

Universidad de las Ciencias Informáticas

Facultad 3



Título: Desarrollo de los módulos Compra y Venta del

Sistema XEDRO-ERP.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autores: Roberto Angel Calaña Nalda
Yaniel Fabian Montenegro Martínez

Tutores: Ing. Saily Oliva Martínez
Ing. William González Obregón
Ing. Leandro Pérez-Borroto Vivero

La Habana, 2014
“Año 56 de la Revolución”

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores:

Roberto Angel Calaña Nalda

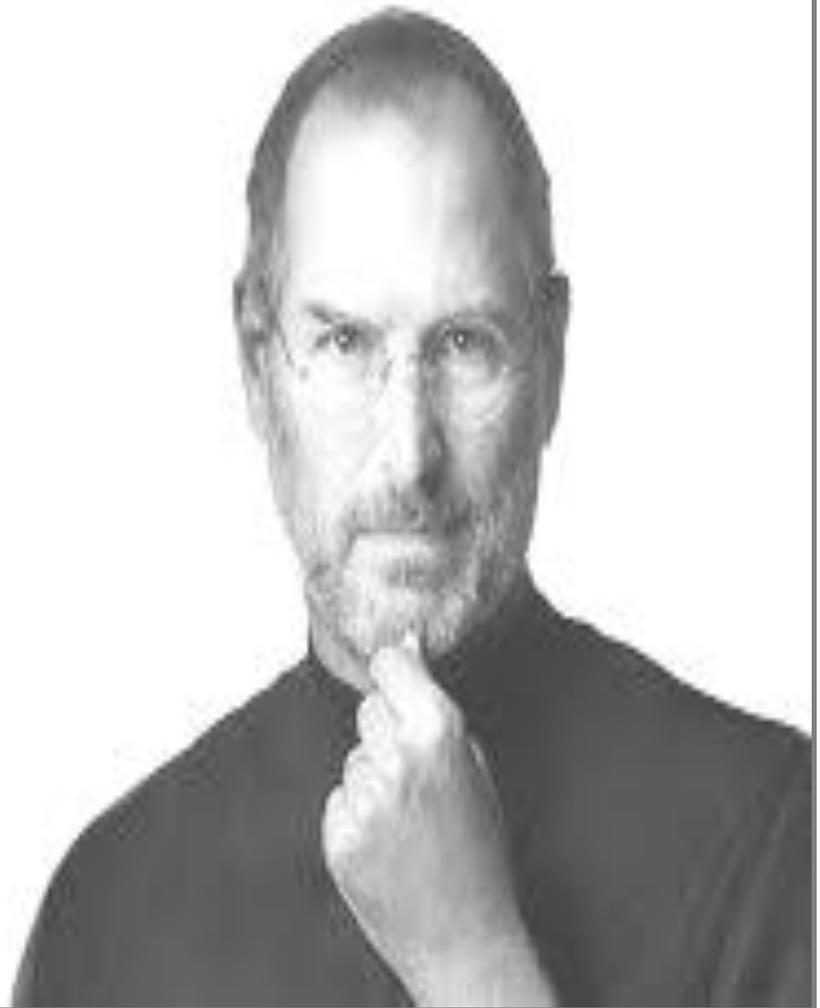
Yaniel Fabian Montenegro Martínez

Tutores:

Ing. Saily Oliva Martínez

Ing. William González Obregón

Ing. Leandro Pérez-Borroto Vivero



“Lo simple puede ser más fácil que lo complejo. Tienes que trabajar duro para mantener tu mente clara para hacer posible las cosas. Vale la pena llegar hasta el final con esto, porque cuando terminas puedes mover montañas.”

Steve Jobs

Dedicatoria

A mis padres por hacer posible que hoy este aquí y por todo su amor, y en especial a mi madre por ser mi inspiración en la vida.

A mi hermana del alma y mi abuelita querida por estar ahí cuando lo necesité y a todo el familión de chaparra.

A todos mis amigos, los viejos y nuevos que han formado parte de mi transitar por la vida.

A todos aquellos que de una forma u otra han hecho posible este momento tan importante.

Roberto

A las personas más importantes que tengo en la vida, mi abuela, mi mamá mi papá y mi hermano por haber hecho de mí un hombre de bien.

A mi tío Eduardito que es como un padre para mí y que es una de las personas que me ha apoyado siempre.

A Lázara, Lellanis, Luis, Carlos y mi ahijadito por permitir ser parte de esa familia y que me apoyaron todos estos años como mi propia familia.

A todas mis amistades las de la universidad y los de siempre, que muchos ya no están pero que no se olvidan.

Yaniel

Quisiera empezar agradeciendo a toda mi familia por haberme apoyado a lo largo de mi vida y hacer de mí un hombre de bien.

A mi madre por ser la persona más especial del mundo, la que más amo y me ama. Por toda una vida llena de sacrificios y dolores de cabeza, por esas noches de insomnio y sobre todo por esos sabios consejos que me has dado a cada minuto de mi vida. Gracias a ti he llegado tan lejos y trataré de seguir haciéndote sentir orgullosa del hijo que tienes.

A mi padre por apoyarme durante toda la vida y en especial desde que vine para la universidad. A pesar de las muchas diferencias entre nosotros y la distancia que nos ha separado, no he dejado de quererte nunca.

A mi hermana del alma, la única que tengo y la que siempre está ahí cuando lo necesito. Gracias por aguantarme todo este tiempo y espero que algún día pueda servirte de guía y ejemplo.

A Mima por ser mi confidente y participe de todas mis locuras. Gracias por encubrir cada una de mis fechorías y darme tanto amor y cariño.

A una personita muy especial que apareció de momento y sin esperarlo se robó todo mi corazón, gracias por tu paciencia y amor.

A mi compañero de tesis, que se ha convertido en un hermano para mí durante esta larga temporada.

A todos mis amigos del aula, de las tunas y todos los demás que conocí en esta etapa tan importante de mi vida. En especial a Elvis y Paul por mostrarme lo importante de una verdadera amistad, siempre podrán contar conmigo.

Roberto

Me gustaría empezar agradeciendo a la persona que más quiero en este mundo mi abuela Lidu y ojala que siempre estés orgullosa de mí ya que tú eres mi razón de ser.

A las personas más especiales que tengo en la vida por darme todo el cariño y amor en todo momento mi mamá Oneida, mi papá Rolando y mi hermanito Daniel, los quiero de todo corazón y espero ser el hijo con el que siempre soñaron.

A toda mi familia que me han brindado todo su apoyo y siempre se han preocupado por mis estudios especialmente a mi tío Eduardito que es como mi segundo padre.

A mi otra familia, la familia de la Habana como todos la conocen, que me acogieron como un miembro más de ella a Lázara, Lellanis, Luis, Carlos y mi ahijadito Fabio, los quiero mucho

A mis amigos de la universidad, los de Trinidad y los de toda la vida que no puedo dejar de mencionar Niorbis y mi primo Asiel que son como mis hermanos.

A Hanny y a toda su familia que formaron parte de mi vida durante estos 5 años y ojala que esta amistad no termine nunca.

A mi compañero de tesis por tener que soportarme todo este tiempo y haber compartido tantos buenos y malos momentos, espero que esta amistad sea para siempre.

A todos los integrantes del grupo 3505 y los que a lo largo de la carrera compartieron momentos felices conmigo y soportaron mis pesadeces.

A todos los mencionados anteriormente y si se me queda alguien, a ese también, mil gracias.

Yaniel

Resumen

Los procesos de compra y venta de productos o servicios, son los responsables de la entrada y salida de capital en las empresas, por lo que representan una de las áreas de mayor peso dentro de la gestión comercial. Siguiendo una estrategia del país, la Universidad de las Ciencias Informáticas cuenta con varias líneas de desarrollo de productos de software para el entorno empresarial, destacándose la encargada de desarrollar el sistema XEDRO-ERP. En su primera versión este sistema no incorpora un conjunto de módulos que informatizan la gestión comercial de las entidades cubanas, trayendo consigo una serie de dificultades que lo ponen en desventaja con respecto a otras soluciones informáticas de similar corte.

El presente trabajo describe el desarrollo de los módulos Compra y Venta, responsables de generar una solicitud de compra y calcular los precios de ventas de los productos destinados a la comercialización, respectivamente. Para ello se realizó un estudio de los principales sistemas informáticos que efectúan este proceso, concluyendo que era necesario construir una solución propia. Las herramientas y tecnologías utilizadas para su confección fueron las definidas por el Centro de Informatización de Entidades. Se identificaron los requisitos funcionales con los que deben cumplir los módulos. Se desarrolló un modelo de diseño que fue implementado posteriormente. Por último se realizaron un conjunto de pruebas con el fin de validar los módulos desarrollados. La nueva solución informática contribuye a la gestión comercial del sistema XEDRO-ERP, cumpliendo con las políticas de migración del país a software libre.

Palabras clave: compra, gestión comercial, venta, XEDRO-ERP.

Tabla de contenidos

INTRODUCCIÓN	7
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	13
1.1 Introducción.....	13
1.2 Principales conceptos.....	13
1.3 Valoración de sistemas que incluyen los procesos de compra y venta.....	17
1.4 Modelo de desarrollo del software	27
1.5 Tecnologías y herramientas para el desarrollo	28
1.6 Arquitectura	32
1.7 Patrones de diseño.....	33
1.8 Conclusiones del capítulo	35
CAPÍTULO 2. MODELADO DEL NEGOCIO, REQUISITOS Y DISEÑO DE LA SOLUCIÓN ...	36
2.1 Introducción.....	36
2.2 Modelado del negocio	36
2.3 Ingeniería de requisitos	39
2.4 Modelo de datos del sistema.....	47
2.5 Diagrama de componentes	49
2.6 Diagrama de clases del diseño.....	53
2.7 Patrones de diseño y arquitectura empleados en la solución.	55
2.8 Conclusiones del capítulo	57
CAPÍTULO 3. IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	58
3.1 Introducción.....	58
3.2 Implementación de los módulos	58
3.3 Diagrama de despliegue	62
3.4 Evaluación del diseño propuesto.....	64
3.5 Pruebas de software.....	68
3.6 Conclusiones del capítulo	73
CONCLUSIONES GENERALES	74
RECOMENDACIONES	75
BIBLIOGRAFÍA	76

Introducción

En medio de un mercado cambiante donde los negocios son cada vez más complejos y las necesidades de información de las entidades son de mayor envergadura, la informática es indispensable para agilizar los procesos empresariales en cualquier entidad. En un entorno competitivo como el actual, un enfoque avanzado en la gestión comercial es imprescindible.

La gestión comercial engloba varias de las disciplinas de negocio presente en las empresas cuyas actividades fundamentales se encuentran enfocadas al marketing, la compra, la venta y la producción de diferentes productos. Tiene entre sus tareas fundamentales la administración de los ingresos y gastos para generar un retorno financiero, provocados generalmente por la compra y venta de los materiales para la producción y los productos ya terminados respectivamente.

Lo anterior planteado evidencia la importancia de los procesos de compra y venta dentro de la gestión comercial. La compra es la actividad efectuada por empresas con necesidades de adquirir los recursos que les permitan realizar sus procesos productivos. Por otra parte, la venta constituye una de las actividades más perseguidas por organizaciones de corte comercial, debido a que su éxito depende directamente de la calidad, rentabilidad y cantidad de veces que se realice esta actividad. Partiendo de la premisa de que la obtención de utilidades es el objetivo principal de una empresa, se puede decir que las compras y las ventas cobran especial importancia dentro de los procesos empresariales.

La aplicación de la informática dentro del ámbito empresarial ha propiciado el surgimiento de nuevas tecnologías de gestión. Estas han convertido los sistemas empresariales de las empresas en las cuales se han implantado en ventajas competitivas, posicionándolas en lugares privilegiados con respecto a sus competidores. Estos sistemas de gestión empresarial permiten controlar, centralizar y compartir la información entre los diferentes módulos del sistema. Además ofrecen una serie de ventajas frente a los procedimientos de registro y control tradicionales, dentro de las que se encuentran:

- manejo de gran cantidad de información.
- gran rapidez de trabajo.
- reducción de errores y facilidad en su resolución.
- integración de la información.

- reducción de costos.

Ejemplo de sistemas implantados en empresas para gestionar sus procesos lo constituyen los sistemas de Planificación de Recursos Empresariales (ERP, por sus siglas en inglés). Estos permiten obtener la información de áreas como finanzas, ventas, compras y logística, centralizando e integrando los procesos presentes en estas áreas con el resto. Facilita la toma de decisiones por parte de los directivos, permitiéndoles anticipar caminos y obstáculos y con ello optimizar el rendimiento comercial.

En el mundo existen varios ERP que son usados en múltiples empresas, de los cuales algunos de sus módulos han sido utilizados en entidades cubanas, además en Cuba se han desarrollado y aplicado sistemas de gestión de buena calidad. Todos estos sistemas han solucionado de una manera u otra los problemas de las empresas en las cuales han sido implantados, pero distan de una solución integral que englobe todas las funcionalidades necesarias para abarcar los procesos de negocio de las entidades cubanas en general. En este sentido, la Universidad de las Ciencias Informáticas, cuenta con varias líneas de producción de desarrollo de productos de software para el entorno empresarial, tanto nacional como para el extranjero. El Centro de Informatización de Entidades (en lo adelante CEIGE), se dedica al desarrollo de proyectos de este corte dentro de la universidad, en el que se encuentra el proyecto ERP-Cuba, cuyo fin es el desarrollo del sistema XEDRO-ERP. Este sistema está estructurado por las siguientes categorías:

- **Subsistema:** Mayor nivel de abstracción dentro de un sistema. Agrupación de módulos y/o componentes que responde a una necesidad de negocio.
- **Módulo:** Nivel de abstracción que incluye agrupación de componentes por cuestión lógica o de negocio.
- **Componente:** Menor nivel de abstracción dentro de un sistema. Agrupación de funcionalidades que responde a una necesidad de negocio y que contribuye a favorecer el mantenimiento, la adaptabilidad y la flexibilidad.

En la actualidad XEDRO-ERP no incorpora un conjunto de módulos que informatizan áreas funcionales de gran impacto dentro del sistema empresarial, esta situación trae un conjunto de dificultades, entre las que se destacan:

- Limitaciones en cuanto a la población de entidades donde puede llegar.

- Limitaciones en cuanto a la ejecución integral de un conjunto de procesos en el sistema.
- Impacto negativo en los ingresos que puede generar el proyecto donde se desarrolla el sistema.

Dentro de los módulos ausentes en XEDRO-ERP, se encuentran los de gestión de compra y venta, no contar con estos, genera la siguiente situación:

- Imposibilidad de desplegar el sistema en empresas que presenten una actividad relacionada a estos procesos, por lo que se pierde mercado y con ello ingresos al proyecto.
- Imposibilidad de sustituir otros sistemas propietarios que hoy mantiene el país en entidades con actividades relacionadas a este proceso y por el cual debe pagar costosas licencias de software.

La situación descrita, demanda la incorporación al sistema XEDRO-ERP, de los módulos que informaticen los procesos de compra y venta, integrándolos al resto de los subsistemas, por lo que se plantea como **problema a resolver**: ¿Cómo lograr la gestión de los procesos de compra y venta a partir de los módulos del sistema XEDRO-ERP?

Para ello la investigación está enfocada en el **objeto de estudio**: Los procesos de gestión comercial, enmarcado en el **campo de acción**: Procesos de compra y venta en el sistema XEDRO-ERP. Para dar solución al problema a resolver se define como **objetivo general**: Desarrollar los módulos de compra y venta para contribuir con la gestión comercial en el sistema XEDRO-ERP.

Para darle cumplimiento al objetivo general, se concibieron los siguientes **objetivos específicos**:

1. Definir el marco teórico de la investigación para determinar los elementos necesarios a tener en cuenta en el desarrollo de la solución.
2. Realizar la modelación del negocio para lograr un mejor entendimiento de los procesos de compra y venta.
3. Realizar el análisis y diseño de los módulos para potenciar la implementación del sistema.
4. Realizar la implementación de los módulos para dar cumplimiento a los requisitos propuestos.
5. Validar la implementación de los módulos desarrollados mediante pruebas de caja negra.

6. Validar la investigación aplicando prueba de aceptación.

Para realizar la presente investigación con éxito, cumpliendo con los objetivos específicos, se proponen las siguientes **tareas de investigación**:

Sobre objetivo 1:

- Realización de una valoración sobre los distintos sistemas de gestión del entorno nacional e internacional que informaticen los procesos de compra y venta.
- Realización de un informe sobre las características generales de XEDRO-ERP.
- Realización de un informe sobre las tecnologías, herramientas y modelo de desarrollo a utilizar en la investigación.

Sobre objetivo 2:

- Realización del modelado de los procesos de compra y venta.
- Realización del modelo conceptual de los datos.
- Realización de la captura, descripción, evaluación y validación de los requisitos funcionales y no funcionales de los módulos.

Sobre objetivo 3:

- Realización del modelo entidad relación de los módulos a desarrollar.
- Realización del diagrama de componentes de los módulos a desarrollar.
- Realización del diagrama de clases del diseño de los módulos a desarrollar.
- Validación de los artefactos generados en estas fases.

Sobre objetivo 4:

- Realización de la codificación de los módulos aplicando los estándares establecidos para el desarrollo en XEDRO-ERP.
- Descripción de los algoritmos realizados mediante la inserción de comentarios en el código fuente.
- Realización del diagrama de despliegue correspondiente.

Sobre objetivo 5:

- Realización de pruebas de caja negra a los módulos aplicando técnicas y métodos de prueba.

Sobre objetivo 6:

- Realización de pruebas de aceptación a los módulos.

Métodos de investigación

Métodos teóricos:

- Analítico-sintético: Se empleó este método para estudiar toda la bibliografía seleccionada sobre el tema, para analizar y comprender la información relacionada con la gestión de compra y venta, las herramientas y tecnologías para el desarrollo y lograr la asimilación de los elementos más importantes relacionados con el objeto de estudio.
- Histórico-lógico: Este método se empleó para conocer los antecedentes y estado actual de los sistemas informáticos relacionados con compra y venta, así como las causas de su evolución.
- Modelación: Se empleó este método para la realización de los modelos correspondientes al ciclo de vida del sistema, ya sea las tareas a cumplir del diseño de los procesos y diagramas que forman parte de la solución.

Métodos empíricos:

- Entrevista: Para consultar con expertos sobre temas relacionados con la gestión de compra y venta, además de obtener conocimiento acerca de las necesidades y exigencias que debe cumplir el sistema a desarrollar.
- Observación: Se exploran los procesos relacionados con el sistema, con el objetivo de percibir las necesidades reales que debe satisfacer. Se empleó este método para analizar el flujo de actividades que se realiza durante los procesos de compra y venta.

Estructura del documento:

Capítulo 1. Fundamentación teórica: se describen los principales conceptos relacionados con los procesos de compra y venta. Se expone una valoración referente a los sistemas informáticos existentes en Cuba y el resto del mundo que gestionen la compra y venta. Se realiza la fundamentación de las herramientas, lenguajes y modelo de desarrollo a utilizar para dar solución a la problemática.

Capítulo 2. Modelado de negocio, requisitos y diseño de la solución propuesta: se realiza un estudio del negocio y de las necesidades de información a través de la disciplina modelado

de negocio. Se definen los requisitos funcionales y no funcionales, describiéndose cada uno de ellos y se exponen las técnicas utilizadas para validarlos. Además, se elabora una propuesta de solución en términos de componentes, diagramas y otros artefactos que guiarán la posterior implementación.

Capítulo 3. Implementación y validación de la solución propuesta: se implementan las funcionalidades identificadas y se valida el diseño propuesto. Se exponen los resultados obtenidos durante la validación de la solución propuesta mediante las pruebas de caja negra y de aceptación.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En el presente capítulo se abordan los fundamentos teóricos asociados a la investigación y se enuncian las definiciones y conceptos relacionados con las actividades de compra y venta. Se realiza un análisis profundo para obtener la mayor cantidad de información actualizada, de sistemas informáticos que gestionan de una forma u otra los procesos de compra y venta, implementados en el ámbito nacional e internacionalmente. Se analiza: la arquitectura, el modelo de desarrollo de software definido y las tecnologías, lenguajes de programación y herramientas a utilizar en la implementación de la solución propuesta.

1.2 Principales conceptos

Para el mejor entendimiento de los procesos de negocio relacionados con las acciones de compra y venta en las entidades, es necesario definir una serie de conceptos que facilite el modelado del negocio, imprescindibles para la posterior implementación de los módulos propuestos en la investigación.

1.2.1 Gestión comercial

La gestión comercial es un término usado para describir varias de las disciplinas de negocios presentes en una empresa u organización, en particular la administración de los ingresos y gastos para generar un retorno financiero. Es una de las tradicionales áreas funcionales de la gestión que se encuentra en cualquier organización y que se ajusta a la ejecución de tareas de ámbito de compras, ventas y marketing, incluyendo: [1]

- El estudio del ambiente externo y de las capacidades de la propia organización con el fin de alcanzar las previsiones de compras y ventas, con el objetivo de determinar las principales amenazas y oportunidades que se presentan a la organización.
- La organización y control de las actividades comerciales y de marketing, incluyendo la definición de estrategias comerciales y políticas de actuación.
- La relación con el cliente incluyendo la definición de la forma de recaudación de fondos, la definición de servicios complementarios y de post-venta y la gestión de reclamaciones.

1.2.2 Gestión de compras

La gestión de compras se define como la operación comercial que tiene como principal objetivo la función de compra, garantizando así la reducción de costos, la obtención de utilidades y la eficiencia en el servicio. En términos generales es obtener, mediante negociaciones, los recursos necesarios que ayuden en el cumplimiento de la misión de la organización. [2]

Compra

Comprar supone el proceso de localización y selección de proveedores, adquisición de productos, luego de negociaciones sobre el precio y condiciones de pago, así como el acompañamiento de este proceso para garantizar el cumplimiento de las condiciones pactadas. En términos de mercadotecnia, comprar es adquirir por un precio en dinero algún bien, derecho o mercancía. [3]

1.2.3 Gestión de ventas

La gestión de ventas permite el seguimiento de las actividades de ventas desde el primer contacto hasta el final de la realización del pedido de venta. Es un proceso dinámico donde interactúan diferentes elementos que contribuyen a que la venta se realice, teniendo como premisa la satisfacción de las necesidades del cliente. [4]

Venta

La venta es la transferencia de algo (un producto, servicio) a un comprador mediante el pago de un precio convenido. Es toda actividad que incluye un proceso personal o impersonal mediante el cual, el vendedor: 1) identifica las necesidades y/o deseos del comprador, 2) genera el impulso hacia el intercambio y 3) satisface las necesidades y/o deseos del comprador (con un producto, servicio) para lograr el beneficio de ambas partes. [5]

1.2.4 Cliente

Es aquel individuo que mediando una transacción financiera adquiere un producto y/o servicio de cualquier tipo (tecnológico, gastronómico, decorativo, mueble o inmueble). Un cliente es sinónimo de comprador o de consumidor y estos se pueden clasificar en activos e inactivos, de compra frecuente u ocasional, de alto o bajo volumen de compra, satisfecho o insatisfecho y si

son o no potenciales. El vendedor o encargado de marketing debe asegurarse de tomar en cuenta tanto las necesidades como las expectativas de cada cliente. [6]

Ciente es la persona, empresa u organización que adquiere o compra de forma voluntaria productos o servicios que necesita o desea para sí mismo o para otra empresa u organización; por lo cual, es el motivo principal por el que se crean, producen, fabrican y comercializan productos y servicios. [7]

1.2.5 Producto

Todo lo que el comprador recibe cuando efectúa un acto de compra: el producto propiamente dicho (bien o servicio), el envase, la garantía y los servicios complementarios. Un producto comercial es algo más que un bien o servicio que satisface una determinada necesidad. Constituye una combinación de atributos que con frecuencia son determinantes del éxito o fracaso comercial. [8]

El producto es el resultado de un esfuerzo creador que tiene un conjunto de atributos tangibles e intangibles (empaquete, color, precio, calidad, marca, servicios y la reputación del vendedor) los cuales son percibidos por sus compradores (reales y potenciales) como capaces de satisfacer sus necesidades o deseos. Por tanto, un producto puede ser un bien, un servicio, una idea, una persona o un lugar. El mismo existe para propósitos de intercambio, la satisfacción de necesidades o deseos y para contribuir al logro de objetivos de una organización (lucrativa o no lucrativa). [9]

1.2.6 Proveedor

Proveedor es la persona o empresa que abastece con algo a otra empresa o a una comunidad. El término procede del verbo proveer, que hace referencia a suministrar lo necesario para un fin. Los proveedores deben cumplir con los plazos y las condiciones de entrega de sus productos o servicios para evitar conflictos con la empresa a la que abastecen. En muchos casos, estas compañías tienen que tener un departamento de soporte o atención técnica, debido a los problemas que causan a los clientes las interrupciones del servicio. [10]

En informática, un proveedor es una entidad física o virtual que tiene el fin de ofrecer un servicio a otra u otras entidades. Los tipos de proveedores pueden ser tan distintos como una empresa que brinda servicios de Internet a clientes en un país, como un sistema informático

que pone aplicaciones y recursos al servicio de otros. En general, se conoce como proveedores a las empresas o particulares que ofrecen servicios tecnológicos. Estos pueden ser acceso y conexión a Internet, telefonía móvil y sitios web, acceso a servicios y cuentas en determinados software o sitios web. [11]

1.2.7 Factura

Una factura es un documento de carácter administrativo que sirve de comprobante de una compraventa de un bien o servicio, además de incluir toda la información de la operación. A través de este una persona que vende puede rendir cuentas, de forma instrumentalizada, al contrato de compraventa comercial. Las facturas además de probar una transacción de compra o venta debe contar con ciertos datos de las partes, así como también, la clase de producto vendido y su cantidad, o bien el tipo de servicio prestado, el número y la fecha de emisión. Además, una factura, debe mostrar el precio total y unitario de la transacción, los diferentes gastos que pueden deberse a diversos conceptos y que deben abonarse al comprador, así como también, en caso de que suceda, los impuestos que la compraventa implique. [12]

1.2.8 Precio

Precio es la expresión de valor que tiene un producto o servicio, manifestado por lo general en términos monetarios, que el comprador debe pagar al vendedor para lograr el conjunto de beneficios que resultan de tenerlo o usarlo [13]. Es el monto de dinero asignado a un producto o servicio, o la suma de los valores que los compradores intercambian por los beneficios de tener, usar o disfrutar un bien o servicio [14].

1.2.8.1 Precio de venta

El precio puede estudiarse desde dos perspectivas diferentes. La perspectiva del cliente que lo utiliza como una referencia de valor y la perspectiva de la empresa que lo toma como una herramienta para generar recursos encaminados a recuperar la inversión realizada y obtener una ganancia. El precio de venta en términos financieros, es aquel valor suficiente para cubrir los costos variables y el margen de contribución. El precio de venta está compuesto por los costos totales (Costo variable más Costo fijo) y la utilidad. [14]

1.2.8.2 Precio de compra

Precio que se paga al vendedor al comprar algo. Se deben deducir los descuentos en factura, pero no el retorno variable. Aunque es práctica habitual, no debe restarse a efectos de precio de coste el de cuenta por pronto pago, pues esta es una decisión financiera y no comercial. [15]

1.2.9 Resoluciones vigentes para la compra y venta

Para el desarrollo de la solución propuesta es necesario el estudio de las leyes que rigen las actividades comerciales de las entidades estatales del país. Para ello se estudiaron las resoluciones dictadas por el Ministerio de Finanzas y Precios (MFP), una de las entidades rectoras encargada de establecer las políticas económicas del país.

Entre estas legislaciones se destaca la Resolución No.11/2007, publicada en la Gaceta Oficial en su Edición Ordinaria No.15 de 12/02/2007, la cual pone en vigor los datos de uso obligatorio que se tendrán en cuenta al momento de diseñar los modelos del Subsistema de Inventario. Dentro de esta se enfatizó en el modelo SC-2-01 - Solicitud de Compras, encargado de solicitar de los proveedores los productos necesarios para el desenvolvimiento de las operaciones de la entidad.

También se estudiaron las Resoluciones No.20/2014 y No.21/2014 publicadas en la Gaceta Oficial en su Edición Extraordinaria No.12 de 06/03/2014. En estas legislaciones el MFP haciendo uso de sus funciones para dirigir y controlar la ejecución de las políticas de formación, fijación y modificación de precios y tarifas, pone en vigor las siguientes metodologías:

- “Metodología general para la formación y modificación de precios mayoristas, tarifas técnico-productivas y tasas de margen comercial.”
- “Metodología general para la formación y modificación de precios minoristas y las tarifas a la población.”

Estas resoluciones son parte de los entregables que conforman el expediente de proyecto de XEDRO-ERP, como parte de la bibliografía consultada para la modelación del negocio.

1.3 Valoración de sistemas que incluyen los procesos de compra y venta

En la actualidad existen diferentes sistemas informáticos que se encargan de la realización de los procesos de compra y ventas. Para una mejor comprensión de las funcionalidades a desarrollar en la solución propuesta, es necesario realizar un estudio de los sistemas de gestión que se describen a continuación.

1.3.1 OpenERP

OpenERP es un sistema de planificación de recursos empresariales (ERP) de licencia libre que cubre las necesidades de las áreas de contabilidad, ventas, compras, almacén, inventario, proyectos, recursos humanos, tiendas virtuales, entre otros. Incorpora funcionalidades de gestión de documentos y conectores con otras aplicaciones. Permite trabajar remotamente mediante una interfaz web o aplicación de escritorio multiplataforma (Windows, Linux y Mac) e incluye un entorno modular de programación/adaptación rápida de aplicaciones. [16]

El módulo de gestión de compras permite: [16]

- Realizar el seguimiento de tarifas de sus proveedores y convertirlas en órdenes de compra.
- Monitorizar factura y realizar el seguimiento de la recepción de materiales solicitados.
- Gestionar entregas parciales del proveedor y mercancías faltantes.
- Gestionar las reclamaciones a proveedor por retrasos en la entrega.
- Generar borradores de pedidos de compra (necesidades de compra) automáticamente.

El módulo de gestión de ventas permite: [16]

- Crear pedidos de venta.
- Revisar los pedidos en sus distintos estados.
- Consultar el flujo de trabajo de los pedidos de venta.
- Confirmar envío.
- Definir una fecha de facturación y las condiciones individualmente en cada pedido.
- Asignar los gastos de envío usando una tabla de tarifas donde se pueden consultar los precios de los diferentes proveedores de transporte.
- Generar albaranes¹ automáticos desde pedido.

1.3.2 Openbravo

Es un sistema ERP de código abierto para la gestión de pequeñas y medianas empresas. Tiene como principales ventajas que sigue un licenciamiento de software libre, lo cual permite adquirirlo sin costo alguno o la necesidad de depender de algún proveedor de servicio.

¹ Documento mercantil que acredita la entrega de un pedido.

Capítulo 1. Fundamentación teórica

Openbravo ha sido diseñado para ser una solución web (montado sobre un servidor), haciendo posible que sea consultado desde cualquier parte del mundo. Entre sus principales módulos se encuentran: Gestión de datos maestros, Del pedido al cobro (Ventas), Del pedido al pago (Compras), Gestión financiera, Gestión de inventario, Gestión de proyectos y servicios y Gestión de la producción. [17]

El módulo de Compras garantiza un control de la cadena de suministros, con funcionalidad alineada para minimizar costos de adquisiciones y optimizar la colaboración con socios comerciales. Este módulo está completamente integrado con los de finanzas, contabilidad y bodegas, permitiendo: [18]

- Maximizar el flujo de efectivo y la estandarización de procesos de Cuentas por pagar.
- Reducir costos de inventarios y distribución, mientras mantiene los niveles de servicio altos.
- Compartir información de inventarios y de uso con proveedores a través de un acceso seguro y basado en roles, así como integración fácil a servicios web.
- Controlar las compras de sus empleados a través de una gestión de requisiciones centralizadas y basada en web.
- Integrar procesos de bodegas para reducir desperdicio, errores y tiempos de ciclo.

El módulo de Ventas gestiona varias de las actividades relacionadas con la venta de productos o servicios; órdenes de venta, facturas, comisiones y comprobantes. También permite hacer fácilmente informes analíticos que ayudarán a ver la evolución de las ventas. Este módulo está interrelacionado con los módulos de gestión de almacenes y económico-financiero, así como con el de gestión de productos y servicios. El módulo permite: [19]

- Gestionar pedidos a través del soporte multi-moneda y multi-idioma.
- Realizar la facturación electrónica para facturar a los clientes de manera más rápida y precisa, además de mejorar la disponibilidad del efectivo.
- Agilizar la introducción de nuevos productos y minimizar los errores en los precios gracias a una gestión centralizada de productos y listas de precios a cualquier nivel de la organización.

- Gestionar el ciclo de pedido a cobro para complejas líneas de productos con número ilimitado de atributos de producto, seguimiento de la cantidad disponible por atributo e inventario cíclico.
- Monitorizar en tiempo real el ciclo del pedido al cobro así como disponer de una visión total del cliente gracias a la potente inteligencia de negocio integrada.

1.3.3 SAP Business One

Es un sistema integrado de gestión empresarial, que ha sido diseñada específicamente para las pequeñas y medianas empresas. Permite la informatización de los procesos empresariales, ofreciendo una visión veraz y unificada de la información empresarial actualizada y crítica que requieren las compañías en todas sus áreas funcionales. La visibilidad total sobre las áreas de ventas, finanzas, compras, gestión de almacenes y producción, entre otras, permite tomar decisiones seguras, incrementando así la rentabilidad y los niveles de control sobre los respectivos procesos.

El módulo de Compras, presente en SAP Business One, ayuda a gestionar todo el ciclo desde el pedido hasta el pago, incluidos los recibos, las facturas y las devoluciones. También puede planificar las necesidades de material para la producción, controlar las listas de materiales y reponer existencias automáticamente. [20]

El módulo Ventas de SAP Business One permite: [20]

- Crear ofertas, introducir pedidos y proporcionar una mejor atención al cliente.
- Realizar un seguimiento de las oportunidades y las actividades de venta desde el primer contacto hasta el cierre de la venta.
- Empezar campañas de marketing utilizando plantillas para el envío masivo de mensajes por correo electrónico.
- Proporcionar soporte para la atención al cliente, contratos de servicio y garantías.

SAP utiliza para su creación software propietario, lo que contrasta con la política de migración a software libre del país. Además se observa que este posee gran cantidad de las funcionalidades requeridas por los módulos que se desean desarrollar, pero no están acorde con lo establecido en las resoluciones ministeriales estudiadas.

1.3.4 VERSAT Sarasola

Es un sistema integrado de gestión económica, diseñado para ser utilizado por el sector empresarial cubano. Tiene como objetivo fundamental ofrecerles a los usuarios la posibilidad de contar con un instrumento seguro, rápido y de fácil manejo para la planificación, control y el análisis de la gestión económica. Permite llevar el control y el registro contable individual de todos los hechos económicos que se originan en las estructuras internas de las entidades, así como obtener los estados financieros y sus respectivos análisis a estos niveles. Es un producto cubano, en el que están incorporados los principios de control interno y demás regulaciones vigentes en el país, con la garantía de continuar desarrollándolo e implementándole nuevas opciones. Entre sus principales módulos se encuentra Contratación y Facturación, los cuales posibilitan controlar y evaluar la ejecución de todo el proceso empresarial, así como las actividades de la entidad con sus clientes y suministradores. Permite además realizar la facturación de las acciones y servicios que comercialicen. Entre sus actividades características están: [21]

- Posibilidad de definir y controlar el proceso de actividades con los clientes y suministradores, a través de la interrelación que se establece con otros subsistemas (Inventario, Activos Fijos, Finanzas y Control).
- Emite prefacturas y facturas de las actividades y servicios, e interactúa para este proceso con los subsistemas antes mencionados, utilizando para ello varios métodos de cálculo que permiten adecuar el sistema para acciones realizadas en el proceso de ventas.
- Trabaja con precios estabilizados y controla los tipos de actividad a las cuales se le aplica.
- Es configurable en su funcionamiento y trabaja en estrecha relación con los subsistemas de Inventarios para la reserva de existencias a los clientes en el proceso de prefacturación y con Finanzas para garantizar el cobro de las deudas de los respectivos clientes mediante un chequeo sistemático del estado de las mismas en el momento de efectuar las ventas.
- Emite variedad de reportes sobre todas las actividades y funciones.

1.3.5 RODAS XXI

Es un sistema multiempresa creado por CITMATEL para la automatización de la gestión empresarial que posibilita automatizar el funcionamiento de cualquier empresa o unidad

presupuestada. Cuenta actualmente con ocho módulos: Finanzas, Contabilidad, Activos Fijos, Nóminas, Inventario, Facturación, Recursos Humanos y Telecobranzas. [22]

El módulo Facturación permite: [23]

- Declarar siete tipos diferentes de clasificadores (productos, servicios, categorías, clientes, proveedores, gestor de venta, tipos de facturas).
- Prefacturar, facturar y refacturar.
- Realizar la facturación en moneda nacional y en divisa.
- De forma opcional, elaborar los comprobantes contables.
- Emitir informes (facturas, prefacturas, productos más vendidos, clientes que más compran, vendedores que más venden, niveles de facturación, otros).

1.3.6 ASSETS NS

Es un sistema comercializado por la firma panameña D'MARCO S.A. y distribuido en Cuba por INFOMASTER, entidad informática perteneciente a la Empresa Nacional de Producción y Servicios a la Educación Superior del MES². ASSETS NS es un sistema de gestión integral estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Taller, Producción, Inventario, Finanzas, Activos Fijos, Útiles y Herramientas, Contabilidad, Presupuesto y Recursos Humanos.

El módulo de Compras permite realizar todas las operaciones vinculadas al proceso de adquisición de mercancías para la venta o para insumo, pasando por un proceso de formación de costo a partir de las facturas de los diferentes proveedores, ya sean en torno al precio unitario del producto o teniendo en cuenta otros gastos incurridos en las gestiones de compras, manipulación, gestiones aduanales, transportación y seguros. Las opciones que brinda son las siguientes: [24]

- Emisión manual o generación automática de órdenes de compras al proveedor.
- Recepciones de mercancías.
- Recepciones de servicios.
- Recepciones de servicios asociadas a productos.
- Formación de costo con la posibilidad de estimar.

² Ministerio de Educación Superior.

- Creación de ficha de costo.
- Actualización de existencia por mercancías recepcionadas.
- Análisis de compras.
- Análisis de pedidos incompletos.

El módulo de Ventas permite realizar cotizaciones a clientes, emitir listados de precios, ejecutar facturas a clientes por ventas directas de mercancías o servicios, así como controlar las facturas que se emiten desde el módulo de Taller. Se enlaza con el módulo de Control de inventario permitiendo extraer las mercancías de los almacenes vinculadas a una orden de trabajo o una orden de entrega. Adicionalmente, posibilita definir y calcular precios de venta a partir de parámetros, así como ejecutar una amplia gama de análisis estadísticos relacionados con el proceso de ventas. Las opciones que brinda este módulo son las siguientes: [24]

- Emisión de cotizaciones a clientes.
- Seguimiento de las cotizaciones a clientes.
- Emisión de facturas de ventas de mercancías o servicios.
- Control de facturas generadas por cierre de órdenes de trabajo.
- Control de facturas generadas por órdenes de entrega.
- Despacho de mercancías.
- Cálculo de precios de venta.
- Cálculo de listas de precios.
- Análisis de ventas.

1.3.7 Valoración de los sistemas estudiados.

A continuación se muestra una tabla comparativa de los sistemas estudiados en cuanto a parámetros relevantes, necesarios para dar solución a la problemática planteada.

Tabla 1. Resultados del estudio de los sistemas analizados

Sistemas estudiados	Plataformas	Tipo de software	Cumplimiento de las resoluciones del MFP ³	Funcionalidades de gestión de compras	Funcionalidades de gestión de ventas	Soporte
---------------------	-------------	------------------	---	---------------------------------------	--------------------------------------	---------

³ Ministerio de Finanzas y Precios.

Capítulo 1. Fundamentación teórica

Open ERP	Windows, Linux, Mac.	Libre	No	Gestionar pedidos de compra, entregas parciales y mercancías faltantes.	Gestionar pedidos de ventas, facturación y condiciones del envío.	No
Openbravo	Linux.	Libre	No	Gestionar cadena de suministros y proceso de Cuentas por Pagar.	Gestionar pedidos de ventas, ciclo del Pedido al Cobro y listas de precios.	No
SAP	Windows.	Privativo	No	Gestionar el ciclo desde el pedido hasta el pago, recibos, facturas y devoluciones.	Crear ofertas e introducir pedidos. Gestionar las oportunidades y las actividades de venta.	Sí
VERSAT	Windows.	Privativo	Sí	Gestionar facturas recibidas por el aprovisionamiento. Reportes sobre todas las actividades y funciones relacionadas con las compras.	Interrelación con los subsistemas de Inventarios para la reserva de existencias a los clientes y con Finanzas para garantizar el cobro de las deudas de los clientes.	Sí
Rodas XXI	Windows.	Privativo	Sí	Declarar siete tipos diferentes de clasificadores de la factura. Permitir prefacturar, facturar y refacturar. La facturación se realiza en moneda		Sí

				nacional y en divisa. De forma opcional elabora los comprobantes contables.	
ASSETS NS	Windows.	Privativo	Sí	Permitir la emisión de órdenes de compras, creación de ficha de costo y análisis de compras.	Gestionar facturas de ventas, calcular precios de venta y despacho de mercancías.

A partir del estudio realizado a los sistemas mencionados anteriormente, se observó que a pesar de poseer varias funcionalidades que permiten realizar los procesos de compra y venta, no abarcan todas las necesidades funcionales para dar respuesta a la problemática. Muchos fueron desarrollados bajo herramientas y software propietarios, siendo ejemplo de esto SAP, VERSAT, Rodas XXI y ASSETS NS. Estos sistemas tienen como ventajas que poseen empresas certificadoras en Cuba que les brindan soporte y también cumplen con las resoluciones establecidas por el MFP, exceptuado de esta última ventaja el sistema SAP. Sin embargo, tienen un costo inicial muy alto por lo que el país al utilizarlos incurre en grandes gastos de licencia y mantenimiento. Los demás sistemas pertenecen a la familia OpenSource⁴, que a diferencia de los propietarios, tienen un costo inicial bajo. Como desventajas de estos sistemas se encuentran la no presencia en el país de una empresa que certifique o garantice la seguridad del producto, además de no cumplir con las resoluciones No.11/2007, No.20/2014 y No.21/2014 dictadas por el MFP.

Se puede concluir que ninguno de los sistemas estudiados cumple con todas las características deseadas, por lo que sería más factible crear un sistema que responda a las particularidades de la economía cubana, que incluya las características positivas de los estudiados y sea consecuente con el paradigma de independencia tecnológica por el cual apuesta el país, para lo cual sería imprescindible la utilización de herramientas y tecnologías libres para el desarrollo de la solución.

1.3.8 Características del producto XEDRO-ERP

⁴ Código abierto, expresión con la que se conoce al software distribuido y desarrollado libremente.

XEDRO-ERP es un paquete de soluciones integrales de gestión para las entidades presupuestadas y empresariales, basada en los principios de independencia tecnológica y con funcionalidades generales de los procesos y las particularidades de la economía cubana. Pone al servicio de las entidades cubanas las potencialidades de la tecnología informática y provee facilidades para la integración de las diferentes áreas productivas y departamentos administrativos. Sus características generales son: [25]

- Independencia tecnológica.
- Plataforma web.
- Interoperabilidad con otros sistemas.
- Multimoneda.
- Multientidades.
- Control de fechas.
- Integralidad funcional.

XEDRO-ERP en su primera fase de explotación abarca en su solución a los subsistemas Configuración y Multimoneda, Estructura y Composición, Contabilidad General, Costo y Procesos, Caja, Banco, Cobros y Pagos, Capital Humano e Inventario y Facturación. Es una solución apta para desplegarse en centros de datos por su arquitectura y puede ser explotada con el uso de la tecnología de clientes ligeros. La utilización de software libre para su desarrollo contribuye a la independencia tecnológica del país. Para el desarrollo de los módulos de Compra y Venta se requiere de una estrecha relación con algunos de los subsistemas antes mencionados, de modo que se logre una integración óptima con el sistema en general. [25] A continuación se muestran las principales características de las funcionalidades que abarcan los subsistemas utilizados durante el desarrollo de la solución propuesta:

Configuración y Multimoneda: se pueden realizar ajustes a los elementos del sistema para adaptarlos a las características de cada entidad. Permite la gestión de moneda contable y de tasas, la reevaluación de cuentas y el desglose de monedas. [25]

Estructura y Composición: brinda la posibilidad al usuario de definir la estructura organizativa en la cual se va a ubicar su entidad y la estructura dentro de dicha entidad. Permite además que se puedan especificar los cargos por la cuales van a estar compuestas las diferentes áreas dentro de las unidades y los medios asociados. [25]

Inventario y Facturación: permite gestionar los documentos primarios que se utilizan en el control de las existencias y los movimientos del producto dentro del almacén: Apertura del almacén, Recepción de productos, Despacho de productos, Inventario físico, Ajuste de inventario, Baja de productos en el almacén y Conciliación entre almacenes. Además posibilita el registro de las facturas realizadas por venta de productos y activos fijos tangibles (AFT), así como por la prestación de servicios. [25]

La solución a desarrollar estará enmarcada dentro de este subsistema, en el que se añadirá un nuevo documento primario (Solicitud de compra) y se formarán los precios de venta de los productos, así como el desglose correspondiente al precio.

1.4 Modelo de desarrollo del software

El modelo de desarrollo de software propuesto por el CEIGE describe el flujo de actividades necesarias para la construcción y desarrollo de soluciones. Este brinda la posibilidad de que todos los proyectos de CEIGE cuenten con un modelo estandarizado que incluye las fases por las que se debe transitar durante el desarrollo de una solución y el conjunto de artefactos a generar en cada una de ellas. Entre sus principales características se encuentran el enfoque de desarrollo basado en componentes, la reutilización, se tiene en cuenta las líneas de productos de software (LPS) y la integración con los distintos subprocesos dictado por el Nivel II del Modelo de Capacidad de Madurez del Software (CMMI, por sus siglas en inglés). Se divide en dos fases de desarrollo: Estudio preliminar y Desarrollo, este último formado por una secuencia de disciplinas necesarias para el desarrollo de una solución. [26]

A continuación se muestran los artefactos que se generarán por cada una de las disciplinas de la fase de Desarrollo, teniendo en cuenta la estructura del presente trabajo: [26]

Modelado de negocios

- Mapa de procesos de negocio.
- Diagrama de procesos de negocio.
- Descripción de procesos de negocio.
- Modelo conceptual.

Requisitos

- Descripción de requisitos.
- Evaluación de requisitos.

Análisis y diseño

- Modelo de datos.
- Diagrama de componentes.
- Descripción de los componentes.
- Diagramas de clases del diseño.
- Diseño de casos de prueba.

Implementación

- Código fuente de los componentes.
- Modelo de despliegue.

Pruebas internas

- Resultados de las pruebas de aceptación.
- Resultados de las pruebas de caja negra.

1.5 Tecnologías y herramientas para el desarrollo

Para llevar a cabo el desarrollo de un sistema se comienza estudiando las principales soluciones existentes que ayudan a dar solución a la problemática planteada. Ejemplo de estas soluciones son los marcos de trabajo, los cuales resuelven muchos de los escenarios que pueden encontrarse en el transcurso del desarrollo, facilitando la obtención de resultados en breves períodos de tiempo.

1.5.1 Marco de trabajo

La solución propuesta se desarrollará utilizando el marco de trabajo Sauxe, implementado por el Departamento de Tecnología del CEIGE. Cuenta con los siguientes frameworks definidos para cada una de las capas como parte del Modelo-Vista-Controlador (MVC): [27]

- Para la Vista: ExtJS-Framework, el cual es muy utilizado en el desarrollo de aplicaciones Web con tecnología AJAX⁵.
- Para el Controlador: Zend-Framework el cual emplea específicamente el estilo Modelo-Vista-Controlador como base de su funcionamiento.
- Para agilizar el acceso a datos en el Modelo: Doctrine, un potente y completo sistema Mapeador de Objeto Relacional (ORM, por sus siglas en inglés).

⁵ Técnica de desarrollo web para crear aplicaciones interactivas.

ExtJS 2.2: Es una biblioteca o conjunto de librerías de JavaScript para el desarrollo de aplicaciones web interactivas, usa tecnologías AJAX, DHTML⁶ y DOM⁷. Permite realizar completas interfaces de usuario, fáciles de usar, muy parecidas a las conocidas aplicaciones de escritorio, posibilitando a los desarrolladores concentrarse en la funcionalidad de las aplicaciones en vez de en las advertencias técnicas. [28]

Zend_Framework 1.7: Se trata de un marco de trabajo para desarrollo de aplicaciones web y servicios web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es de código abierto y trabaja con PHP 5. Está formado por una serie de métodos estáticos y varios componentes que pueden usarse múltiples veces. [29]

Doctrine 1.2.1: Potente y completo sistema de mapeo relacional de objetos para PHP 5.2 con una capa de abstracción de bases de datos incorporada. Brinda la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, en otras palabras, convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. Al trabajar con este es necesario informar a su motor interno cuál es el modelo de la aplicación haciendo ingeniería inversa de la base de datos existente, o si se empieza la aplicación desde cero, crear el modelo en la sintaxis específica que propone Doctrine y luego generar toda la base de datos. [30]

1.5.2 Lenguajes de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas a desarrollar por un equipo. Los lenguajes de programación representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona. Es el proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo. [31]

JavaScript: Es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. Gran parte de su programación está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de

⁶ HTML Dinámico (del inglés Dynamic HTML).

⁷ Interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

páginas, entre otros. Permite la programación de pequeños scripts y de programas más grandes orientados a objetos, con funciones y estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página web para acceder a ellos y modificarlos dinámicamente. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado, es soportado por Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. [32]

PHP 5.2.6: Es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. [33]

1.5.3 Lenguajes y herramientas para el modelado

Notación de Modelado de Procesos de Negocio (BPMN⁸): Es un patrón de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso. Su principal objetivo es proveer una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio. Entre estos interesados están los analistas de negocio (quienes definen y redefinen los procesos), los desarrolladores técnicos, responsables de implementar los procesos, los gerentes y administradores del negocio (quienes monitorizan y gestionan los procesos). En síntesis, BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. [34]

Lenguaje Unificado de Modelado (UML⁹): Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Permite modelar sistemas de información y su objetivo es lograr modelos que, además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los clientes o usuarios de aquello que se modela. Posibilita el intercambio de modelos entre las distintas herramientas de modelado orientadas a objetos del mercado, constituyendo un lenguaje estándar al definir una notación y semántica común para todas estas herramientas. [35]

⁸ Business Process Modeling Notation.

⁹ Unified Modeling Language.

Visual Paradigm 6.4: Es una herramienta de diseño UML libre y profesional, diseñado para contribuir al desarrollo de software. Soporta los principales estándares como UML, SysML¹⁰, BPMN y XML¹¹. Ofrece un completo conjunto de herramientas a los equipos de desarrollo de software, necesarios para la captura de requisitos, la planificación de software, la planificación de pruebas, el modelado de clases y el modelado de datos. [36]

1.5.4 Herramientas para el desarrollo colaborativo

Subversion 1.6.6: Es un software de sistema de control de versiones OpenSource y se le conoce también como **svn** por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que los archivos versionados no tienen cada uno un número de revisión independiente y puede registrar la historia de archivos de código fuente y documentos. En cambio, todo el repositorio tiene un único número de versión. Se coloca un árbol de archivos en un repositorio central. El repositorio es como un servidor de archivos ordinario, excepto en que recuerda todos los cambios que se realizan sobre los archivos y los directorios que contiene. [37]

1.5.5 Herramientas de programación

NetBeans 7.4: NetBeans es un producto libre y gratuito, sin restricciones de uso. Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) disponible para Windows, Mac, Linux y Solaris, de código abierto, escrito completamente en Java. Es una plataforma de aplicaciones que permite a los desarrolladores crear rápidamente aplicaciones, utilizando un número importante de módulos para extenderlo a otros lenguajes como PHP (Zend y Symfony), JavaScript, Ajax, Groovy y Grails y C / C + +. [38]

1.5.6 Gestor de Base de Datos

PostgreSQL 8.3: Es un sistema gestor de base de datos. Tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y joins. Permite la definición de tipos de datos personalizados e incluye un

¹⁰ Systems Modeling Language.

¹¹ eXtensible Markup Language.

modelo de seguridad completo. Cuenta con una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno y es multiplataforma. [39]

1.5.7 Servidor Web

Apache 2.2.9: Es un servidor gratuito de código fuente abierto, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Se ejecuta en varios Sistemas Operativos, característica que lo hace prácticamente universal. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es altamente configurable en la creación y gestión de logs¹². [40]

1.5.8 Navegador Web

Mozilla Firefox 2.17: Presenta una forma rápida y eficiente de navegar por la web, que permite abrir varias páginas en una misma ventana mediante el empleo de pestañas separadas. Contiene el plugin Firebug que se utiliza para ver los errores del código. Firefox es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y otros sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL. [41]

1.6 Arquitectura

Para el desarrollo del marco de trabajo Sauxe, en el Departamento de Tecnologías del CEIGE se definió el estilo arquitectónico **N capas**, con cuatro capas fundamentales que se describen a continuación: [42]

- **Capa de presentación:** esta es la capa encargada de elaborar y mostrar las interfaces de los usuarios. La capa de presentación está compuesta principalmente por el marco de trabajo ExtJS y centra su desarrollo en tres componentes fundamentales archivos JS, archivos CSS, páginas clientes y páginas servidoras.
- **Capa de control o negocio:** en esta capa se encuentra ubicado el marco de trabajo Zend, el cual implementa el patrón Modelo-Vista-Controlador.
- **Capa de acceso a datos:** está ubicado el marco de trabajo Doctrine, como ORM para la comunicación con el servidor de datos mediante el protocolo PDO¹³.

¹² Registro de errores.

¹³ Protocolo de conexión a base de datos basado en objetos.

- **Capa datos:** en esta capa se encuentran ubicado los servidores de base de datos y los archivos XML donde se almacena la información de la aplicación.

En la capa de control o negocio de la arquitectura descrita anteriormente para Sauxe, se implementa el patrón arquitectónico MVC, el cual se encarga de separar los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos: [42]

- **Modelo:** está compuesto por datos, reglas de negocio y las funcionalidades correspondientes para la comunicación con el encargado de persistir los datos usando Doctrine.
- **Controlador:** gestiona las entradas y las respuestas del sistema al usuario.
- **Vista:** muestra la información del modelo al usuario.

1.7 Patrones de diseño

Una vez definida la arquitectura, los patrones constituyen una parte indispensable en el desarrollo del software, estos hacen la producción más resistente al cambio, establecen parejas problema-solución y ayudan a especificar interfaces. Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software, por lo que es de suma importancia aplicarlos en la construcción del diseño de un sistema. Con su uso, se pretende establecer un lenguaje común entre los programadores, contribuir a la reutilización, ahorrar tiempo en la implementación y obtener un producto con calidad. Brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. [43]

1.7.1 Patrones GRASP

GRASP es el acrónimo para General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignación de Responsabilidades). Los patrones GRASP representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones. Dentro de este grupo de patrones se destacan: Experto, Creador, Controlador, Bajo acoplamiento y Alta cohesión, los cuales se analizarán a continuación: [44]

Experto: Ofrece una analogía con el mundo real. Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe incurrir sobre la clase que conoce toda la información necesaria para ejecutar la tarea. Este patrón posibilita una adecuada asignación de

responsabilidades facilitando la comprensión del sistema, su mantenimiento y adaptación a los cambios con reutilización de componentes.

Creador: Guía la asignación de responsabilidades que tienen relación con la creación de objetos. Ayuda a identificar quién debe ser el responsable de la instancia o de la creación de nuevas clases u objetos. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia es útil contar con un principio general para la asignación de las responsabilidades de creación.

Controlador: Se encarga de asignar la responsabilidad a una clase en el momento de manejar mensajes correspondientes a eventos en un sistema a clases específicas, facilitando la centralización de actividades. Esta clase es la responsable de manejar la lógica de negocio, decidiendo así que clase es la encargada de ejecutar una tarea determinada.

Bajo acoplamiento: Consiste en tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases.

Alta cohesión: Importante en todas las decisiones del diseño, es la meta principal que debe buscarse en todo momento. Indica que la información que recopila una clase debe ser coherente, de modo que todos sus métodos tengan un comportamiento bien definido. Es una medida que determina cuán relacionadas y adecuadas están las responsabilidades de una clase, de manera que no realice un trabajo colosal. Una clase con baja cohesión realiza un trabajo excesivo, haciéndola difícil de comprender, reutilizar y conservar.

1.7.2 Patrones GoF

Los patrones Gang Of Four (GoF), son patrones de diseño publicados en el libro Design Patterns: Elements of Reusable Object-Oriented Software por Gamma, Helm, Jonson y Vlissides, conocidos por Banda de los cuatro. Están divididos fundamentalmente en tres grandes grupos: [45]

Patrones creacionales: Se encargan de las formas de crear instancias en los objetos. Su objetivo es: abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados. Ejemplos: Fábrica abstracta e Instancia única.

Patrones estructurales: Estudian cómo se relacionan los objetos en tiempo de ejecución y se aprecia cómo las clases y los objetos se combinan para formar nuevas estructuras más complejas y proporcionar nuevas funcionalidades. Ejemplos: Decorador y Fachada.

Patrones de comportamiento: Están relacionados con algoritmos y asignación de responsabilidades a los objetos. Describen no solamente patrones de objetos o clases, sino también patrones de comunicación entre ellos. Ejemplos: Cadena de responsabilidad y Observador.

1.8 Conclusiones del capítulo

Luego de valorar y analizar varios conceptos relacionados con los procesos de compra y venta, estudiar sistemas de gestión empresarial tanto nacionales como extranjeros, así como varias herramientas utilizadas para el desarrollo del software, se pudo arribar a las siguientes conclusiones:

- El estudio de las resoluciones dictadas por el MFP para la regulación de los procesos de compra y venta en las entidades cubanas, permitió identificar los principales elementos establecidos para el desenvolvimiento de las operaciones de compra y de formación de precios, a tener en cuenta en la implementación de la solución.
- El análisis de los sistemas de gestión empresarial estudiados, fundamenta la necesidad de desarrollar nuevos módulos que se adapten a los requerimientos del producto XEDRO-ERP. Estos incluirán las características positivas de las funcionalidades de los sistemas estudiados de importancia para la solución.
- La descripción de las tecnologías, el modelo de desarrollo y las herramientas a utilizar en la construcción de la solución contribuyen a un mejor entendimiento de estas, mostrando además que su utilización favorece el paradigma de independencia tecnológica por el cual apuesta el país.

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

2.1 Introducción

En el presente capítulo se realiza la modelación de los procesos de compra y venta. Se elabora un mapa de procesos en el cual se muestran los flujos de actividades y artefactos relacionados con la gestión de compra y venta. Se elabora el modelo conceptual, registrando los conceptos fundamentales relacionados con los módulos propuestos. Son descritos y validados los requisitos funcionales y no funcionales que fueron identificados a partir del uso de técnicas de captura de requisitos. Se obtiene el modelo de datos, en el que se describen cada una de las tablas creadas. Además se estructura la implementación de la solución mediante el diagrama de componentes y los diagramas de clases del diseño.

2.2 Modelado del negocio

El primer paso de la disciplina modelado del negocio, consiste en capturar los procesos de negocio presentes en las actividades de los clientes. La definición del conjunto de procesos del negocio es una tarea crucial, pues define los límites del proceso de modelado posterior. [46]

Para el modelado de negocio se realizó la identificación de los procesos presentes en las tareas de compra y venta. Esta actividad se ejecutó en conjunto con la analista principal de XEDRO-ERP, obteniendo un mapa de procesos que detalla los procesos, sus entradas y salidas, además del diagrama y la descripción de cada proceso de negocio identificado.

2.2.1 Mapa de procesos

Un mapa de procesos muestra cómo se relacionan los procesos de negocio, especificando las entradas y salidas de cada uno. Las entradas son los documentos que se necesitan para realizar el proceso, las salidas la constancia de lo que se hizo, además de las condiciones en que quedó conformado (Ver Figura 1).

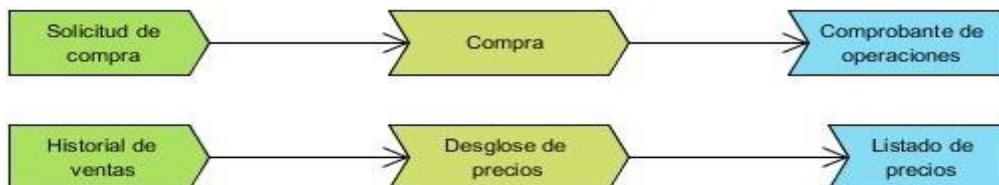


Figura 1. Mapa de procesos del negocio

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

En el desarrollo del Mapa de procesos del negocio se identificaron y describieron los principales procesos que se realizan para gestionar las actividades de compra y venta. Estos fueron los siguientes:

Proceso de negocio Compra

En este proceso se realizan las compras necesarias para garantizar el abastecimiento de las actividades de ventas e insumos. Para realizar una compra es necesario un documento que contenga las especificaciones de la compra a realizar, así como todos los datos y requisitos que se deben cumplir para efectuar esta operación (solicitud de compra). Una vez recepcionada la compra solicitada se conforma un documento que recoja los detalles de las operaciones efectuadas por la misma (comprobante de operaciones).

A través de este proceso, se establece un flujo organizativo de documentos primarios, que según las normativas establecidas por el MFP, se utilizan en un proceso normal de compras o adquisición de productos. Este flujo, de forma muy resumida, puede definirse en el siguiente orden:

1. Solicitar y aprobar una compra de productos.
2. Enviar solicitud de compra al proveedor y copia del documento al almacén.
3. Realizar la recepción detallada de los productos adquiridos.
4. Actualizar inventario por productos recepcionados o emitir informe de reclamación al proveedor.
5. Actualizar submayor de inventario y generar un comprobante de operaciones.

Proceso de negocio Configuración de precios

En este proceso se realiza un desglose de los precios necesarios para llevar a cabo el cálculo de los precios de venta de los productos, tanto mayoristas como minoristas. Para realizar este cálculo es necesario consultar un documento que recoja el historial de los precios de este producto (Historial de ventas). Después de realizado el cálculo se conforma un documento que recoja los precios, los cuales se utilizarán para conformar las facturas de venta.

A través de este proceso, se establece un flujo organizativo de documentos primarios, que según las normativas establecidas por el MFP, se utilizan en un proceso de formación de

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

precios en las entidades estatales. Este flujo, de forma muy resumida, puede definirse en el siguiente orden:

1. Consultar el historial de ventas.
2. Escoger tipo de precio a calcular y el método de formación de precios correspondiente.
3. Elaborar la matriz de precios del producto.
4. Elaborar el listado de precios.

2.2.2 Descripción de procesos de negocio

Para obtener información sobre la descripciones correspondientes a los procesos de negocio Compra (Ver Anexo 1) y Configuración de precios (Ver Anexo 2), consultar el expediente de proyecto de CEIGE.

2.2.3 Validación del modelado de negocio

Para la validación de los artefactos generados como parte de la modelación del proceso de negocio se aplicó la técnica Revisión Técnica Formal. Para ello la analista principal del XEDRO-ERP realizó un proceso de revisión riguroso de cada uno de los documentos generados con el objetivo de verificar que se ajustaran a sus especificaciones y estándares establecidos, señalando las posibles desviaciones detectadas.

2.2.4 Modelo conceptual

Un modelo conceptual explica los conceptos más significativos en un dominio del problema, identificando los atributos y las asociaciones. Representa elementos del mundo real, no componentes del software. En estos diagramas se muestran conceptos (objetos), asociaciones entre conceptos (relaciones) y atributos de conceptos (atributos). [47]

La Figura 2 muestra el modelo conceptual con las clases, relaciones y atributos identificados, generados a partir del análisis de los procesos compra y configuración de precios para XEDRO-ERP. Para mayor información sobre el modelo conceptual consultar Anexo 3.

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

principal de XEDRO-ERP acerca de los procesos de compra y venta, con el fin de tomar conocimiento del problema y comprender los objetivos de la solución buscada.

Tormenta de ideas: es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. Las sesiones de tormentas de ideas suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el jefe de la sesión, encargado más de comenzar la sesión que de controlarla. [49] Se seleccionaron en el centro CEIGE un grupo de especialistas a ser partícipes en la realización de la técnica, entre los que se destacan la analista principal, el arquitecto del sistema y el administrador de configuraciones del sistema XEDRO-ERP, así como la analista y el arquitecto de la línea Logística. De este grupo se eligió a la analista principal como responsable de las dos sesiones en la que se reunieron todos estos profesionales y los miembros del equipo de desarrollo, con el objetivo de generar ideas. Se escogieron las más significativas y que daban respuesta a la problemática planteada, las cuales pasaron a conformar los requisitos del sistema. Como constancia de estos encuentros se generaron dos minutas de reuniones que forman parte del expediente de proyecto de XEDRO-ERP, recogiendo los puntos tratados y los acuerdos tomados por cada uno de ellos.

2.3.2 Requisitos de software

La actividad de captura de requisitos tiene gran importancia en el proceso de desarrollo del software, pues a través de estos se identifican las funcionalidades del sistema que dan respuesta a las necesidades del cliente.

Los requisitos de software, según Sommerville, se pueden clasificar en:

- **Funcionales:** son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.
- **No funcionales:** son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Estos a menudo se aplican al sistema en su totalidad.

2.3.2.1 Requisitos funcionales

Los requisitos funcionales identificados de los procesos de Compra y Venta son:

RF 1. Agrupación de requisitos: Configuración de precios.

RF 1.1 Calcular el precio de venta mayorista.

RF 1.2 Calcular el precio de venta minorista.

RF 1.3 Modificar precio de venta mayorista.

RF 1.4 Modificar precio de venta minorista.

RF 1.5 Mostrar matriz de precio por producto.

RF 1.6 Mostrar el desglose de precio por método de formación según tipo de precio.

RF 1.7 Imprimir listado de precios de venta por producto.

RF 2. Agrupación de requisitos: Gestionar solicitud de compra.

RF 2.1 Adicionar solicitud de compra.

RF 2.2 Modificar solicitud de compra.

RF 2.3 Eliminar solicitud de compra.

RF 2.4 Listar solicitudes de compra.

RF 2.5 Consultar solicitud de compra.

RF 2.6 Buscar solicitud de compra.

RF 2.7 Realizar búsqueda avanzada de solicitud de compra.

RF 2.8 Imprimir solicitud de compra.

RF 2.9 Imprimir listado de solicitudes de compra.

RF 2.10 Confirmar solicitud de compra.

RF 2.11 Cancelar estado de la solicitud de compra.

RF 2.12 Aprobar solicitud de compra.

RF 3. Agrupación de requisitos: Gestionar producto de la solicitud de compra.

RF 3.1 Adicionar producto a la solicitud de compra.

RF 3.2 Modificar producto de la solicitud de compra.

RF 3.3 Eliminar producto de la solicitud de compra.

RF 3.4 Listar productos de la solicitud de compra.

RF 3.5 Buscar producto en la solicitud de compra.

RF 4. Agrupación de requisitos: Gestionar informe de recepción por solicitud de compra.

RF 4.1 Adicionar informe de recepción a partir de la solicitud de compra.

RF 4.2 Adicionar producto al informe de recepción a partir de la solicitud de compra.

Descripción de los requisitos funcionales

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

La descripción de los requisitos identificados se realizó siguiendo los pasos descritos en la plantilla conformada por el CEIGE. Para más información sobre de los requisitos consultar el expediente de proyecto de CEIGE en el que se encuentran todas las descripciones de requisitos funcionales realizadas. A continuación se muestra como ejemplo la descripción del requisito Adicionar solicitud de compra (ver Tabla 2).

Tabla 2. Descripción de requisito Adicionar solicitud de compra

Precondiciones	Existe un proveedor y un depósito registrados en el sistema.
Flujo de eventos	
Flujo básico	
1	Se introducen los datos de la solicitud de compra: Fecha de vencimiento. Nombre del proveedor. Depósito.
2	El sistema valida (ver validación 1) los datos introducidos.
3	Si los datos son correctos el sistema los registra.
4	El sistema confirma el registro de datos.
5	Concluye el requisito.
Pos-condiciones	
6	Se registró en el sistema una nueva solicitud de compra.
Flujos alternativos	
Flujo alternativo 1.a El usuario cancela la acción	
1	Concluye el requisito.
Pos-condiciones	
1	No se registran los datos.
Flujo alternativo 2.a Información errónea	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.
Pos-condiciones	
N/A	
Flujo alternativo 2.b Información incompleta	
1	El sistema señala los datos vacíos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

Pos-condiciones

N/A

Validaciones

1 Se validan los datos según lo establecido en el Modelo conceptual

Conceptos	Solicitud de	de	Visibles en la interfaz:
	Compra		Fecha de emisión del modelo. Número del modelo. Nombre del proveedor. Fecha de vencimiento. Depósito. Observaciones. Utilizados internamente: Código del proveedor. Código del solicitante. Código del depósito. Código del documento.

Prototipo de interfaz de usuario

Adicionar Solicitud de Compra

Fecha de emisión del modelo: Número del modelo:

Proveedor: Solicitante:

Entrega

Vía: Fecha: Lugar de entrega:

Listado de Productos

Total de la Solicitud:

Adicionar Cancelar

2.3.2.2 Requisitos no funcionales

Los requisitos no funcionales de los módulos de Compra y Venta fueron aplicados guiándose por el documento CIG-CDX-N-INV-i2301, perteneciente al expediente de proyecto de CEIGE, en el cual se especifican los requisitos del subsistema Logística. Los requisitos no funcionales aplicados a los módulos propuestos son:

- **Funcionalidad**

El sistema debe poseer la capacidad para proporcionar efectos o resultados correctos con un grado de exactitud necesario y la interacción equitativa con uno o más sistemas específicos.

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

RNF 1: Las respuestas de los módulos ante búsquedas corresponderán en un 100% a lo solicitado en los criterios.

RNF 2: El sistema será capaz de leer datos provenientes de los módulos: Configuración, Estructura y composición, Inventario y Facturación.

- **Seguridad**

El sistema debe poseer la capacidad para proteger la información y los datos; donde se garantizará el acceso de personas o sistemas y los que no tengan una previa autorización no podrán leer ni modificar.

RNF 3: El sistema concederá el acceso a cada usuario autenticado solo a las funciones que le estén permitidas, de acuerdo a la configuración del sistema.

RNF 4: El sistema registrará las trazas de operaciones realizadas por cada usuario en todo momento.

- **Confiabilidad**

El sistema debe poseer la capacidad de proteger la información ante acciones indebidas para el negocio.

RNF 5: Ninguna información que se haya ingresado en el sistema y se haya asociado a alguna operación, será eliminada físicamente de la base de datos.

RNF 6: Los errores cometidos por el usuario al realizar alguna operación les serán notificados y los mensajes incluirán sugerencias a operaciones alternativas.

RNF 7: El sistema no permitirá la entrada de datos incorrectos.

RNF 8: El sistema impondrá campos obligatorios para garantizar la integridad de la información que se introduce por el usuario.

- **Usabilidad**

El sistema debe poseer la capacidad para permitirle al usuario entender si el software es idóneo, operarlo y controlarlo para las tareas y condiciones de uso particulares.

RNF 9: Todos los mensajes de error del sistema deberán incluir una descripción textual del error.

RNF 10: El idioma de todas las interfaces de la aplicación será el español.

RNF 11: Las etiquetas de cada funcionalidad y los campos de cada interfaz tendrán títulos asociados a su función.

RNF 12: El sistema expondrá el menú general en todo momento para que pueda ser utilizado por el usuario cuando lo desee.

- **Soporte**

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

El sistema debe poseer la capacidad para permitir la aplicación de una modificación especificada.

RNF 13: El sistema permitirá agregar nuevas funcionalidades o modificar alguna existente sin romper la estructura y consistencia de los módulos.

- **Portabilidad**

El sistema debe poseer la capacidad del producto de software de ser adaptado a los ambientes especificados y de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.

RNF 14: El sistema podrá ser visualizado en todos los navegadores que estén soportados por el framework Extjs en su versión 2.2.

RNF 15: El sistema podrá ser instalado en el ambiente especificado en los requisitos tecnológicos para servidores.

- **Interfaz**

Interfaz de hardware.

RNF 16: El sistema interactuará con impresoras para imprimir los diferentes documentos que genere la aplicación como respuesta a las funcionalidades del sistema.

RNF 17: El sistema para su instalación en las maquinas clientes requiere:

Procesador: 1.40 GHZ

RAM: 512 MB (recomendado 1024 MB)

Tarjeta de Red: 1

RNF 18: El sistema para su instalación en el servidor de aplicaciones y el servidor de base de datos:

Procesador: 3.00 GHZ

RAM: 1GB (recomendado 4GB)

Disco duro: 160 GB (recomendado 500GB)

UPS: 1

Lector de CD: 1

Tarjeta de Red: 1

2.3.3 Evaluación de requisitos

Para la evaluación de los requisitos identificados en los procesos Compra y Configuración de precios del sistema XEDRO-ERP se utilizó la planilla "Evaluación de requisitos", definida por el

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

CEIGE. Siguiendo los criterios de complejidad que en ella se establecen, se determinan como requisitos de mayor prioridad aquellos que posean complejidad Alta.

Criterios de complejidad:

- **Diferentes comportamientos:** Un mismo requisito se comporta de manera diferente ante determinadas circunstancias.
- **Consultas a fuentes de almacenamientos:** Los requisitos pueden presentar diversidad en la cantidad y complejidad de la interacción con la fuente de datos (Base de datos, Ficheros, otros).
- **Restricciones de validación:** Complejidad de todas las validaciones que lleve un requisito, tanto las validaciones en el lado del cliente, como en el servidor.
- **Grado de reutilización:** Complejidad de un requisito, para poder ser reutilizado por otros.
- **Lógica de negocio:** Los requisitos pueden presentar diferentes niveles de complejidad para la implementación de la lógica de negocio que contienen (operaciones y métodos matemáticos).

La Tabla 3 muestra los resultados arrojados de la evaluación de los requisitos de acuerdo a los criterios de complejidad a los que se hicieron referencia anteriormente. Para más detalles consultar documento entregable CIG-CDX-N-LOG-i6101.

Tabla 3. Resultado complejidad

Resumen	
Cantidad de requisitos con complejidad Alta	5
Cantidad de requisitos con complejidad Media	7
Cantidad de requisitos con complejidad Baja	14

2.3.4 Validación de requisitos

Para llevar a cabo este proceso se propone aplicar las siguientes técnicas para la validación de requisitos:

Revisión técnica formal: Esta técnica fue desarrollada a partir de las revisiones que realizó el equipo de trabajo en conjunto con la Analista principal del proyecto XEDRO-ERP. A través de diferentes encuentros y entregas de artefactos para su revisión, se examinaron las

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

especificaciones y se realizó una búsqueda de errores en el contenido, en la interpretación, en la información faltante, inconsistencias, conflictos entre requisitos y la existencia de requisitos inalcanzables.

Prototipos: Esta técnica se aplicó a partir de las especificaciones de requisitos, mediante las cuales el equipo de trabajo conformó prototipos de interfaz de usuario y a través de estos, los especialistas (analista principal y arquitecto del sistema XEDRO-ERP) visualizaron de forma general una posible versión inicial de los módulos, ayudando a revelar errores u omisiones en los requisitos propuestos. Cada uno de los prototipos le permitió a los especialistas tener una idea de cada una de las interfaces gráficas del componente, estos se realizaron de forma no funcional con la herramienta Visual Paradigm for UML.

2.4 Modelo de datos del sistema.

El modelado de análisis comienza a menudo con el modelado de datos. El ingeniero o analista de software define todos los objetos de datos que se procesan dentro del sistema y las relaciones existentes entre dichos objetos. [48]

El objetivo de construir un modelo de datos es identificar y representar las tablas (entidades) de importancia para el funcionamiento del negocio, sus propiedades (atributos) y la forma en que estas tablas se comunican entre sí (relaciones). Este modelo se desarrolla para facilitar el diseño de la base de datos y mostrar los datos que contendrá el sistema.

El modelo de datos propuesto en la solución está estrechamente relacionado con los esquemas “documento”, “inventario” y “facturación” debido que las tablas creadas para el modelo se encuentran dentro de estos esquemas. En la Figura 3 se muestra el diagrama del modelo perteneciente al proceso de Compra, en el que se diferencian las tablas existentes de las añadidas para dar solución a los módulos propuestos. Se creó una tabla denominada “mod_documento.dat_compra” en la que se guardan los datos de la solicitud de compra en el proceso de Compra. Esta tabla se relaciona directamente con “mod_documento.dat_documento”, que establece los atributos generales de los documentos del sistema. También se relaciona con “mod_documento.dat_informerecepcion”, que representa el informe de recepción que se genera a partir de la solicitud de compra.

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

Para el proceso de compra de productos se creó la tabla “mod_inventario.dat_prodcompra”, en la que se establece la relación entre los productos y la solicitud de compra, además de contener otros atributos necesarios para la solicitud. Se relaciona con las tablas “mod_documento.dat_compra” y “mod_inventario.dat_producto”, esta última contiene la información de los productos existentes en el inventario.

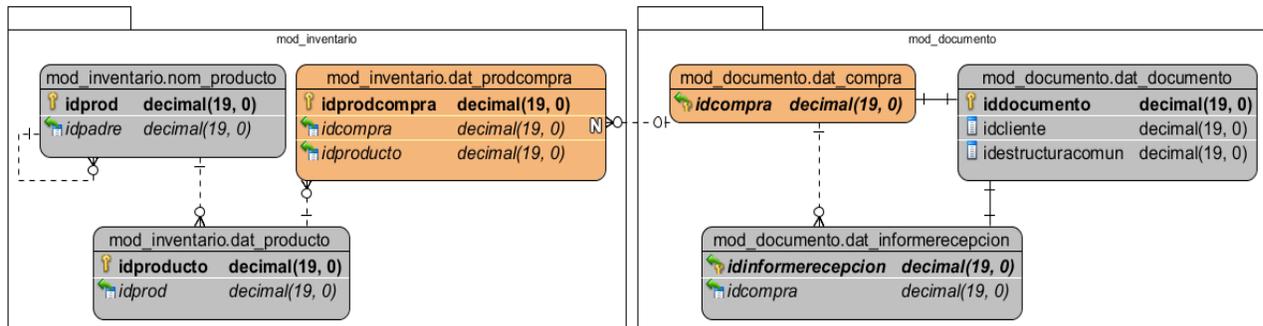


Figura 3. Modelo de datos del proceso Compra

La Figura 4 muestra el modelo de datos elaborado para el proceso de Configuración de precios, diferenciándose de igual forma las tablas creadas y las existentes. Se crearon las tablas nomencladores “mod_facturacion.nom_tipoprecio”, “mod_facturacion.nom_metodoprecio” y “mod_facturacion.nom_categoriaprecio”, que registran los datos correspondientes a los tipos de precios calculables, los métodos de formación de precios vigentes y las categorías de precios respectivamente. Se creó la tabla “mod_facturacion.dat_desgloseprecio”, la cual se relaciona con los nomencladores antes descritos, guardando así la relación que existe entre estos. Como resultado de la relación de muchos a muchos entre la categoría de precio y el desglose de precio se creó la tabla “mod_facturacion.dat_desglosecategoria”. La tabla “mod_facturacion.dat_desgloseprecio” también se relaciona con la tabla “mod_facturacion.dat_matrizprecio”, la cual establece la relación entre los productos, el tipo de precio a calcular y el método de formación de precio.

Para realizar el cálculo de los precios de venta de un producto por el método de correlación se creó la tabla “mod_facturacion.nom_correlacion”, la cual guarda los parámetros necesarios para establecer una comparación entre productos. Como resultado de la relación entre correlación y desglose de precio se creó la tabla “mod_facturacion.dat_desglosecorrelacion”, encargada de guardar el valor del parámetro resultante de la correlación que representa una parte del total del precio de venta del producto.

Para obtener mayor información sobre el modelo de datos general y las descripciones de las tablas creadas, consultar Anexo 4 y Anexo 5 respectivamente.

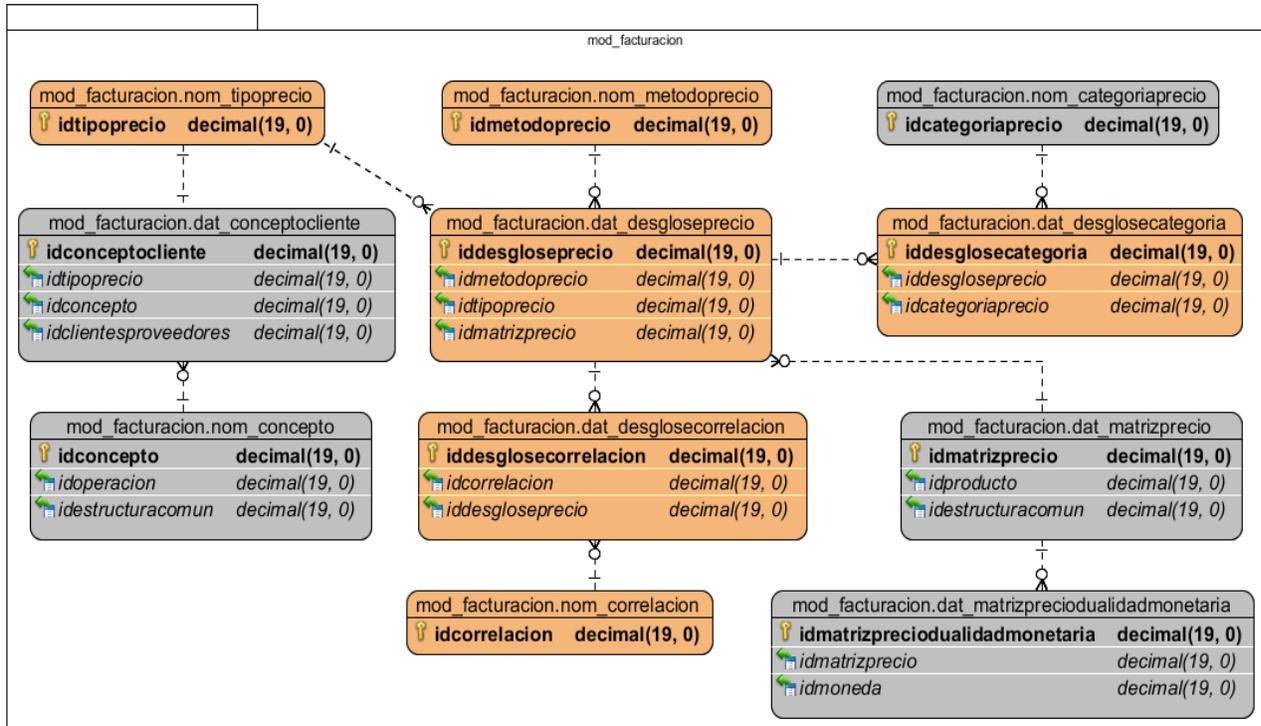


Figura 4. Modelo de datos del proceso Configuración de precios

2.5 Diagrama de componentes

Un modelo de componentes representa los componentes, sus interfaces y las relaciones con las interfaces que utilizan [50]. En la Figura 5 se expone el diagrama de componentes correspondiente a la solución propuesta, que muestra las relaciones que fueron definidas para dar solución arquitectónica a los módulos a desarrollar (Compra y Venta). Estos módulos están asociados a varios componentes, entre los cuales se encuentran: Estructuraop, Clientes y proveedores, Documentos, Multimoneda, Recepción y Productos.

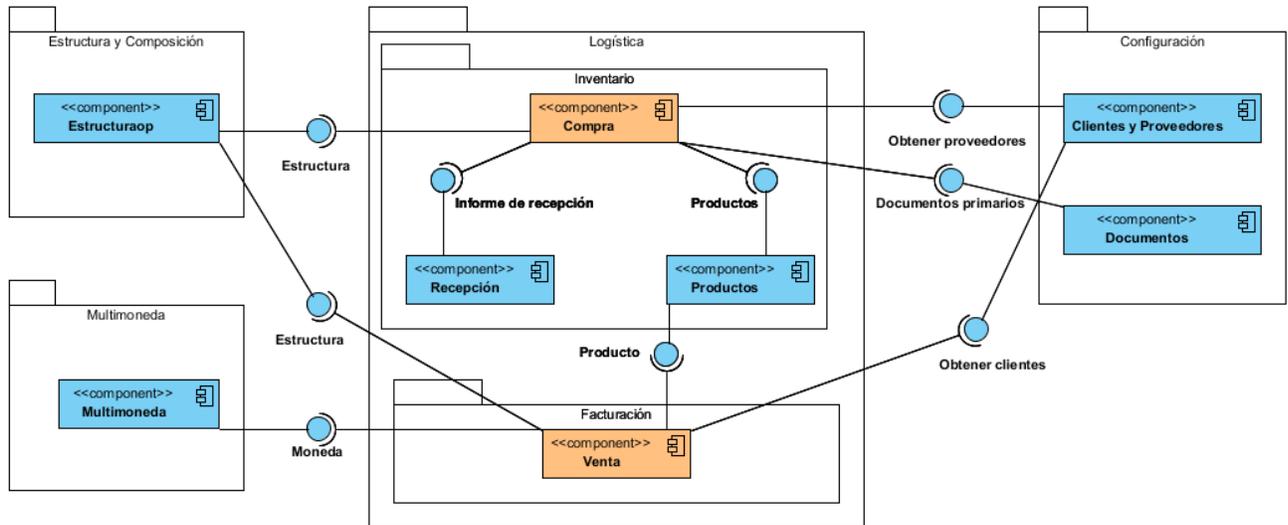


Figura 5. Diagrama de componentes

A continuación, una explicación más puntualizada de la funcionalidad de cada uno de estos componentes a partir de los servicios que brindan y reciben:

- **Gestionar estructura:** permite poner en práctica el concepto de la multientidad. Componente encargado de definir la estructura de la entidad y sus distintos niveles de formación.
- **Clientes y Proveedores:** contiene el listado de todos los clientes y proveedores de la entidad.
- **Multimoneda:** define todos los tipos de monedas usadas por una entidad.
- **Recepción:** encargado de definir un informe de recepción, el cual se elabora a partir de una solicitud de compra llevada a cabo por una entidad.
- **Productos:** define los productos con los que cuenta la entidad.
- **Documentos:** Gestionar el nomenclador Documentos primarios.
- **Compra:** crea una solicitud de compra para el proceso de aprovisionamiento de una entidad. Permite elaborar un informe de recepción a partir de la solicitud de compra creada.
- **Venta:** permite calcular los precios de venta de los productos destinados a la venta de una empresa. Elabora un desglose de precios del producto al cual se le calcula su precio de venta a partir del tipo de precio a calcular, el método de formación de precios y las categorías de precios que los conforman.

2.5.1 Servicios utilizados

Para lograr comunicación entre diferentes módulos, se utiliza el patrón inversión de control (IoC)¹⁴. La utilización de este patrón se evidencia en el fichero .xml con igual nombre (ioc.xml) encargado de registrar las funcionalidades que ofrecen los componentes del sistema para su utilización desde otros componentes. Existe un fichero ioc.xml interno contenido en la carpeta apps del subsistema Logística para la integración entre componentes de este subsistema. En la Figura 6 se muestra un fragmento de este, correspondiente al servicio Cargarproducto del componente Inventario. Para la integración entre subsistemas se utiliza el ioc.xml externo, la Figura 7 muestra un fragmento de este, correspondiente al servicio ObtenerProveedores del subsistema Configuración.

```
<?xml version="1.0" encoding="utf-8"?>
<ioc>
  <productos src="logistica/inventario/productos">
    <Cargarproducto reference="">
      <inyector clase="ProxyProdService" metodo="cargarprod"/>
      <prototipo>
        <parametro nombre="start" tipo="enterospos" default="0"/>
        <parametro nombre="limit" tipo="enterospos" default="0"/>
        <parametro nombre="idestructuraop" tipo="numerosletras" default="0"/>
        <parametro nombre="codigo"/>
        <parametro nombre="pieza" />
        <parametro nombre="desc" tipo="numerosletras" default="0"/>
        <parametro nombre="idcategoriaproducto"/>
        <parametro nombre="existencia"/>
        <parametro nombre="idgrupo"/>
        <parametro nombre="tipo"/>
        <parametro nombre="cantidad"/>
        <parametro nombre="signo"/>
        <parametro nombre="desde"/>
        <parametro nombre="hasta"/>
        <parametro nombre="activo"/>
        <parametro nombre="verexist" />
        <parametro nombre="verdisp" />
        <parametro nombre="vervalor" />
        <resultado tipo="array"/>
      </prototipo>
    </Cargarproducto>
  </productos>
</ioc>
```

Figura 6. Contenido del fichero ioc.xml interno

¹⁴ IoC: es un concepto junto a unas técnicas de programación en las que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos o funciones. Tradicionalmente el programador especifica la secuencia de decisiones y procedimientos que pueden darse durante el ciclo de vida de un programa mediante llamadas a funciones. En su lugar, en la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir.

```
<?xml version="1.0" encoding="utf-8"?>
<ioc>
  <parametros src="configuracion">
    <BuscarMoneda reference="multimoneda">
      <inyector clase="MultimonedaService" metodo="Monedas" />
      <prototipo>
        <parametro nombre="idmoneda" tipo="enterospos"/>
        <parametro nombre="identidad" tipo="enterospos"/>
        <resultado tipo="array" />
      </prototipo>
    </BuscarMoneda>
  </parametros>
</ioc>
```

Figura 7. Contenido del fichero ioc.xml externo

Servicios que consumen los módulos Compra y Venta:

BuscarPadrePorSubordinacion: servicio utilizado para obtener el id de la estructura a la cual pertenecen los productos a los que se les desea calcular los precios de venta.

Cargarproducto: servicio consumido para cargar los productos a los cuales se les desea calcular sus precios de venta.

datosProducto: servicio utilizado para obtener los datos de los productos a los cuales se les desea calcular sus precios de venta.

Conceptosparentidad: servicio consumido para obtener los conceptos asociados a la entidad.

BuscarMoneda: servicio consumido para obtener las monedas con que cuenta la entidad.

FechaContableSubsistema: fecha contable de creación de un documento.

ObtenerProveedores: carga los datos de los proveedores asociados a la entidad.

ObtenerProductoDadoldEstructura: carga los productos de una entidad determinada dado su identificador.

BuscarDatosDoc: carga los datos de un documento dado un criterio de búsqueda determinado.

CargarDatosEncabezado: carga los datos del encabezado de un documento dado su identificador.

TipoDocumentoPorId: devuelve el tipo de documento dado su identificador.

NombreUsuario: devuelve el nombre del usuario registrado en el sistema.

ObtenerClientesProveedorEntidad: devuelve los clientes y proveedores asociados a una entidad.

ObtenerClientes: devuelve los clientes asociados a una entidad.

Servicios que brinda el módulo Compra:

BuscarEntidadSuministradoresPor: devuelve los proveedores por un criterio de búsqueda determinado.

BuscarProductoSolicitudCompraPorId: devuelve los productos asociados a una solicitud de compra.

AdicionarDocCompra: adiciona una solicitud de compra.

BuscarSolicitudesCompraPorEntidad: devuelve las solicitudes de compra asociadas a una entidad.

EliminarSolicitudDadold: elimina una solicitud de compra dado su identificador.

2.6 Diagrama de clases del diseño.

Los diagramas de clases según la clasificación UML, son diagramas de estructura estática donde la representación de los requisitos se lleva a cabo a través de las clases del sistema y sus interrelaciones [51]. Su principal utilidad radica en mostrar a través de sus atributos y métodos la estructura de las clases que después serán escritas en algún lenguaje de programación (PHP, en este caso).

En la Figura 8 se muestra de manera general el diagrama de clases del diseño del proceso Compra, que será informatizado con el desarrollo del componente Compra y en la Tabla 4 su respectiva descripción. Para obtener mayor información sobre este diagrama y el correspondiente al proceso de Configuración de precios, consultar Anexo 6 y Anexo 7 respectivamente.

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

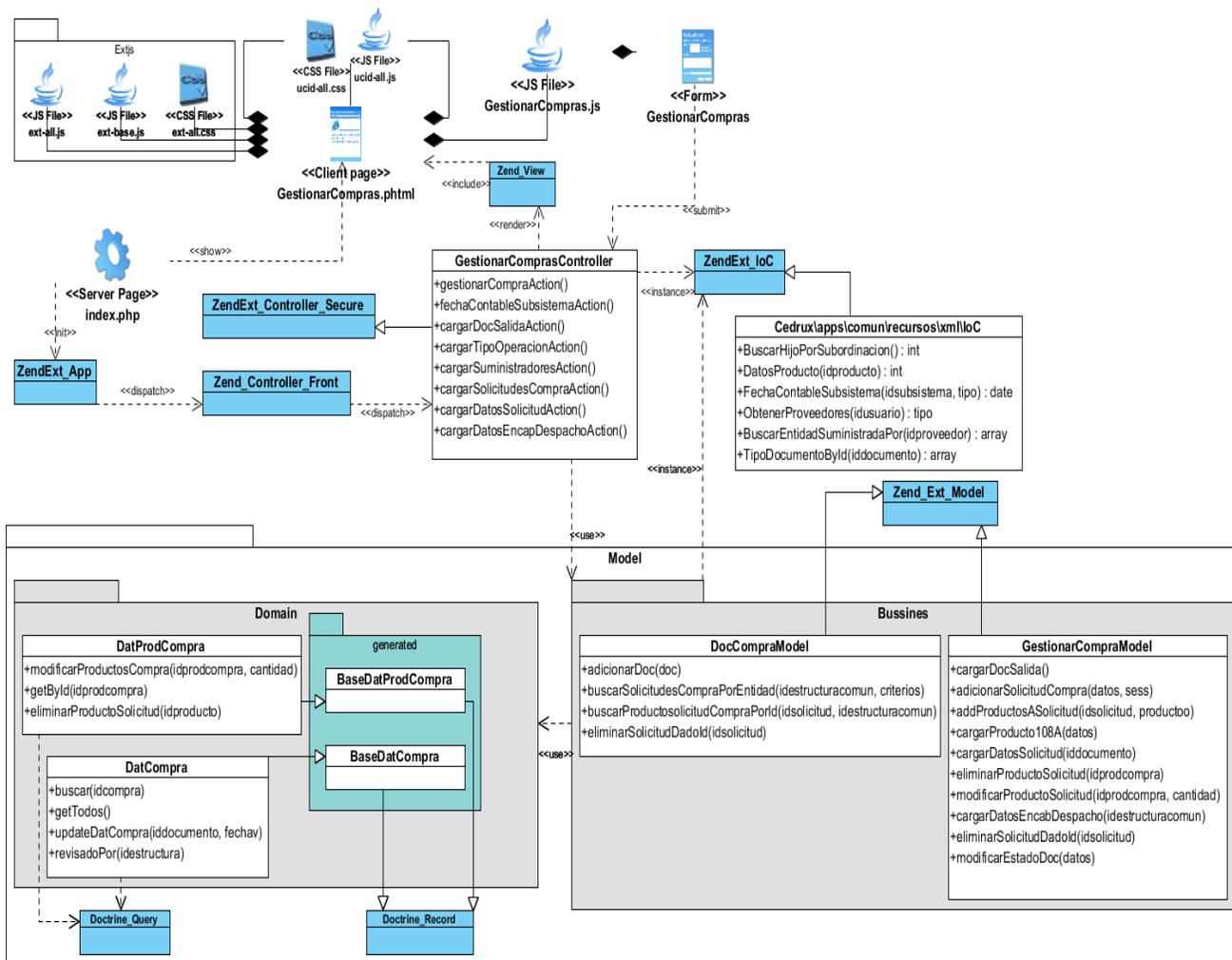


Figura 8. Diagrama de clases del diseño del proceso Compra

Tabla 4. Descripción del diseño de clases del proceso Compra

Clases	Descripción
GestionarCompras.phtml	Encargada de visualizar, a través de los js que debe incluir, la información necesaria del proceso Compra.
GestionarCompras.js	Encargada de generar de forma dinámica, a través del DOM y utilizando la librería ExtJS, los componentes necesarios a través de los cuales se puedan realizar las operaciones para generar una solicitud de compra. Responsable de enviar y recibir los datos de la controladora utilizando tecnología Ajax.

Capítulo 2. Modelado del negocio, requisitos y diseño de la solución

GestionarCompras	Encargada de recoger los datos entrados por el usuario necesarios para conformar una solicitud de compra.
Index.php	Encargada de redireccionar el sistema, comienza con el levantamiento del sistema y va al componente Portal.
GestionarComprasController	Clase controladora encargada de manejar la comunicación entre la programación del negocio y la vista. A partir de ella se deben ejecutar las diferentes funcionalidades, según las peticiones del usuario.
IoC	Permite fomentar la reutilización de componentes mediante los servicios que brindan. Esta representada en el diagrama por dos clases(IoC general del XEDRO-ERP e IoC del subsistema Logística) pues se utilizan servicios de ambas.
Paquete Model	Encargado de manejar los datos persistentes dentro del componente. Contiene el Bussines y el Domain.

2.7 Patrones de diseño y arquitectura empleados en la solución.

XEDRO-ERP está implementado usando el marco de trabajo Sauxe. La arquitectura y el diseño a través del correcto uso de patrones de diseño en la generación de los artefactos necesarios para el desarrollo, facilitaron crear una entrada apropiada como punto de partida a las actividades de implementación, con el fin de lograr una mayor calidad del producto y la satisfacción del cliente. La selección de los patrones de diseño y arquitectura estuvo basada en el marco de trabajo, evidenciando su uso como se describe a continuación.

Modelo Vista Controlador

El Patrón Arquitectónico Modelo-Vista-Controlador (MVC) se empleó en la solución de la forma siguiente: La Vista engloba la capa de presentación, en la misma es manejado todo el flujo web. Le brinda al usuario la posibilidad de interactuar con el Controlador mediante los mensajes de eventos y le permite visualizar las respuestas a sus peticiones. El Controlador por su parte representa el gestor de eventos, el cual está compuesto por las clases controladoras. Este atiende los pedidos que llegan desde la Vista accediendo al Modelo que se encuentra dividido en los paquetes: business y domain. El primero encierra las clases correspondientes a la lógica del negocio y el segundo las clases del dominio, las que se encargan de leer y/o escribir datos en las tablas de la base de datos.

Patrones GRASP

Controlador: El empleo de este patrón se puede observar en la creación de una clase controladora para las diferentes acciones que se realizan en el módulo de Compras. Ejemplo: GestionarComprasController.

Experto: Se evidencia su uso en todas las clases del paquete Models, pues cada clase conoce su información y es la encargada de implementar las funcionalidades que le corresponde. Ejemplo: DocCompraModel.

Alta Cohesión: Su empleo está dado en que fueron asignadas responsabilidades a las clases de forma tal que la cohesión siguiera siendo alta, o sea, cada clase se encarga de realizar solamente las funciones que están en correspondencia con la responsabilidad que posee. Un ejemplo de esto se evidencia en la clase controladora GestionarComprasController, la cual delega responsabilidades a otras clases, encargándose solo de definir las acciones a realizar.

Bajo Acoplamiento: Este patrón se pone de manifiesto al no existir una conexión directa entre las clase controladora GestionarComprasController y las clases modelos de dominio como DatCompra creadas por Doctrine.

Creador: Este patrón se evidencia en las clases del paquete Domain, las que crean los objetos de tipo Doctrine_Query, posibilitando el acceso a la información almacenada a nivel de datos.

Patrones GoF

Singleton: Este patrón asegura que una clase tendrá solo una instancia y provee un punto de acceso global a la misma. Se utiliza para las clases en las que se necesita que exista un solo ejemplar de ellas, evitando de esta forma sobrecargas o problemas de seguridad. El empleo de este patrón se evidencia con el uso del objeto pIntegrator, como instancia única para acceder a los servicios del IoC desde cualquier clase perteneciente al sistema.

Fachada: El uso de este patrón está evidenciado en el empleo del archivo IoC. Debido a la necesidad de integración de los módulos a desarrollar con los ya existentes en XEDRO-ERP, fue necesario la utilización de una clase fachada donde son publicados los servicios a consumir entre ellos, facilitando de este modo su integración.

2.8 Conclusiones del capítulo

Tras aplicar las disciplinas de Modelado de negocio, Requisitos y Análisis y diseño se pudo llegar a las siguientes conclusiones:

- El modelado del negocio facilitó la delimitación de los procesos Compra y Configuración de precios a tener en cuenta en la solución, como paso previo a la identificación de los requisitos funcionales.
- La definición, descripción y validación de los requisitos de los módulos, posibilitó un mayor entendimiento de las funcionalidades a implementar, para dar solución al problema planteado.
- El diseño del modelo de datos y el diagrama de componentes permitió conocer las relaciones existentes entre las diferentes tablas de la base de datos y la relación de los módulos a desarrollar con los existentes, respectivamente.
- El diagrama de clases de diseño permitió conocer la estructura y las relaciones entre las clases que se manejan en los módulos.

Capítulo 3. Implementación y validación de la solución propuesta

3.1 Introducción

El presente capítulo mostrará una breve descripción de la estructura del marco de trabajo Sauxe necesaria para llevar a cabo la implementación de los módulos, así como los estándares de codificación definidos para el marco de trabajo. Se elaborará el diagrama de despliegue para obtener una perspectiva más clara sobre el hardware empleado en las implementaciones de sistemas y las relaciones entre sus componentes. Se realiza la validación del diseño a través de métricas definidas para ello y la validación de la solución propuesta para la cual se aplicarán pruebas de caja negra y de aceptación.

3.2 Implementación de los módulos

3.2.1 Estructura del Marco de trabajo Sauxe

A continuación se mostrará la estructura del marco de trabajo Sauxe, definiéndose así cómo se organizará la implementación de los módulos a desarrollar, de manera que facilite la organización y claridad durante el desarrollo.

Contenido dentro de la carpeta apps:

En la carpeta denominada apps se encuentran todos los subsistemas de XEDRO-ERP y cada uno de estos está compuesto por los módulos correspondientes. Dentro del subsistema Logística se encuentran los módulos Inventario y Facturación, dentro de los cuales se ubican Compra y Venta respectivamente.



Figura 9: Contenido de la carpeta de la aplicación apps

Los módulos Compra y Venta van a contener un conjunto de carpetas que serán especificadas a continuación:

Capítulo 3. Implementación y pruebas de la solución propuesta

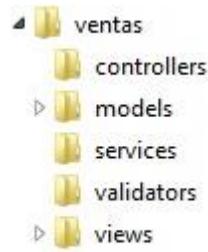


Figura 10: Contenido del componente Ventas dentro de la carpeta app

En la carpeta **controllers** se encontrarán las clases controladoras encargadas de gestionar las funcionalidades del componente.

La carpeta **models** se estructura de la siguiente forma:

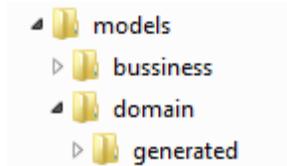


Figura 11: Contenido de la carpeta models

Esta carpeta contiene otras dos, las cuales a su vez agrupan clases y otras carpetas que serán explicadas a continuación:

- **bussines:** contiene las clases necesarias para implementar la lógica del negocio.
- **domain:** contiene las clases generadas por el ORM Doctrine Generator a partir de cada una de las tablas existentes en la base de datos y las clases de acceso a los datos que persisten en la base de datos.

Cada una de estas clases mencionadas anteriormente heredará de una clase generada igualmente por el Doctrine Generator las cuales se ubican en otra carpeta, dentro de esta, llamada generated.

La carpeta **services** contiene las funcionalidades que dan respuesta a los servicios que va a brindar el componente y que referencian a la implementación del negocio en la Model correspondiente. Para solicitar un servicio que está en otro dominio, se accede a través de la carpeta services, esta analizará la solicitud e irá a la clase que tiene dicha funcionalidad y la devolverá a services, que a su vez se la entregará al dominio solicitante.

Capítulo 3. Implementación y pruebas de la solución propuesta

La carpeta **views** contiene las carpetas idioma y scripts, que se encargan de contener el idioma en que se va a mostrar la aplicación y las páginas clientes respectivamente.

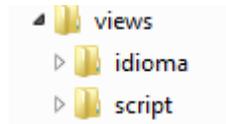


Figura 12: Contenido de la carpeta views

Contenido dentro de la carpeta web:

Al mismo nivel de la carpeta apps mencionada inicialmente debe existir además una carpeta (denominada web) que contenga las vistas de los subsistemas y componentes.



Figura 13: Carpeta de diseño correspondiente al componente Ventas

Dentro de la carpeta ventas se encuentra el archivo **index**: este fichero contiene la dirección del archivo de configuración y además inicializa la aplicación para que se carguen en la misma un conjunto de componentes necesarios para su funcionamiento. Su código permanece igual para todos los componentes.

La carpeta **views**, que se sitúa al mismo nivel que el archivo antes mencionado, contendrá los css y los js que se explican a continuación.



Figura 14: Contenido dentro de la carpeta views

- **css**: incluye las clases necesarias para estructurar gráficamente el componente, separando de esta forma el estilo del contenido.

Capítulo 3. Implementación y pruebas de la solución propuesta

- **js:** comprenderá las clases Java Script necesarias para que el usuario interactúe con el sistema y obtenga los resultados necesarios. Está compuesta por carpetas con los nombres de las funcionalidades del componente.

3.2.2 Estándares de codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. A continuación se especifican los estándares de codificación que fueron utilizados en el desarrollo de los módulos de Compra y Venta, normados por el documento “Estándares de codificación para Proyectos con el marco de trabajo Sauxe del CEIGE”:

Nomenclatura de las clases

Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing¹⁵. Con solo leerlo se reconoce el propósito de la misma. Ejemplo: GestionarVentas.

- Según el tipo de clases:

Las clases controladoras después del nombre llevan la palabra "Controller". Ejemplo: GestionarVentasController.

Como parte de las clases del modelo se encuentran las pertenecientes al Business (Negocio). Las clases que se encuentran dentro de Business después del nombre llevan la palabra "Model". Ejemplo: GestionarPreciosModel.

Como parte de las clases del modelo también se encuentran las clases del Domain (Dominio). Las clases que se encuentran dentro de Domain el nombre que reciben es el de la tabla en la Base de Datos. Ejemplo: DatDesglosePrecio.

Dentro de la carpeta Domain también se encuentran las clases del Generated, para las que se define la siguiente nomenclatura: el nombre comienza con la palabra "Base" y seguido el nombre de la tabla en la Base de Datos. Ejemplo: BaseDatDesglosePrecio.

¹⁵ PascalCasing es un procedimiento de programación común en el lenguaje Java y .Net. La nomenclatura está compuesta por tantas palabras como sean necesarias. La letra inicial del identificador y la primera letra de cada una de las palabras que lo forman irán siempre en mayúsculas y se obvia el uso de artículos.

Nomenclatura de las funcionalidades

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing¹⁶. Con solo leerlo se reconoce el propósito de la misma. Ejemplo: calcularPreciosVentas.

En caso de ser funcionalidades de la clase controladora se le pone el nombre y seguida la palabra "Action". Ejemplo: calcularPreciosVentasAction.

Nomenclatura de los comentarios

Es una necesidad comentar todo lo que se haga dentro del desarrollo, es decir, establecer las pautas que conlleven a lograr un código más legible y reutilizable, de manera que se pueda aumentar su mantenibilidad a lo largo del tiempo. Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se está desarrollando.

3.3 Diagrama de despliegue

El diagrama de despliegue es utilizado para modelar el hardware empleado en las implementaciones de sistemas y las relaciones entre sus componentes. Constituye un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El despliegue de los módulos Compra y Venta se rige por la especificación dada para el subsistema Logística. A continuación se muestran los posibles escenarios con los requerimientos de software.

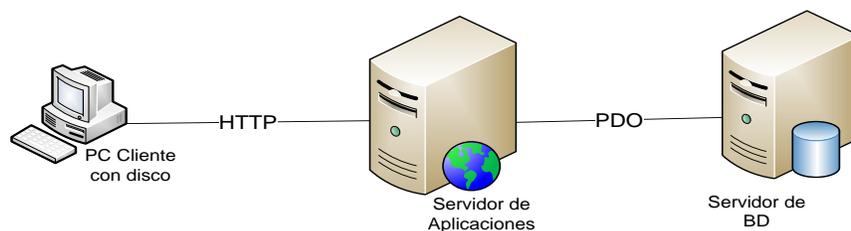


Figura 15. Diagrama de despliegue de escenario para PC cliente con disco

¹⁶ CamelCasing es un estilo de escritura que se aplica a frases o palabras compuestas. La nomenclatura está compuesta por tantas palabras como sean necesarias. La letra inicial del identificador irá en minúscula y la primera letra del resto de las palabras que lo forman irán siempre en mayúsculas y se obvia el uso de artículos.

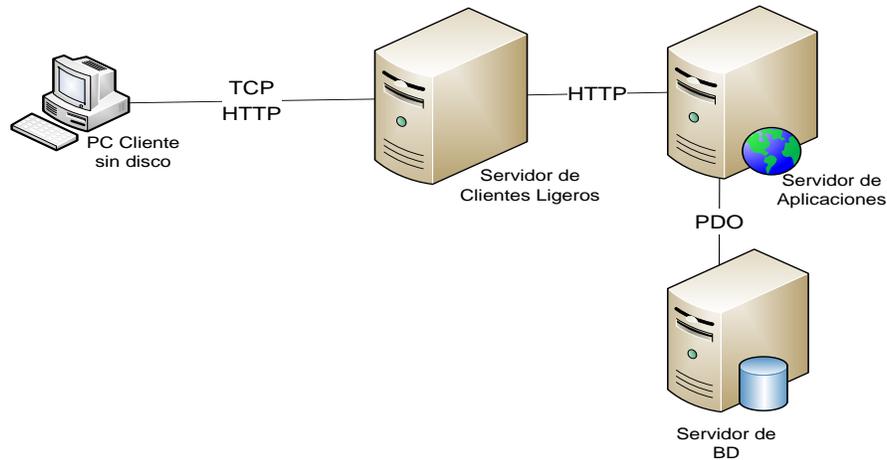


Figura 16. Diagrama de despliegue de escenario para PC cliente sin disco

Servidores

Servidor de Aplicaciones Web

- Sistema Operativo: Ubuntu Server
- Servidor Web: Apache 2.0
- Librerías Adicionales: PHP 5

Servidor de Base de Datos

- Sistema Operativo: Ubuntu Server
- Sistema Gestor de Base de Datos: PostgreSQL 8.3.8

Servidor de Clientes Ligeros

- Sistema Operativo: Nova Server
- Navegador Web: Mozilla Firefox 2.17 ó superior
- Herramientas Ofimáticas
- Visualizador de ficheros pdf
- Herramienta de administración de clientes ligeros

Clientes

PC Cliente con disco duro

- Sistema Operativo: Linux
- Navegador Web: Mozilla Firefox v2.2 ó superior
- Herramientas Ofimáticas
- Visualizador de ficheros pdf

PC Cliente sin disco duro

- Todo se instala en el servidor de clientes ligeros

3.4 Evaluación del diseño propuesto

Con el objetivo de comprobar cuán bien están definidas las clases, se emplean las métricas Tamaño operacional de la clase y Relaciones entre clases, diseñadas para evaluar los siguientes atributos de calidad:

Responsabilidad: Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto de la problemática propuesta.

Complejidad de implementación: Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.

Reutilización: Consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.

Acoplamiento: Consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras, está muy ligada a la característica de Reutilización.

Complejidad del mantenimiento: Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta pero fuertemente en los costes y la planificación del proyecto.

Cantidad de pruebas: Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (unidad) del producto (componente, módulo, clase, conjunto de clases, etc.) diseñado.

3.4.1 Métrica tamaño operacional de clases (TOC) propuesta por Lorenz y Kidd:

Las métricas orientadas a tamaño operacional de clases se centran en el cálculo de atributos y de operaciones para una clase individual. Este tamaño operacional de clases está dado por el número de métodos asignados a una clase.

Si existen valores grandes de TOC, estos mostrarán que una clase puede tener demasiadas responsabilidades, lo cual reducirá la reusabilidad de la clase y complicará la implementación. Por otra parte, cuanto menor sea el valor medio para el tamaño, más probable es que las clases existentes dentro del sistema se puedan reutilizar ampliamente. En la Tabla 5 se muestran los parámetros de calidad para valores grandes de TOC, propuestos por Lorenz. [52]

Capítulo 3. Implementación y pruebas de la solución propuesta

Tabla 5. Parámetros de calidad para valores grandes de TOC

Atributos de calidad que afecta	Valores Grandes de ToC
Responsabilidad	Implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Implica un aumento de la complejidad de implementación de la clase.
Reutilización	Implica una disminución en el grado de reutilización de la clase.

De forma general, operaciones y atributos deben ser ponderados al determinar el tamaño de la clase. Para valores pequeños de TOC para una clase existe mayor posibilidad de que la clase pueda ser reutilizada. [48]

Tabla 6. Umbrales para la TOC

Clasificación	Valores de los umbrales
Pequeño	\leq Promedio de operaciones (Prom)
Mediano	$>$ Prom y $\leq 2^*$ Prom
Grande	$> 2^*$ Prom

Para determinar el valor de los atributos a evaluar, se calcula el promedio de la cantidad de procedimientos de cada clase del sistema. Este promedio (en este caso el resultado obtenido fue 3.96) es el que se emplea en la Tabla 6 en la columna valores de umbrales para clasificar cada una de las clases.

La gráfica que se muestran a continuación muestran los resultados obtenidos para cada uno de los atributos medidos.

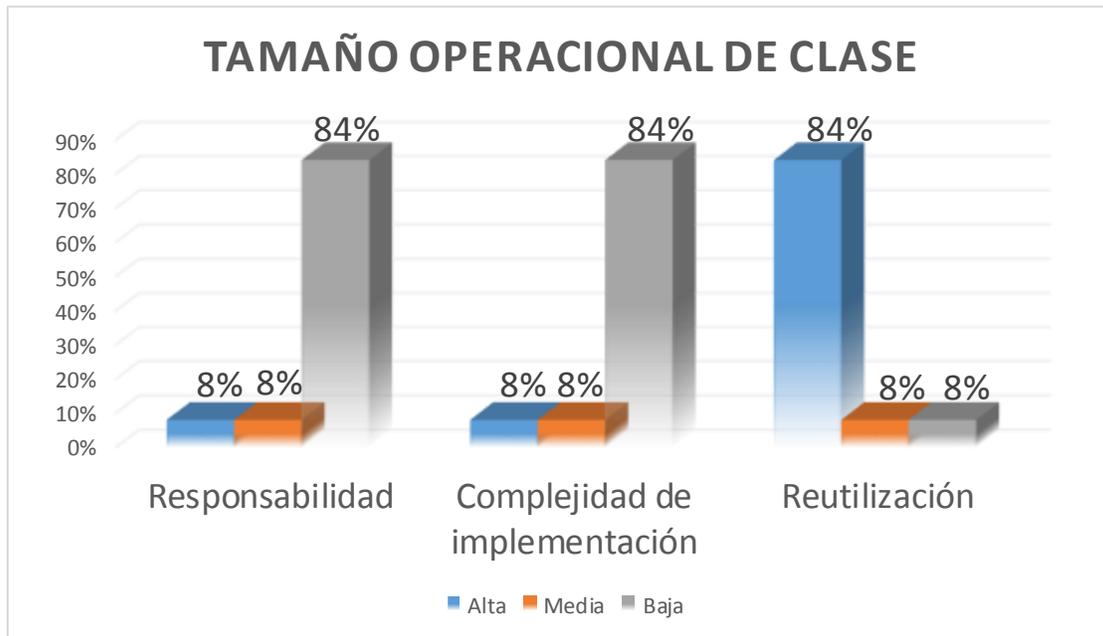


Figura 17. Representación en % de los resultados de la evaluación de la métrica TOC

Tras un análisis de los resultados obtenidos aplicando la métrica TOC en la evaluación del instrumento para el atributo Responsabilidad, se observó que la mayoría de las clases que conforman el sistema poseen una baja responsabilidad con un 84% del total, lo que promueve el bajo acoplamiento entre las clases y facilitará la implementación. Para el atributo Complejidad de implementación se obtuvo un 84% de clases con baja complejidad, característica que permite mejorar el mantenimiento y soporte de estas clases. El resultado de la incidencia de la métrica TOC para el atributo Reutilización arrojó un 84% de las clases con alta reutilización, potenciando el bajo acoplamiento y disminuyendo la dependencia entre clases. Lo anterior planteado muestra que los módulos cuentan con una elevada reutilización, baja complejidad y responsabilidad en el diseño propuesto, por lo que se concluye que los resultados obtenidos según esta métrica son positivos.

3.4.2 Métrica relaciones entre clases (RC)

Está dado por el número de relaciones de uso de una clase con otras. En la Tabla 7 se muestran los parámetros de calidad para valores grandes de RC, propuestos por Lorenz. [52]

Tabla 7. Parámetros de calidad para valores grandes de RC.

Atributos de calidad que afecta	Valores Grandes de RC
Acoplamiento	Implica un aumento del acoplamiento de la clase.

Capítulo 3. Implementación y pruebas de la solución propuesta

Complejidad de mantenimiento	Implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 8. Umbrales para RC.

Atributos de calidad	Categoría	Valores de umbrales
Acoplamiento	Ninguna	0
	Baja	1
	Media	2
	Alta	>2
Complejidad de mantenimiento	Baja	\leq Promedio (Prom)
	Media	$>$ Prom y $\leq 2^* \text{ Prom}$
	Alta	$> 2^* \text{ Prom}$
Reutilización	Baja	$> 2^* \text{ Prom}$
	Media	$>$ Prom y $\leq 2^* \text{ Prom}$
	Alta	\leq Promedio
Cantidad de pruebas	Baja	\leq Promedio
	Media	$>$ Prom y $\leq 2^* \text{ Prom}$
	Alta	$> 2^* \text{ Prom}$

Para determinar el valor de los atributos, se calcula el promedio de la columna cantidad de relaciones de uso y este promedio (en este caso el resultado obtenido fue 4.5) es el que se emplea en la Tabla 8 en la columna valores de umbrales para clasificar cada una de las clases.

Las gráficas que se muestran a continuación muestran los resultados obtenidos para cada uno de los atributos medidos.

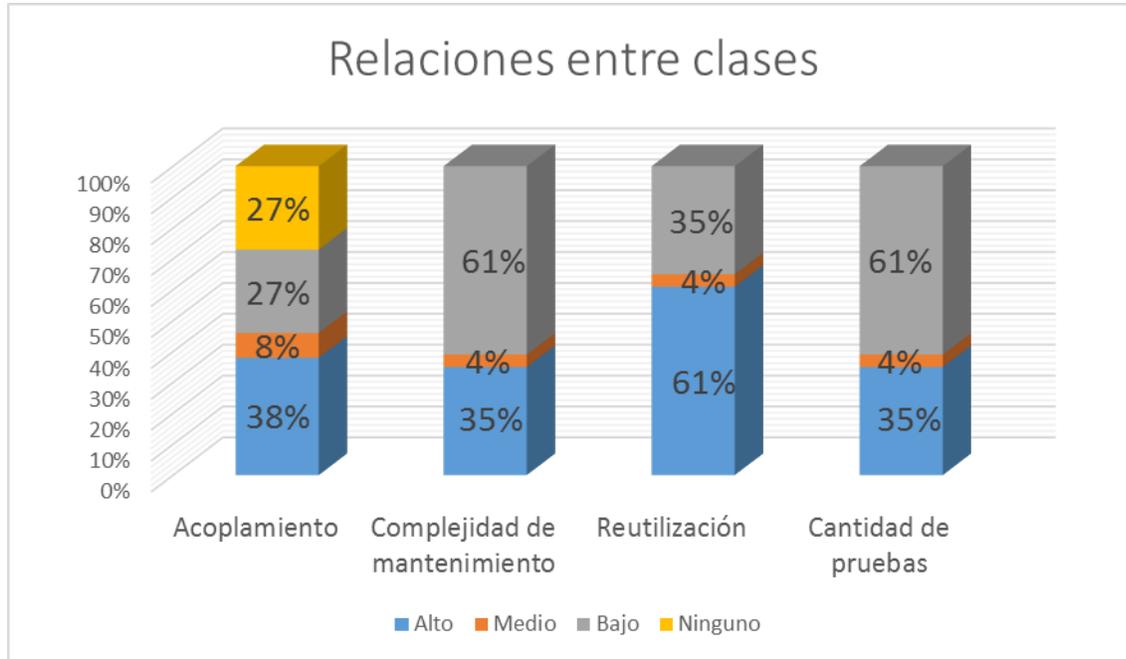


Figura 18. Representación en % de los resultados de la evaluación de la métrica RC

Tras un análisis de los resultados obtenidos aplicando la métrica RC en la evaluación del instrumento para el atributo Acoplamiento se concluyó que las clases del diseño poseen un bajo acoplamiento, pues para este atributo las categorías ninguno y bajo sumaron un 54% del total, aumentando el grado de reutilización de estas clases. Los atributos complejidad de mantenimiento y cantidad de pruebas, sumaron un 65 % en las categorías baja y media, lo que demuestra que no es necesario un elevado esfuerzo en el momento de realizar cambios, rectificaciones y pruebas al software. Para el atributo de reutilización la métrica arrojó un resultado de un 65% del total de las clases entre las categorías baja y media, constituyendo un factor importante en el desarrollo del software. La utilización de más de 15 servicios para dar cumplimiento a solo 26 requisitos funcionales podía atender contra la correcta aplicación de los atributos que evalúa la métrica RC en la solución. A pesar de esto, los resultados obtenidos se encuentran todos por encima del 50%, por lo cual se consideran favorables.

3.5 Pruebas de software

Las pruebas de software no garantizan que un sistema esté libre de errores, sino que se detecten la mayor cantidad de defectos posibles para su debida corrección. Estas son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa, probando el comportamiento del

Capítulo 3. Implementación y pruebas de la solución propuesta

componente. Permiten determinar la calidad del producto, detectar todo posible mal funcionamiento y comprobar el grado de cumplimiento de las especificaciones iniciales del sistema. Se considera una prueba exitosa, si se demuestran deficiencias en el software. [53]

3.5.1 Aplicación de las Pruebas de caja negra

Para realizarlas no es necesario conocer los detalles internos del programa y su objetivo fundamental es probar que el software está acorde a los requisitos. [53]

Para la ejecución de este tipo de pruebas se realizaron un conjunto de casos de prueba que tenían como objetivo principal determinar si los requisitos estaban parcial o completamente cumplidos. Los casos de prueba elaborados para los requisitos funcionales se encuentran en el expediente de proyecto de XEDRO-ERP. A continuación se representa el diseño de caso de prueba para el requisito funcional Adicionar solicitud de compra (Ver Tabla 9, Tabla 10 y Tabla 11).

Descripción general

Se adiciona una nueva solicitud de compra con los atributos: número, fecha, proveedor, producto y estado.

Condiciones de ejecución

- Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- Se debe seleccionar el subsistema Logística.
- Se debe seleccionar la opción Inventario/Compras/Solicitud de compras.
- Se debe seleccionar la opción Adicionar.

Tabla 9. Posibles escenarios al adicionar una solicitud de compra.

Nombre del requisito	Escenarios de pruebas	Flujo del escenario
1: Adicionar Solicitud de compra.	EP 1.1: Adicionar una nueva solicitud de compra, introduciendo datos válidos.	<ul style="list-style-type: none">– Se seleccionan y/o introducen los datos de una solicitud de compra correctamente.– Se presiona el botón Aceptar.

Capítulo 3. Implementación y pruebas de la solución propuesta

		<ul style="list-style-type: none"> - Se muestra un mensaje de confirmación. - Se presiona el botón Aceptar.
	EP 1.2: Adicionar una nueva solicitud de compra, introduciendo datos válidos presionando el botón Aplicar .	<ul style="list-style-type: none"> - Se seleccionan y/o introducen los datos de una solicitud de compra correctamente. - Se presiona el botón Aplicar. - Se muestra un mensaje de información. - Se presiona el botón Aceptar.
	EP 1.3: Adicionar una nueva solicitud de compra, introduciendo datos inválidos.	<ul style="list-style-type: none"> - Se introducen y/o seleccionan los datos dejando algún campo en blanco. - Se presiona el botón Aceptar. - Se muestra un mensaje informando del error. - Se presiona el botón Aceptar.
	EP 1.4: Cancelar	<ul style="list-style-type: none"> - Se introducen o no los datos de la solicitud de compra. - Se presiona el botón Cancelar.

Tabla 10. Descripción de las variables del Adicionar solicitud de compra.

No	Nombre de campo	Tipo	Válido	Inválido	Inválido
1	Suministrador	Campo de selección (No editable).	Letras y números	Vacío	N/A

Capítulo 3. Implementación y pruebas de la solución propuesta

2	Fecha	Campo de selección (No editable).	de Letras y números	Vacío	N/A
3	Depósito	Campo de selección (No editable).	de Letras y números	Vacío	N/A

Tabla 11. Juego de datos a probar del Adicionar solicitud de compra.

Id del escenario	Escenario	Suministrador	Fecha	Depósito	Respuesta del sistema
EP 1.1	Adicionar una nueva solicitud de compra, introduciendo datos válidos.	V (CONAVANA)	V(28/05/2014)	V(Almacén 1)	Se adiciona la solicitud de compra. Se muestra un mensaje informando que “El documento ha sido adicionado satisfactoriamente.”. El sistema cierra la interfaz.
EP 1.2:	Adicionar una nueva solicitud de compra, introduciendo datos válidos presionando el botón	V (UCI)	V(01/06/2014)	V(Almacén Caribe)	Se adiciona la solicitud de compra. Se muestra un mensaje informando que “El documento ha sido adicionado satisfactoriamente.”. El sistema mantiene abierta la interfaz.
EP 1.3	Adicionar una nueva solicitud de compra, introduciendo datos inválidos.	V (Vacío)	V(28/05/2014)	V(Almacén 1)	El sistema subraya el campo vacío mostrando un mensaje de error: “Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s).”.
		V (CONAVANA)	V(Vacío)	V(Almacén 1)	

Capítulo 3. Implementación y pruebas de la solución propuesta

		V (CONAVANA)	V(28/05/2014)	V(Vacío)	
EP1.4	Cancelar	N/A	N/A	N/A	Se cancela la operación. El sistema cierra la interfaz sin realizar ninguna operación.

Resultados de las pruebas.

Los módulos fueron sometidos a tres iteraciones de pruebas de caja negra realizadas por el grupo de calidad de CEIGE. En la primera iteración se detectaron 27 no conformidades, de las cuales 22 fueron de aplicación y 5 de documentación. Dentro de las no conformidades referentes a las de aplicación se encuentran 12 de validación, 8 de interfaz y 2 de funcionalidad. En cuanto a las de documentación se encontraron 3 no conformidades de redacción, 1 de ortografía y 1 de correspondencia. En la segunda iteración se detectaron 10 no conformidades, de las cuales 8 fueron de aplicación y 2 de documentación. Dentro de las no conformidades de aplicación se encuentran 6 de validación y 2 de interfaz. En cuanto a las no conformidades de documentación se encontraron 1 de redacción y 1 de concordancia. Todas estas no conformidades fueron resueltas seguidamente de haberlas detectado, lo que permitió que el componente pasara a una tercera iteración en la que no se detectaron no conformidades. Una vez concluido el proceso de pruebas funcionales se obtuvo una solución libre de no conformidades y con un grado de calidad aceptable. Prueba de ello lo constituye el acta de liberación expedida por el grupo de calidad de CEIGE (Ver Anexo 8).

3.5.2 Validación de la solución

Los módulos desarrollados fueron presentados además a la Analísta principal del CEIGE para realizar la aceptación de los mismos, certificando la adecuación de las funcionalidades implementadas para la realización de parte de los subprocesos Distribución y reabastecimiento y Comercio minorista. Los subprocesos antes mencionados constituyen dos de los subprocesos identificados en el documento Visión-Anexo # 1 del proyecto ERP-Cuba para el producto XEDRO-ERP, como parte de la Gestión comercial que persigue para su segunda fase. Como consta en el Acta de aceptación que se refleja en el Anexo 9, la solución de los

Capítulo 3. Implementación y pruebas de la solución propuesta

módulos desarrollados aporta en al menos 25% al sistema para la ejecución de los subprocesos mencionados anteriormente de la siguiente manera:

- Compra implementa aproximadamente el 35% del subproceso Distribución y reabastecimiento mediante funcionalidades para gestionar las solicitudes de compra y generar informes de recepción a partir de las anteriores.
- Ventas implementa aproximadamente el 25% del subproceso Comercio minorista mediante funcionalidades para realizar el cálculo de los precios de ventas y sus respectivos desgloses de precios.

Lo planteado anteriormente confirma que los módulos desarrollados contribuyen en un alto grado a la Gestión comercial reflejada en la Visión del proyecto, dando cumplimiento al objetivo general planteado en la investigación. Además, actualmente no se cuenta con ningún componente que haga estas funcionalidades del área de Gestión comercial en el sistema XEDRO-ERP.

3.6 Conclusiones del capítulo

Una vez concluido el proceso de implementación y pruebas de la solución se obtuvieron las siguientes conclusiones:

- La implementación de los módulos de Compra y Venta permitió contar con una solución capaz de dar respuesta a la problemática planteada.
- El diagrama de despliegue permitió identificar dos posibles escenarios en los cuales puede ser desplegado el sistema, así como los requerimientos de software necesarios para su correcto funcionamiento.
- La validación del diseño a través de métricas arrojó resultados satisfactorios sobre el diseño realizado, demostrando que este era simple y los atributos de calidad alcanzaban niveles favorables.
- La realización de pruebas de caja negra y aceptación a los módulos arrojaron resultados satisfactorios, demostrándose con ello la correcta implementación de los requisitos funcionales definidos.

Conclusiones generales

Una vez terminado el presente trabajo de diploma se puede concluir que se desarrollaron todas las tareas a fin de cumplir los objetivos propuestos, resaltando que:

- El estudio de los sistemas de gestión empresarial revisados, fundamenta la necesidad de desarrollar nuevos módulos que se adapten a los requerimientos del producto XEDRO-ERP y sean consecuentes con las políticas del país.
- La modelación del negocio de los procesos de Compra y Configuración de precios permitió su comprensión para su posterior desarrollo e integración al sistema XEDRO-ERP.
- El diseño de la estructura de componentes, clases y datos garantizó una comprensión exacta sobre las funcionalidades a implementar.
- La implementación de los módulos Compra y Ventas permitió contar con una solución que se integra al sistema XEDRO-ERP para dar respuesta al problema planteado.
- La realización de pruebas de caja negra a los módulos arrojaron resultados satisfactorios, demostrándose con ello la correcta implementación de los requisitos funcionales definidos.
- Con la prueba de aceptación aplicada se pudo comprobar que la solución contribuye a la gestión comercial en el sistema XEDRO-ERP, adaptándose a las necesidades identificadas y dando cumplimiento al objetivo general de la presente investigación.

Recomendaciones

Para futuras mejoras de la solución se recomienda:

- Agregar funcionalidades al módulo Ventas que permitan el cálculo de las tarifas de los servicios prestados por las entidades, guiándose por las regulaciones y normas del MFP.
- Realizar el despliegue de los módulos propuestos como parte del subsistema Logística de XEDRO-ERP.

Bibliografía

- [1]. Todo sobre la gestión y administración empresarial. [En línea]. Disponible en: <http://www.gestionyadministracion.com/>. [Accedido: 15-feb-2014].
- [2]. Leenders, Michiel R. Fearon, Harold E. England, Wilbur. Purchasing and Materials Management. México, D. F. MX. 2 ed.Continental. 1997.
- [3] Compras | GestioPolis. [En línea]. Disponible en:
- [4]. Gestión De Ventas | Portal web sobre OpenERP en España | Información, noticias y eventos. [En línea]. Disponible en: <http://openerpspain.com/tag/gestion-de-ventas/>. [Accedido: 17-feb-2014].
- [5]. DEFINICIÓN DE VENTA - Promonegocios.net. [En línea]. Disponible en: <http://www.promonegocios.net/mercadotecnia/definicion-concepto-venta.htm>. [Accedido: 15-feb-2014].
- [6]. Definición de Cliente. Concepto en Definición ABC. [En línea]. Disponible en: <http://www.definicionabc.com/general/cliente.php>. [Accedido: 17-feb-2014].
- [7]. DEFINICIÓN DE CLIENTE - Promonegocios.net. [En línea]. Disponible en: <http://www.promonegocios.net/clientes/cliente-definicion.html>. [Accedido: 17-feb-2014].
- [8]. PRODUCTO - Enciclopedia de Economía. [En línea]. Disponible en: <http://www.economia48.com/spa/d/producto/producto.htm>. [Accedido: 17-feb-2014].
- [9]. DEFINICIÓN DE PRODUCTO - Promonegocios.net. [En línea]. Disponible en: <http://www.promonegocios.net/mercadotecnia/producto-definicion-concepto.html>. [Accedido: 17-feb-2014].
- [10]. Definición de proveedor - Qué es, Significado y Concepto. [En línea]. Disponible en: <http://definicion.de/proveedor/>. [Accedido: 17-feb-2014].
- [11]. Definición de Proveedor. Concepto en Definición ABC. [En línea]. Disponible en: <http://www.definicionabc.com/tecnologia/proveedor.php>. [Accedido: 17-feb-2014].
- [12]. Qué es una factura [En línea]. Disponible en: <http://www.misrespuestas.com/que-es-una-factura.html>. [Accedido: 18-feb-2014].
- [13]. DEFINICION DE PRECIO - Promonegocios.net. [En línea]. Disponible en: <http://www.promonegocios.net/mercadotecnia/precio-definicion-concepto.html>. [Accedido: 18-feb-2014].
- [14]. Precio | Gerencie.com. [En línea]. Disponible en: <http://www.gerencie.com/precio.html>. [Accedido: 18-feb-2014].

- [15]. Definición de precio de compra. [En línea]. Disponible en: <http://www.diclib.com/precio%20de%20compra/show/es/alkonaeconomia/P/732/360/0/0/4853>. [Accedido: 18-feb-2014].
- [16]. MODULOS | OpenERP: ERP Open Source / Software Libre. [En línea]. Disponible en: <http://www.openerpsite.com/erp-openerp-modulos/>. [Accedido: 20-feb-2014].
- [17]. Características de la Plataforma ERP de Openbravo | Openbravo. [En línea]. Disponible en: <http://www.openbravo.com/es/features-erp-platform>. [Accedido: 20-feb-2014].
- [18]. Openbravo, Funcionalidad de software ERP - Mintecsys. [En línea]. Disponible en: <http://mintecsys.com/funcionalidad/>. [Accedido: 20-feb-2014].
- [19]. De Pedido al Cobro | Características de la Plataforma ERP de Openbravo | Openbravo. [En línea]. Disponible en: <http://www.openbravo.com/es/features-erp-platform-order-to-cash>. [Accedido: 20-feb-2014].
- [20]. Sap Bussines One. [En línea]. Disponible en: <http://www.mgssoft.com/pdf/sap/Optimice%20su%20empresa%20acelere%20el%20crecimiento%20rentable.pdf>
- [21]. Forum.villaclara.cu - Inicio. [En línea]. Disponible en: <http://www.forum.villaclara.cu/>. [Accedido: 21-feb-2014].
- [22]. Rodas XXI. Sistema Integral Económico Administrativo. [En línea]. Disponible en: <http://www.rodasxxi.cu/>. [Accedido: 21-feb-2014].
- [23]. Facturación. [En línea]. Disponible en: <http://www.rodasxxi.cu/facturacion.php>. [Accedido: 21-feb-2014].
- [24]. ASSETS: Sistema de Gestión Integral. [En línea]. Disponible en: <http://www.assets.co.cu/assets.asp>. [Accedido: 21-feb-2014].
- [25]. Tamara Rodríguez Sánchez, Mairelys Fernández González, Eliecer Cabrera Casas. Revista Infociencia Vol.15, No.2, abril-junio, 2011. La gestión empresarial de las entidades cubanas. Cedrux a la vuelta de la esquina. Extraído el 22 de febrero del 2014 desde <http://www.infociencia.idict.cu/infociencia/article/download/16/12?.pdf>.
- [26]. William González Obregón. Modelo de Desarrollo de Software del CEIGE v1.2.
- [27]. Oiner Gómez Baryolo, y otros. Plantilla Registro de la Propiedad intelectual (Sauxe). Habana: s.n., 2008.
- [28]. Gómez Baryolo, Oiner, Morejón Borbón, Yoandry y García Tejo, Darien. 2012. Arquitectura tecnológica para el desarrollo de software. 2012.

- [29]. ZendFramework. [En línea]. Disponible en: <http://framework.zend.com/about/>. [Accedido: 25-feb-2014].
- [30]. Home — Doctrine Project. [En línea]. Disponible en: <http://www.doctrine-project.org/>. [Accedido: 25-feb-2014].
- [31] Lenguajes de programación - Investigaciones - Utria93. [En línea]. Disponible en: <http://www.buenastareas.com/ensayos/Lenguajes-De-Programacion/246119.html>. [Accedido: 25-feb-2014].
- [32]. Flanagan, David. Java Script. The definitive guide.
- [33]. Home — PHP.Net. [En línea]. Disponible en: <http://www.php.net/manual/es/intro-what-is.php>. [Accedido: 25-feb-2014].
- [34]. Andrés, Rolando Rodríguez. 2008. Lenguajes, notaciones y herramientas para el Modelado. 2008.
- [35]. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. El proceso Unificado de desarrollo de software. 2000.
- [36]. Visual Paradigm. [En línea]. Disponible en: <http://www.visual-paradigm.com/product/vpuml/>. [Accedido: 27-feb-2014].
- [37]. Subversion - doc.ubuntu-es. [En línea]. Disponible en: <http://doc.ubuntu-es.org/Subversion>. [Accedido: 27-feb-2014].
- [38]. Net Beans. [En línea]. Disponible en: https://netbeans.org/community/releases/61/index_es.html. [Accedido: 27-feb-2014].
- [39]. Sobre PostgreSQL | www.postgresql.org.es. [En línea]. Disponible en: http://www.postgresql.org.es/sobre_postgresql. [Accedido: 28-feb-2014].
- [40]. About the Apache HTTP Server Project - The Apache HTTP Server Project. [En línea]. Disponible en: http://httpd.apache.org/ABOUT_APACHE.html. [Accedido: 28-feb-2014].
- [41]. Inicio del Proyecto Mozilla — Mozilla. [En línea]. Disponible en: <http://www.mozilla.org/es-ES/>. [Accedido: 28-feb-2014].
- [42]. Oiner Gómez Baryolo. Solución informática de autorización en entornos multientidad y multisistema. 2010.
Universidad de las Ciencias Informáticas. La Habana: s.n., 2010. Tesis de Maestría.
- [43] ¿Qué es un Patrón de Diseño? [En línea]. Disponible en: <http://msdn.microsoft.com/es-es/library/bb972240.aspx>. [Accedido: 03-mar-2014].