pago seleccionado.	
Flujos alternos		
Nº Evento: Selecciona la opción “*Cancelar*”.		

	Actor	Sistema
1.	Selecciona la opción “*Cancelar*”.	
2.		Regresa al paso 1 del Flujo Básico: Efectuar pago. Termina el caso de uso.

Flujos alternos
Nº Evento: Campo obligatorio vacío.

	Actor	Sistema
3.1		Muestra un mensaje de error indicando que hay campos vacíos. Regresa al paso 1 del Flujo Básico: Seleccionar modo de pago.

Flujo alterno
Nº Evento: Transacción incorrecta.

	Actor	Sistema
5.1		Muestra un mensaje de error indicando que no se pudo realizar la transacción. Regresa al paso 1 del Flujo Básico: Efectuar pago.

Relaciones	**CU Incluidos**	Ninguno.
	CU Extendidos	Ninguno
Requisitos no funcionales	RNF3, RNF4	