



Universidad de las Ciencias Informáticas
XETID



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

Título: “Módulo de comercio electrónico utilizando la plataforma cubana (EnZona) de pago en línea para la Plataforma de Formación a Distancia (DEDUKO)”.



Autor:

Ariel Alejandro Méndez Achón

Tutor:

Ing. Vidal Orlando Acosta García

DrC. NemurySilega Martínez

La Habana, 2020 “Año 62 de la Revolución”.

Declaración autoría:

Declaramos ser autor(es) de la presente tesis y reconocemos a la XETID, Empresa de Tecnologías de la Información para la Defensa, y a la Universidad de las Ciencias Informáticas, los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ___ días del mes _____ del año _____.

Firma de Autor

Ariel Alejandro Méndez Achón

Firma de Tutor

Ing. Vidal Orlando Acosta García

Firma de Tutor

DrC. Nemury Silega Martínez

Datos del contacto:

Graduado en el año 2015 en la Universidad de las Ciencias Informática, con 4 años de experiencia como Ingeniero en Ciencias Informática. Ha cursado los postgrados de Gestión de Alcance y Tiempo, Arquitectura y componentes tecnológicos para Symfony 2, Gestión de la Calidad, Gestión de Recursos Humanos, Actualidad socioeconómica y política de Cuba en el contexto latinoamericano (edición 2), Preparación básica en seguridad y defensa nacional (edición 7), inglés (edición 5), La mejora de procesos de software. Una estrategia para lograr la calidad en el producto final. (edición 3) y Ciencia, Tecnología y Sociedad. Ha realizado varias publicaciones y ponencias científicas como autor en los eventos 7mo Taller de Información y Gestión Tecnológica, en la Peña Tecnológica del MININT, y la revista Serie Científica de la Universidad de las Ciencias Informáticas. Ha tutorado 3 tesis de pregrado dentro del área del conocimiento.

Agradecimientos:

A mis padres, mi hermano y mis tíos que dan su vida para que todo salga adelante y lo mejor posible. A toda mi familia en especial a mi tía Lien que estuvo en los últimos momentos de la batalla ayudándome, a mi novia y a mi cuñado Pacheco que siempre me dieron su mano en los momentos más difíciles. A todas mis amistades tanto de la UCI como de fuera. Al claustro de profesores que de una manera u otra aportaron su granito de arena. A la Revolución por darme esta gran oportunidad de graduarme en unas de sus grandes universidades. A la Universidad de Ciencias Informáticas por facilitarme las vías para formarme como ingeniero. A todos muchísimas gracias.

Dedicatoria:

A mi familia, principalmente a mis padres, mi hermano y mis tíos por ser mi principal fuente de motivación e inspiración para poder superarme cada día más, por ser mis guías, por confiar siempre en mí, por ayudarme a cumplir mis metas, por todo su sacrificio, comprensión y amor.

Resumen

El comercio electrónico permite mejorar la comunicación con los clientes, atender sus peticiones y preguntas generando una mayor fidelización. Actualmente su uso ha alcanzado un mayor impacto en los diferentes sectores de la sociedad debido a su horario comercial de 24 horas. La investigación realizada permitió desarrollar un módulo de pago en línea para la Plataforma de Formación a Distancia DEDUKO, de la Empresa de Tecnologías de la Información para la Defensa (XETID), que garantiza la comercialización de sus cursos en línea. Se utilizó el LMS Opigno, el IDE NetBeans y el Visual Paradigm como herramientas de desarrollo. Además, se utilizó la tecnología web PHP, HTML5, JavaScript y CSS3 y como tecnología de integración WSO2. Además, se recogen los resultados del trabajo investigativo realizado, presentando los distintos resultados obtenidos en cada una de las etapas definidas por el proceso de desarrollo de software PRODESOFTE de la XETID. El módulo desarrollado posee funcionalidades como: la gestión del precio de los cursos o capacitaciones, la selección del modo de pago por los usuarios y visualización del registro de los usuarios a cursos o capacitaciones luego de haber efectuado el pago. El desarrollo de este módulo contribuye a incrementar la comercialización de los productos ofrecidos por la Plataforma de Formación a Distancia DEDUKO lo cual le permite a la XETID una mayor visión en el campo comercial ofreciendo alternativas de pago en línea para los cursos de la plataforma, posibilitando realizar nuevas experiencias de cara al producto a personas ubicadas en diferentes latitudes.

Palabras clave: comercio electrónico, LMS Opigno, pago en línea, WSO2, XETID.

Abstract

The electronic commerce allows improving communication with customers, responding to their requests and questions, generating greater loyalty. Currently its use has reached a greater impact on different sectors of society due to its 24-hour business hours. The research carried out allowed the development of an online payment module for the DEDUKO Distance Training Platform of the Defense Information Technology Company (XETID), which guarantees the marketing of its online courses. The LMS Opigno, the IDE NetBeans and the Visual Paradigm were used as development tools. In addition, PHP, HTML5, JavaScript and CSS3 web technology was used and WSO2 as integration technology. In addition, the results of the research work carried out are collected, presenting the different results obtained in each of the stages defined by the XETID PRODESOFIT software development process. The developed module has functionalities such as: the price management of the courses or training, the selection of the payment method by the users and the visualization of the registration of the users to the courses or training after having made the payment. The development of this module contributes to increasing the commercialization of the products offered by the DEDUKO Distance Training Platform, which allows XETID a greater vision in the commercial field by offering online payment alternatives for the platform's courses, enabling carry out new experiences regarding the product to people located in different latitudes.

Keywords:e-commerce, Opigno LMS,online payment, WSO2, XETID.

Índice

Introducción - 1 -

1 Capítulo 1: Fundamentos teóricos sobre el comercio electrónico y el pago en línea - 6 -

 1.1 Elementos asociados al dominio del problema - 6 -

 Comercio electrónico..... - 6 -

 Sistemas de pagos..... - 2 -

 Pasarelas de pagos..... - 3 -

 1.2 Estudio de módulos de comercio electrónico existentes. - 4 -

 WooCommerce - 4 -

 VirtueMart..... - 5 -

 Drupal Commerce - 5 -

 PrestaShop - 7 -

 Magento - 7 -

 1.3 Valoración de los módulos estudiados - 2 -

 1.4 Metodología de desarrollo..... - 1 -

 1.5 Herramienta de modelado..... - 2 -

 1.6 Lenguaje Unificado de Modelado - 2 -

 1.7 Entorno de Desarrollo Integrado - 2 -

 1.8 Servidores web - 3 -

 1.9 Sistema gestor de base de datos..... - 3 -

 1.10 Marco de trabajo - 4 -

 1.11 Lenguajes de programación - 5 -

 1.12 API WSO2..... - 6 -

 1.13 Conclusiones parciales..... - 7 -

2 Capítulo 2: Análisis y diseño del módulo de comercio electrónico para la Plataforma Integral de Formación a Distancia DEDUKO..... - 8 -

2.1	Modelo de dominio.....	- 8 -
	Descripción de las clases del modelo de dominio.....	- 8 -
	Descripción general del modelo de dominio	- 9 -
2.2	Especificación de requisitos del módulo de videoconferencia para la Plataforma de Formación a Distancia DEDUKO	- 2 -
	Requisitos funcionales del módulo de comercio electrónico	- 2 -
	Requisitos no funcionales del módulo de comercio electrónico	- 3 -
2.3	Diseño de la arquitectura propuesta.....	- 5 -
	Arquitectura de software	- 5 -
	Patrón arquitectónico	- 5 -
2.4	Modelo de diseño.....	- 6 -
2.5	Diagrama de clases del diseño	- 6 -
2.6	Diseño de la base de datos.....	- 7 -
	Diagrama entidad – relación.....	- 7 -
	Descripción de las tablas del diagrama entidad – relación	- 8 -
2.7	Patrones de diseño	- 8 -
	Patrones GRASP	- 9 -
2.8	Conclusiones parciales	- 11 -
3	Capítulo 3: Implementación y prueba del módulo de videoconferencia para la Plataforma Integral de Formación a Distancia DEDUKO.....	- 13 -
	3.1 Diagrama de despliegue	- 13 -
	3.2 Diagrama de componentes	- 15 -
	3.3 Estándares de codificación	- 16 -
	3.4 Pruebas del módulo de comercio electrónico y pago en línea.....	- 2 -
	Estrategias de prueba	- 4 -
	3.5 Resultados de la investigación.....	- 5 -
	3.6 Conclusiones parciales	- 5 -

4 Conclusiones generales - 6 -

Referencias bibliográficas - 8 -

1. Anexos: - 11 -

 Anexo 1: Descripción de los Requisitos Funcionales del sistema..... - 11 -

 Anexo 2: Diagrama general de clases del diseño..... - 13 -

Índice de tablas

Tabla 1: Características de sistemas analizados..... - 3 -

Tabla 2: Descripción de la tabla " method of payment " - 8 -

Tabla 3: Descripción de los requisitos funcionales del módulo de videoconferencia. - 11 -

Tabla 4: Descripción de la tabla " user "..... - 17 -

Tabla 5: Descripción de la tabla " course " - 17 -

Tabla 6: Descripción de la tabla " capacitation " - 17 -

Tabla 7: Descripción de la tabla " price " - 18 -

Tabla 8: Descripción de la tabla " digital certificate " - 18 -

Índice de figuras

Ilustración 1: Fases de la metodología de desarrollo PRODESOF(T(Xetid 2014). - 1 -

Ilustración 2: Modelo de Dominio - 9 -

Ilustración 3: Patrón arquitectónico Modelo-Vista-Controlador. - 5 -

Ilustración 4: Diagrama de clases del diseño: Gestionar modalidad de pago. - 7 -

Ilustración 5: Diagrama entidad – relación. - 8 -

Ilustración 6: Patrones - 11 -

Ilustración 7: Diagrama de Despliegue..... - 13 -

Ilustración 8: Diagrama de Componente - 15 -

Ilustración 9: Diagrama General ClasesAnexo 3: Descripción de las tablas del diagrama entidad
– relación. - 13 -

Introducción

Las nuevas tecnologías de la información y la comunicación (TIC) tienen un papel fundamental en la sociedad actual y su vida cotidiana. Estas han tenido un impacto tal, que no solo han cambiado la forma de interactuar entre las personas, sino que además han sido una revolución para el sector empresarial y para la sociedad en general. Han permitido el desarrollo de sistemas cada vez más complejos que logran disminuir las distancias que separan a las personas constituyendo una herramienta de desarrollo para garantizar un aumento en la cantidad y calidad de la información intercambiada.

El comercio electrónico como parte de las tecnologías de la información y comunicación ha alcanzado un vertiginoso auge permitiendo que su uso sea cada vez más frecuente. Es utilizado por empresas u organizaciones para la realización de compra-venta a través de Internet, trayendo consigo un mayor impacto en la esfera económico político y social. Además, permite obtener un mayor número de clientes tanto *online*¹ como *offline*² gracias al aumento de visibilidad que permite Internet. Ofrece ventajas como comodidades para los clientes que desde cualquier lugar pueden acceder a sus servicios y ofrecer compras o ventas en cualquier momento y sin importar muchas veces sus localizaciones, debido a que no tiene que desplazarse ya que la tienda está abierta las 24 horas del día. Permite el acceso a un mercado global debido a que no conoce fronteras geográficas, sino que todo está a un clic y una mayor interactividad con los clientes gracias a la gran cantidad de información que proporciona Internet. Posibilita ofrecer mucha más información al comprador y ahorro de tiempo a la hora de realizar las compras.

Sin embargo, no se puede dejar de considerar que algunas desventajas del comercio electrónico se deben tener en cuenta antes de iniciar o ampliar cualquier negocio tales como la seguridad de la información y la privacidad y protección de los datos de los usuarios. Además, la necesidad de un dispositivo con acceso a Internet y la seguridad de los datos pueden hacer que los clientes no se sientan seguros al comprar en línea debido a que su información financiera corra peligro o se divulgue.

¹ En línea

² Desconectado

En Cuba se trabaja intensamente en la ampliación de la informatización de la sociedad cubana y como consecuencia se trabaja en el fortalecimiento de las herramientas tales como es las de comercio electrónico que se experimentan soluciones en la esfera de la economía. La empresa XETID es un pilar en el desarrollo de software, cuenta con la División de Tecnologías Bancarias donde se desarrolló la integración de la tienda virtual de 5ta y 42 con la pasarela de pago EnZona.

En la empresa XETID se encuentra la División de Tecnologías Bancarias la cual cuenta con la plataforma EnZona; la cual es un sistema electrónico que dentro de sus funcionalidades se encuentran las de transferencia de dinero, conocer saldo y ver últimas operaciones de tarjetas magnéticas cubanas con el objetivo de impulsar el comercio electrónico (Xetid 2019). Otra de las divisiones existentes es la División de Servicios Profesionales (SP), la cual se especializa en el desarrollo de portales web, audiovisuales, gestión de centros de datos, diseño de procesos internos de una empresa u organismo y prestación de servicios técnicos. Cuenta con 4 centros productivos, dentro de los cuales se encuentra el Centro de Audiovisuales y Portales (CENAP), el cual cuenta actualmente con la Plataforma de Formación a Distancia (DEDUKO) que integra un conjunto de tecnologías de desarrollo (web, de software, etc.) entre las que se encuentran Drupal como gestor de contenido.

Los cursos dentro de la plataforma brindan una serie de modalidades de pagos. La primera modalidad de pago es al iniciar el curso, cuando el cliente se registra en el mismo. La otra modalidad de pago que existe es en el momento que el usuario terminó el curso y desea adquirir su certificado digital. En la actualidad la plataforma no cuenta con la posibilidad de comercializar sus cursos en línea lo cual no brinda una serie de facilidades al cliente como realizar el pago en línea en tiempo real. Esto trae consigo que gaste recursos financieros en la contratación para certificarse en cursos o capacitaciones, así como el tiempo que tiene que utilizar para visitar la empresa en busca del título lo cual afectaría su horario laboral ya aumenta los gastos debido al viaje. Además, no puede realizar el pago de una forma por lo que al hacerlo manualmente puede hacer que los empleados pierdan muchas horas. Esto no solo es tedioso, sino que también es propenso al error humano. Por lo que resulta necesario disponer de la información desde cualquier lugar y en cualquier momento, aspecto que se ve obstaculizado sin la automatización.

A partir de la situación problemática descrita, se plantea el siguiente **problema a resolver**: ¿Cómo garantizar el pago en línea de la certificación de los cursos para la plataforma de Formación a distancia DEDUKO? En consecuencia, con el problema planteado se determinó como **objeto de estudio** de la investigación: El comercio electrónico en plataformas web, enmarcado en el **campo de acción**: Los procesos asociados a la compra y pago en línea en la plataforma de Formación a Distancia DEDUKO. Para el desarrollo de la investigación se define como **objetivo general**: Desarrollar un módulo de comercio electrónico utilizando la plataforma cubana (EnZona) de pago en línea para la Plataforma de Formación a Distancia (DEDUKO) que garantice el pago en línea de los cursos.

Para dar cumplimiento al objetivo general se proponen los siguientes **objetivos específicos**:

1. Construir el marco teórico conceptual y el estado del arte respecto a las tecnologías actuales relacionadas con el comercio electrónico y la integración con sistemas de pagos en línea.
2. Analizar las herramientas y tecnologías apropiadas para la realización del pago en línea dentro de la Plataforma de Formación a Distancia DEDUKO.
3. Realizar el análisis y diseño del módulo de comercio electrónico y pago en línea para la Plataforma de Formación a Distancia DEDUKO.
4. Implementar el módulo de comercio electrónico y pago en línea para la Plataforma de Formación a Distancia DEDUKO.
5. Validar la solución desarrollada a partir de una estrategia de pruebas.

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Se clasifican en teóricos y empíricos, los cuales están dialécticamente relacionados (Rolando Alfredo Hernández León 2012).

Para la investigación se han utilizado diferentes tipos de **métodos científicos de investigación**, los cuales se describen a continuación:

Métodos teóricos

Histórico- Lógico: el método histórico analiza la trayectoria completa del fenómeno, su condicionamiento a las diferentes fases de la historia, revela las etapas principales de su desenvolvimiento y las conexiones históricas fundamentales. El método lógico se basa en el estudio histórico del fenómeno, pone de manifiesto la lógica interna de su desarrollo, de su

teoría y encuentra el conocimiento más profundo de su esencia(**Rolando Alfredo Hernández León** 2012). Este método se utilizó para el estudio de la situación actual del tema (estado del arte),

analizando posibles soluciones existentes, además permitió seguir la evolución y desarrollo, incluyendo las tendencias actuales, de los mecanismos de pago electrónico para la solución del problema anterior. Además, se utiliza para estudiar la evolución de los conceptos asociados al comercio electrónico y pago en línea permitiendo así la definición de términos propios.

Analítico-Sintético: el análisis permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio. La síntesis por su parte establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas(**Rolando Alfredo Hernández León** 2012).

Este método se utiliza para seleccionar las herramientas y tecnologías a utilizar durante el desarrollo de la aplicación, además permite la evaluación de otras soluciones que respondan al problema, lo que permite realizar una valoración crítica y detallada de cada una de ellas. Además, ayudó a sintetizar el análisis realizado a toda la documentación consultada para el desarrollo de la investigación.

Otros métodos de investigación

Conflicto de diálogo: se basa en crear posiciones opuestas entre dos o más personas para estimular el debate y conocer los criterios de los interlocutores sobre el tema, abierta y espontáneamente, pues el conflicto que produce el debate estimula a los contendientes a dar información que en otros momentos no expresarían(**Rolando Alfredo Hernández León** 2012). Permitted realizar un conflicto de diálogo con los especialistas y directivos del Centro de audiovisuales y portales con el fin de comprender el proceso de funcionamiento de la Plataforma de Formación a Distancia DEDUKO. Además, permitió describir e identificar las necesidades a solucionar, lo que posibilitó obtener información necesaria para la especificación de los requisitos.

El resultado final de la investigación será un módulo de comercio electrónico integrado con la pasarela de pago cubana(EnZona) que permitirá el pago en línea de la contratación para la certificación o capacitación de los cursos disponibles en la Plataforma de Formación a Distancia (DEDUKO), brindándole al cliente un mejor servicio con el pago en línea y

ahorrándole tiempo en dicha operación sin afectarle el horario laboral. Además, no llevará a cabo un gasto financiero ya que no se realizará ningún viaje a la empresa lo cual creará más confianza con el cliente y le dará una mayor visión acerca de las distintas ofertas que existen en la plataforma.

El presente trabajo de diploma está estructurado en los siguientes capítulos:

Capítulo 1: Fundamentos teóricos sobre el comercio electrónico y pago en línea: En este capítulo se establecen las bases teóricas generales de la investigación. Se exponen los principales conceptos asociados al dominio del problema. Se realiza un análisis detallado del estado del arte relacionado con los diferentes módulos de comercio electrónico existentes. Se muestran las herramientas, la metodología y los lenguajes utilizados en el desarrollo del módulo.

Capítulo 2: Análisis y diseño del módulo de comercio electrónico y pago en línea para la Plataforma Integral de Formación a Distancia DEDUKO: En este capítulo se representa el modelo de dominio con los conceptos más significativos del negocio y la relación existente entre ellos. Se describe la solución propuesta y se define el listado de los requisitos funcionales y no funcionales. Se diseña la arquitectura de software a utilizar.

Capítulo 3: Implementación y prueba del módulo de comercio electrónico y pago en línea para la Plataforma Integral de Formación a Distancia DEDUKO: En este capítulo se exponen elementos de la metodología específica a usar para la implementación de la solución. Además, se definen los tipos de pruebas y los casos de prueba realizados al componente para medir su nivel de cumplimiento con los objetivos especificados.

1 Capítulo 1: Fundamentos teóricos sobre el comercio electrónico y el pago en línea

En el capítulo se define la base teórica y conceptual para el desarrollo de la propuesta de una solución. Se abordan los conceptos relacionados con el proceso de pago y transacciones de los cursos. También se realiza un estudio del estado del arte para obtener más información actualizada acerca de los módulos de comercio electrónico existentes actualmente y conocer que elementos deben ser tomados para el desarrollo de una solución. Se analizan las tecnologías, metodologías y herramientas más factibles para utilizar en el proceso de desarrollo.

1.1 Elementos asociados al dominio del problema

Para lograr una mejor comprensión del dominio del problema se abordan los conceptos y elementos básicos que son fundamentales para el desarrollo de la investigación, y que le brindan solidez al mismo.

Comercio electrónico

El comercio electrónico (también conocido como e-Commerce) se podría definir como la compraventa de bienes y/o servicios a través de Internet, o utilizando Internet únicamente como una sola parte del proceso (Carranza 2019).

La definición en cuanto al término según la Comisión Europea es: "el comercio electrónico como todo tipo de negocio, transacción administrativa o intercambio de información que utilice cualquier tecnología de la información y de las comunicaciones". En este sentido vemos como el comercio electrónico se relaciona con la educación virtual en la medida en que se establecen los contratos electrónicos (Velazco et al. 2017).

A continuación, se mostrarán ventajas y desventajas de las cuales se debe tener muy en cuenta acerca del comercio electrónico. Para las ventajas se encuentran (Mora Barreto Miguel Alfredo 2018):

- ✓ Crea compañías totalmente virtuales.
- ✓ Mejora el posicionamiento de la compañía.
- ✓ Innovación del proceso tradicional de compra y venta a un proceso virtual.
- ✓ Incrementa notablemente la tecnificación de las empresas.
- ✓ Favorece y modifica la relación que se establece entre el fabricante, el distribuidor y los consumidores.

En las desventajas se encuentran(**Mora Barreto Miguel Alfredo** 2018):

- ✓ Transmite inseguridad en las transacciones.
- ✓ Desconfianza en la forma de pago.
- ✓ Nos ayuda en la comunicación para poder tramitar quejas o devoluciones.

Existen bastantes modelos de comercio electrónico que se pueden clasificar mediante la forma en la que interaccionan los clientes o empresas bien entre ambos(empresa-cliente) o bien entre ellos mismos (cliente-cliente y empresa-empresa). Los modelos de negocio más importantes son los siguientes(**Serra** 2015):

Business to Business (B2B) El modelo de comercio electrónico B2B es un modelo de negocio en el cual las transacciones de bienes o servicios se realizan entre 2 empresas.

Peer to Peer (P2P) El modelo de comercio electrónico P2P es un modelo de negocio el cual va a tener estructura de red, esto quiere decir que carece de clientes y servidores fijos. En su lugar va a poseer una serie de nodos que se pueden comportar o como clientes, o como servidores de otros nodos. La filosofía del Peer to Peer consiste en que todos los usuarios deberán compartir entre ellos.

Consumer to Consumer (C2C) El modelo de comercio electrónico C2C es un modelo de negocio en el cual se establece un contacto directo entre usuarios particulares de bienes y servicios, los cuales realizarán entre ellos una transacción comercial sin ningún tipo de intermediario.

Business to Consumer (B2C) El modelo de comercio electrónico B2C es un modelo de negocio en el cual se encuentran tanto las empresas como los consumidores. Es la forma más popular en la red para realizar comercio electrónico. Se caracteriza por una venta directa a través de Internet. La venta tiene como característica fundamental que es al por menor, ya que un intermediario se encarga de realizar la función de “tendero”, es decir, de comprar el producto que necesita su cliente y hacérselo llegar a un destino. Este modelo de negocio refleja claramente la alta globalización que existe en el mundo ya que la entrega será a distancia y podría ser en cualquier parte del mundo.

El comercio electrónico se relaciona con la educación virtual en la medida en que se establecen los contratos electrónicos que permiten establecer relaciones y compromisos que tengan por

objeto, bienes y servicios informáticos como son la adquisición de hardware y software, así como los servicios de asistencia de programas.(Velazco, Bustamante and Pérez 2017).En este orden se ponen de manifiesto elementos que permiten la relación entre el comercio electrónico y el aprendizaje electrónico.

Otro aspecto importante es la integración de los medios de pago para el comercio electrónico, dentro de los cuales se pueden encontrar los sistemas de pago online y diferentes pasarelas de pago que utilizan las tarjetas de crédito, monederos electrónicos, las transferencias bancarias, etc. Es un punto fundamental puesto que es la forma en la que el cliente pagará por el artículo o servicio y es necesario que estos métodos sean seguros, confiables y sencillos tanto para el cliente como para el vendedor.

Sistemas de pagos

Los sistemas de pago constituyen la infraestructura básica para que los mercados puedan operar eficientemente, por lo que su adecuado funcionamiento depende de los procedimientos y reglas establecidas por las autoridades para que los usuarios puedan liquidar la adquisición de bienes, servicios y activos de manera sencilla. De esta manera, el sistema de pagos es definido como el conjunto de instrumentos y procedimientos para que las transferencias de dinero puedan circular eficientemente(Figueroa and Martínez 2016).Estos son los sistemas de pago más habituales:

Pagos en línea a través de tarjetas de crédito

Las tarjetas de créditos son herramientas de pago plásticas, numeradas y magnetizadas, emitidas por una entidad bancaria que autoriza a su portador a emplearlas como medio de pago en los negocios adheridos a este sistema. El instrumento que se utiliza para materializar estos pagos es la pasarela de pago.

Pagos en línea a través del móvil

El pago móvil es el medio que permite realizar transacciones financieras mediante teléfonos móviles, incluyendo el pago de productos y servicios, compras a distancia y la transferencia de dinero.Existe un gran número de aplicaciones que permiten realizar compras desde un teléfono móvil. Por ejemplo: Google Wallet, Visa payWave, PayPal, Buyster(CCM 2020).

Transferencias bancarias

Las transferencias bancarias son envíos de dinero realizados a la orden de un cliente desde su cuenta bancaria en una entidad a otra designada. En caso de realizarse entre cuentas del mismo banco se denomina traspaso (BBVA 2019).

El método de pago utilizado en la investigación es a través de tarjetas de créditos y no otro porque las operaciones de pago mediante este método permiten que los datos asociados al pago, como el número de tarjeta, se transmitan de manera segura, obteniendo confidencialidad. Tampoco permite controlar el uso que el comerciante hace de los datos de pago recibidos, lo que obliga al cliente a confiar en el comerciante. Además, le ofrece una mayor comodidad y le permite realizar el pago en tiempo real sin tener presente que día y hora es. Asociado al método de pago utilizado existen las pasarelas de pago las cuales brindan la posibilidad que una página web procese una tarjeta de crédito o algunos otros medios de pago y así cobrar la compra que hace un cliente en el caso de una comercialización directa de su empresa.

Pasarelas de pagos

Las pasarelas de pagos son sistemas de pago electrónico que permiten la realización de pagos y transferencias entre tiendas electrónicas y entidades bancarias de manera segura. Se encargan de cifrar la información confidencial que se requiere para ejecutar transacciones bancarias por las redes (Damaris Solis Fonseca 2013). Distintas pasarelas de pagos que existen hoy en la actualidad son:

PayPal

La plataforma de pago más utilizada en todo el mundo es PayPal, se puede pagar en sitios web y también es posible transferir dinero a otros usuarios. Puede entenderse como una alternativa a métodos más tradicionales, como los giros postales o cheques. Es la plataforma favorita para todos aquellos que compran *online*, gracias a la comodidad que ofrece. El mantenimiento de la plataforma se logra a través del cobro de un porcentaje al vendedor por cada transacción (PayPal 2020).

Google Wallet

Google Wallet puede realizar transferencias de dinero de la misma forma que se utiliza PayPal. Además, incorpora una tarjeta monedera e incluso, está previsto que exista una versión en plástico para poder pagar en los comercios físicos. Google Wallet está conectada a la cuenta del usuario de Google (como la de Gmail o Google Plus) y sus posibilidades son infinitas (YEEPLY 2019).

Sequra

Sequra, plataforma cuyo funcionamiento está basado en el de PayPal, es totalmente segura y muy recomendable para aquellos comercios que vendan productos con un coste elevado, ya que solo así el pago a plazos tendría sentido. A la hora de crearse una cuenta para implementar

Sequra en un comercio electrónico, hay que tener en cuenta que no permiten direcciones de correo de las plataformas gratuitas, como Gmail o Yahoo (SEQURA 2019).

Mangopay

Mangopay es una pasarela de pago que llegó al mercado para ofrecer soluciones a las plataformas que proporcionan servicios a terceros. Ofrece servicio a quienes cobran en nombre de otros, como en el caso de la venta de entradas y tiene una API³ un tanto complicada de utilizar puesto que está enfocada a plataformas (MANGOPAY 2020).

1.2 Estudio de módulos de comercio electrónico existentes.

WooCommerce

WooCommerce es el plugin de comercio electrónico gratuito de WordPress que te permite vender online de manera simple y escalable. Uno de los grandes beneficios de WooCommerce, es que es un plugin totalmente gratuito lo cual lo convierte en el plugin más poderoso y flexible que existe en el mercado. Es un plugin *Open Source*⁴, es actualizado constantemente por miles de programadores que colaboran para mejorarlo a diario, introduciendo mejoras, corrigiendo vulnerabilidades y añadiendo prestaciones (WordPress 2019).

³Interfaz de Programación de Aplicaciones

⁴Código Abierto

Es usable, sus extensiones son gratuitas y de bajo costo. Está diseñado para permitir a los gestores de tiendas administrar ellos mismos un sitio de comercio electrónico, sin necesidad de tener un desarrollador web. Permite la integración de diferentes plataformas de pago en línea y como sistemas de pago utiliza las transferencias bancarias, cheque, el pago en entrega y la pasarela de pago PayPal, además de otras formas de pago que se le puede incorporar como la plataforma en línea TPV Online y Stripe (**WooCommerce** 2020).

VirtueMart

VirtueMart es una solución de comercio electrónico de código abierto diseñada como una extensión de Mambo o Joomla (sistemas de gestión de contenidos). Está escrito en PHP y requiere el entorno de base de datos MySQL para el almacenamiento. Es el más adecuado para sitios web de tráfico de nivel bajo a medio. Basado en el framework Joomla MVC con todas sus ventajas como anulación de plantilla, implementación de complementos de Joomla para pago, envío, cupones, reglas de cálculo y campos de productos personalizados e instalación simple e integración de otras extensiones de Joomla (**VirtueMart** 2008).

El sistema es totalmente configurable y modular en función de todas las necesidades y presupuestos. VirtueMart soporta un número ilimitado de productos y categorías, con productos que puedan ser asignados a varias categorías. También permite la venta de productos descargables, y ofrece un modo de catálogo en el que la función del carrito está apagada. VirtueMart soporta múltiples precios para un solo producto, en torno a los grupos de comprador o un rango de cantidad (**VirtueMart** 2011). Es fácil de instalar, toneladas de extensiones y la integración sin fisuras con Joomla. Tiene carencia de características de carrito de compras. Todas las formas y medios de pago existentes (a elección) como por ejemplo tarjetas de crédito y siguientes pasarelas de pago como PayPal, 2Checkout, eWay, Worldpay, PayMate y NoChex, entre otras (**MGSCREATIVE** 2020).

Drupal Commerce

DrupalCommerce es un software de comercio electrónico de código abierto que aumenta el sistema de gestión de contenidos Drupal. En el contexto de un sitio basado en Drupal, DrupalCommerce presenta productos para la compra; camina con los clientes a través del proceso de compra, seguimiento de facturas, recibos, pedidos y pagos, facilita el envío y el pago, y realiza otras funciones que necesitan los comerciantes en línea. Drupal Commerce se

utiliza para crear sitios web de comercio electrónico y aplicaciones de todos los tamaños. En esencia, es simple y estricto, aplicando estrictos estándares de desarrollo y aprovechando las mejores características de Drupal 7 y los principales módulos como Vistas y Reglas para una máxima flexibilidad (**Drupal** 2019).

El módulo facilita la creación de diversos tipos de producto con atributos personalizables, así como la integración vía API⁵ de diversas pasarelas de pago. Mediante una gran variedad de módulos adiciones es posible añadir funciones extra como cupones de descuento, descarga de archivos, métodos de envío y opciones de cobro vía PayPal, cheque, SagePay, Google Wallet o transferencia bancaria, entre muchas otras manifestándose así su característica de multi-moneda (**Drupal** 2019).

El módulo E-Commerce para Drupal 5 y sus subsiguientes versiones del CMS incluye software personalizable para el carrito de compra, el cual nos permite vender productos físicos y también digitales. El módulo genera notificaciones y recibos por email e informes de ventas y

⁵Interfaz de Programación de Aplicaciones

transacciones. También nos permite añadir tasas, impuestos o descuentos; así como recibir donativos y pagos periódicos. Incluye plugins con pasarelas de pagos como PayPal, eWay, entre otros. Es fácil de instalar, toneladas de extensiones, la integración sin fisuras con Drupal, pero posee falta de características de abandono de carrito de compras (Drupal 2019).

PrestaShop

PrestaShop es un gestor de contenidos libre, de código abierto compatible con las pasarelas de pago como DirecPay, Google Checkout, Authorize.net, Moneybookers, PayPal. También se ofrecen módulos de pago adicionales en el mercado. Para su instalación se debe tener un nombre de dominio y un servidor web con las siguientes características: servidor web Apache 1.3 o posterior, PHP 5.1 o superior, MySQL 5 o superior y Linux, Unix o Windows instalado como sistema operativo. No incluye hosting⁶, abandono de carrito, ni cuentas de email incluidas (ECN 2019).

Permite la creación de cuentas de cliente, es instalable en tu propio servidor y da acceso a su código fuente. Es multi-tienda, permite edición por FTP y tiene web móvil incluida. La posibilidad de usar varios temas y módulos diferentes es otro factor determinante para la elección de este software a la hora de la creación de una tienda online. Otro punto a favor para el éxito de esta herramienta es que resulta muy rápida y ligera en su funcionamiento. La selección de moneda con la que se va a trabajar permite trabajar y aceptar la moneda que se use en tu país, sea cual sea. (Webescuela 2018).

Magento

Magento es un gestor de contenidos web de código abierto para comercio electrónico desarrollado en PHP. Brinda muchas funcionalidades entre las que destacan especialmente la gran flexibilidad en el diseño y un panel de control. Gracias a su concepción de sistema modular, es altamente flexible y escalable. Soporta multi-idioma, gestión de clientes e incluye funcionalidad para aceptar órdenes de compra, cheques y giros postales, pero tendrá que integrarse con una pasarela de pago para aceptar pagos con tarjeta de crédito como

⁶Un servicio en línea que te permite publicar un sitio o aplicación web en Internet.

Capítulo I

Authorize.net y PayPal, Payflow Pro. Existe la creación de cuentas de clientes y descarga del código fuente. Además, admite múltiples monedas y una web móvil incluida. No incluye hosting⁷, ni copia de seguridad automática, ni cuentas de email incluidas(**E-Commerce** 2019).

⁷Un servicio en línea que te permite publicar un sitio o aplicación web en Internet.

Como una plataforma de comercio electrónico permite funciones incluyendo opciones de visualización de productos, plantillas móviles, transacción, múltiples tiendas, funcionalidad multi-sitio, cuentas de usuario y programas de fidelización. La categorización de productos, el filtrado de comprador, el descuento y las normas de promoción también se manifiestan como funciones. Además, los clientes pueden hacer la compra sin registro o con registro y hacer un seguimiento de sus pedidos. Tiene amplias funcionalidades debido a que tiene casi todo lo que puedas desear, de manera estándar. Es escalable porque puede crecer con tu negocio. Es personalizable debido a que puedes personalizar cualquier aspecto de tu tienda, pero tienes que tener conocimientos de programación y su descarga es absolutamente gratuita, aunque su implementación no (ECN 2019).

1.3 Valoración de los módulos estudiados

A continuación, se muestra una tabla comparativa donde se permite una rápida comprensión de las principales características identificadas para cada sistema:

Algunas de las características presentes en la mayoría de los módulos estudiados fueron la integración con pasarelas de pagos, el uso de tarjetas de crédito, cuentas de emails incluidas, así como multi-tienda y multi-moneda. Por tal razón, se tomaron en cuenta estos aspectos para el análisis y comparación de los sistemas. Debido a que el código fuente de la solución seleccionada sea de código abierto se incluye esta característica en el análisis. Un sistema de código abierto permite que los programadores puedan leer, modificar y redistribuir el código fuente, permitiendo que este evolucione, se desarrolle y mejore. Además, los usuarios pueden adaptarlo a sus necesidades y corregir sus errores con un tiempo de espera menor a la aplicada en el desarrollo de software convencional o cerrado, dando como resultado la producción de un mejor software. Por último, se analiza si las soluciones encontradas utilizan licencia gratuita para la realización de dicho módulo. Dicha característica brinda a la investigación un conjunto de permisos para desarrollar el módulo otorgándole al usuario la posibilidad de distribuir, usar o modificar el producto bajo un tiempo definido, el cual depende de la zona geográfica donde se aplica la misma.

Tabla 1: Características de sistemas analizados

Después de analizados los módulos anteriormente descritos se concluye que los mismos no constituyen una solución para la problemática planteada en la presente investigación. A pesar que permiten la integración con pasarelas de pagos, con EnZona (pasarela de pago cubana

Características de interés Módulos detectados	Código abierto	Integración con pasarelas de pagos	Multi-moneda	Uso de tarjetas de crédito	Multi-tienda	Cuentas de email incluidas	Licencia gratuita
Woocommerce	X	X	X	X			X
VirtueMart	X	X	X	X			X
Drupal Commerce	X	X	X	X	X	X	X
PrestaShop	X	X	X	X	X		X
Magento	X	X	X	X	X		X

creada por la XETID) no, permiten el uso de tarjetas de crédito y son multi-monedas tienen sus diferencias como por ejemplo **Woocommerce** y **VirtueMart** no son multi-tienda lo cual no les permite administrar varias tiendas en línea independientes entre sí o compartir diferentes características, con un solo sistema de administración. **Woocommerce, VirtueMart, PrestaShop y Magento** no tienen cuentas de email incluidas, aunque si cumplen con la mayoría de los requisitos que debe tener la solución.

Se debe implementar un módulo de comercio electrónico que realice la venta de cursos y capacitaciones en la plataforma educativa garantizando así el pago en línea de los mismos. Vale destacar que las soluciones estudiadas aportaron características a tener en cuenta para el desarrollo de la solución propuesta, aunque una de las más fundamentales es que ninguno de los sistemas anteriores acepta tarjetas magnéticas cubanas lo cual es un requisito indispensable para la solución.

1.4 Metodología de desarrollo

El proceso de desarrollo de software como el ciclo de vida de un software, es un conjunto de procedimientos o actividades a seguir para la correcta construcción del programa y que este posea la calidad requerida. Dichos procedimientos definen el enfoque que se toma cuando se diseña el software y proporciona una interacción entre los usuarios, diseñadores y las herramientas en evolución. Este proceso incorpora cinco actividades estructurales: comunicación, planeación, modelado, construcción y despliegue que son aplicables a todos los proyectos de software (Pressman and Maxim 2015).

En la XETID, es usado actualmente el proceso de desarrollo de software PRODESOF en su versión 1.5 elaborado en la misma entidad. Por tanto, es el seleccionado para guiar el proceso de desarrollo de la propuesta de solución. PRODESOF cuenta con cinco fases en el desarrollo del ciclo de vida de un software: inicio, modelación, construcción, explotación experimental y despliegue. Con estas fases se logra tener más control sobre los softwares que se desarrollan, garantizando la calidad de los mismos. Como herramienta CASE⁸ de modelado PRODESOF utiliza el Visual Paradigm for UML.



Ilustración 1: Fases de la metodología de desarrollo PRODESOF (Xetid 2014).

⁸CASE: Ingeniería de Software Asistida por Ordenador, del inglés Computer Aided Software Engineering.

1.5 Herramienta de modelado

Visual Paradigm for UML es una herramienta CASE. Es de gran ayuda para desarrollar una aplicación ya que permite desde la planificación, análisis y diseño hasta la generación de códigos fuente de los programas y documentos. Soporta el lenguaje de modelado UML en su versión 2.0, permite el desarrollo multiusuario, soporta aplicaciones web y es multiplataforma(**Paradigm 2017**).

Este software es de gran ayuda para desarrollar una aplicación, maximizando su calidad y minimizando su costo de desarrollo. Posee capacidades de ingeniería directa e inversa. Soporta el lenguaje de modelado UML en su versión 2.0, permite el desarrollo multiusuario, soporta aplicaciones web y es multiplataforma(**Diagrams 2018**).

Por lo planteado anteriormente, se decide utilizar la herramienta Visual Paradigm for UML 8.0 para la construcción de los artefactos correspondientes a cada una de las etapas del ciclo de vida del componente a desarrollar. También se tuvo en cuenta que esta herramienta sirve de puente entre los analistas, arquitectos y diseñadores de sistemas de información.

1.6 Lenguaje Unificado de Modelado

El Unified Modeling Language [™] (UML[®]) ayuda a especificar, visualizar y documentar modelos de sistemas de software, incluida su estructura y diseño, de una manera que cumpla con todos estos requisitos(**UML 2019**). El lenguaje de modelado asumido en la presente investigación para modelar y diseñar la solución propuesta es UML 2.0. Se selecciona dicho lenguaje ya que mejora el desarrollo de los proyectos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios. Además, es la utilizada por Visual Paradigm for UML 8.2.(**Paradigm 2017**)

1.7 Entorno de Desarrollo Integrado

El IDE⁹ o Entorno De Desarrollo Integrado, es una aplicación informática que brinda servicios integrales que le facilitan al programador el desarrollo de software. Están diseñados para sacar el máximo provecho a la productividad del programador.

⁹*Integrated Development Environment*

NetBeans es un programa que sirve como IDE (un entorno de desarrollo integrado) que nos permite programar en diversos lenguajes. Es ideal para trabajar con el lenguaje de desarrollo JAVA (y todos sus derivados), así como también nos ofrece un excelente entorno para programar en PHP (**NetBeans** 2017).

Se decide utilizar como Entorno de Desarrollo Integrado NetBeans 8.2 ya que permite trabajar con los lenguajes de programación seleccionados, en este caso utilizamos PHP y HTML5. Una de las herramientas más importantes que NetBeans ofrece para el desarrollo de la solución propuesta es su editor de código. Dicho editor de código es multilenguaje, posee coloreado y sugerencias de código, acceso a clases desde el código, comprobaciones sintácticas y semánticas, entre muchas otras funcionalidades y beneficios que harán el desarrollo mucho más fácil.

1.8 Servidores web

Apache es un servidor HTTP de código abierto que permite servir contenido a las peticiones que vienen desde los clientes web (navegadores). Permite una instalación y configuración sencilla. Es altamente extensible y adaptable mediante módulos y soporta lenguajes como Perl, PHP y Python. En la presente investigación se propone utilizar servidor Apache 2.4.23 ya que la Plataforma Integral de Formación a Distancia DEDUKO se encuentra alojada en dicho servidor.

1.9 Sistema gestor de base de datos

El uso de un Sistema Gestor de Base de Datos (SGBD) permitirá el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que usa y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas (**PostgreSQL** 1996-2019). Se propone como SGBD a utilizar ya que la base de datos de la Plataforma Integral de Formación a Distancia DEDUKO está implementada sobre dicho gestor. Además, es compatible con PHP y es multiplataforma.

Se utiliza la herramienta PgAdmin III ya que es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. El interfaz gráfico soporta todas las características de PostgreSQL y facilita

enormemente la administración y la conexión al servidor puede hacerse mediante conexión TCP/IP(**PostgreSQL** 1996-2019).

1.10 Marco de trabajo

Opigno LMS es un sistema de gestión de aprendizaje basado en Drupal. Está escrito en PHP y cuenta con una amplia y activa comunidad de usuarios y desarrolladores que colaboran conjuntamente en su mejora y ampliación. Contiene un conjunto de librerías, una API¹⁰ y un plugin

¹⁰Interfaz de Programación de Aplicaciones

que permiten crear contenido HTML5 enriquecido, específicamente contenido didáctico. Esta distribución incluye las últimas versiones del núcleo de Opigno y un montón de otros módulos que permiten ofrecer herramientas de formación en línea eficientes e innovadoras como el módulo de comercio, de calendario, de correo, entre otros lo cual la hace muy interactiva para el usuario. Proporciona una forma de crear una plataforma funcional de e-learning de forma inmediata, con usuarios, cursos, certificados, entre otros(OPIGNO 2020).

Se utiliza Opigno LMS para la creación del módulo a la Plataforma de Formación a Distancia(DEDUKO) debido dicho sistema de gestión de aprendizaje es muy completo e incluye muchas funcionalidades por defecto con lo que no se necesita instalar extensiones o módulos para conseguir crear algunas de las funcionalidades. Además, ofrece la posibilidad de adaptarse a muchos formatos de una manera sencilla y gran capacidad de administración de usuarios.

1.11 Lenguajes de programación

PHP (*Hypertext Preprocessor*) es un lenguaje de programación interpretado que se utiliza para la generación de páginas web de forma dinámica. Este código se ejecuta del lado del servidor y se incrusta dentro del código HTML. Cabe destacar que es un lenguaje de código abierto, gratuito y multiplataforma. Permite la conexión a numerosas bases de datos de forma nativa tales como PostgreSQL, MySQL, Oracle, Microsoft SQL Server, entre otras, lo cual permite la creación de aplicaciones web muy robustas. Es orientado para desarrollar aplicaciones web donde la información esté en una base de datos(PHP 2020). Según lo expuesto anteriormente, se decide utilizar el lenguaje PHP 7.0 debido a las facilidades ya presentadas, la experiencia alcanzada sobre el desarrollo de aplicaciones web y la posibilidad de utilizarlo libremente.

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas web. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto. Es multiplataforma, o sea, puede ser visualizado por cualquier navegador de cualquier sistema operativo. Puede ser creado y editado con cualquier editor básico de textos. Utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y demás elementos en la pantalla del ordenador(HTML 2019).En el

presente trabajo se hace uso de **HTML5** porque de todos los estándares utilizados para escribir las páginas web es el de código más sencillo y simplificado permitiendo que las páginas se carguen más rápido.

JavaScript (JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js, Apache CouchDB y Adobe Acrobat. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. Leer más sobre JavaScript(**Foundation** 2020). El uso del lenguaje JavaScript es básico con el marco de trabajo propuesto ya que es utilizado tanto del lado del cliente como del lado del servidor.

CSS3¹¹ es un lenguaje utilizado en la presentación de documentos HTML, siendo coloquialmente una página web. La idea original detrás de CSS es la de separar la información de su representación visual, de modo que cambiar el diseño no implique modificar el contenido(**CSS** 2020). Entonces podemos decir que el lenguaje CSS se utilizó para organizar la presentación y aspecto del módulo.

1.12 API WSO2

Para la integración con la plataforma de pago se utiliza la plataforma ágil de integración WSO2 debido que permite desarrollar, reutilizar, ejecutar y administrar las integraciones. WSO2 se trata de una oferta de código totalmente abierto cuyos componentes pueden usarse por separado o en conjunto. Es una compañía que desarrolla aplicaciones de software abierto enfocadas en proveer una arquitectura orientada a servicios (SOA). Sus capacidades de integración híbrida simplifican aún más los proyectos que abarcan entornos tradicionales y de microservicio. A diferencia de otros productos de administración de ciclo de vida y API¹², es completamente de código abierto lo que permite la extensibilidad y la personalización(**X4COMX** 2019).

¹¹Hojas de estilo en cascada

¹²Interfaz de Programación de Aplicaciones

1.13 Conclusiones parciales

El estudio de los principales conceptos relacionados con el comercio electrónico y las pasarelas de pagos, permitió sentar las bases para el desarrollo de la propuesta de solución, además del porqué y la necesidad de la realización del presente trabajo. Se realizó un estudio de los módulos de comercio electrónico existentes para sistemas homólogos. Se comprobó que las soluciones existentes no constituyen una solución para la problemática de la investigación. Tras delimitar las características de la presente investigación se decidió desarrollar un módulo de comercio electrónico para la Plataforma de Formación a Distancia DEDUKO conllevando al uso de PRODESOFTE como metodología de desarrollo, los lenguajes de programación PHP, porque Drupal está escrito en PHP, además de cualquier modificación que se quiera hacer deberá hacerse con códigos en PHP, Java Script para realizar las validaciones de la aplicación, como herramientas una distribución de Drupal 8, CMS en el cual se realizó el módulo, PostgreSQL por ser un sistema gestor de bases de datos objeto-relacional que Drupal utiliza, NetBeans IDE por ser compatible con los lenguajes de programación utilizados, Visual Paradigm como herramienta CASE para modelar el sistema y PgAdmin III debido a que es una aplicación gráfica que trabaja con el gestor de base de datos utilizado.

2 Capítulo 2: Análisis y diseño del módulo de comercio electrónico para la Plataforma Integral de Formación a Distancia DEDUKO

En el presente capítulo se exponen las principales características del sistema a implementar, mediante la identificación de los requisitos funcionales y no funcionales con los que debe de cumplir la aplicación que dará solución a la problemática de la presente investigación. Se realiza el modelo de dominio, donde se definen las clases a utilizar en la solución y la relación que existe entre ellas. Se diseña la arquitectura del sistema, definiendo el patrón arquitectónico a utilizar y explicando la estructura de la Plataforma Educativa DEDUKO, para un mejor entendimiento de dicho sistema de gestión de aprendizaje. Además, se definen y detallan las funcionalidades y los prototipos de interfaz de usuario funcional.

2.1 Modelo de dominio

El modelo de dominio es el área de experiencia o aplicación que necesita conocerse para resolver un problema. En el ámbito de los sistemas de información son el conjunto de conceptos interrelacionados que son necesario conocer para entender el negocio del cliente y, por lo tanto, para poder entender sus necesidades y proponer una solución adecuada. El objetivo principal es obtener un conocimiento suficiente del dominio del problema como para poder comunicarse eficazmente con clientes y usuarios y comprender su negocio. Al no tener bien definidos los procesos necesarios del negocio ni un manual de procedimientos se realiza el modelo de dominio.

Descripción de las clases del modelo de dominio

A continuación, se identifican los conceptos fundamentales que se emplean en el modelo de desarrollo del módulo de comercio electrónico para la Plataforma de Formación a Distancia DEDUKO a través de un glosario de términos:

- ✓ **Plataforma educativa:** Plataforma que contiene los cursos.
- ✓ **Usuario:** Usuarios que interactúan con la plataforma.
- ✓ **Estudiante:** Usuario que sigue los cursos en la plataforma.

- ✓ **Profesor:** Usuario encargado de impartir los cursos y gestionar las actividades dentro de los cursos.
- ✓ **Curso:** Cursos que se ofertan en la plataforma.
- ✓ **Capacitación:** Capacitaciones que se ofertan en la plataforma.
- ✓ **Entidad:** Entidad que el cliente tiene que visitar para realizar la compra del curso o capacitación.
- ✓ **Pago:** Es la acción que realiza manualmente el usuario para pagar un curso o capacitación de la plataforma.

Descripción general del modelo de dominio

Para modelar el desarrollo del módulo de comercio electrónico a La Plataforma de Formación a Distancia DEDUKO se utilizó el LMS Opigno para la gestión los cursos que se ofertan en la misma. La plataforma cuenta con cursos y capacitaciones. Los usuarios que acceden a los cursos se clasifican en estudiantes y profesores. Los estudiantes son los que reciben el contenido que se imparte en los cursos. Los profesores son los encargados de impartir los contenidos de los cursos. Cada estudiante para poder adquirir cada curso debe visitar la entidad a la cual está asociada la plataforma educativa. En este modelo de dominio se relacionan los principales conceptos identificados en el dominio del problema, que se utilizan durante el desarrollo del módulo de comercio electrónico.

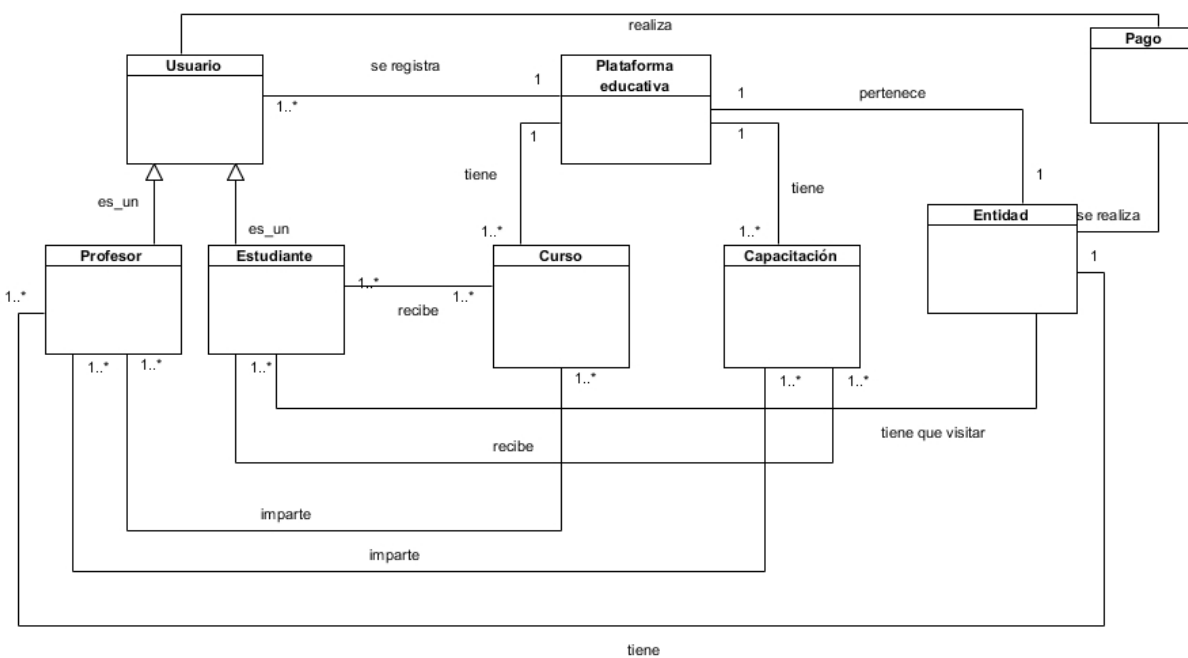


Ilustración 2: Modelo de Dominio

2.2 Especificación de requisitos del módulo de videoconferencia para la Plataforma de Formación a Distancia DEDUKO

Según la IEEE-STD-830 define un requisito como una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Se utilizan como datos de entrada en la etapa de diseño del producto y establecen qué debe hacer el sistema, pero no cómo hacerlo. Estos se clasifican en funcionales y no funcionales.

Requisitos funcionales del módulo de comercio electrónico

Los requisitos funcionales (RF) son la definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares. Deben ser lo más completos, claros y precisos posible (Pressman and Maxim 2015). A continuación, se especifican los requisitos funcionales que dan respuesta a la propuesta solución.

RF.1 Asociar usuario a un curso o capacitación.

El módulo debe permitir al estudiante asociarse en un curso o capacitación.

RF.2 Eliminar usuario de un curso o capacitación.

El módulo debe permitir al profesor eliminar al estudiante del curso o capacitación deseada.

RF.3 Configurar modalidad de pago.

Seleccionar tipo de pago

- Al registrarse en el curso o capacitación.
- Al culminar el curso o capacitación.

Ejecutar tipo de pago

RF.4 Adicionar precio de un curso o capacitación.

El módulo debe permitir al administrador adicionar precio a los cursos o capacitaciones que se ofertan en la plataforma.

RF.5 Modificar precio a un curso o capacitación

El módulo debe permitir al administrador modificar el precio a los cursos o capacitaciones que se ofertan en la plataforma.

RF.6 Eliminar precio a un curso o capacitación.

El módulo debe permitir al administrador eliminar el precio a un curso o capacitación en caso de que esta se desee ofertar gratis a los usuarios.

RF.7 Listar precio de un curso o capacitación.

El módulo debe permitir listar los precios de los cursos y capacitaciones.

RF.8 Adicionar curso o capacitación.

El módulo debe permitir adicionar cursos o capacitaciones.

RF.9 Modificar curso o capacitación.

El módulo debe permitir modificar cursos o capacitaciones.

RF.10 Listar curso o capacitación.

El módulo debe permitir listar los cursos o capacitaciones.

RF.11 Eliminar curso o capacitación.

El módulo debe permitir eliminar cursos o capacitaciones que no estén en venta.

RF.12 Visualizar curso o capacitación donde el usuario se ha registrado.

El módulo debe permitir al usuario visualizar los cursos o capacitaciones en que se ha registrado.

RF.13 Generar certificado digital.

El módulo debe permitir generar el certificado digital.

RF.14 Firmar certificado digital.

RF.15 Enviar notificaciones de errores.

El módulo debe permitir enviar notificaciones de errores.

RF.16 Enviar notificación al correo de activación de la cuenta

El módulo debe permitir enviar notificaciones al correo de activación de la cuenta.

RF.17 Configurar conexión con la pasarela de pago EnZona.

El módulo debe permitir configurar la conexión integrada con la pasarela de pago EnZona.

Para ver la descripción de los requisitos funcionales consultar [Anexo 1](#).

Requisitos no funcionales del módulo de comercio electrónico

Los requisitos no funcionales (RNF) son restricciones que afectan a los servicios o funciones del sistema, tales como restricciones de tiempo de respuesta, rendimiento y la seguridad. Además,

este tipo de requisitos garantizan que el producto sea fácil de usar, seguro y atractivo (Pressman and Maxim 2015). Los requisitos no funcionales del componente son:

RNF.1 Rendimiento.

- ✓ Los tiempos de respuesta y velocidad de procesamiento de la información no serán mayores de quince segundos para hasta cien usuarios concurrentes.

RNF.2 Seguridad.

- ✓ El sistema debe permitir definir los roles de cada usuario con el objetivo de separar las funcionalidades que podrá realizar cada uno.
- ✓ Garantizar la edición de la información solo al personal que tiene permiso a realizar esta actividad.

RNF.3 Software.

- ✓ Para las PC clientes:
- ✓ Se debe contar con navegador: Mozilla Firefox 69 o superior
- ✓ Para los servidores:
 - Servidor Web Apache 2.4.23 o superior.
 - PostgreSQL 9.4 o superior como Sistema Gestor de Base de Datos.
 - PHP 7.0.10 o superior.

RNF.4 Hardware

- ✓ Para las PC Clientes:
 - Se requiere tarjeta de red ethernet.
 - Al menos 2GB de memoria RAM.
 - Se requiere al menos 80GB de disco duro.
 - Procesador 1.80 GHz como mínimo.
- ✓ Para los servidores:
 - Se requiere tarjeta de red ethernet.
 - Se requiere al menos 80GB de disco duro.
 - Al menos 4GB de memoria RAM.
 - Procesador 3.0 GHz como mínimo.

RNF.5 Fiabilidad.

- ✓ El sistema debe registrar cualquier fallo en su funcionamiento para que los administradores puedan resolverlo y de la misma forma notificarle al usuario sin que se pueda comprometer la seguridad e integridad del sistema.
- ✓ Los usuarios deben de autenticarse para poder acceder a las listas de cursos o capacitaciones.

2.3 Diseño de la arquitectura propuesta

Arquitectura de software

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos(Pressman and Maxim 2015).

Patrón arquitectónico

El patrón arquitectónico utilizado por la Plataforma Integrada de Formación a Distancia DEDUKO es el Modelo-Vista-Controlador (MVC), el cual se evidencia en el desarrollo del componente de integración.

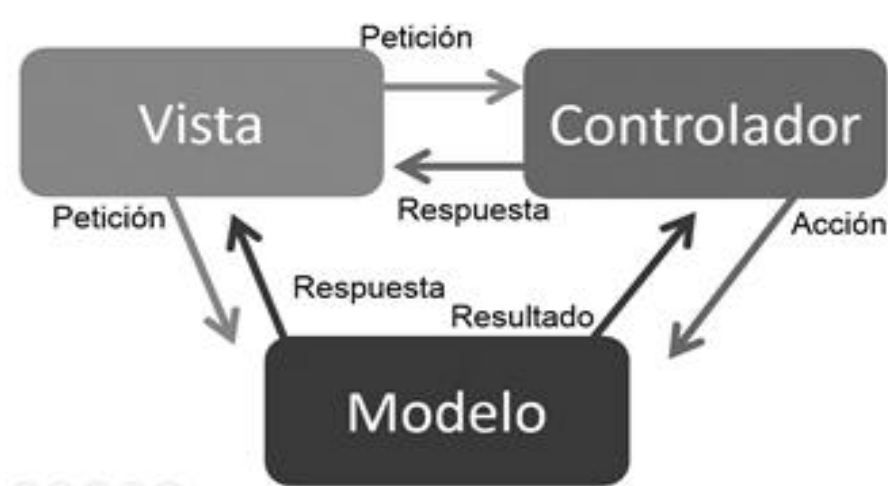


Ilustración 3: Patrón arquitectónico Modelo-Vista-Controlador.

Los tres componentes que conforman el patrón son el modelo, la vista y el controlador. El modelo administra el comportamiento y los datos del dominio de la aplicación, es un

componente totalmente independiente de las representaciones específicas de salidas o del comportamiento

de la entrada. La vista maneja la visualización de la información, implementa los procedimientos de actualización, además presenta la información del modelo al usuario. El controlador recibe la entrada, normalmente como eventos que codifican los movimientos del mouse o entrada del teclado, los eventos son traducidos para servir a las demandas del modelo o las vistas. El usuario interactúa con el sistema solamente a través de los controladores.

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, lo que provoca como ventajas, sencillez para crear distintas representaciones de los mismos datos, permite la reutilización de los componentes, simplicidad en el mantenimiento de la aplicación y facilidad para desarrollar prototipos rápidos.

2.4 Modelo de diseño

Según Pressman, el modelo de diseño proporciona detalles sobre la arquitectura del software, estructuras de datos, interfaces y componentes que se necesitan para implementar el sistema. La actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software y representaciones de interfaz y algoritmos.

2.5 Diagrama de clases del diseño

Los diagramas de clases del diseño son un tipo de diagrama estático que describe gráficamente la estructura de una aplicación. Para contribuir a la calidad del módulo de comercio electrónico se hace uso de diagramas de clases del diseño, ya que sirven de guía a los desarrolladores al constituir una aproximación del sistema que se desea implementar (Pressman 2010). En estos tipos de diagramas se utilizan las clases “Server Page o Página Servidora” para la programación lógica del servidor, las “Client Page o Página Cliente” para la programación de clientes, los “Forms o formularios” para los formularios generados por las clases “Client Page”, las “Clases Controladoras” que son las encargadas de manejar los eventos del sistema y la “Clase de Acceso a Datos” mediante la cual se obtiene la información de la base de datos.

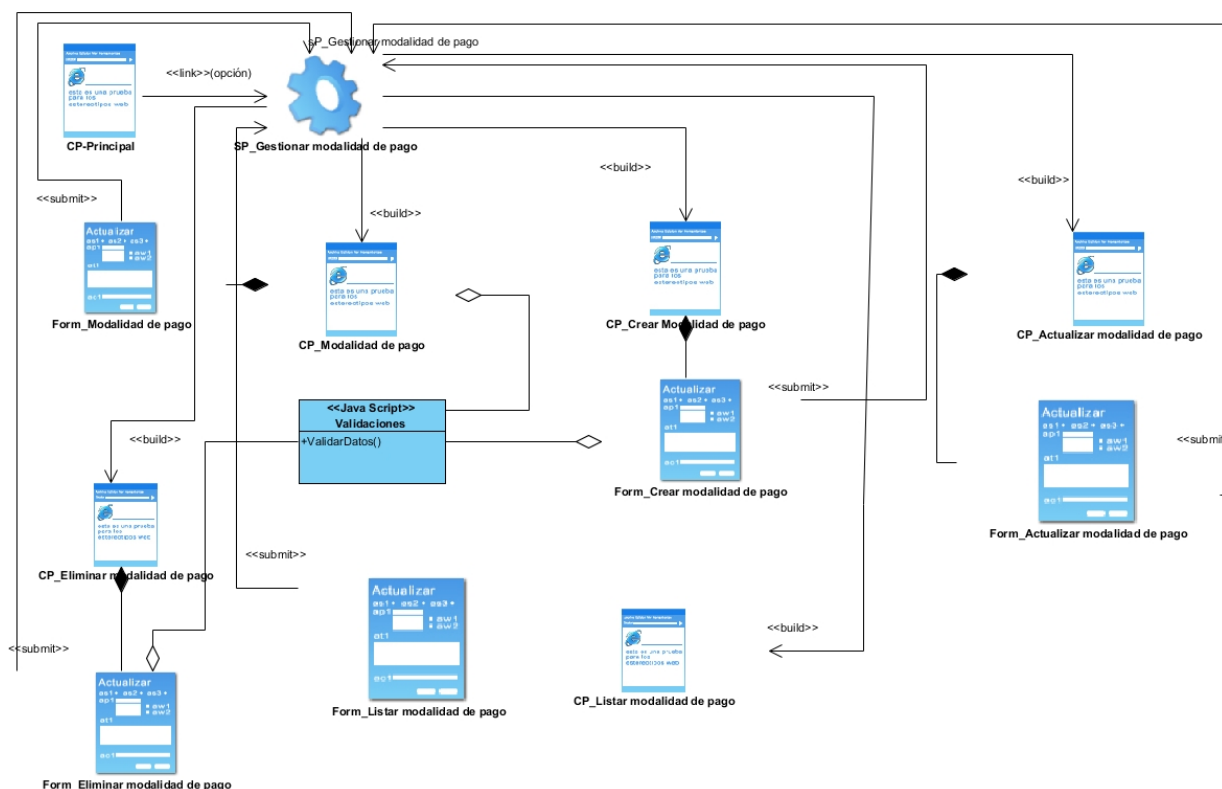


Ilustración 4: Diagrama de clases del diseño: Gestionar modalidad de pago.

Para ver el diagrama general de clases de diseño consultar [Anexo 2](#).

2.6 Diseño de la base de datos

Una base de datos bien diseñada permite guardar grandes cantidades de información de forma organizada para poder en un futuro encontrarla y utilizarla fácilmente. Independientemente de que la propuesta está basada en el LMS Opigno, el cual consta con una base de datos ya diseñada y probada por una comunidad internacional de varios miles de usuarios, se agregarán algunas entidades necesarias para el funcionamiento de la aplicación.

Diagrama entidad – relación

El modelo entidad relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos clásicos llamados entidades y relaciones entre objetos, implementándose en forma gráfica a través del diagrama entidad relación (Adjunta 2012). En el siguiente diagrama solo se contemplan las entidades de significativa relevancia para el módulo. No será necesario el análisis de todas las entidades pues el LMS se encarga de la gestión de las mismas.

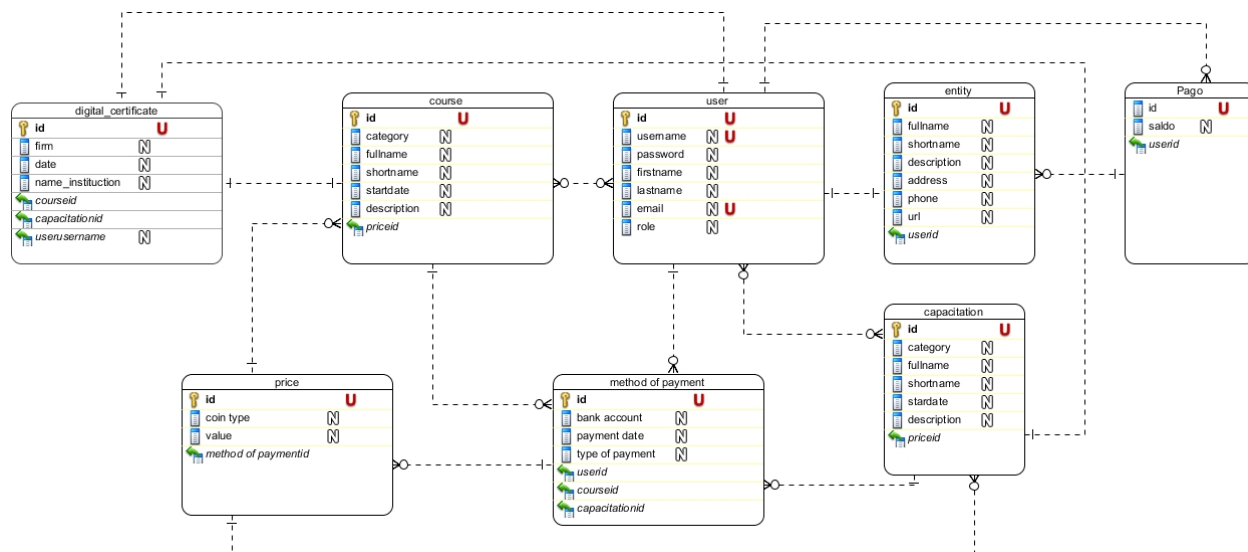


Ilustración 5: Diagrama entidad – relación.

Descripción de las tablas del diagrama entidad – relación

Tabla 2: Descripción de la tabla " method of payment "

Nombre	method of payment	
Descripción	En esta tabla se almacenan todas las instancias de los métodos de pagos vinculados a los cursos o capacitaciones.	
Atributo	Tipo	Descripción
id	integer	En este campo se almacena el identificador del método de pago de un curso o capacitación
bank account	integer	En este campo se almacena la cuenta bancaria asociada al método de pago de un curso o capacitación
payment date	date	En este campo se almacena la fecha de pago asociada al método de pago de un curso o capacitación
type of payment	varchar	En este campo se almacena el tipo de pago asociado al método de pago de un curso o capacitación

Para ver la descripción de las tablas restantes consultar [Anexo 3](#).

2.7 Patrones de diseño

Un patrón es un conjunto de información que proporciona respuesta a un conjunto de problemas similares. Para ello se aíslan los aspectos comunes y su solución y se añaden cuantos comentarios y ejemplos sean oportunos. Los patrones ayudan a capturar conocimiento

y a crear un vocabulario técnico, hacen el diseño orientado a objetos más flexible, elegante y en algunos casos reusable. Para el diseño de software, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. El patrón de diseño incorpora el conocimiento de diseño pragmático, ganando con dificultad, en una forma que permite que otros lo reutilicen “un millón de veces sin elaborarla dos veces de la misma forma”(Pressman and Maxim 2015).

Patrones GRASP

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP¹³) nos da unos principios generales para asignar responsabilidades, se utiliza sobre todo en la realización de diagramas de interacción y describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones(Larman 2003).

El patrón **Controlador**, asigna a clases específicas la responsabilidad de controlar el flujo de eventos de la aplicación. Ello facilita la centralización de actividades (validaciones, seguridad, entre otros.). El controlador no las realiza, sino las delega en otras clases. Este patrón se ve reflejado en la clase “CursoCP” donde se realiza el control de los objetos que son creados, editados o eliminados, que serán dirigidos a la base de datos, así como el uso correcto de mensajes de información dirigidos al usuario, en dicha clase se delega la responsabilidad del trabajo directo con la base de datos a la clase controladora del servidor “CursoCS”.

Otro patrón utilizado es el **Creador (Creator)** que tiene la información necesaria para realizar la creación del objeto. Es decir, este patrón ayuda a identificar quién debe ser el responsable de la creación (o instancia) de nuevos objetos o clases. Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creadora. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. El mismo se ve aplicado en cualquier caso de agregación o composición de clases.

Para la asignación de responsabilidades en la aplicación se utilizó el patrón **Experto (Expert)** el cual indica que la clase que cuenta con la información necesaria para cumplir la responsabilidad es la responsable de manejar la información. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento). Sus ventajas se centran en que mantiene el encapsulamiento, los objetos

¹³General Responsibility Assignment Software Patterns.

utilizan su propia información para llevar a cabo sus tareas, se distribuye el comportamiento entre las clases que contienen la información requerida y son más fáciles de entender y mantener.

Para fomentar la reutilización de código se tuvo en cuenta el patrón **Bajo Acoplamiento (Low Coupling)** que indica una menor dependencia entre clases. De tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases. También es utilizado el patrón **Alta Cohesión (High Cohesión)** debido a que los conceptos de cohesión y acoplamiento están íntimamente relacionados. Un mayor grado de cohesión implica un menor grado de acoplamiento. Maximizar el nivel de cohesión intramodular en todo el sistema resulta en una minimización del acoplamiento intermodular. Este patrón plantea que la información que almacena una clase debe de ser coherente y debe estar (en la medida de lo posible) relacionada con la clase. Estos dos últimos patrones no deben tratarse de forma independiente pues poseen relaciones que mantienen el equilibrio entre clases, equilibra las responsabilidades y garantiza que las clases sean creadas con un buen diseño donde los objetos sean capaces de interactuar.

Patrones GoF

La línea base en el tema de patrones de diseño la impone el catálogo "Design Patterns: Elements of Reusable Object-Oriented Software", en este libro se presenta un conjunto de 23 patrones de diseño identificados a partir del estudio y la experiencia del grupo llamado Banda de los Cuatro o The Gang of Four (GOF): Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, quienes se dedicaron a analizar los problemas recurrentes en el desarrollo de software y realizaron una clasificación y agrupación a partir de dos criterios, su propósito y alcance, las categorías definidas son (Carlos A Guerrero 2013):

- Patrones Creacionales: Inicialización y configuración de objetos.
- Patrones Estructurales: Separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes.
- Patrones de Comportamiento: Más que describir objetos o clases, describen la comunicación entre ellos.

A partir de las categorías mencionadas se describen los patrones GOF empleados en el desarrollo del software.

Patrones de Comportamiento.

El patrón **Observador (Observer)** define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. En la aplicación se utiliza el Observador en la clase controladora “Curso”, al utilizar el método “ListarCurso(curso)” que se ejecuta cuando un usuario con el rol profesor entra a los cursos y muestra a los demás usuarios que están registrados en el mismo.

También se utilizó el patrón de comportamiento **Orden (Command)** que permite encapsular una petición de un objeto y luego realizar operaciones con el mismo. Dicho patrón se pone de manifiesto en la interfaz “Comprar.html”.

```
public function validateForm(array &$form, FormStateInterface $form_state){
    if(empty($form_state->getValue('message'))){
        $form_state->setErrorByName('message',$this->t('Introduce un nombre.'));
    }
    if(empty($form_state->getValue('message'))){
        $form_state->setErrorByName('message',$this->t('Introduce un apellido.'));
    }
    if(empty($form_state->getValue('message'))){
        $form_state->setErrorByName('message',$this->t('Introduce un correo.'));
    }
    if(empty($form_state->getValue('message'))){
        $form_state->setErrorByName('message',$this->t('Introduce los números del carnet de identidad.'));
    }
    if(empty($form_state->getValue('message'))){
        $form_state->setErrorByName('message',$this->t('Introduce los números de la tarjeta bancaria.'));
    }
}

public function submitForm(array &$form, FormStateInterface $form_state){
    foreach ($form_state->getValues() as $key => $value) {
        drupal_set_message($key . ': ' . $value);
    }
}
```

Ilustración 6: Patrones

Patrones Creacionales.

Durante el desarrollo del módulo se utilizó como patrón relacional Instancia única (**Singleton**) el cual garantiza que una clase sólo tenga una única instancia y proporciona un punto de acceso global a ella. Para inicializar la conexión a la base de datos en la clase controladora “UsuarioCP” para iniciar la conexión.

2.8 Conclusiones parciales

En el presente capítulo con la realización del modelo de dominio se definieron los principales conceptos del entorno y sus relaciones, obteniendo una mejor descripción del dominio del problema. Mediante la realización de la disciplina levantamiento de requisitos, se consiguió

Capítulo II

identificar las funcionalidades que el módulo de comercio electrónico y pago en línea para la Plataforma de Formación a Distancia DEDUKO debe brindar. Se representó la arquitectura que será usada para el desarrollo de la solución propuesta. Se modelaron los diagramas de clases para permitir visualizar la relación existente entre los conceptos u objetos significativos en el dominio del problema, así como se establecieron los patrones de diseño utilizados en el módulo.

3 Capítulo 3: Implementación y prueba del módulo de videoconferencia para la Plataforma Integral de Formación a Distancia DEDUKO

En el presente capítulo se tendrán en cuenta todos los aspectos del diseño del sistema con el fin de llevar a cabo el desarrollo de los flujos de trabajo de implementación y prueba. Se muestra la organización del sistema mediante el modelo de componentes el cual representa la vista estática del sistema y la situación física de los distintos componentes lógicos desarrollados a través del modelo de despliegue. Se describen los estilos de programación y los estándares de codificación empleados. Y por último se define el proceso de pruebas en el cual se verifica que todos los requisitos hayan sido implementados y funcionen correctamente.

3.1 Diagrama de despliegue

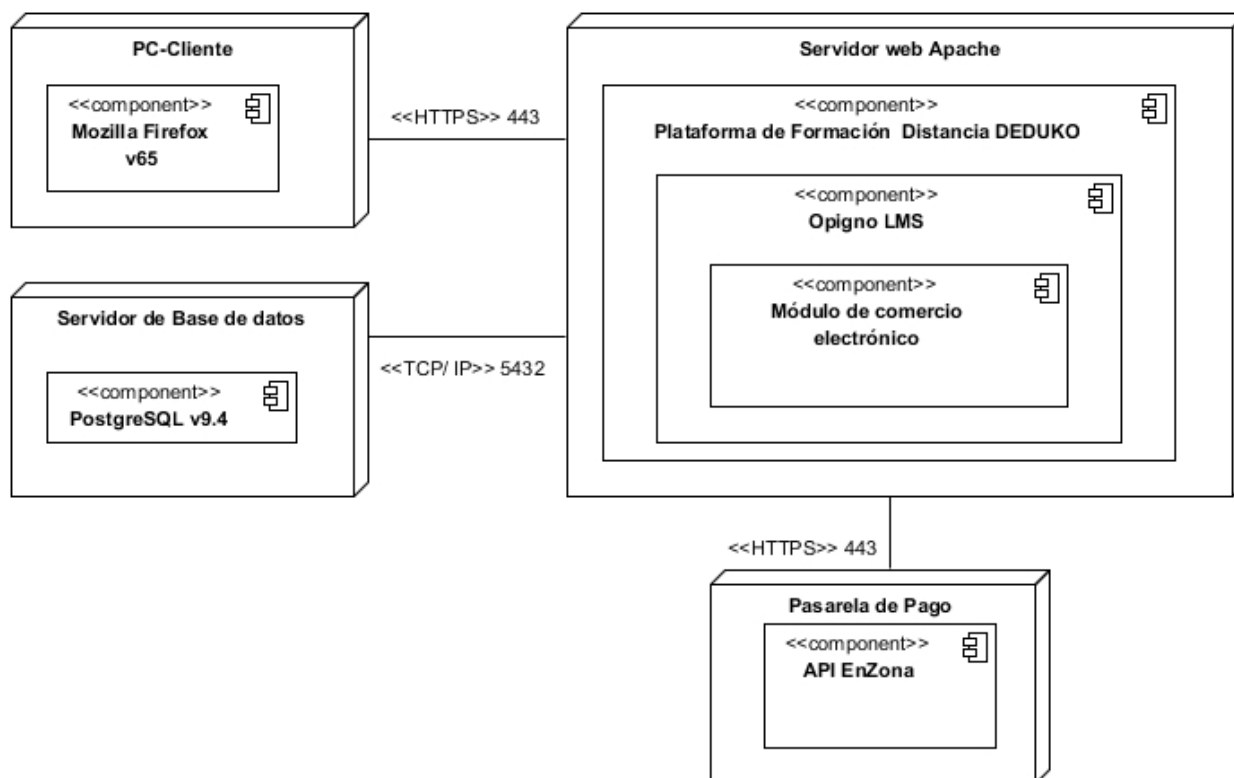


Ilustración 7: Diagrama de Despliegue

El nodo *PC-Cliente* representa las estaciones de trabajo que el usuario utiliza para acceder a la actividad de comercio electrónico de la Plataforma de Formación a Distancia DEDUKO, usando un navegador de *Mozilla Firefox*. Mediante el protocolo HTTPS y el puerto 443 se conecta con el servidor web Apache donde se encuentra alojada la plataforma.

El nodo *Servidor Web Apache* representa el servidor sobre el cual se encuentra desplegada la *Plataforma de Formación a Distancia DEDUKO* que utiliza el *LMS Opigno* para la gestión de los cursos y que a su vez contiene el módulo de comercio electrónico. Accede al servidor de base de datos *PostgreSQL* para el manejo de información mediante el protocolo TCP/IP y el puerto 5432.

EL nodo Pasarela de pago representa la API¹⁴ EnZona por la cual el sistema va a consumir los servicios para realizar las distintas operaciones definidas anteriormente. Mediante el protocolo HTTPS y el puerto 443 se conecta con el servidor web Apache donde se encuentra alojada la plataforma.

¹⁴Interfaz de Programación de Aplicaciones

3.2 Diagrama de componentes

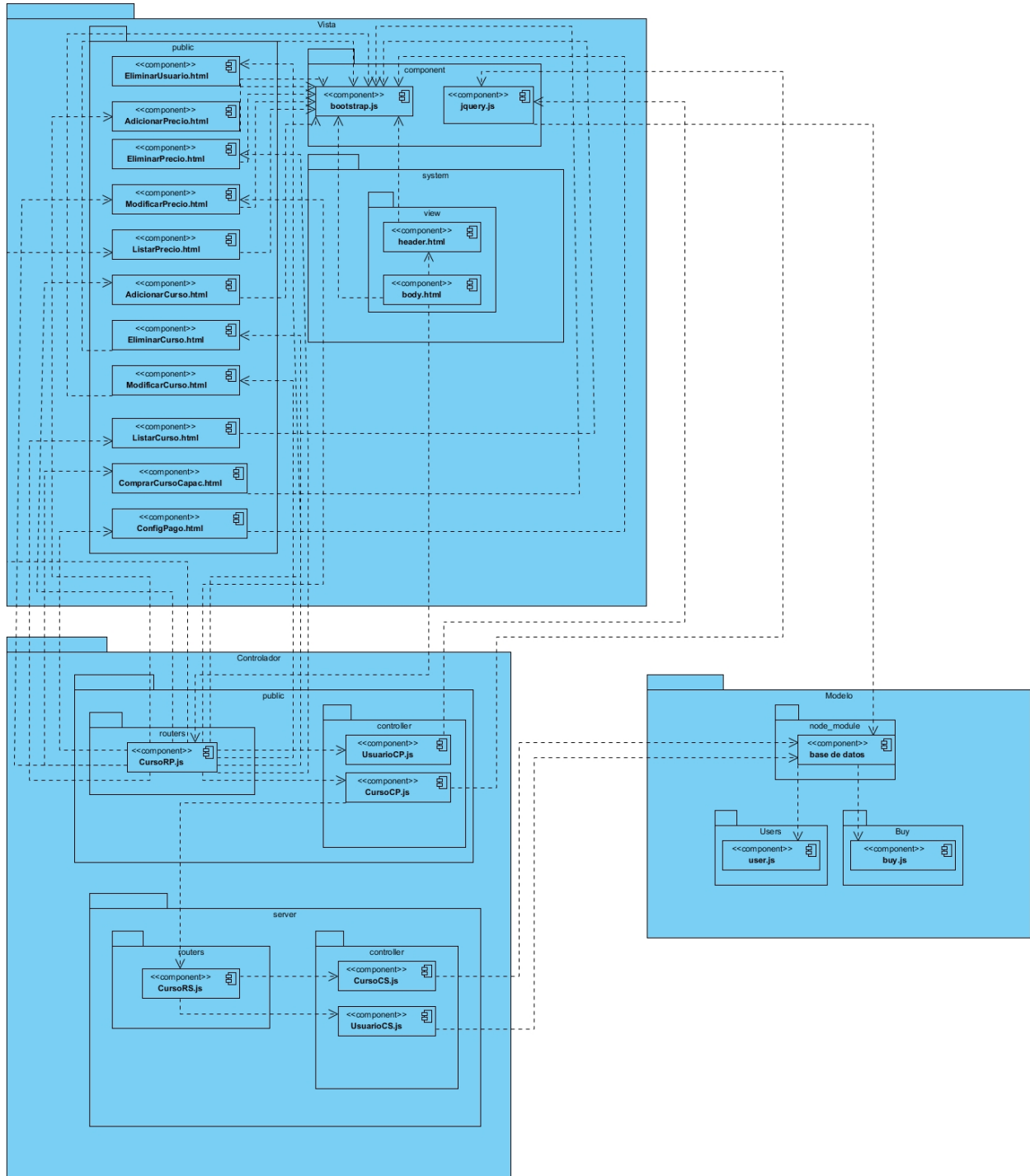


Ilustración 8: Diagrama de Componentes

3.3 Estándares de codificación

Identación

Para que el código sea más legible se puede insertar espacios en blanco. En Drupal se usa indentar con 2 espacios, y nunca con tabulaciones, también se debe tener presente que nunca se debe dejar espacios en blanco al final de cada línea, si se desea continuar en la siguiente línea simplemente se presiona la tecla <ENTER>.

Estructuras de control

Las estructuras de control deben tener el siguiente formato:

```
if (condición_1 || condición_2) {  
  acción_1;  
}  
elseif (condición_3 && condición_4) {  
  acción_2;  
}  
else {  
  acción predeterminada;  
}
```

- Espacio entre las palabras reservadas (if, while, for, else, entre otros) y el paréntesis de apertura.
- Entre el paréntesis de cierre y la llave de apertura {también tiene su correspondiente espacio.
- Siempre se debe usar las llaves de apertura y cierre.
- Las estructuras else y elseif deben ir en líneas aparte, debajo del cierre de la estructura, en este caso debajo de la llave de cierre.

Ejemplo de switch:

```
switch (condición) {  
  caso_1:  
  acción_1;  
  break;  
  
  caso_2:  
  acción_2;
```

```
break;
```

```
default:
```

```
acción predeterminada;
```

```
}
```

Ejemplo de do:

```
do {
```

```
acción;
```

```
} while ($condición);
```

Etiquetas de apertura y cierre de PHP

Al escribir código PHP, se debe utilizar las etiquetas de apertura `<?php` y la etiqueta de cierre `?>`, en los archivos. `.module` y `.inc` se puede omitir la etiqueta de cierre siempre y cuando se trabaje solo con PHP, omitir la etiqueta de cierre evita que queden líneas en blanco lo que podría dar como resultado un error típico "Cannot modify header information - headers already sent by". Cuando se trabaja con PHP puro se puede evitar la etiqueta de cierre, pero dentro de HTML se debe utilizar la etiqueta de apertura de PHP `<?php` y la etiqueta de cierre de PHP `?>`.

Uso de comillas

Se puede utilizar comillas simples ('texto') o comillas dobles ("texto") para delimitar las cadenas, las comillas dobles permiten interpretar el valor de una variable, entonces para `"<h1> $title </h1>"`, el resultado sería el valor de la variable `$title`.

Uso de punto y coma

En Drupal siempre se debe finalizar las líneas de código PHP con el punto y coma.

Operadores

Todos los operadores binarios (operadores que se utilizan entre dos valores), deben tener un espacio de separación entre ellos, esto aplica a operadores como `+`, `-`, `*`, `/`, `=`, `==`, `!=`, `<`, `>`, `..`, `.=`, `+=`, `+-`, entre otros.

Ejemplo de una asignación:

```
$numero = 3;
```

Los operadores unarios como el `++` y el `--` no deben tener separación.

Ejemplo de operador unario

```
$numero++;
```

Funciones

Los nombres de las funciones deben estar escritas en minúsculas, y las palabras separadas por un guión bajo. Siempre se debe escribir como prefijo el nombre del módulo, tema, entre otros para evitar así generar nombres duplicados. Entre el nombre de la función y el paréntesis de inicio de los argumentos no se debe dejar espacios, cada argumento debe estar separados por una coma y luego un espacio. Se debe colocar un espacio entre el cierre del paréntesis y la apertura de la llave que contiene el bloque de código.

Ejemplo:

```
function forum_help($path, $arg) {
bloque de código
....
}
```

Arreglos

Dentro del arreglo o matriz, se separa con un espacio (luego de la coma) los valores. El operador => debe separarse por un espacio a ambos lados. Cuando se superes los 80 caracteres en la declaración del arreglo se debe escribir cada uno de los elementos en una única línea. La indentación debe tener 2 espacios.

Ejemplos:

```
$some_array = array('hello', 'world', 'foo' => 'bar');
$form['title'] = array( '#type' => 'textfield',
'#title' => t('Title'),
'#size' => 60,
'#maxlength' => 128,
'#description' =>t('Título del nodo.'),
);
```

Constantes

Siempre se debe escribir en mayúsculas los nombres de las constantes, si se componen de varias palabras entonces se pueden separar con guiones bajos, al igual que en las funciones se debe colocar de prefijo el nombre del módulo o tema a donde pertenece, el objetivo es evitar crear nombres duplicados, el prefijo debe estar también en mayúsculas.

```
<?php
/**
```

```
* The current system version.
*/
define('VERSION', '8.9');

/**
 * Minimum supported version of PHP.
 */
define('DRUPAL_MINIMUM_PHP', '7.2.4');
?>
```

Variables globales

Si se necesita definir variables globales se debe comenzar con el nombre con un guión bajo seguido por el nombre del módulo o tema y otro guión bajo antes del nombre de la variable.

```
Global $_forum_página_foros;
```

Comentar el código

Se debe diferenciar los comentarios que se utilizan para aclarar determinados bloques de código, ya que estos se pueden colocar donde se crea que sea necesario, de los comentarios de documentación que son los que se escriben al principio del archivo, o antes de declarar alguna función, estos últimos son sumamente importantes ya que por medio de una API¹⁵ se genera la documentación de ayuda desde las etiquetas que se emplean. En el tipo de documentación que

se usa para aclarar algún bloque de código se utiliza las etiquetas de apertura /* y de cierre */, entre ellas podemos colocar varias líneas de comentario, o se puede utilizar las etiquetas // para un comentario de una simple línea.

```
// comentario de una simple línea
```

```
/**
 * Comentario de varias
 * líneas de texto
 */
function system_init() {
```

```
// Otro comentario de una sola línea
```

¹⁵Interfaz de Programación de Aplicaciones

```
... mi bloque de código  
}
```

3.4 Pruebas del módulo de comercio electrónico y pago en línea

La prueba es un conjunto de actividades que se elaboran con anticipación y se realizan de manera sistemática (Pressman and Maxim 2015). Las pruebas se ejecutan dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que cumple con los requerimientos. En las pruebas se utilizan casos de prueba que permiten detectar el mayor número de no conformidades y corregirlas antes de la entrega del software.

Técnicas de diseño de caso de prueba

El objetivo principal para el diseño de casos de prueba es derivar un conjunto de pruebas que tienen la mayor probabilidad de descubrir errores en el software. Para lograr este objetivo, se usan categorías diferentes de técnicas de diseño de caso de prueba: pruebas de caja blanca

y pruebas de caja negra (Pressman and Maxim 2015). La prueba de caja negra se realiza a las distintas interfaces del software. Examina algunos aspectos fundamentales de un sistema. La prueba de caja blanca se basa en el examen cercano de los detalles de procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y bucles.

3.4.1.1 Prueba de caja blanca

La prueba de caja blanca es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba (Pressman and Maxim 2015). Al utilizar el método de prueba de caja blanca, se pueden derivar casos de prueba que garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez. Dichos casos de pruebas pueden revisar todas las decisiones lógicas en sus lados verdaderos y falsos, y ejecutar todos los bucles en sus fronteras y dentro de sus fronteras operativas. Además, pueden revisar estructuras de datos internas para garantizar su validez.

En la presente investigación se utilizó la prueba de camino básico, la cual permite al diseñador de casos de prueba derivar una medida de complejidad lógica de un diseño de procedimiento y

usarla como guía para definir un conjunto básico de rutas de ejecución. Los casos de prueba derivados para revisar el conjunto básico, tienen garantía para ejecutar todo enunciado en el programa al menos una vez durante la prueba.

Para realizar la prueba de camino básico es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática. Este se realiza de tres formas diferentes:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- La complejidad ciclomática $V(G)$ de un grafo de flujo G se define como: $V(G) = A - N + 2$
- donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
- $V(G)$ también se calcula como el resultado de $P+1$ donde P es el número de nodos predicados (nodos de los cuales parten dos o más aristas) que tiene contenido el grafo de flujo G .

3.4.1.2 Prueba de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software. Permiten derivar conjuntos de condiciones de entrada que son las encargadas de revisar por completo todos los requerimientos funcionales para un programa (Pressman and Maxim 2015).

Las pruebas de caja negra intentan encontrar errores como: funciones incorrectas o faltantes, errores de interfaz, errores en las estructuras de datos o en el acceso a bases de datos externas, errores de comportamiento o rendimiento, y errores de inicialización y terminación.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas se encuentran:

Partición de equivalencia: La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógicas.

Pruebas de rendimiento: Las pruebas de rendimiento es un tipo no funcional de prueba para determinar la respuesta del sistema, es decir, velocidad, estabilidad, confiabilidad y escalabilidad. Entre los tipos de prueba de rendimiento se encuentra la prueba de carga. Dicha prueba se lleva a cabo para poner a prueba el comportamiento de la aplicación en diversos

niveles de carga dentro de sus límites aceptables. Uno de los principales datos a centrarse durante la prueba de carga es "tiempo de respuesta".

Pruebas de estrés: las pruebas de estrés se realizan para determinar el punto de equilibrio donde el rendimiento de aplicaciones deteriora. Principales parámetros para centrarse en las pruebas de tensión son "tiempo de respuesta" y "rendimiento".

Estrategias de prueba

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software(Pressman and Maxim 2015). Inicialmente, las pruebas se enfocan en cada componente de manera individual hasta alcanzar el sistema en su totalidad.

Prueba de seguridad

Las pruebas de seguridad intentan verificar que los mecanismos de protección que se construyen en un sistema en realidad lo protegerán de cualquier penetración impropia(Pressman and Maxim 2015). Su objetivo es evaluar la confidencialidad, integridad y disponibilidad de los datos por lo que permiten validar que dichos datos o el sistema, solamente sean accedidos por los actores definidos y según sus niveles de acceso.

Resultado de la prueba de seguridad

Prueba de integración

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño. Las pruebas de integración verifican el correcto ensamblaje entre los distintos componentes con el fin de comprobar que interactúan correctamente a través de sus interfaces externas e internas, y cumplen con las funcionalidades establecidas.

Existen dos tipos de integración, no incremental e incremental. No incremental es cuando se combinan todos los módulos y se prueba todo el programa en su conjunto. La integración incremental es cuando el programa se construye y se prueba en pequeños segmentos del mismo. Existen dos estrategias de integración incremental: ascendente y descendente(Pressman and Maxim 2015).

Prueba de rendimiento

Las pruebas de rendimiento están diseñadas para probar la resistencia del software en tiempo de ejecución dentro del contexto de un sistema integrado (Pressman and Maxim 2015).

3.5 Resultados de la investigación

La XETID cuenta con la Plataforma de Formación a Distancia DEDUKO. Los cursos dentro de la plataforma cuentan con una serie de recursos didácticos y herramientas de comunicación que permiten la interacción entre profesores y estudiantes tales como cuestionarios, encuestas, chats y foros, para compartir información y mejorar el trabajo colaborativo. Los estudiantes en muchas ocasiones requieren que el pago de los cursos o capacitaciones sea de forma online. Dicha manera podría enriquecer la manera de obtener el certificado digital mejorando así el proceso de obtención del mismo.

Con el resultado de la presente investigación, se implementó un módulo que permite la comercialización de los cursos y capacitaciones, así como el pago en línea mediante la plataforma de pago cubana Enzona posibilitando una mayor comodidad al usuario. Además permite al profesor poder administrar todas las operaciones realizadas por el estudiante y luego dar los resultados.

3.6 Conclusiones parciales

Mediante el diagrama de componentes se representó una vista estática del módulo, mostrando la organización y dependencia que existen entre los componentes físicos que se necesitan para ejecutar el mismo. Se estableció la distribución física del componente propuesto a partir del diagrama de despliegue, donde se definieron sobre qué elementos de hardware se deben ejecutar los elementos de software. Se desarrolló satisfactoriamente la propuesta de solución, logrando así la implementación del módulo para la Plataforma Integral de Formación a Distancia. La utilización de los estándares de codificación establecidos por Drupal, permitió la obtención del código del componente de forma organizada, siendo este entendible para cualquier programador que requiera de su utilización.

4 Conclusiones generales

Una vez culminada la investigación se puede afirmar que se cumplieron los objetivos planteados, arribando a las siguientes conclusiones:

- El estudio de los principales conceptos relacionados al comercio electrónico asociado al pago en línea, permitió sentar las bases para el desarrollo del módulo de comercio electrónico y pago en línea para la Plataforma de Formación a Distancia DEDUKO. El estado del arte, referido a los módulos de comercio electrónico y pago en línea, permitió afirmar que las soluciones que existen, que de alguna manera tributan a la investigación, no resolvían la problemática planteada, evidenciándose la necesidad de esta investigación. Las herramientas, tecnologías y lenguajes propuestos para la construcción de la aplicación, se corresponden con las políticas de soberanía tecnológica que impulsa la universidad y el país.
- Los artefactos generados durante el proceso de desarrollo del software permitieron un mejor entendimiento del módulo de comercio electrónico y pago en línea en caso de mejoras en el futuro.
- La utilización de patrones y estándares de codificación promovieron buenas prácticas en el desarrollo de la solución, siendo más entendible y proporcionando uniformidad en la implementación.

Como resultado se obtuvo el módulo de comercio electrónico y pago en línea el cual garantiza la venta de cursos y capacitaciones de la Plataforma de Formación a Distancia DEDUKO. La utilización de dicho módulo permitirá a los usuarios de la Plataforma de Formación a Distancia DEDUKO comprar en tiempo real los cursos y capacitaciones que se estén ofertando para así lograr un mejor intercambio entre estudiantes y profesores.

Recomendaciones

Se recomienda incorporar al módulo de comercio electrónico y pago en línea efectuar el pago mediante el escaneo de un código de barras o un código QR que identifica a dicho producto.

Referencias bibliográficas

- ADJUNTA, G.** *Diagrama Entidad Relación*. Edtion ed., 2012.
- BBVA.** ¿Qué es una transferencia bancaria? In., 2019.
- CARLOS A GUERRERO, J. M. S. Y. L. E. G.** Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. . In., 2013.
- CARRANZA, P. M.-V.** Nuevos modelos de negocio: Analítica web, comercio electrónico y marketing en un entorno digital. Universidad Politécnica de Madrid, enero 2019 2019.
- CCM.** ¿Qué es el pago móvil? . In., 2020.
- CSS, L.** Lenguaje CSS. In., 2020.
- DAMARIS SOLIS FONSECA, W. R. P., MARÍA LOURDES MORILLA FAURÉS** Pasarela de pagos para la seguridad de transacciones bancarias en línea Dialnet, 2013, No.2.
- DIAGRAMS, V. P. O.** In., 2018.
- DRUPAL** Drupal Commerce. Drupal.org, 2019.
- E-COMMERCE, S.** Comparativa Software E-Commerce. In., 2019.
- ECN.** Qué es Magento: Análisis y opinión de esta plataforma ecommerce. In., 2019.
- FIGUEROA, J. G. AND F. V. MARTÍNEZ** Impacto de los medios electrónicos de pago sobre la demanda de dinero. Universidad Nacional Autónoma de México (UNAM, México), ene./mar. 2016 2016.
- FOUNDATION, M.** JavaScript. In., 2020.
- HTML, L.** Lenguaje HTML. In., 2019.
- LARMAN, C.** *UML y patrones*. Edtion ed., 2003.
- MANGOPAY.** Solución de pago integral para plataformas de mercado, de financiación colaborativa y de FinTech In., 2020.
- MGSCREATIVE.** e-Commerce VirtueMart. In., 2020.
- MORA BARRETO MIGUEL ALFREDO, F. G. M. E., YUQUILEMA YUMAGLLA FRANKLIN GEOVANNY, GONZÁLEZ MÁRQUEZ JOSÉ LUIS, SUÁREZ LÓPEZ SERGIO CRISTÓBAL** El comercio electrónico y su implicación en las transacciones comerciales Observatorio de la Economía Latinoamericana, marzo 2018.
- NETBEANS.** In., 2017.
- OPIGNO.** OPIGNO LMS THE DRUPAL LMS. In., 2020.
- PARADIGM, V.** In., 2017.
- PAYPAL.** La forma sencilla y segura de pagar y que te paguen. In., 2020.
- PHP.** PHP. In., 2020.
- POSTGRESQL.** PostgreSQL. In., 1996-2019.

Referencias Bibliográficas

PRESSMAN, R. S. *Ingeniería del software Un enfoque practico*. Edtion ed., 2010.

PRESSMAN, R. S. AND B. R. MAXIM *Software engineering : a practitioner's approach, EIGHTH EDITION. Séptima Edición.* . Edtion ed., 2015.

ROLANDO ALFREDO HERNÁNDEZ LEÓN , S. C. G. *El Proceso De Investigación Científica*. edited by E.U.D.M.D.E. SUPERIOR. Edtion ed., 2012.

SEQURA Lo natural es adaptarse 2019.

SERRA, L. Diferentes modelos de negocio para comercio electrónico 22 de abril 2015 2015

UML. Unified Modeling Language. In., 2019.

VELAZCO, S. Y., Á. I. BUSTAMANTE AND J. A. PÉREZ Sinergia entre e-learning y e-commerce Universidad Distrital Francisco Jose de Caldas, 2017.

VIRTUEMART What is VirtueMart? virtuemart.net, 2008.

VIRTUEMART. Virtuemart is a Free Open Source E-commerce solution. In., 2011.

WEBESCUELA ¿Qué es PrestaShop y por qué usarlo para crear mi propio eCommerce? 2018.

WOOCOMMERCE. WooCommerce, The eCommerce Platform In., 2020.

WORDPRESS Plugins. WordPress.org, 2019.

X4COMX. Tecnologías WSO2. In., 2019.

XETID *Proceso de Desarrollo de Gestión de Proyectos de Software (PRODESOF)*. Edtion ed., 2014.

XETID. ENZONA In., 2019.

YEEPLY. Métodos de pago online en m-commerce y en aplicaciones móviles. In., 2019.

Bibliografía

- Ayoze, Alberto Castillo. 2017.**Curso de Programación Web: JavaScript, Ajax y jQuery. 2017.
- BILLY, R. 2005.**Introducción a la Arquitectura de Software. 2005.
- Díaz, Yanette González y Fernández, Yanisleidy Romero. 2012.**Patrón Modelo-Vista-Controlador. 2012. págs. 47-57.
- drupalcommerce.2013.**<http://www.drupalcommerce.org/>
- EnZona. 2020.** Enzona. [En línea] 2020. <https://bulevarsandbox.enzona.net/>.
- EnZona. 2020.** Enzona. [En línea] 2020. <https://bulevar.enzona.net/>.
- Fernández, Luis Felipe. 2006.**Arquitectura de Software. 2006.
- Gauchat, Juan Diego.2012.**El gran libro de HTML5, CSS3 y JavaScript. 2012.
- Guerrero, Carlos A., Suárez, Johanna M. y Gutiérrez, Luz E. 2013.**Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. s.l. : Grupo de Investigación en Ingeniería del Software-GRIS, 2013.http://www.scielo.cl/scielo.php?pid=S071807642013000300012&script=sci_arttext. 0718-0764.
- Hernández, Rolando Alfredo León y Coello, Sayda González. 2012.**El proceso de investigación científica. La Habana : Editorial Universitaria, 2012. ISBN 978-959-16-1557-2.
- HTML5. 2015.** CSS3 HTML5. [En línea] 2015. [Citado el: 10 de 01 de 2015.] <http://html5.dwebapps.com/que-es-css3/>.
- JQUERY. 2018.** Jquery. [En línea] 2018. <https://jquery.com/>.
- Juca, F. J. Maldonado. 2016.**La educación a distancia, una necesidad para la formación de los profesionales. 2016.
- Larman, Craig. 2003.**UML y Patrones. s.l. : PEARSON EDUCATION, SA, 2003.
- Mozilla Foundation. 2015.** developer.mozilla.org. [En línea] 2015. [Citado el: 25 de enero de 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- NetBeans. 2017.** 2017.
- Opigno.** OPIGNO LMS The Drupal Lms, 2020.
- Peñalvo, Francisco José García y Holgado, Alicia García.2018.**Modelo de dominio, Ingeniería de software I, 2º grado en ingeniería informática:Universidad de Salamanca. 2017/2018.
- PgAdmin. 2017.** 2017.
- PHP. 2017.** 2017.
- PostgreSQL. 2017.** 2017.
- Pressman, Roger S. y Maxim, Bruce R. 2015.**Software engineering : a practitioner's approach, EIGHTH EDITION. Séptima Edición. s.l. : McGraw-Hill Education, 2015. ISBN 978-0-07-802212-8.

RAE. 2014. Diccionario de la Real Academia Española. [En línea] 2014. www.rae.es/comercio_electrónico.

Silva Murillo, Roxana.2009. Beneficios del comercio electrónico, Universidad Católica Boliviana San Pablo Cochabamba, Bolivia. 2009.

Sommerville. 2005.Ingeniería del software. Madrid : s.n., 2005.

UML. 2017. 2017.

Visual Paradigm. 2017. 2017.

W3SCHOOLS.2020. [En línea] 2020. www.w3schools.com.

XETID. 2014.PRODESOF. La Habana : s.n., 2014.

YouTube. 2020. YouTube. [En línea] 2020. www.youtube.com.

1. Anexos:

Anexo 1: Descripción de los Requisitos Funcionales del sistema.

Tabla 3: Descripción de los requisitos funcionales del módulo de videoconferencia.

Requisito Funcional	Descripción	Atributos
Asociar usuario a un curso o capacitación.	El módulo debe permitir al estudiante asociarse en un curso o capacitación.	
Eliminar usuario de un curso o capacitación.	El módulo debe permitir al profesor eliminar al estudiante del curso o capacitación deseada.	
Configurar modalidad de pago.	El módulo debe permitir configurar las modalidades de pagos existentes en el sistema.	Tipo de pago
Adicionar precio de un curso o capacitación.	El módulo debe permitir adicionar un precio a un curso o capacitación.	Precio

Eliminar precio de un curso o capacitación	El módulo debe permitir eliminar un precio a un curso o capacitación.	Precio
Modificar precio de un curso o capacitación	El módulo debe permitir modificar un precio a un curso o capacitación.	Precio
Listar precio de un curso o capacitación	El módulo debe permitir listar un precio a un curso o capacitación.	Precio
Adicionar curso o capacitación.	El módulo debe permitir adicionar un curso o capacitación.	
Eliminar curso o capacitación	El módulo debe permitir eliminar un curso o capacitación.	
Modificar curso o capacitación	El módulo debe permitir modificar un curso o capacitación.	
Listar curso o capacitación	El módulo debe permitir listar un curso o capacitación.	
Visualizar curso o capacitación donde el usuario se ha registrado	El módulo debe permitir al usuario visualizar los cursos o capacitaciones en los que se ha registrado.	
Generar certificado digital del curso o capacitación.	El módulo debe permitir generar el certificado digital de un curso o capacitación.	
Firmar certificado digital	El módulo debe permitir firmar el certificado digital.	
Enviar notificaciones de errores.	El módulo debe permitir enviar notificaciones de errores cuando ocurra algún error.	
Enviar notificación al correo de activación de la cuenta	El módulo debe permitir enviar notificaciones al correo de activación de la cuenta.	
Configurar conexión con la pasarela de pago EnZona.	El módulo debe permitir configurar la conexión integrada con la pasarela de pago EnZona.	



Anexo 2: Diagrama general de clases del diseño.

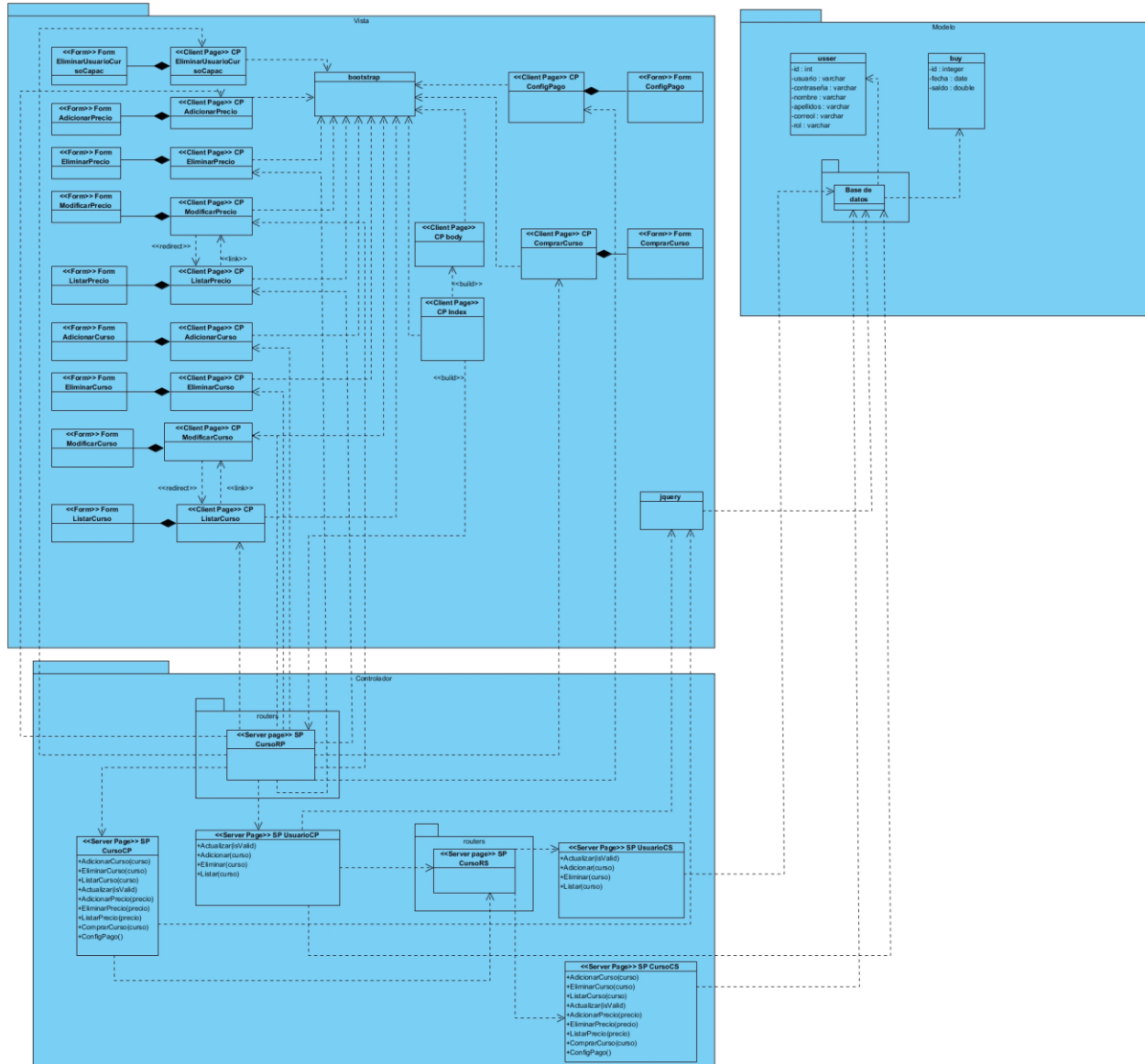
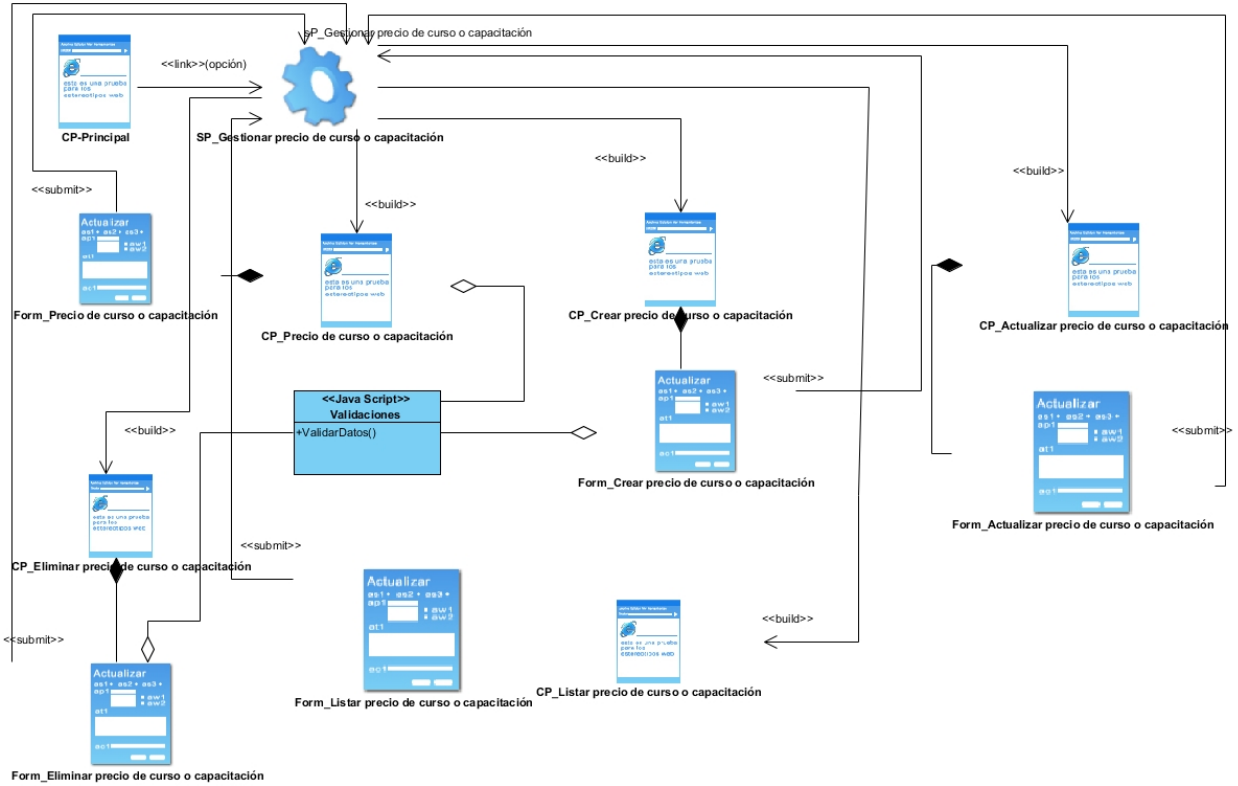
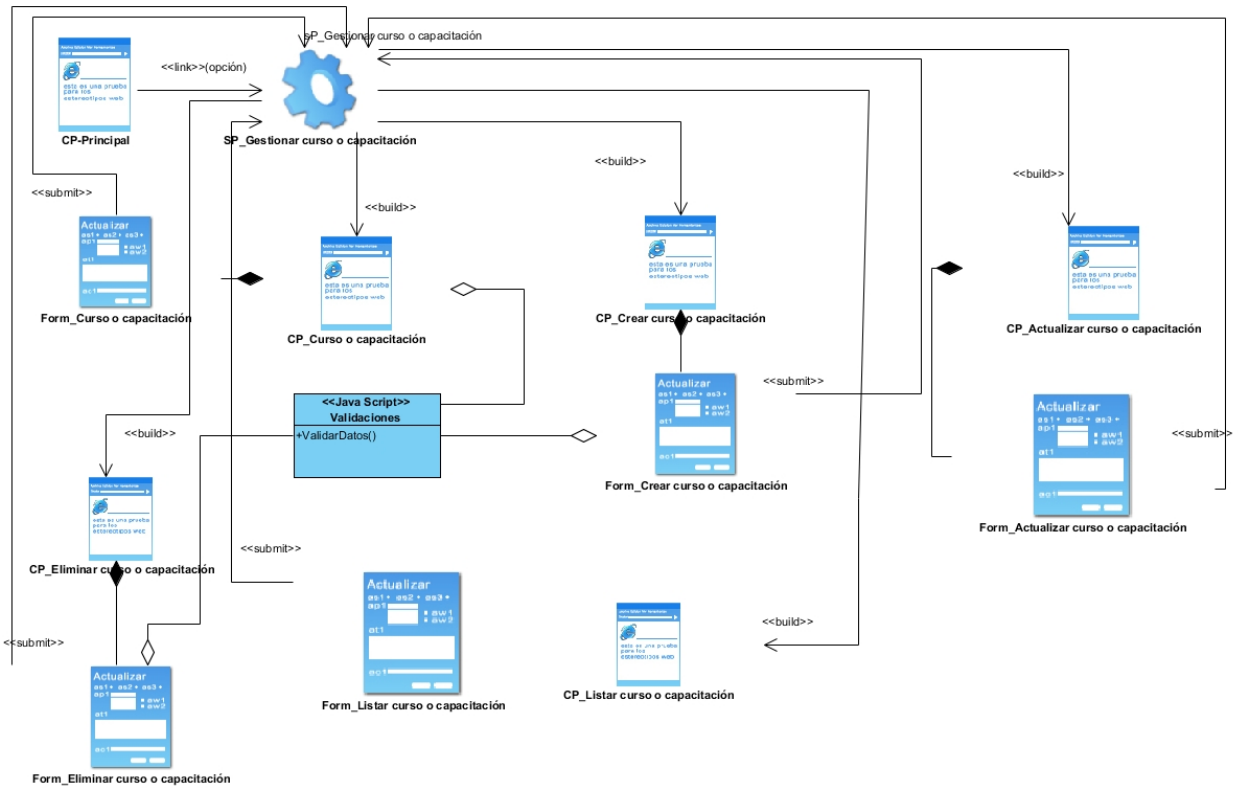


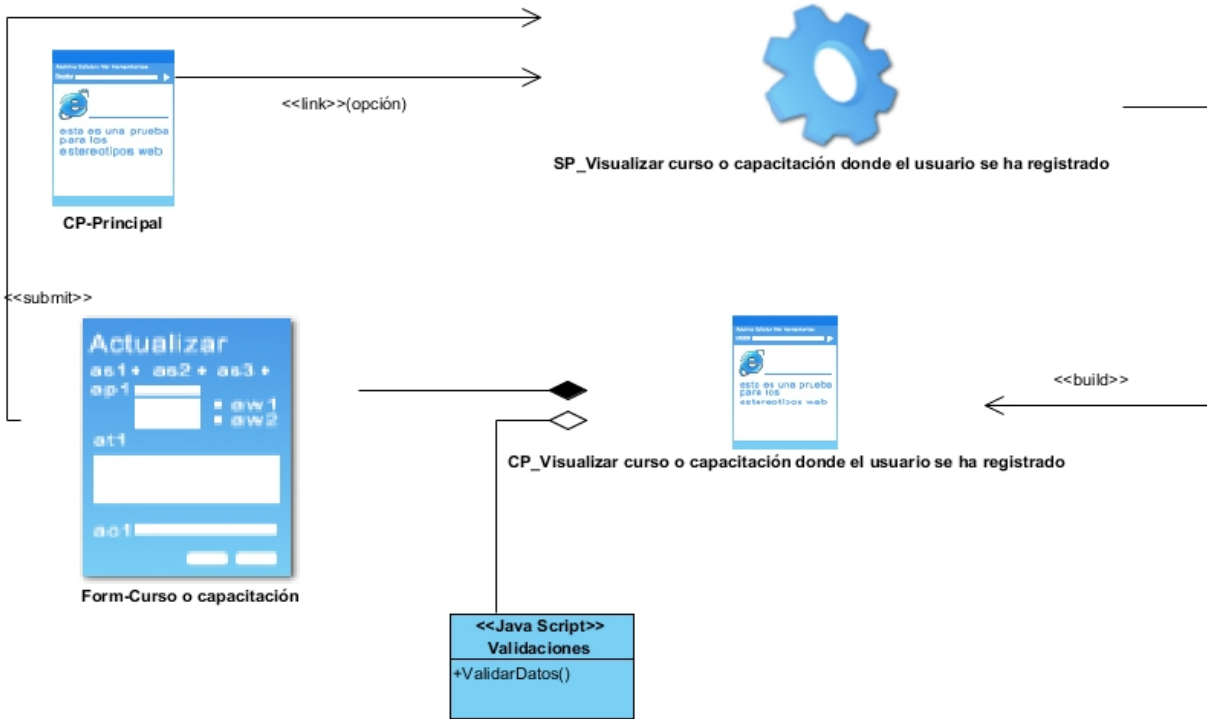
Ilustración 9: Diagrama General Clases Anexo 3: Descripción de las tablas del diagrama entidad – relación.



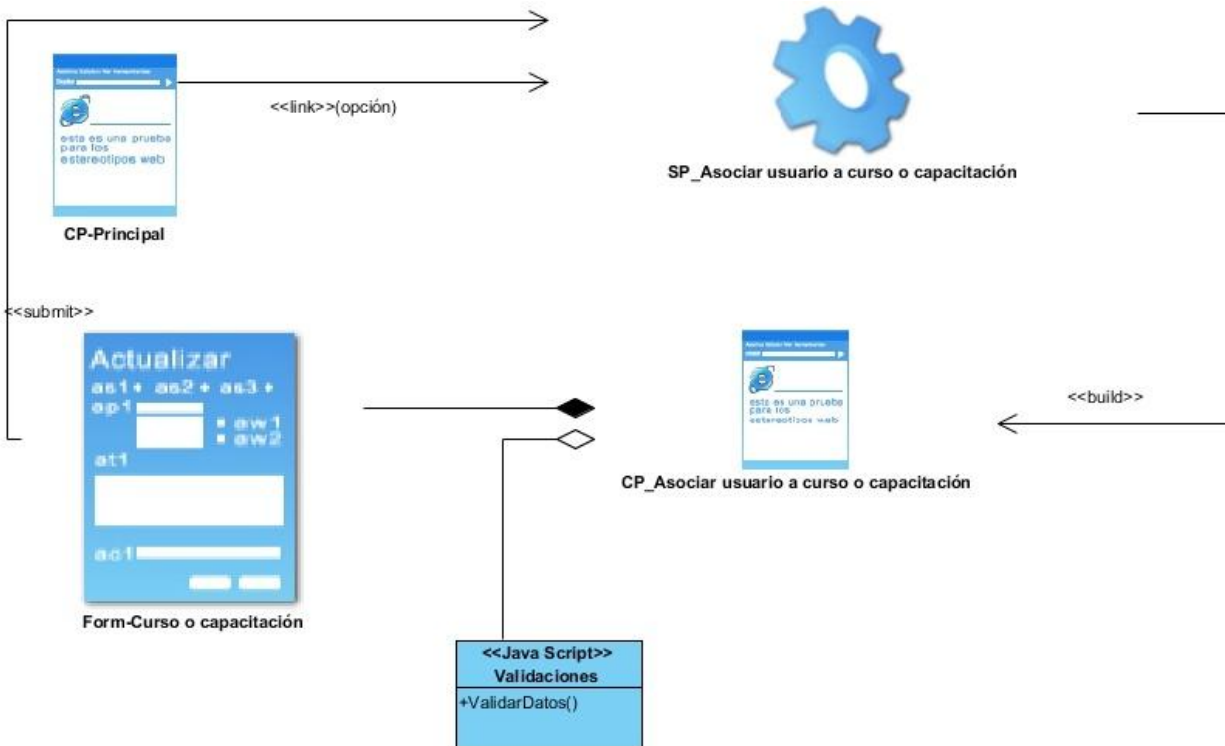
Gestionar precio de un curso o capacitación



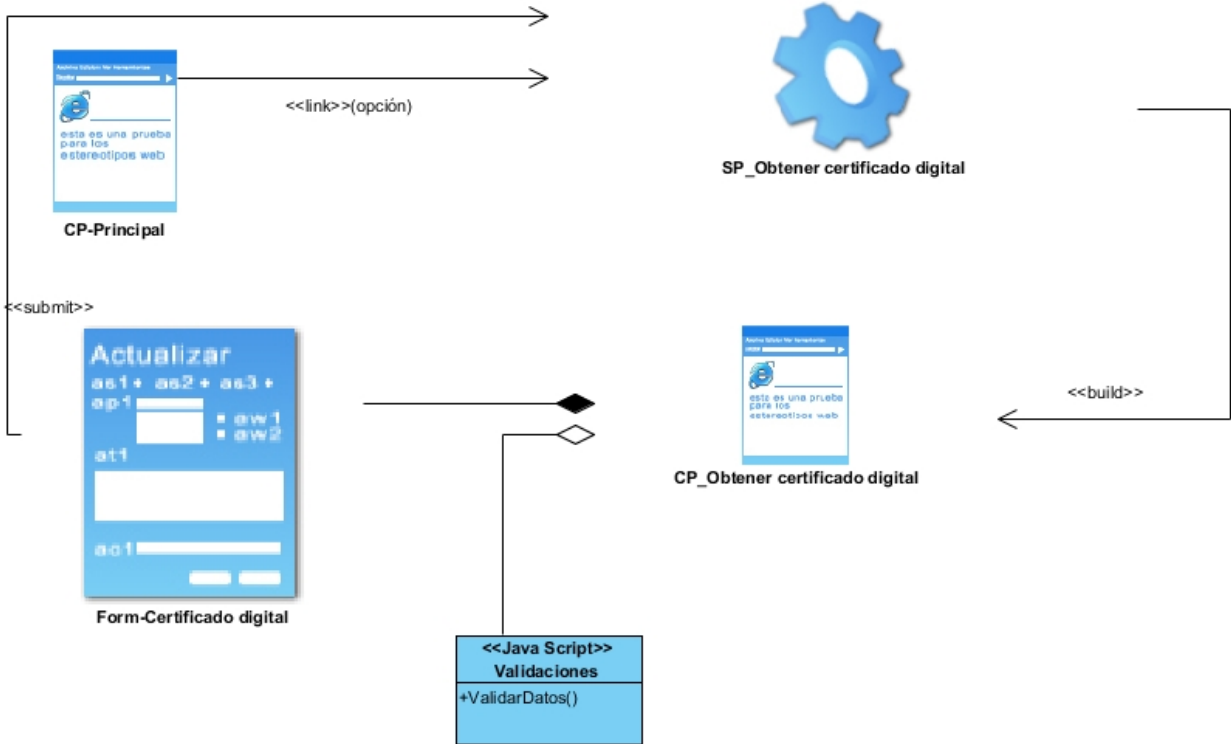
Gestionar curso o capacitación



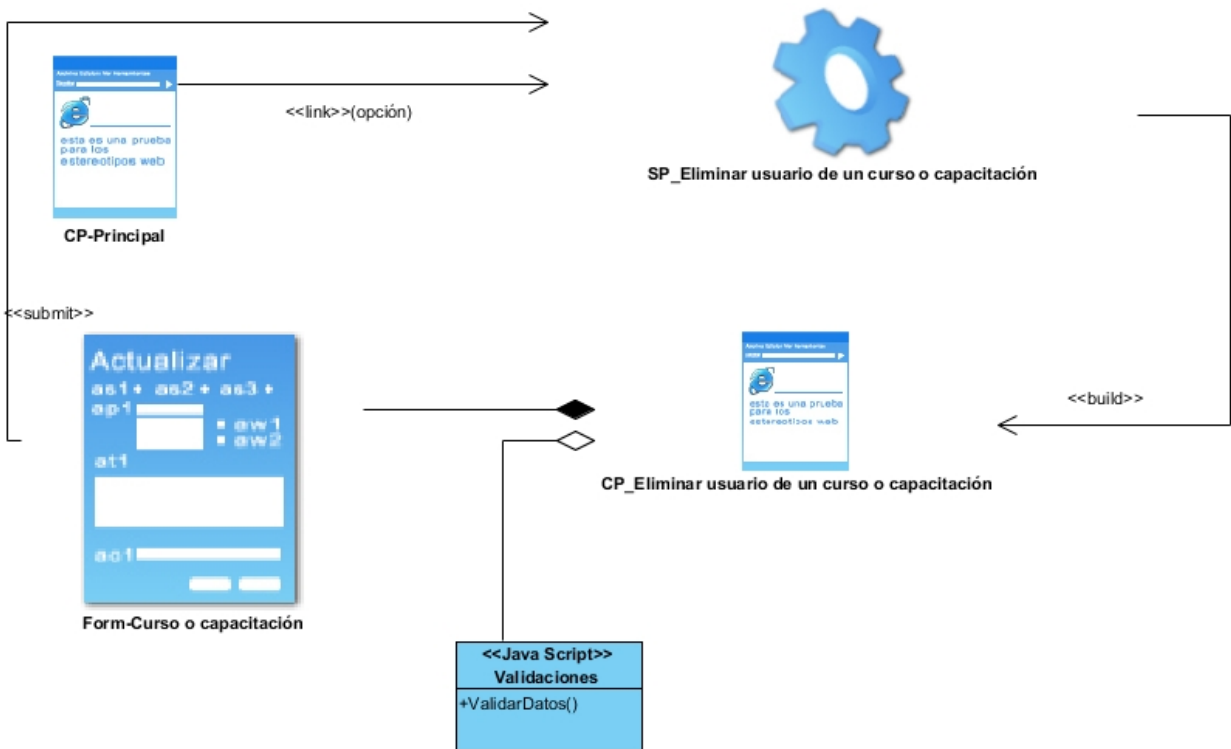
Visualizar curso o capacitación donde el usuario se ha registrado



Asociar usuario a un curso o capacitación



Obtener certificado digital



Eliminar usuario de un curso o capacitación

Tabla 4: Descripción de la tabla " user "

Nombre	user	
Descripción	En esta tabla se almacenan todos los datos del usuario que se registra en un curso o capacitación	
Atributo	Tipo	Descripción
id	integer	En este campo se almacena el identificador del usuario.
username	varchar	En este campo se almacena el nombre del usuario.
password	varchar	En este campo se almacena la contraseña del usuario.
firstname	varchar	En este campo se almacena el primer nombre del usuario.
email	varchar	En este campo se guarda el correo electrónico del usuario.
role	varchar	En este campo se guarda el rol de usuario.

Tabla 5: Descripción de la tabla " course "

Nombre	course	
Descripción	En esta tabla se almacenan toda la información de un curso.	
Atributo	Tipo	Descripción
id	integer	En este campo se almacena el identificador del curso.
category	varchar	En este campo se almacena la categoría del curso.
fullname	varchar	En este campo se almacena el nombre completo del curso.
shortname	varchar	En este campo se almacena el nombre abreviado del curso.
startdate	date	En este campo se guarda la fecha de inicio del curso.
description	varchar	En este campo se almacena la descripción del curso.

Tabla 6: Descripción de la tabla " capacitation "

Nombre	capacitation	
Descripción	En esta tabla se almacenan toda la información de una capacitación.	
Atributo	Tipo	Descripción
id	integer	En este campo se almacena el identificador de la capacitación.

Anexos

category	varchar	En este campo se almacena la categoría de la capacitación.
fullname	varchar	En este campo se almacena el nombre completo de la capacitación.
shortname	varchar	En este campo se almacena el nombre abreviado de la capacitación.
startdate	date	En este campo se guarda la fecha de inicio de la capacitación.
description	varchar	En este campo se almacena la descripción de la capacitación.

Tabla 7: Descripción de la tabla " price "

Nombre	price	
Descripción	En esta tabla se almacenan toda la información del precio de un curso o capacitación.	
Atributo	Tipo	Descripción
id	integer	En este campo se almacena el identificador del precio de un curso o capacitación.
coin type	varchar	En este campo se almacena el tipo de moneda.
value	integer	En este campo se almacena el valor del precio.

Tabla 8: Descripción de la tabla " digital certificate "

Nombre	digital certificate	
Descripción	En esta tabla se almacenan toda la información acerca del certificado digital de un curso o capacitación.	
Atributo	Tipo	Descripción
id	integer	En este campo se almacena el identificador del certificado digital.
firm	varchar	En este campo se almacena la firma del certificado digital.
date	date	En este campo se almacena la fecha de entrega del certificado digital.
name institution	varchar	En este campo se almacena el nombre de la institución.

Bibliografía

Nombre	pago	
Descripción	En esta tabla se almacenan toda la información acerca de la acción del pago	
Atributo	Tipo	Descripción
id	integer	En este campo se almacena el identificador del pago.
saldo	double	En este campo se almacena el saldo de la acción de pago.

Tabla 9: Descripción de la tabla " pago "