

Universidad de las Ciencias Informáticas
Facultad 3



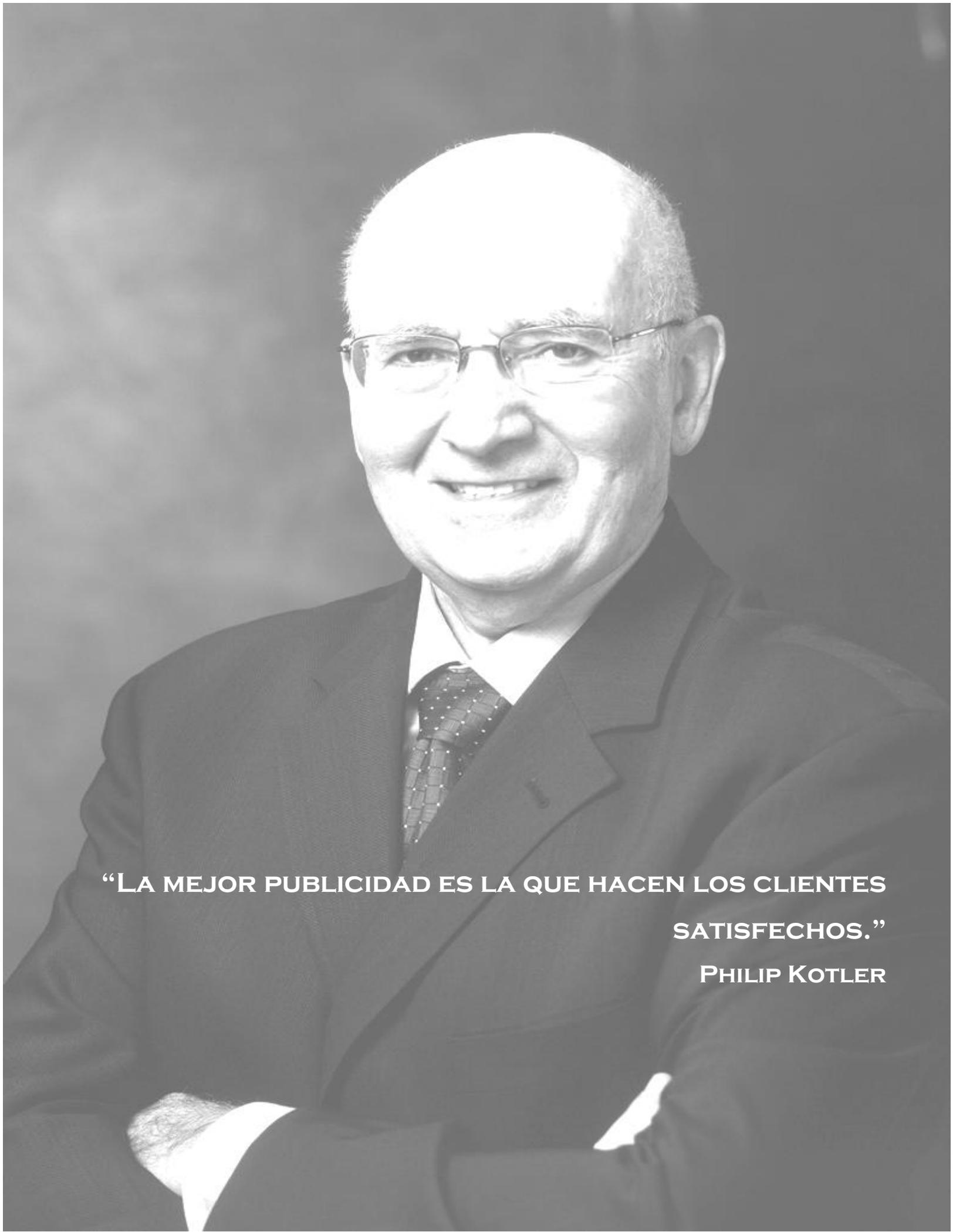
**Título: Análisis y diseño del sistema para la gestión del portafolio de
productos del CEIGE**

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

Autora: Laura Rodríguez Leveque

Tutoras: Ing. Leidy Ramos González

Ing. Ileana Centelles Rubiera



**“LA MEJOR PUBLICIDAD ES LA QUE HACEN LOS CLIENTES
SATISFECHOS.”**

PHILIP KOTLER

DECLARACIÓN DE AUTORÍA

Declaro que yo, Laura Rodríguez Leveque, con número de identidad 88110110171, soy el único autor de este trabajo y autorizo al Centro de Informatización de la Gestión de Entidades de la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con este trabajo.

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

Laura Rodríguez Leveque

Firma del Autor

Ing. Leidy Ramos González

Firma del Tutor

Ing. Ileana Centelles Rubiera

Firma del Tutor

Agradecimientos

Quiero agradecer a mi familia, la principal motivación de este trabajo. A mi madre por darme la vida y apoyarme siempre aunque muchas veces no estuviera de acuerdo conmigo. A mi abuela porque me ha enseñado a ser la persona que soy hoy, por su constancia y perseverancia. A mi padre, que aunque no hemos compartido muchos momentos juntos, siempre está pensando en mí y apoyándome. A Irene por cuidar de mi padre y estar a su lado en todo momento.

A Gabriela, que me ha demostrado que los amigos verdaderos existen y no solo para los momentos buenos, sino en los malos también. Ella ha hecho suya esta batalla y la ha peleado junto a mí, este triunfo también es tuyo.

A mi pareja, por su fidelidad incondicional, por aguantar mis malcriadeces y majaderías. Por estar conmigo aún cuando todo conducía al fracaso.

A todos mis compañeros de carrera, gracias por haberme dejado entrar en sus vidas y por haber pasado por la mía, dejándome lo mejor de cada uno.

A Magdanis y Sasha que me ayudaron en la recta final para que pudiera terminar la tesis.

A mi Tata, que donde quiera que esté e cuida siempre, este triunfo es para ti.

A todas las personas que creyeron en mí gracias y a los que no también porque gracias a ellos pude probarme a mí misma que si podía vencer.

A todos mis vecinos, amigos de la familia, gracias por el apoyo y la preocupación.

Dedicatoria

Dedico esta tesis a mi familia, la de sangre y la de sentimiento. Ustedes son el impulso de mi vida. Gracias por existir y por apoyarme siempre

RESUMEN

Con el surgimiento de la industria de bienes y servicios, la producción se llevaba a cabo de manera artesanal y sólo satisfacía las necesidades de un cliente específico. Con el paso de los años los avances tecnológicos propiciaron un cambio hacia la producción en masa, permitiendo la creación de una mayor cantidad de productos para satisfacer las demandas de un amplio número de clientes.

Las empresas dedicadas a la comercialización tanto de bienes como servicios cuentan con mecanismos para ofertar sus productos; estos pueden ser sitios web como las tiendas virtuales, propagandas tanto radiales, televisivas, de divulgación digital o en formato duro y también existe el portafolio de productos que organiza toda la información y la divide por categorías. El portafolio es una nueva herramienta que ayuda en el análisis y la toma de decisiones. Además su gestión ofrece nuevas oportunidades empresariales y de mercado al combinar la información existente de manera que se pueden separar los productos de éxito de los que fracasan.

Este trabajo propone un sistema que facilite la identificación de productos desde el portafolio de los productos que se desarrollan en el Centro de Informatización de Entidades (CEIGE) mediante la administración de sus características. Con este fin se lleva a cabo el análisis y diseño que incluye entender el dominio de la información del problema, definir las funciones que debe efectuar el sistema y confeccionar los modelos que describan la información, funciones y comportamiento de cada elemento de la solución que se plantea.

Palabras claves: administración de características, análisis de dominio, portafolio de productos.

ÍNDICE DE CONTENIDOS

Resumen	6
introducción	1
capítulo 1. Fundamentación teórica	5
introducción.....	5
1.1. Portafolio de productos	5
1.2. Sistemas para la gestión del portafolio	6
1.3. Línea de productos de software (lps)	9
1.4. Modelos de características	10
1.5. Análisis automatizado de modelos de características	12
1.6. Herramientas para el análisis automatizado de modelos de características.....	13
1.7. Herramienta a emplear en la solución.....	15
1.8. Modelo de desarrollo	16
1.9. Ingeniería de requisitos.....	16
1.10. Patrones empleados para el diseño de la solución.....	20
1.11. Métricas de validación del diseño.....	21
1.12. Lenguaje de modelado.....	23
1.13. Notación de modelado de procesos de negocio (bpmn).....	23
1.14. Herramienta de modelado	23
1.15. Herramienta para diseño de interfaces de usuario	24
1.16. Marco de trabajo	24
conclusiones parciales	24
capítulo 2. Propuesta de solución	26
introducción.....	26
2.1. Proceso de negocio	26
2.3. Modelo conceptual	29
2.4. Requerimientos funcionales	32
2.5. Diagrama de clases del diseño	44
2.6. Modelo de datos	47
conclusiones parciales	48
capítulo 3. Validación de la propuesta.....	49
introducción.....	49

3.1. Validación de requisitos	49
3.2. Validación del diseño	50
3.3. Aplicación de la métrica toc	50
3.4. Aplicación de la métrica rc	53
3.5. Resultado de la evaluación de las métricas	58
Conclusiones parciales	58
Conclusiones generales.....	60
Recomendaciones	62
Referencias bibliográficas	63
Glosario de términos.....	65
Anexos.....	68
Anexo 1. Modelo de características básico	68
Anexo 2. Modelo de características con cardinalidad y multiplicidad.....	68
Anexo 3. Modelo de características extendido	68
Anexo 6. Diagrama de clases realizar reporte de características más comunes	69
Anexo 7. Diagrama de clases realizar reporte de productos vendidos	69
Anexo 8. Diagrama de clases realizar reporte de productos solicitados	70
Anexo 9. Diagrama de clases realizar reporte de mejores clientes	70
Anexo 10. Complejidad de los requisitos.....	71

INDICE DE TABLAS

Tabla I. Comparación entre sistemas de gestión de portafolio.....	7
Tabla II. Herramientas para el análisis automatizado de modelos de características.....	15
Tabla III. Rango de valores para evaluación de la métrica TOC.....	22
Tabla IV. Rango de valores para la evaluación de la métrica RC.....	22
Tabla V. Descripción del procesos de negocio Comercializar productos.....	26
Tabla VI. Descripción de las clases conceptuales.....	30
Tabla VII. Descripción textual del requisito Crear perfil de cliente.....	35
Tabla VIII. Descripción textual del requisito Adicionar solicitud.....	38
Tabla IX. Descripción textual del requisito Realizar reporte de solicitudes pendientes.....	42
Tabla IX. Descripción textual del requisito Integrar FaMa.....	43
Tabla XI. Aplicación de las métricas de especificación de requisitos.....	49
Tabla XI. Aplicación de la métrica TOC.....	50
Tabla XII. Resultados del atributo Responsabilidad de la métrica TOC.....	50
Tabla XIII. Resultados del atributo Complejidad de implementación de la métrica TOC.....	51
Tabla XIV. Resultados del atributo Reutilización de la métrica TOC.....	52
Tabla XV. Aplicación de la métrica RC.....	53
Tabla XVI. Resultados del atributo Acoplamiento de la métrica RC.....	54
Tabla XVII. Resultados del atributo Cantidad de pruebas de la métrica RC.....	55
Tabla XVIII. Resultados del atributo Complejidad de implementación de la métrica RC.....	56
Tabla XIX. Resultados del atributo Reutilización de la métrica RC.....	57

INDICE DE FIGURAS

Figura 1. Estructura del archivo del modelo extendido.....	16
Figura 2. Diagrama bpmn del proceso de negocio comercializar producto.....	29
Figura 3. Modelo conceptual.....	30
Figura 4. Prototipo de interfaz de usuario crear perfil de cliente.....	38
Figura 5. Diagrama de clases del diseño gestionar perfiles de clientes.....	45
Figura 6. Diagrama de clases del diseño gestionar solicitudes.....	46
Figura 7. Modelo de datos.....	47
Figura 8. Gráfico del atributo responsabilidad de la métrica toc.....	51
Figura 9. Gráfico del atributo complejidad de implementación de la métrica toc.....	52
Figura 10. Gráfico del atributo reutilización de la métrica toc.....	53
Figura 11. Gráfico del atributo acoplamiento de la métrica rc.....	55
Figura 12. Gráfico del atributo cantidad de pruebas de la métrica rc.....	56
Figura 13. Gráfico del atributo complejidad de implementación de la métrica rc.....	57
Figura 14. Gráfico del atributo reutilización de la métrica rc.....	58

INTRODUCCIÓN

El proceso de producción de bienes y servicios ha experimentado cambios a lo largo de su desarrollo. La fabricación artesanal de un producto para un cliente específico, ha evolucionado hacia una producción masiva para satisfacer las necesidades de un mayor número de clientes. Varios sectores de la industria adoptan esta forma de producción, incluyendo la industria del software.

Las empresas dedicadas a la comercialización de bienes y servicios utilizan propagandas, catálogos y portafolios para ofertar sus productos. Los portafolios constituyen una nueva herramienta de la que disponen las empresas comercializadoras para el análisis y la toma de decisiones. En la creación de portafolios se emplean las técnicas de Líneas de Productos de Software (LPS) que consisten en organizar los productos de manera tal que todos aquellos que comparten características similares se encuentren agrupados.

La Industria Cubana del Software no se encuentra ajena a los cambios ocurridos a nivel mundial en cuanto a la forma de producción y comercialización, evidenciándose en la Universidad de Ciencias Informáticas (UCI), donde el desarrollo de soluciones y la comercialización de las mismas forman parte de su misión productiva. La UCI cuenta con centros de desarrollo como el Centro de Informatización de la Gestión de Entidades (CEIGE). La función fundamental del centro es la producción de software para automatizar procesos que se realizan en diferentes empresas del país como la Aduana General de la República, el Banco Central de Cuba, entre otras. También brinda servicios de asesoría a entidades nacionales y extranjeras en torno a la definición, adopción de procesos y soluciones para su gestión integral. El Centro busca oportunidades para potenciar el desarrollo de sistemas y contribuir a la independencia tecnológica del país mediante el desarrollo de software libre.

Debido a las oportunidades que brindan las LPS, el CEIGE se encamina a la adopción de estos beneficios, orientando su estructura actual hacia una migración progresiva hacia esta forma de producción. Actualmente, cada línea elabora sus productos de forma independiente lo que ocasiona que los activos de software (código fuente, interfaces o funcionalidades) estén duplicados, evidenciándose un bajo nivel de reutilización; por ejemplo, en el proyecto Aduana se desea implementar un módulo para gestionar recursos humanos e incluirlo en el sistema GINA. El producto Cedrux, del proyecto ERP, ya posee este módulo, pero al no existir interrelación entre los proyectos, Aduana tiene que desarrollar el módulo desde cero en lugar de reutilizar el que tiene Cedrux. El CEIGE posee un sistema de información

de mercadotecnia que contiene un módulo de registros internos, que almacena los datos sobre los productos desarrollados, pero no posibilita el análisis de los componentes que pueden ser reutilizados, lo que aumenta el tiempo de respuesta a las solicitudes de los clientes, disminuye la calidad del servicio y el nivel de comercialización.

De la misma manera, se introducen los conceptos del análisis de dominio, con el objetivo de identificar activos de software que puedan ser creados una vez y reutilizados tantas veces como se necesite. Al conformar productos con partes de software, el portafolio de productos no solamente se compone de productos completos, sino de varios activos que de conjunto pueden significar un valor de uso a un cliente, como documentación de un negocio, datos de carga inicial, datos y estrategias de prueba, script de bases de datos, entre otros.

Actualmente el proceso de comercialización se ejecuta de manera presencial, exponiéndole al cliente las soluciones desarrolladas, sin permitir que el mismo personalice en tiempo real, o tenga la certeza de un conjunto de funcionalidades desarrolladas en un breve tiempo para la configuración y ensamblaje de sus productos, a pesar de que los productos que se desarrollan son altamente configurables y personalizables. En el mercado actual, lograr que el cliente pueda acceder a sus necesidades con un nivel de certeza confiable, gana espacio en la comercialización y oportunidades de venta de una organización.

Lo anteriormente expuesto permite definir como **problema a resolver**: ¿cómo contribuir a la identificación de productos por parte del cliente en el portafolio de productos del CEIGE?

Se plantea como **objeto de estudio**: el proceso de gestión del portafolio de una organización mediante las técnicas de análisis de dominio.

Como **objetivo general**: realizar el análisis y diseño del sistema de gestión del portafolio de productos del CEIGE a partir del análisis de dominio basado en características.

El **campo de acción** se enmarca en el proceso de identificación de productos de una organización mediante técnicas de análisis de dominio basado en características.

Como **objetivos específicos** se definieron:

- Elaborar el marco teórico de la investigación que permita conocer los elementos que componen la gestión del portafolio de productos y su relación con el análisis de dominio.
- Realizar la ingeniería de requisitos para el sistema de gestión del portafolio de productos del CEIGE.
- Elaborar el diseño del sistema para la gestión del portafolio de productos del CEIGE.
- Validar la solución propuesta.

Como **idea a defender** se plantea: si se realiza el análisis y diseño del sistema de gestión del portafolio de productos del CEIGE a partir del análisis de dominio basado en características, se contribuiría con la identificación de productos por parte del cliente en el portafolio de productos del CEIGE.

Para el desarrollo de este trabajo de diploma se emplean los métodos de investigación científica siguientes:

Analítico-Sintético: mediante este método se analiza la documentación bibliográfica seleccionando los elementos de interés para la investigación. También se analiza los el estado del arte a utilizar para la gestión del portafolio de productos del CEIGE

Histórico-Lógico: con este método se realiza un estudio de los sistemas que se dedican a la gestión del portafolio de productos y que emplean modelos de características.

Modelación: este método contribuye a la construcción de los modelos que se generan durante el diseño del sistema para la gestión del portafolio de productos del CEIGE.

Observación: se utiliza para conocer a través de la percepción directa realizada a otros sistemas, características que necesita o puede tener el sistema que se desarrollará, así como las vías para lograrlo.

Entrevista: la aplicación de este método permite conocer el funcionamiento de la comercialización de productos en el CEIGE a través de la voz de los especialistas que conocen sobre el tema, permitiendo identificar los requisitos funcionales de la aplicación.

El documento está estructurado de la siguiente manera:

Capítulo 1 Fundamentación Teórica: en este capítulo se realiza una investigación sobre los sistemas existentes que se dedican a la gestión del portafolio de productos y que emplean modelos de características. Además se estudian los conceptos relacionados con las LPS y las herramientas para el análisis de modelos de características. Se fundamenta la selección de la metodología de desarrollo a aplicar, las fases y técnicas de la ingeniería de requisitos, los patrones de diseño a emplear, las métricas para su validación, el uso de las herramientas y lenguajes en el desarrollo de la propuesta.

Capítulo 2 Propuesta de Solución: en este capítulo se realiza el modelado de negocio del proceso de comercialización de productos, se identifican y describen los requisitos funcionales. Se generan artefactos como: diagrama de proceso de negocio, modelo conceptual, diagrama de clases del diseño y modelo de datos.

Capítulo 3 Validación de la Propuesta: en este capítulo se validan los resultados obtenidos en el capítulo anterior, mediante la aplicación de las técnicas revisiones técnicas formales y construcción de prototipos de interfaz de usuario, se validan los requisitos funcionales. Además se emplean las métricas de tamaño operacional de clases y relaciones entre clases para validar el diseño.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se realiza un estudio sobre los sistemas de gestión de portafolio de productos que existen a nivel mundial. Se hace un análisis sobre los modelos de características y su vinculación a la identificación de productos desde el portafolio de una organización. Se definen las herramientas, lenguajes, notación de modelado, patrones y técnicas de validación del diseño a utilizar, también se muestran técnicas para la captura y validación de los requisitos, así como los patrones para su descripción.

1.1. Portafolio de productos

Un portafolio de productos es el conjunto de todos los bienes y/o servicios que se producen en una empresa. Su gestión es una de las funciones más importantes de la actividad empresarial. Permite definir con claridad hacia qué mercado estará orientado un determinado producto, apoya la toma de decisiones sobre las estrategias y operaciones que se llevarán a cabo en la empresa para lograr incrementar las ventas, asignar correctamente los recursos con los que cuenta (materiales o humanos) y desarrollar productos que se ajusten a las necesidades de los clientes. Además permite obtener información sobre los productos que lo componen, conocer las limitaciones y oportunidades tanto empresariales como de mercado y separar los que tienen buena aceptación de los que no la tienen facilitando a los directivos de empresas, decidir si se mantiene o no un producto en el portafolio a partir de los beneficios que pueda reportar. (Corporation, 2006)

La gestión de portafolio de una empresa permite a los especialistas encargados de la comercialización de los productos conocer cuáles son los más solicitados, las principales características en explotación, la orientación del mercado, las tecnologías de mayor desarrollo, entre otros indicadores. Esto se logra a partir del análisis de la información que aportan los clientes así como de las expectativas o necesidades planteadas por ellos. (Corporation, 2006)

Después de un análisis de los principales conceptos que definen un portafolio de productos, se considera como la posibilidad o capacidad de agrupar bienes y servicios que se producen en la organización, permitiendo la gestión de los mismos.

1.2. Sistemas para la gestión del portafolio

Para la gestión de portafolios, las empresas cuentan con sistemas automatizados capaces de facilitar los procesos de administración de productos. A continuación se plasma el análisis de algunos de estos sistemas.

Changepoint

La Gestión del Portafolio de Proyectos (PPM, por sus siglas en inglés) de Changepoint permite detectar y considerar ideas para nuevos productos, comprender qué productos necesitan más inversión y cuáles simplemente deberían mantenerse, proporcionando una visibilidad integral a la organización para controlar sus necesidades de aplicaciones, productos, inversiones y proyectos. Además, se puede responder a las necesidades exclusivas de cada cliente realizando un seguimiento de los problemas y respuestas mejoradas para facilitar una relación con el cliente más cooperativa (Corporation, 2009). El software de PPM de Changepoint es una solución de gestión del portafolio con un enfoque integrado que incorpora las disciplinas de Planificación de portafolios, Gestión del portafolio de proyectos (PPM), Gestión del Portafolio de Aplicaciones (APM, por sus siglas en inglés) y la gestión de los recursos que intervienen en todos los aspectos de su ciclo de vida. (Corporation2009)

Solución Gestión de Carteras de Productos y Planificación de Temporadas de Dassault Systèmes (DS)

La solución Gestión de Requisitos y Cartera de Productos de Dassault Systèmes permite que el fabricante de maquinaria industrial establezca procedimientos para la gestión de productos en todo el ciclo de vida. Brinda robustas funcionalidades para la gestión de requisitos, que fomentan la colaboración entre los administradores del producto y los equipos técnicos, capturando y procesando la opinión del cliente, así como las especificaciones internas y externas que pudieran existir. Basándose en dichos requisitos, el ingeniero podrá establecer un listado de características que representen el punto de vista del cliente y construir estructuras genéricas de productos para establecer una biblioteca de características estándar que ayuda a aumentar la variabilidad de la oferta, pues permite que las características se adapten a los distintos mercados. (Systemes, 2012)

Durante todo el ciclo de vida se siguen las especificaciones funcionales y estéticas, y las pruebas de cada nueva característica del producto. También permite planificar el portafolio de productos según las necesidades del cliente, tendencias del mercado e innovación tecnológica. Esto confiere la flexibilidad necesaria para crear nuevos Indicadores Clave de Rendimiento (KPI, por sus siglas en inglés) y vistas

de cuadros de mando configurados para el usuario, a fin de reaccionar rápido ante los problemas y las nuevas oportunidades. Al ofrecer a los directivos una visibilidad en tiempo real del estado de desarrollo del portafolio de productos, el análisis preliminar de situaciones puede realizarse antes de emplear recursos e incurrir en gastos. (Systemes, 2012)

SE Portfolio

SE Portfolio ayuda a la empresa a identificar, priorizar, equilibrar y administrar la combinación de iniciativas que ofrezca la mayor contribución y valor agregado a los intereses estratégicos de la organización. Además ofrece más visibilidad a los portafolios de la organización: permite monitorear y comunicar el progreso en términos de valores reales y previstos en criterios clave, como costo, plazo, beneficios y riesgos. Con la solución, es posible identificar, por medio de un *ranking*, las inversiones que compiten entre sí y optimizar la aplicación de los recursos a las oportunidades de mayor valor. (Expert, 2012)

El sistema permite analizar diferentes casos por medio de simulaciones; a través de las tablas financieras los gerentes pueden representar como maximizar las ganancias y reducir los costos. Las tablas pueden personalizarse para ajustarlas al portafolio, de acuerdo con las necesidades de la organización. Es posible aplicar listas de verificación en la evaluación para elevar el nivel de confianza y la calidad de los resultados. (Expert, 2012)

Gears

El sistema Gears fue creado por la compañía BigLever Software. Es un configurador de LPS que suministra un cambio en la administración y la producción automatizada basada en la metodología de tres capas que incluye modelado de capacidades y un mecanismo uniforme para implementar los cambios en los requisitos de software. Incluye requerimientos, arquitectura, código fuente, diseño y documentación del producto, reúne aquellos que son comunes y los ordena por líneas de productos, lo que permite automatizar la producción y responder con rapidez a los cambios que surgen en el ciclo de vida del software. Uno de los parámetros que recibe como entrada es un perfil de características obtenido a partir de las necesidades de los clientes. (Software, 2007)

Tabla I. Comparación entre sistemas de gestión de portafolio

Indicadores/Sistemas	Changepoint	SE Portfolio	DS	Gears
Operabilidad	Sí	Sí	No	Sí

Interoperabilidad	Sí	Sí	Sí	Sí
Flexibilidad	Sí	Sí	Sí	Sí
Cognoscibilidad	No	Sí	No	Sí
Traceabilidad	Sí	Sí	Sí	Sí
Sistema web	No	Sí	No	No
Informe con gráficos	No	Sí	No	Sí
Análisis de características	No	No	Sí	Sí
Licencia GPL	No	No	No	No

En la tabla anterior se realiza una comparación entre los sistemas para la gestión de portafolios atendiendo a los siguientes indicadores, donde Si indica la presencia del indicador y No indica su ausencia:

Operabilidad: capacidad de permitir al usuario interactuar con el software y controlarlo cuando lo solicite. Mide la utilidad de la información y la facilidad de desplazamiento en la ventana de ayuda.

Interoperabilidad: capacidad de interactuar con uno o más sistemas especificados (que la aplicación se relacione correctamente con otros sistemas, que funcione en diferentes navegadores).

Flexibilidad: capacidad de permitir una modificación determinada.

Cognoscibilidad: capacidad de permitir al usuario entender el software y utilizarlo a través de un demo para el producto de software y/o tutoriales.

Traceabilidad: capacidad de monitorear el ciclo de vida de todo el proyecto.

Sistema web: si es una aplicación web o una aplicación de escritorio.

Informe con gráficos: si permite llevar el control de la información mediante gráficos.

Análisis de características: si realizan algún análisis basado en las características de los productos para la gestión del portafolio.

Licencia GPL: si son sistemas libres o privativos.

Con el estudio realizado a los sistemas para la gestión de portafolios, se logró identificar que Gears cumple con la mayoría de los indicadores propuestos, pero no es una herramienta de mercado sino propia de una organización. Además al igual que el resto de los sistemas analizados es una solución privativa y comprar su licencia o pagar por el soporte resultaría muy costoso. Sin embargo, en la presente investigación se utilizarán como puntos de reutilización, las características que brinda Gears

relacionadas con la recopilación de requerimientos y la agrupación de los mismos en líneas de productos.

1.3. Línea de productos de software (LPS)

Estudios e investigaciones recientes, sugieren que la reutilización de componentes en el desarrollo de software debe experimentar un cambio de rumbo hacia la producción de familias de sistemas, en lugar de sistemas individuales. Las familias de sistemas o líneas de productos de software se centran en la búsqueda de lo común entre diferentes sistemas, dentro de un mismo dominio, para también controlar los puntos de variabilidad entre esos sistemas de una forma más o menos automática. De esta forma, es posible obtener nuevos sistemas basados en la reutilización de componentes ya existentes. (Cano, 2008)

Las líneas de productos de software distingue dos procesos de desarrollo: ingeniería del dominio e ingeniería de la aplicación. La ingeniería del dominio tiene como función principal producir elementos que serán reutilizados posteriormente y al igual que en el desarrollo de sistemas particulares o individuales, se atraviesan las diferentes fases de análisis, diseño e implementación, aunque teniendo en cuenta que se está desarrollando para un conjunto de sistemas y no uno sólo. (Cano, 2008)

El análisis del dominio consiste en determinar el ámbito o contexto de la línea de productos, de modo que se identifiquen los elementos comunes dentro del dominio, así como la variabilidad existente; mientras que el diseño del dominio se encarga de producir una arquitectura común para toda la línea de productos y planificar la forma de producción de cada uno de los miembros individuales de la familia, haciendo uso de los componentes reutilizables. (Cano, 2008)

El otro proceso, ingeniería de aplicación, utilizando los recursos creados por la ingeniería de dominio reúne el conjunto de tareas necesarias para construir aplicaciones concretas reutilizando componentes, atraviesa por la fase de especificación de requisitos, principal fuente de información para la obtención de diferentes sistemas dentro del dominio. Con esta información, se introducen variaciones en la arquitectura de referencia y en la implementación, obteniendo como resultado un producto de software único. (Cano, 2008)

El principal beneficio de las LPS es la reducción en tiempo y esfuerzo en desarrollo para realizar productos específicos pertenecientes a la misma familia. También introduce nuevas características como el diseño de variabilidad, que se refiere a la incorporación de mecanismos de variación (componentes

plugin, mejoras orientadas a aspectos) y la derivación del producto que es el proceso de construcción de productos software, dada una configuración específica. (Ramírez, 2009)

1.4. Modelos de características

Para describir todos los productos que una LPS es capaz de producir, es necesario disponer de un modelo que capture los elementos comunes y diferentes entre los sistemas pertenecientes a una LPS, para ello existen los modelos de características, cuyo objetivo principal es recoger las divergencias y aspectos comunes entre los distintos productos. Los modelos de características son modelos de variabilidad propuestos por el análisis de dominio. Los mismos representan gráficamente una LPS y se clasifican en tres tipos que se explican a continuación.

1.4.1. Modelos de características básicos

Los modelos básicos (Ver Anexo 1) fueron propuestos en 1990 como parte del método de análisis FODA (*Feature-Oriented Domain Analysis*). Están compuestos por dos elementos: las características, que se representan en forma de árbol y las relaciones entre ellas que pueden ser jerárquicas o no.

La relación jerárquica se define entre una característica padre y sus hijas. Una hija solamente puede formar parte de los productos en los que la característica padre aparece. En FODA se proponen tres tipos de relaciones jerárquicas. (Ross, 2009)

Obligatorias: la característica hija tiene que formar parte de un producto.

Opcionales: la característica hija puede o no formar parte del producto.

Alternativas: solo una de las características hijas puede formar parte del producto.

La relación no jerárquica suelen ser del tipo: si la característica A aparece, entonces la característica B se debe incluir o excluir según el criterio de funcionalidad de los elementos que componen el producto. Estas relaciones pueden ser:

Excluye: cuando la presencia de una característica en un producto depende de la ausencia de otra.

Requiere: cuando la existencia de una característica en un producto requiere de la presencia de otra.

En 1998 se presentan los modelo Feature-RSEB (*Reuse-Driven Software Engineering Business*) como una extensión de FODA, los cuales incluían una nueva relación jerárquica, OR indicando que cuando la

característica padre hace parte de un producto particular, una o más de las características del grupo de hijas deben hacer parte del producto. (Benavides, 2009)

1.4.2. Modelos basados en cardinalidad y multiplicidad

En los modelos basados en cardinalidad y multiplicidad (Ver Anexo 2) se emplea la multiplicidad para sustituir las relaciones alternativas y OR, mientras que la cardinalidad se utiliza para sustituir las relaciones opcionales y obligatorias.

La multiplicidad equivalente a la relación alternativa es $\langle 1-1 \rangle$, lo que significa que solo una de las hijas del grupo puede hacer parte del producto. La multiplicidad equivalente a la relación or es $\langle 1- n \rangle$, siendo “n” el número de características del grupo, lo que significa que una o n de las características del grupo pueden hacer parte del producto. La cardinalidad equivalente a la relación obligatoria es $[1..1]$, lo que significa que la característica hija debe hacer parte del producto. La cardinalidad equivalente a la relación opcional es $[0..1]$, lo que significa que la característica hija, puede o no hacer parte del producto. (Ross, 2009)

1.4.3. Modelos extendidos

La necesidad de representar informaciones adicionales en los modelos descritos anteriormente condujo a extender los modelos de características, (Ver Anexo 3) añadiéndoles atributos compuestos por:

Nombre: el identificador de atributo.

Dominio: especifica un rango de posibles valores para un atributo.

Valor: constituye el valor del atributo que pertenece al dominio especificado.

Además de poder especificar atributos para las características del modelo, también se pueden especificar relaciones entre los atributos. De esta manera, una característica podría requerir o excluir otras características dependiendo del valor de sus atributos. (Benavides, 2009)

1.4.4. Análisis de los tipos de modelos de características

Con el análisis de los tipos de modelos de características existentes y la necesidad de expresar las características funcionales de los productos de la LPS, se selecciona el modelo extendido para recopilar la información, debido que es el único que permite representar atributos de las características.

1.5. Análisis automatizado de modelos de características

El análisis automatizado es una actividad importante en el ambiente de LPS, porque es muy difícil realizarla de forma manual, ya que provoca que los resultados estén sujetos a errores. Debido a que los modelos de características son uno de los artefactos principales que se obtiene de la ingeniería de dominio, su análisis desde las primeras etapas del desarrollo contribuye al éxito de la línea de productos. Para llevar a cabo esta tarea se emplean los llamados razonadores lógicos y a continuación se describen tres de ellos.

CPS

Problema de Satisfacción de Restricciones o CSP (Constraint Satisfaction Problem, por sus siglas en inglés) es un paquete de software que recibe como entrada un problema modelado como un CSP y determina si existe alguna solución para el mismo. Este razonador representa a las entidades de un problema como una colección homogénea de restricciones finitas sobre variables, las cuáles son resueltas por métodos de satisfacción de restricciones. CSP se utiliza intensamente en investigación tanto en la inteligencia artificial, investigación operativa, bases de datos o sistemas de recuperación. (Ramírez, 2009)

La programación de restricciones es la propuesta más flexible. Puede ser usada para desempeñar la mayoría de las operaciones actualmente identificadas en los modelos de características. Sin embargo, la programación de restricciones revela cierta debilidad cuando calculan ciertas operaciones en modelos de características de medio y gran tamaño como por ejemplo al calcular el número de posibles combinaciones de características. (Ramírez, 2009)

SAT

El problema de Satisfacción proposicional o SAT(Boolean Satisfiability Problem, por sus siglas en inglés) consiste en decidir cuándo una fórmula proposicional se puede satisfacer, por ejemplo, unos valores lógicos pueden ser asignados a sus variables de una forma que hagan la fórmula verdadera. Los resultados de rendimiento para los razonadores SAT son ligeramente mejores en resultados que CSP, la expresividad que obtenemos con CSP es mucho mayor, por lo que podemos definir las restricciones de manera muy simple. (Ramírez, 2009)

BDD

Un diagrama de decisión binario o BDD (Binary Decision Diagram, por sus siglas en inglés) es una estructura de datos usada para representar una función booleana. Se representa como un grafo acíclico

con una raíz, dirigido y compuesto por un grupo de nodos de decisión y dos nodos terminales llamados terminal-0 y terminal-1. Cada nodo en el grafo simboliza una variable en una función booleana, y tiene dos nodos hijos que representan la asignación de la variable a 0 o 1. Todos los caminos desde la raíz al terminal-1 representan las asignaciones de variables para los que la función booleana es verdadera en cambio todos los caminos al terminal-0 representan las asignaciones de variables para los que la función booleana devuelve falso.

Aunque el tamaño de los BDD pueden ser reducidos de acuerdo a algunas reglas establecidas, la debilidad de este tipo de representación es el tamaño de la estructura de datos la cual puede variar entre rango lineal o exponencial dependiendo del orden de la variable. Calcular el mejor ordenamiento de las variables puede ocasionar grandes consumos de memoria, este problema es compensado por los tiempos resultantes. (Ramírez, 2009)

1.5.1. Comparación entre razonadores

El razonador CPS tiene un tiempo de ejecución débil cuando realiza ciertas operaciones sobre medianos o grandes modelos de características. Por su parte el razonador SAT tiene resultados de rendimiento ligeramente mejores que los que se obtienen con CSP, sin embargo, no son una alternativa muy poderosa porque no hay ninguna propuesta donde los atributos de los modelos de características puedan ser traducidos a SAT para realizar operaciones como minimizar o maximizar valores de atributos. Los razonadores SAT y CSP son incapaces de encontrar el número total de soluciones de los modelos de características de mediano y gran tamaño en un tiempo razonable, BDD resuelve este problema en un tiempo insignificante, lo que justifica su uso en operaciones de conteo. (Benavides, 2009)

Para la solución de la presente investigación se propone utilizar una herramienta capaz de emplear una combinación entre los razonadores SAT, CSP y BDD con el objetivo de garantizar un mejor desempeño de las operaciones de análisis y aprovechar las ventajas que cada uno de estos proporciona.

1.6. Herramientas para el análisis automatizado de modelos de características

Existen muchas herramientas para el análisis automatizado de modelos de características, la mayoría han sido desarrolladas en distintas universidades como proyectos de investigación y publicadas relativamente hace poco tiempo (Ramírez, 2009). A continuación se realiza un análisis de las mismas atendiendo a los razonadores que emplean, licencia, interfaz gráfica de usuario (GUI), configuraciones y soporte para modelos extendidos.

1.6.1. FeatureIDE

Está basado en Eclipse y bajo licencia Open-Source. Permite realizar configuraciones y la creación de restricciones. Emplea el razonador SAT para realizar operaciones de análisis y soporta varios lenguajes como: Java, C, C++, C# y XML. Es un editor gráfico y textual que no permite representar modelos de características extendidos. (Ramírez, 2009)

1.6.2. Requiline

Es una aplicación para Windows que requiere de conexión con una base de datos Oracle o MySQL. Dispone de un validador de restricciones, gestor de usuarios con posibilidad de personalizar los perfiles e intercambio de información a través de ficheros XML. Soporta modelos extendidos y permite realizar configuraciones de productos. (Ramírez, 2009)

1.6.3. XFeature

XFeature es una herramienta para el modelado de LPS desarrollada como un plugin de Eclipse bajo licencia Open-Source. Permite crear modelos de características mediante un asistente, ofreciendo facilidades para expresar restricciones. La aplicación posee archivos para su configuración que definen su comportamiento y utilizan tecnología estándar como XML. (Zürich, 2004)

1.6.4. Pure::Variants

Pure::Variants es una herramienta comercial, desarrollada como plugin de Eclipse, que soporta el modelado y configuración de características comunes y variables a través de una interfaz gráfica. Realiza operaciones de análisis por medio de razonadores de restricciones basados en Prolog¹. Puede integrarse en cualquier proceso de desarrollo independientemente del lenguaje de programación que se emplee. Utiliza una base de datos centralizada para el almacenamiento de los modelos. (David Benavides, 2007)

1.6.5. FaMa (Feature Model Analyser)

FaMa ha sido implementado como un plugin de Eclipse desarrollado bajo licencia Open-Source. Integra diferentes razonadores con el objetivo de aprovechar lo mejor de cada uno en términos de rendimiento. Soporta modelos de características básicos, extendidos y basados en cardinalidad. Los modelos básicos

¹ El Prolog (o PROLOG), proveniente del francés PROgrammation en LOGique, es un lenguaje para programar artefactos electrónicos mediante el paradigma lógico con técnicas de producción final interpretada. Es bastante conocido en el área de la Ingeniería Informática para investigación en Inteligencia Artificial.

y basados en cardinalidad se guardan en formato XMI y los extendidos con extensión .afm y .emf, también se pueden importar y exportar modelos en formato XML. Posee un editor visual para el modelado de características y un motor para el análisis. (David Benavides, 2007)

Comparación entre herramientas para el análisis automatizado de modelos de características

Tabla II. Herramientas para el análisis automatizado de modelos de características

Indicadores/Herramientas	Feature1 DE	Requiline	XFeature	Pure::Variants	FaMa
BDD	No	No	No	No	Sí
SAT	Sí	No	No	No	Sí
CPS	No	No	No	No	Sí
Modelos extendidos	No	Sí	Sí	No	Sí
GUI	Sí	Sí	Sí	Sí	Sí
Licencia Open-Source	Sí	No	Sí	No	Sí
Configuraciones	Sí	Sí	Sí	Sí	Sí

Con el estudio realizado de las herramientas anteriores se determina que FaMa reúne todos los aspectos que se establecen en la comparación, al ser la única que emplea los tres razonadores: BDD, SAT y CPS. Fama se utiliza como referencia para el desarrollo de la solución.

1.7. Herramienta a emplear en la solución

Para emplear la herramienta FaMa existen dos tipos de formato para la entrada de los datos que forman el modelo de características: XML y texto plano, seleccionándose el último porque permite representar el modelo extendido de características. El archivo de texto plano debe tener extensión .afm o .efm y en su estructura debe constar de tres secciones: una sección para las relaciones, una para las restricciones y otra para los atributos.

La sección de relaciones se enuncia %Relationships, contiene la estructura del árbol y los tipos de relaciones. El primer dato que se define es la raíz del modelo, cualquier característica después de los dos puntos es un hijo y los nombres pueden estar formados por letras y el caracter especial “_”. Las características opcionales se representan entre corchetes ([]). Los grupos de relaciones se representan con cardinalidad y un grupo de características. En la sección de restricciones, que se declara %Constraints, se definen las restricciones de inclusión (REQUIRES) y exclusión (EXCLUDES), además

se pueden usar operadores aritméticos (+,-,*,/,mod), operadores relacionales (>,>=,<,<=,==,!=) y operadores lógicos (AND, OR,NOT,IMPLIES, IFF (if only if)) para definir restricciones complicadas. En la sección de atributos, %Attributes, se definen los atributos de las características, con su tipo, dominio y valor por defecto (cuando la característica está seleccionada). (ISA, 2007) A continuación se representa el formato de texto plano correspondiente al Anexo 3.

```
% Relationships

Software: Dominio Gestor BD [Marco de Trabajo];

Dominio: [1,1] { Aduanero Empresarial Bancario};

Gestor BD: [1,1] {Apache Oracle};

[Marco de Trabajo]: [1,3] {Doctrine EXT JS Sauxe};

% Attributes

Apache.version: [2.0,2.5], 2.5;

Oracle.version: [9,10, 11g], 11g;
```

Figura 1. Estructura del archivo del modelo extendido

1.8. Modelo de desarrollo

Para el desarrollo del presente trabajo de diploma se emplea el modelo de desarrollo establecido por el CEIGE (Obregón, 2012). Este modelo se caracteriza por ser centrado en la arquitectura, orientado a componentes, iterativo e incremental, ágil y adaptable al cambio. De las fases que define, la presente investigación se centra en el modelado del negocio, donde se generan los artefactos: descripción y diagrama del proceso de negocio, modelo conceptual y glosario de términos; la fase de requisitos donde se generan los documentos de especificación de requisitos; la fase de análisis y diseño donde se realizan los diagramas de clases del diseño y el modelo de datos.

1.9. Ingeniería de requisitos

La ingeniería de requisitos (IR) es el uso sistemático de procedimientos, técnicas, lenguajes y herramientas para obtener con un coste reducido el análisis, documentación, evolución continua de las necesidades del usuario y la especificación del comportamiento externo de un sistema que satisfaga las necesidades del usuario (Pressman, 2002). Se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación, validación y mantenimiento de los requisitos para un producto de

software determinado, donde es muy importante tomar en cuenta que el aporte de la IR ayudará a determinar la viabilidad del desarrollo del software. Para llevar a cabo la IR se transitará por la etapa de desarrollar y administrar los requisitos que define el modelo de desarrollo del CEIGE. En esta etapa se identifican los requisitos, se describen y validan, además de describir las salidas del sistema y se elaboran los prototipos de interfaz de usuario. (Obregón, 2012)

1.9.1. Técnicas para la captura de requisitos

Las técnicas de recopilación de requisitos son utilizadas para identificar las necesidades del cliente y requieren una adecuada planificación y preparación. Para el desarrollo de la investigación se utilizan:

- Lluvia de ideas: a esta técnica se le conoce también como tormenta de ideas y permite la libre expresión de las ideas de los participantes, con el propósito de producir el mayor número de datos, opiniones o soluciones sobre algún tema, pues mientras más ideas se generen, mejores resultados se conseguirán, además, la producción de ideas en grupos puede ser más efectiva que la individual, ya que permite analizar diferentes puntos de vista. (Sommerville, 2009)
- Entrevistas: las entrevistas sirven para obtener una comprensión general de lo que hacen los usuarios, como podrían interactuar con el sistema y las dificultades a las que se enfrentan los sistemas actuales. (Sommerville, 2009)
- Sistemas existentes: esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, se puede analizar las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada, por otro lado también es útil analizar las distintas salidas que los sistemas producen (listados, consultas, entre otros), porque siempre pueden surgir nuevas ideas sobre la base de estas. (Hernández, 2011)
- Inspección de Registros: ésta técnica permite obtener información sobre algún tema en específico, a través de documentos que proporcionen especialistas en la materia o que se obtengan mediante la búsqueda de información de un determinado tema. (Sommerville, 2009)

1.9.2. Técnicas de especificación de requisitos

Para la especificación de requisitos de la propuesta, se emplean las siguientes técnicas:

- Lenguaje natural: resulta una técnica muy ambigua para la definición de los requisitos. Consiste en definir los requisitos en lenguaje natural sin usar reglas para ello. Pero, a pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando. (Orta, 2012)

- Plantillas o patrones: Esta técnica tiene como objetivo describir los requisitos mediante el lenguaje natural, pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida que el equipo de desarrollo va cumplimentando, usando para ello el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información; cuanto más estructurada sea esta menos ambigüedad ofrece (Orta, 2012). Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Un sistema bien estructurado está lleno de patrones. Cada patrón describe un problema que ocurre una y otra vez, y luego describe el núcleo de la solución a ese problema, de tal manera que se puede usar esa solución un millón de veces más sin hacer jamás la misma cosa dos veces (Larman, 2004). Para la especificación de los requisitos de la propuesta se utilizará la plantilla definida por el CEIGE para este fin y los patrones del análisis propuestos por Craig Larman siguientes:
- Nombres que revelan la intención (*IntentionRevealingName*): El nombre debe reflejar la intención del requerimiento y reflejar un único objetivo e intención que se está intentando lograr. Se debe asignar un nombre apropiado que facilite el manejo del requerimiento, permitiendo tener una vista general del trabajo en su conjunto.
- Escenario más Fragmentos (*ScenarioPlusFragments*): El usuario debe ser capaz de seguir el camino básico en el que está interesado, en caso contrario le sería muy difícil entender la información que busca. Para ello se describen las acciones del flujo básico como un escenario simple, sin tener en cuenta posibles fallos. Debajo se describen los eventos que explican qué flujo alternativo puede ocurrir.
- Condiciones detectables (*DetectableConditions*): Un sistema no puede manejar eventos que no pueda detectar, los desarrolladores necesitan conocer qué situaciones detectar. Se debe capturar cada posibilidad razonable, de otra manera, el sistema nunca estará capacitado para ejecutar el escenario. Descubrir una condición olvidada después de que el sistema ha introducido servicios es incluso más caro.
- Pasos Nivelados (*LeveledSteps*): Excesivamente largos o excesivamente pequeños pasos en el requisito oscurecen la meta y lo hacen difícil de leer y comprender. Cada escenario debe tener de tres a nueve pasos. Idealmente son todos a niveles similares un nivel de la abstracción

justamente debajo de la meta del requerimiento. Esto no quiere decir que un requisito no pueda tener más de 9 pasos si lo requiere.

1.9.3. Técnicas de validación de requisitos

Una vez definidos los requisitos, estos necesitan ser validados. La validación tiene como misión demostrar que la definición de los mismos define realmente el sistema que el usuario necesita o el cliente desea. Para la validación de los requisitos de la propuesta se emplean las técnicas siguientes:

- Revisión técnica formal: esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida, pero es muy fructífera pues se obtiene una documentación clara para clientes y desarrolladores. (Sommerville, 2009)
- Prototipos: Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final. Permiten determinar si el sistema diseñado, sobre la base de los requisitos recolectados, le permite al usuario realizar su trabajo de manera eficiente y efectiva. Además, le posibilita a los desarrolladores validar la corrección de la especificación, mejorar el producto, examinar la viabilidad y usabilidad de la aplicación. (Sommerville, 2009)

1.9.4. Criterios para la evaluación y aceptación de los requisitos mediante métricas

Existen varias métricas para verificar la calidad de la especificación de los requisitos levantados. En la solución se emplean las siguientes:

- Correctitud: Esta métrica permite determinar si los requisitos fueron capturados correctamente, de forma que puedan satisfacer las necesidades del negocio y los interesados. Para aplicarla es necesario contar con un registro que contenga las observaciones del negocio realizadas, en el cual se encuentran reflejadas especificidades del mismo como por ejemplo: atributos de un concepto erróneos, omisión de validaciones y restricciones, omisión de un requisito. Para que la correctitud sea aceptable tiene que tomar un valor menor que 0.10.
- Completitud: Esta métrica se utiliza para determinar si la especificación de requisitos es completa, para esto se cuenta con un registro de las observaciones de formato realizadas, que son las generadas, porque se han omitido secciones que se requieren en la plantilla. Para que la completitud sea aceptable su valor debe ser mayor que 0.9.
- Consistencia: Con esta métrica es posible determinar si la especificación de requisitos es consistente, para lo cual se crea un registro de las observaciones de consistencia realizadas que

contiene las observaciones realizadas por contradicciones existentes entre los requisitos. Para que la consistencia sea aceptable debe tomar un valor menor que 0.20.

1.10. Patrones empleados para el diseño de la solución

En las soluciones informáticas es común encontrar aspectos que se repiten y que, realizados por diferentes personas, puede que no sean óptimos, por eso se utilizan los patrones de diseño que constituyen una solución estándar para un problema común de programación, una manera práctica de describir ciertos aspectos de la organización de un programa y una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. (Lago 2007)

1.10.1. Patrones de asignación de responsabilidades (GRASP)

Los patrones GRASP (Patrones Generales de Software para Asignación de Responsabilidades) del acrónimo General Responsibility Assignment Software Patterns, describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades (Vázquez, 2011). En la solución se utilizan los siguientes:

- **Experto:** es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele ser útil en el diseño orientado a objetos. Este tiene como beneficio que se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. Además el comportamiento se distribuye entre las clases que cuentan con la información requerida, alentando con ello definiciones de clases sencillas y más cohesivas que son más fáciles de comprender y mantener.
- **Creador:** este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos.
- **Alta cohesión:** asigna una responsabilidad de modo que la cohesión siga siendo alta. En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.
- **Bajo acoplamiento:** asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Una clase con bajo (o débil) acoplamiento no depende de muchas otras.

- **Controlador:** asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones: el sistema global (controlador de fachada), la entidad u organización global (controlador de fachada), algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).

1.10.2. Patrón arquitectónico Modelo-Vista-Controlador

El patrón de arquitectura conocido como Modelo-Vista-Controlador (MVC), separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario; es decir separa en tres capas diferentes los datos de una aplicación, la interfaz de usuario, y la lógica de control:

- **Modelo:** esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- **Vista:** esta capa maneja la visualización de la información, es decir que presenta el modelo en un formato adecuado para interactuar, que usualmente es la interfaz de usuario.
- **Controlador:** esta capa controla el flujo de datos entre la vista y el modelo; es decir que responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases.

1.11. Métricas de validación del diseño

Una métrica es un instrumento que cuantifica un criterio y persigue comprender mejor la calidad del producto, estimar la efectividad del proceso y mejorar la calidad del trabajo realizado. Las métricas para sistemas Orientados a Objetos (OO) deben de ajustarse a las características de los mismos (Ivisate, 2011). Para evaluar la calidad del diseño se emplean las métricas tamaño operacional de clases y relaciones entre clases.

1.11.1. Tamaño operacional de clase (TOC)

La métrica tamaño operacional de clase permite medir la calidad de las clases según los atributos Responsabilidad, Complejidad de implementación y Reutilización. De estos atributos de calidad, Responsabilidad y Complejidad de implementación son inversamente proporcionales a la Reutilización,

por lo que mientras mayor sea la responsabilidad y complejidad de implementación de una clase, menor será su nivel de reutilización.

Tabla III. Rango de valores para evaluación de la métrica TOC

	Categoría	Criterio
Responsabilidad	Baja	\leq Prom
	Media	Entre Prom y $2 \cdot$ Prom
	Alta	$> 2 \cdot$ Prom
Complejidad de implementación	Baja	\leq Prom
	Media	Entre Prom y $2 \cdot$ Prom
	Alta	$> 2 \cdot$ Prom
Reutilización	Baja	$> 2 \cdot$ Prom
	Media	Entre Prom y $2 \cdot$ Prom
	Alta	\leq Prom

1.11.2. Relaciones entre clases (RC)

Esta métrica está dada por el número de relaciones de uso de una clase con otras. Se le aplica a las mismas clases que le fueron aplicada la métrica TOC, pero en este caso los atributos a medir son: Acoplamiento, Complejidad de mantenimiento, Reutilización y Cantidad de pruebas.

Tabla IV. Rango de valores para la evaluación de la métrica RC

	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	> 2
Cantidad de pruebas	Baja	\leq Prom
	Media	Entre Prom y $2 \cdot$ Prom
	Alta	$> 2 \cdot$ Prom
Complejidad de implementación	Baja	\leq Prom
	Media	Entre Prom y

		2*Prom
	Alta	>2*Prom
Reutilización	Baja	>2*Prom
	Media	Entre Prom y 2*Prom
	Alta	<=Prom

1.12. Lenguaje de modelado

Lenguaje de Modelado Unificado o UML por sus siglas en inglés, es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. El mismo permite modelar sistemas utilizando técnicas orientadas a objetos, especificar todas las decisiones de análisis, diseño e implementación, construyendo así modelos precisos, no ambiguos y completos, también se utiliza para documentar todos los artefactos de un proceso de desarrollo y conectarse con lenguajes de programación (Ingeniería directa e inversa) (Larman, 2004). Con UML se realizará el modelo conceptual además de los diagramas de clases del diseño y el modelo de datos del sistema de gestión del portafolio de productos del CEIGE.

1.13. Notación de modelado de procesos de negocio (BPMN)

Se emplea BPMN como notación para el modelado de los procesos de negocio. Es una notación fácil, legible y entendible para cualquier persona, ya sea el analista del proceso, el desarrollador técnico y hasta el cliente, así como para crear un puente estandarizado entre el diseño de procesos de negocio y su implementación. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. (BigAgi, 2010)

1.14. Herramienta de modelado

Visual Paradigm es una herramienta CASE² que soporta el ciclo de vida completo del desarrollo de un software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Tiene como ventaja la capacidad de ejecutarse sobre diferentes sistemas operativos permitiéndole ser multiplataforma. Esta herramienta proporciona características tales como generación de código, ingeniería inversa y generación de informes. Visual Paradigm facilita la comunicación debido al uso de un lenguaje estándar, común a todo el equipo de desarrollo, ofrece modelo y código que permanece sincronizado en todo el

²CASE: Ingeniería de Software Asistida por Computadora del inglés Computer-Aided Software Engineering.

ciclo de desarrollo (Work, 2012). Se escoge ésta herramienta para el modelado del negocio, la realización del modelo conceptual, el modelo de datos, los diagramas de clases del diseño, por las facilidades que brinda en su integración con la notación BPMN y porque la UCI posee la licencia para su uso.

1.15. Herramienta para diseño de interfaces de usuario

Pencil es una herramienta gratuita y de código abierto para crear interfaces de usuario, desarrollada por Evolus basada en las tecnologías de Mozilla. Permite exportar a HTML, PNG, documento de Openoffice.org, documento de Word y PDF, incorpora plantillas de diagramas y prototipado. Se decidió utilizar dicha herramienta en el diseño de las interfaces de usuario por las ventajas mencionadas anteriormente y además porque permite importar colecciones externas de objetos, funcionalidad que la diferencia del Visual Paradigm. Esto permite la utilización de íconos que indican actividades típicas como: Adicionar, Eliminar, Modificar, Buscar, etc.; contribuyendo así a que las interfaces sean más intuitivas y sencillas. (Evolution Solutions Co., 2012)

1.16. Marco de trabajo

La solución será implementada sobre el marco de trabajo Sauxe. El mismo contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo. Es la unión de ZendExt, Doctrine y ExtJS. Utiliza en la capa de acceso a datos el Lenguaje de Consulta de Datos (DQL) implementado por Doctrine el cual es un potente y completo sistema Mapeador de Objeto Relacional (ORM) para PHP 5.2.3+, además incorpora una Capa de Abstracción a Base de Datos (DBL), utiliza ExtJS en la capa de presentación, por la gran gama de componentes que se pueden reutilizar para agilizar el proceso de desarrollo y mostrarle al usuario una interfaz más amigable y funcional. (Pupo, 2010)

Conclusiones parciales

- El análisis de los sistemas de gestión del portafolio de productos, permitió determinar que los mismos no serán utilizados para el desarrollo de la investigación pero se identificaron sus ventajas y funcionalidades comunes que sirven de soporte a la solución propuesta.
- El estudio de las técnicas de análisis de dominio basada en características se consideran de utilidad para la gestión dinámica de la identificación de productos.

- Se seleccionó FaMa como herramientas para el análisis automatizado de modelos de características por ser la única que emplea los tres razonadores: BDD, SAT y CPS
- Para guiar el proceso de desarrollo de software se seleccionó el Modelo de desarrollo orientado a componentes del CEIGE, el cual permitió generar los artefactos necesarios para desarrollar la propuesta de solución.
- Se definió el entorno de desarrollo a utilizar para la propuesta de solución, incluyendo las herramientas, lenguajes y tecnologías necesarias.

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

Introducción

En este capítulo se realiza el modelado de negocio del proceso de comercializar productos, son identificados y descritos los requisitos funcionales de la solución. Se generan artefactos como: diagrama de procesos de negocio, descripción del proceso de negocio, modelo conceptual y documentos de especificación de requisitos. Se realizan los diagramas de clases y se confecciona el modelo de datos a emplear en la propuesta.

2.1. Proceso de negocio

Un proceso de negocio es un conjunto de tareas lógicamente relacionadas que se llevan a cabo para obtener un determinado resultado. Estos procesos son la base para comprender mejor la forma en que opera un negocio en sus diferentes áreas y son una herramienta fundamental para acceder a modelos de calidad y eficiencia. Su principal objetivo es satisfacer las necesidades de los clientes, dentro de los mismos, se combinan las personas, los equipos, los recursos materiales y los procedimientos de negocio con objeto de producir un resultado concreto. (Ivisate, 2011) Para la presente propuesta se estudian los procesos fundamentales que se llevan a cabo en el CEIGE para vender sus productos, centrándose en el proceso Comercializar productos. A continuación se describe este proceso.

Tabla V. Descripción del procesos de negocio Comercializar productos

Objetivo	Llevar a cabo las actividades relacionadas con la solicitud de un producto y la satisfacción de dicha solicitud.
Evento(s) que lo genera(n)	Solicitar producto
Pre condiciones	N/A
Marco legal	N/A
Clientes externos	N/A
Entradas	Solicitud de producto
Flujo de eventos	
Flujo básico Comercializar producto	
1.	Realizar solicitud: el cliente realiza una selección de características de los productos que componen el portafolio de productos del CEIGE y que desea que conformen la solución

	a obtener.
2.	Recibir solicitud: el CEIGE recibe la solicitud realizada por el cliente con las características de la solución deseada.
3.	Elaborar oferta: si la solicitud es de un producto nuevo y se puede satisfacer, el CEIGE elabora una oferta para el cliente.
4.	Enviar oferta: el CEIGE envía la oferta elaborada al cliente para su aprobación.
5.	Revisar oferta: el cliente estudia la oferta para determinar si se ajusta a sus intereses.
6.	Firmar oferta: si se ajusta a sus intereses el cliente firma la oferta y la envía al CEIGE.
7.	Elaborar contrato: una vez recibida la oferta firmada el CEIGE procede a elaborar el contrato correspondiente.
8.	Enviar contrato: el contrato elaborado es enviado al cliente.
9.	Firmar contrato: una vez que recibe el contrato el cliente lo firma y lo envía al CEIGE.
10.	Desarrollar producto: al recibir el contrato firmado comienza el desarrollo de la solución solicitada.
11.	Entregar producto: cuando la solución está terminada es entregada al cliente.
12.	Recibir producto: el cliente recibe la solución solicitada.
Pos-condiciones	
1	Ha sido atendida la solicitud del cliente
Salidas	
1	Oferta
2	Contrato
Flujos paralelos	
N/A	
Pos-condiciones	
N/A	
Salidas	

N/A	
Flujos alternos	
Flujo alternativo 3a. La solicitud es de un producto nuevo y no se puede satisfacer	
1	Informar a cliente: el CEIGE informa al cliente que su solicitud no puede ser atendida por no contar con los medios para ello
2	Cancelar solicitud: el cliente cancela la solicitud
3	Concluye el proceso
Pos-condiciones	
1	No se ha tendido la solicitud del cliente
Flujo alternativo 3b. La solicitud no es de un producto nuevo	
1	Realizar el flujo básico 3
Pos-condiciones	
1	Ha sido atendida la solicitud del cliente
Flujo alternativo 6a. La oferta no se ajusta a los intereses del cliente	
1	Solicitar modificación: el cliente solicita modificar el contenido de la oferta definiendo los parámetros que desea cambiar.
2	Volver al flujo básico 3.
Pos-condiciones	
1	Se ha modificado la oferta realizada al cliente
Salidas	
1	N/A
Asuntos pendientes	
N/A	

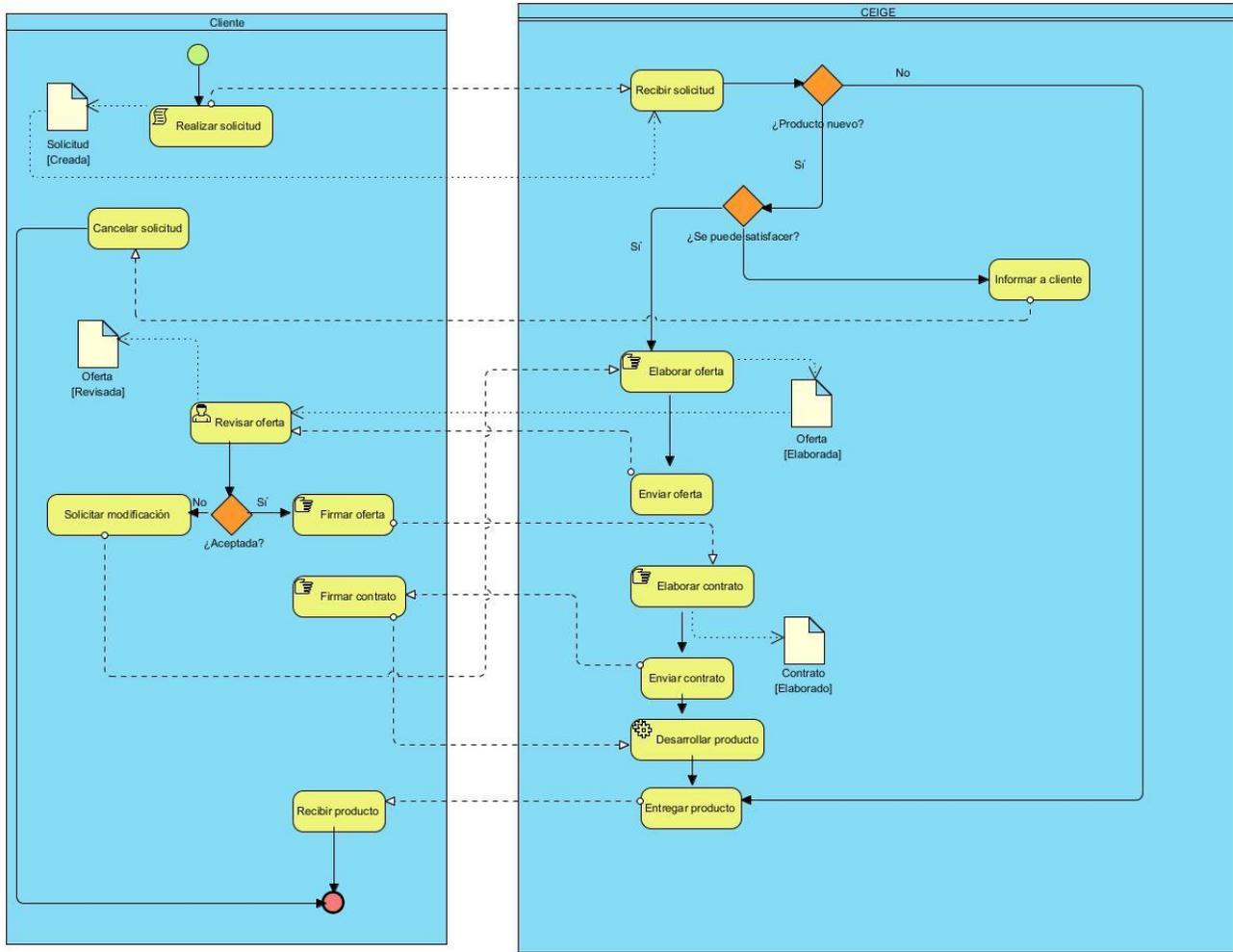


Figura 2. Diagrama BPMN del proceso de negocio Comercializar producto

2.3. Modelo conceptual

El modelo conceptual, también conocido como modelo de dominio, explica cuáles son y cómo se relacionan los conceptos relevantes en la descripción del problema. Mediante diagramas UML, específicamente diagramas de clases conceptuales significativas en el dominio del problema, se muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras a través de asociaciones (Cruz, 2011). Para la realización del modelo conceptual del sistema para la gestión del portafolio de productos del CEIGE se identificaron las clases del negocio, los atributos y las relaciones existentes entre dichos conceptos.

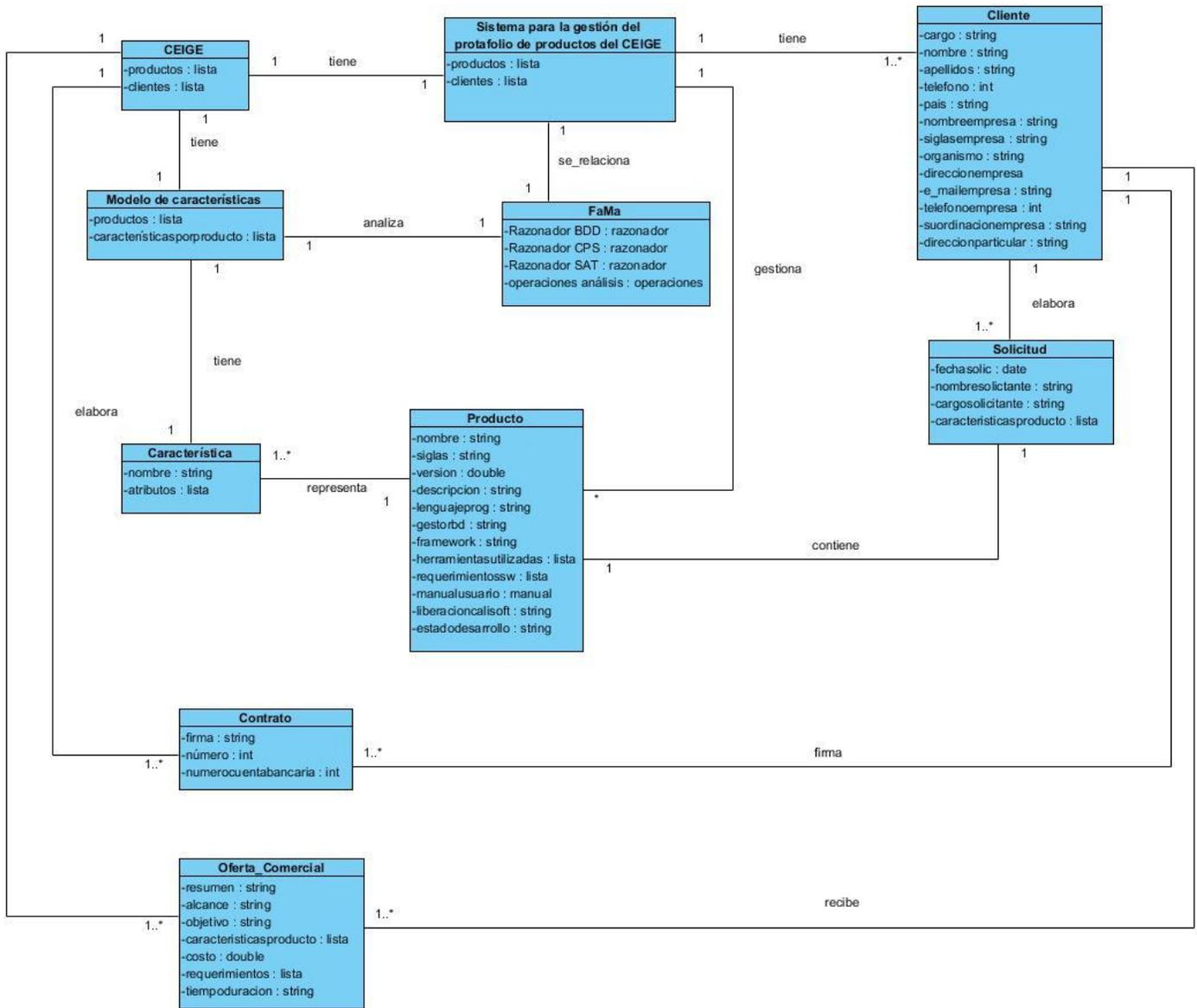


Figura 3. Modelo conceptual

Tabla VI. Descripción de las clases conceptuales

Clase conceptual	Atributos	Descripción
Característica	nombre id atributos	Concepto que almacena los datos relacionados con las características de los productos.

CEIGE	productos clientes	Concepto que representa el Centro de Informatización de la Gestión de Entidades.
Cliente	nombreenpresa siglasempresa organismo subordinación dirección e-mail teléfono nombre apellidos cargo ciudad país	Concepto que almacena los datos relacionados con los clientes.
Contrato	firma número númerocuentabancaria	Concepto que almacena los datos referentes a los contratos que se firman antes de comenzar a desarrollar los productos que se soliciten.
FaMa	Razonador BDD Razonador SAT Razonador CPS operacionesanálisis	Concepto que representa la herramienta de análisis de modelos de características a emplear en la solución.
Modelo de características	productos características	Concepto que representa la estructura donde se representan los productos del portafolio del CEIGE.
Oferta comercial	resumen alcance objetivo descripcion	Concepto que almacena los datos relacionados con una oferta de producto.

	costo requerimientos tiempoduración	
Producto	nombre siglas versión descripcion lenguajeprog gestorbd framework herramientasutilizadas requerimientossw manualusuario liberacioncalisoft estadodesarrollo	Concepto que almacena los datos referentes a los productos que se desarrollan dentro de la línea de productos.
Sistema para la gestión del portafolio de productos del CEIGE	productos clientes	Concepto que representa el sistema propuesto en la solución.
Solicitud	fechasolic nombresolicitante cargosolicitante asunto caracteristicasproducto	Concepto que almacena los datos referentes a las solicitudes de productos que se realizan al CEIGE.

2.4. Requerimientos funcionales

Los requerimientos funcionales describen con detalle la función del sistema, sus entradas, salidas y excepciones. Dependiendo del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general tomado por la organización se redactan los requerimientos. En principio la especificación de requerimientos funcionales de un sistema debe estar completa y ser consistente. La completitud significa

que todos los servicios solicitados por el usuario deben estar definidos y la consistencia se refiere a que los requerimientos no deben tener definiciones contradictorias. (Sommerville, 2009)

En la fase de Requisitos, se emplearon las técnicas de tormenta de ideas y entrevistas las cuales fueron ejecutadas a través de reuniones donde participaron el asesor de mercadotecnia del CEIGE y el jefe del centro, con el objetivo de obtener una vista general de las necesidades del cliente. Como resultado de la aplicación de las técnicas mencionadas se obtuvo un total de 16 requisitos funcionales. Al aplicar un procedimiento para calcular la complejidad (Msc. Karina Pérez Teruel, 2010), se lograron 6 requisitos de complejidad media y 10 de complejidad baja. Estos requisitos dan cumplimiento a las funcionalidades que el sistema brinda tanto al cliente como al administrador, estableciéndose así permisos asociados a dichos roles. La aplicación de las técnicas Sistemas existentes e Inspección de Registros permitió construir los prototipos de interfaz de usuario correspondientes a cada requisito funcional. Durante la administración de requisitos se realizaron las modificaciones a los prototipos de interfaz de usuario según las observaciones realizadas por el cliente, así como en las especificaciones de requisitos.

2.4.1. Lista de requisitos funcionales

Agrupación de requisitos Gestionar perfiles de clientes

- RF1.1. Crear perfil de cliente
- RF1.2. Modificar perfil de cliente
- RF1.3. Eliminar perfil de cliente
- RF1.4. Listar perfiles de clientes
- RF1.5. Buscar perfil de cliente

En esta agrupación de requisitos el cliente crea su perfil de cliente, en el cual introducirá sus datos personales y los de la empresa a la que pertenece. Podrá también modificar dicho perfil (modificar los datos) y eliminar su perfil (darse baja del sistema). A partir de los perfiles de clientes registrados, el administrador podrá consultar el listado de clientes y buscar un cliente determinado.

RF2. Gestionar solicitudes de productos

- RF2.1. Adicionar solicitud
- RF2.2. Modificar solicitud
- RF2.3. Cancelar solicitud

RF2.4. Listar solicitudes

RF2.5. Buscar solicitud

En esta agrupación de requisitos el cliente podrá solicitar un producto que ya está desarrollado en el CEIGE o conformar un producto a partir de las características de los ya existentes. Estas características son obtenidas del modelo de características mediante la herramienta de análisis FaMa. A partir de esta selección de características, el sistema le mostrará un listado de los productos que reúnen las características solicitadas para que el cliente decida con cuál desea quedarse, esta lista de productos también se obtiene de la herramienta FaMa. El cliente podrá modificar la selección de características (ya sea seleccionando o deseleccionando) y cancelar la selección. Para los tres casos: adicionar, modificar y eliminar, la selección se almacena en una solicitud de producto junto con la fecha en que se realizó, el nombre del cliente y la empresa a la cual pertenece este último. Por su parte el administrador podrá consultar el listado de solicitudes realizadas y buscar una solicitud determinada.

RF3. Realizar reporte de solicitudes pendientes

Este requisito le permitirá al administrador, tener control sobre las solicitudes de productos a las cuales no se le ha dado respuesta, a partir de las características que haya seleccionado el cliente para su producto, se puede determinar la complejidad de desarrollo y asignar correctamente los recursos (materiales o humanos) con los que cuenta el centro.

RF4. Realizar reporte de productos vendidos

Con este requisito el administrador podrá obtener información sobre la cantidad de cada producto que ha sido comercializado. Estos datos servirán de apoyo a la hora de decidir si un producto se mantiene en el portafolio, basándose en el nivel de comercialización del mismo.

RF5. Realizar reporte de productos solicitados

Este requisito brinda información sobre los productos que han sido solicitados, los clientes que los han solicitado y las empresas a las cuales pertenecen estos últimos, permitiendo identificar el mercado al cual enfocar el desarrollo de los productos del centro.

RF6. Realizar reporte de mejores clientes

Con este requisito el administrador podrá conocer cuáles son sus mejores clientes atendiendo a la cantidad de productos que han comprado.

RF7. Realizar reporte de características más comunes

Con este requisito el administrador puede obtener, a partir de las solicitudes realizadas, las características comunes entre ellas y en cuántas solicitudes están presentes, posibilitando trazar estrategias para potenciar el desarrollo de productos que las contengan y a su vez incrementar las ventas.

RF8. Integrar FaMa

Este requisito describe como se integrará el sistema con la herramienta FaMa. Cuando el usuario realiza una solicitud, la herramienta carga el modelo de características y a partir de él obtiene todos los productos que componen el portafolio de productos del CEIGE. Además cuando el administrador desear realizar un reporte de las características más comunes, la herramienta obtiene del modelo de características el número de productos en los que cada una de las características identificadas aparece.

Los requisitos relacionados con la gestión de la seguridad que se han incluido en la solución ya han sido definidos previamente en el trabajo de diploma de la tesista Leidy Villanueva Orta, titulado “Análisis y diseño de un Sistema para la Administración de un Repositorio de Activos de Software aplicable al proceso de desarrollo del CEIGE”. (Orta, 2012)

2.4.2. Especificación de requisitos

A continuación se muestran las especificaciones de requisitos de crear perfil, adicionar solicitud, realizar reporte de solicitudes pendientes e integrar FaMa. El resto de las especificaciones de requisitos se encuentran en el expediente de proyecto.

Para el rol de cliente:

Tabla VII. Descripción textual del requisito Crear perfil de cliente

Precondiciones	El cliente debe estar autenticado en el sistema
Flujo de eventos	
Flujo básico	
1	El usuario presiona el botón Perfil de la interfaz principal
2	El sistema muestra la interfaz para que el usuario introduzca sus datos. Se muestra:

Nombre
 Apellidos
 Dirección particular
 Teléfono
 País
 Ciudad
 Nombre de la empresa
 Siglas
 Subordinación
 Cargo del cliente
 Dirección de la empresa
 Teléfono de la empresa
 E-mail

- 3 Se introducen los datos y se selecciona el botón Aceptar
- 4 El sistema valida los datos introducidos (ver validación 1)
- 5 Si los datos son correctos el sistema registra el nuevo perfil del cliente
- 6 El sistema confirma el registro de los datos a través de un mensaje que se muestra en pantalla.
- 7 Concluye el requisito

Pos-condiciones

- 1 Se crea un nuevo perfil del cliente

Flujos alternativos

Flujo alternativo 5 a. Información errónea

- 1 El sistema señala los datos erróneos y permite corregirlos
- 2 El usuario corrige los datos señalados
- 3 Volver al paso 2 del flujo básico

Pos-condiciones

- 1 Se crea un nuevo perfil del cliente

Flujo alternativo 5 b. Información incompleta

- 1 El sistema señala los datos vacíos y permite corregirlos
- 2 El usuario completa la información
- 3 Volver al paso 2 del flujo básico

Pos-condición

- 1 Se crea un nuevo perfil del cliente

Flujo alternativo 3 a. El usuario cancela la acción

- 1 Concluye el requisito

Pos-condiciones

1 No se registran los datos del cliente

Validaciones

1 Se validan los datos según el modelo conceptual CIG-GPP-N-09Modelo Conceptual.doc

Relaciones	Requisitos Incluidos	NA
	Extensiones	NA
Conceptos	Cliente	Atributos visibles en la interfaz: Nombre Apellidos Dirección particular Teléfono País Ciudad Nombre de la empresa Siglas Subordinación Cargo del cliente Dirección de la empresa Teléfono de la empresa E-mail Atributos utilizados internamente: IdCliente
Requisitos especiales	NA	
Asuntos pendientes	NA	

Prototipo elemental de interfaz de usuario

The image shows a web form titled "Perfil" with a sub-header "Adicionar". It is divided into two main sections:

- Información personal:**
 - Nombre: Laura
 - Apellidos: Rodríguez Leveque
 - Dirección particular: Ave. 33 #22009
 - Teléfono: 2053319
 - País: Cuba (dropdown menu)
 - Ciudad: La Habana
- Información corporativa:**
 - Nombre empresa: Universidad de las Ciencias Informáticas
 - Siglas: UCI
 - Subordinación: Nacional Provincial Municipal
 - Cargo: Analista
 - Dirección empresa: Carretera a San Antonio
 - Teléfono empresa: 8374536
 - E-mail: laura@uci.cu

At the bottom right, there are two buttons: "Aceptar" and "Cancelar".

Figura 4. Prototipo de interfaz de usuario Crear perfil de cliente

Para el rol de cliente:

Tabla VIII. Descripción textual del requisito Adicionar solicitud

Precondiciones	El usuario debe estar autenticado en el sistema El modelo de características debe estar cargado
Flujo de eventos	
Flujo básico	
1.	El usuario selecciona la opción Productos en la interfaz principal
2.	El sistema muestra un listado de los productos desarrollados en el centro y ofrece la posibilidad de solicitar uno de ellos o realizar una selección de características a partir de las existentes

3. Si el usuario selecciona uno o varios productos de los desarrollados, el sistema registra la solicitud.
4. Concluye el requisito

Pos-condiciones

- 1 El sistema registra la selección de características en un archivo .amf.

Flujos alternativos**Flujo alternativo 3 a. El usuario solicita una selección de características**

- 1 El usuario selecciona la opción Adicionar solicitud en la interfaz principal
- 2 El sistema muestra un listado de las líneas de productos que componen el portafolio.
Se muestra:
Nombre de la línea
- 3 El usuario selecciona la línea que se ajusta a su negocio y presiona el botón Siguiente.
- 4 El sistema muestra el listado las características correspondientes a la línea seleccionada.
(Ver validaciones 1, 2, 3)
Se muestra:
Nombre de característica
- 5 El cliente selecciona las características deseadas y presiona el botón Siguiente.
- 6 El sistema muestra una interfaz para que el usuario seleccione el idioma y la plataforma en la que debe ejecutarse el producto deseado.
- 7 El usuario selecciona las características deseadas y presiona el botón Siguiente.
- 8 El sistema registra las características seleccionadas y muestra un listado de los productos existentes que cumplen con las características seleccionadas.
- 9 El usuario selecciona el producto que desea y presiona el botón Confirmar.
- 10 Concluye el requisito

Flujo alternativo 4 a. El usuario presiona el botón Terminar

- 1 Volver al paso 7 del flujo alternativo 1a.

Pos-condiciones

- 1 Se registra una solicitud de producto
- 2 El sistema registra la selección de características en un archivo .amf.

Validaciones

- 1 Validar que las características obligatorias se muestren marcadas y deshabilitadas
- 2 Validar que las características opcionales se muestren con checkbox
- 3 Validar que las características alternativas se muestren con radio button

Relaciones	Requisitos	Mostrar listado de productos: Integrar FaMa
	Incluidos	Mostar características: Integrar FaMa
	Extensiones	NA

Conceptos	Solicitud	Atributos visibles en la interfaz: Fecha solicitud Nombre cliente Empresa del cliente Listado de características del producto Atributos utilizados internamente: IdProducto estado
Requisitos especiales	NA	
Asuntos pendientes	NA	

Prototipo elemental de interfaz de usuario

The screenshot shows a web form titled "Solicitud" with a blue header. The form contains the following text and elements:

- Fecha solicitud: 05/03/2013
- Nombre cliente: Esteban Álvarez Sánchez
- Empresa: ETECSA
- A section titled "Seleccione el sector en el que se desarrolla su empresa:" containing three radio button options:
 - Sistema para la gestión aduanera
 - Sistema para la gestión bancaria
 - Sistema integral de gestión de entidades
- At the bottom right, there are three buttons: "Aceptar", "Terminar", and "Siguiete".

Solicitud

Fecha solicitud: 05/03/2013
Nombre cliente: Esteban Álvarez Sánchez
Empresa: ETECSA

Seleccione las características que desea para su producto:

- Estructura y Composición
- Contabilidad general
- Costos y procesos
- Cobros y Pagos
- Caja y Banco
- Inventario y Facturación
- Capital Humano

Solicitud

Fecha solicitud: 05/03/2013
Nombre cliente: Esteban Álvarez Sánchez
Empresa: ETECSA

Seleccione el idioma en el desea su producto:

- Español
- Inglés
- Francés
- Alemán

Seleccione la plataforma sobre la que debe ejecutarse su producto:

- Linux
- Windows
- Multiplataforma

Productos existentes

Fecha solicitud: 05/03/2013
 Nombre cliente: Esteban Álvarez Sánchez
 Empresa: ETECSA

El CEIGE cuenta con 12 productos que cumplen con las características seleccionadas:

Nombre	Descripción	
Cedrux	Desarrollo de 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.	<input type="button" value="Ver"/>
Quarxo	Quarxo, es una novedosa solución informática que se construyó sobre la base del sistema SABIC (Sistema automatizado para la banca internacional y de comercio) adaptado para soportar los procesos que se realizan en los bancos con la responsabilidad de canalizar recursos financieros hacia sectores productivos, a través de otras instituciones financieras que actúan como intermediarios.	<input type="button" value="Ver"/>
GINA	El Sistema de Control de Medios de Transporte Internacional desarrollado para la Aduana General de la República de Cuba, por un equipo de profesionales de la Universidad de Ciencias Informáticas y que comercializa la empresa ALBET, el sistema que se encarga de gestionar el control de los medios de transporte internacional marítimos y aéreos en la Aduana General de la República de Cuba.	<input type="button" value="Ver"/>

Para el rol de administrador:

Tabla IX. Descripción textual del requisito Realizar reporte de solicitudes pendientes

Precondiciones	El usuario debe estar autenticado en el sistema y deben existir solicitudes en estado Creado o Modificado.
Flujo de eventos	
Flujo básico	
1	El sistema muestra un listado de las solicitudes cuyo estado es Pendiente (Ver validación 1). Se muestra: Fecha Nombre de cliente Descripción
2	Concluye el requisito
Pos-condiciones	
1	NA
Flujos alternativos	
1	NA
Pos-condiciones	
1	NA
Validaciones	
1	NA

Relaciones	Requisitos Incluidos	NA
	Extensiones	NA
Conceptos	Solicitud	Atributos visibles en la interfaz: Fecha Nombre de cliente Descripción Atributos utilizados internamente: IdSolicitud
Requisitos especiales	NA	
Asuntos pendientes	NA	

Prototipo elemental de interfaz de usuario

Reporte de solicitudes pendientes		
Fecha	Cliente	Descripción
27/10/2012	Laura Rodríguez Leveque	Solicitud de un sistema para la gestión de entidades CEDRUX
26/12/2012	Esteban Álvarez Santos	Solicitud de un sistema para la gestión de entidades CEDRUX
09/06/2013	Roberto Sánchez Torres	Solicitud de un sistema para la gestión financiera
09/06/2013	Luisa Fernández Torres	Solicitud de un sistema para la gestión aduanera
09/06/2013	Isela Pérez Rodríguez	Solicitud de un sistema para la gestión financiera

Tabla X. Descripción textual del requisito Integrar FaMa

Precondiciones	Se ha solicitado un producto
Flujo de eventos	
Flujo básico	
1	El usuario presiona la opción Productos en la interfaz principal
2	El sistema obtiene de la herramienta FaMa todos los productos que están representados en el modelo de características para mostrarlos al cliente
3	El usuario selecciona la opción adicionar solicitud en la interfaz principal
4	El sistema obtiene de la herramienta FaMa las características que el usuario puede

	seleccionar	
5	El usuario selecciona las características deseadas	
6	El sistema registra la selección en un archivo .amf y con él, obtiene de la herramienta FaMa un listado de los productos que reúnen las características seleccionadas	
7	El usuario solicita ver el reporte de características más comunes	
8	El sistema obtiene a partir de cada característica la cantidad de productos en los que aparece.	
9	Concluye el requisito	
Pos-condiciones		
1	NA	
Flujos alternativos		
1	NA	
Pos-condiciones		
1	NA	
Validaciones		
1	NA	
Relaciones	Requisitos	NA
	Incluidos	
	Extensiones	NA
Conceptos	FaMa	Atributos visibles en la interfaz: NA Atributos utilizados internamente: Razonadores Operaciones
Requisitos especiales	NA	
Asuntos pendientes	NA	

2.5. Diagrama de clases del diseño

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. La figuras 5, 6 y 7 muestran el diagrama de clases del diseño correspondiente a la agrupación de requisitos Gestionar perfiles de clientes, Gestionar solicitudes

de productos y Realizar reporte de solicitudes pendientes. Las páginas cliente incluyen la librería ExtJs que contiene los ficheros de las vistas del sistema así como el archivo .js correspondiente a cada página cliente. Estas páginas están compuestas por formularios para la entrada de datos, los mismos son enviados a la página index.php que gestiona el flujo de operaciones entre las vistas y el controlador. Las clases Zend_Controller_Secure y ZendExt_Model gestionan la seguridad en la clase controladora y la del modelo (encargada del acceso a la base de datos).

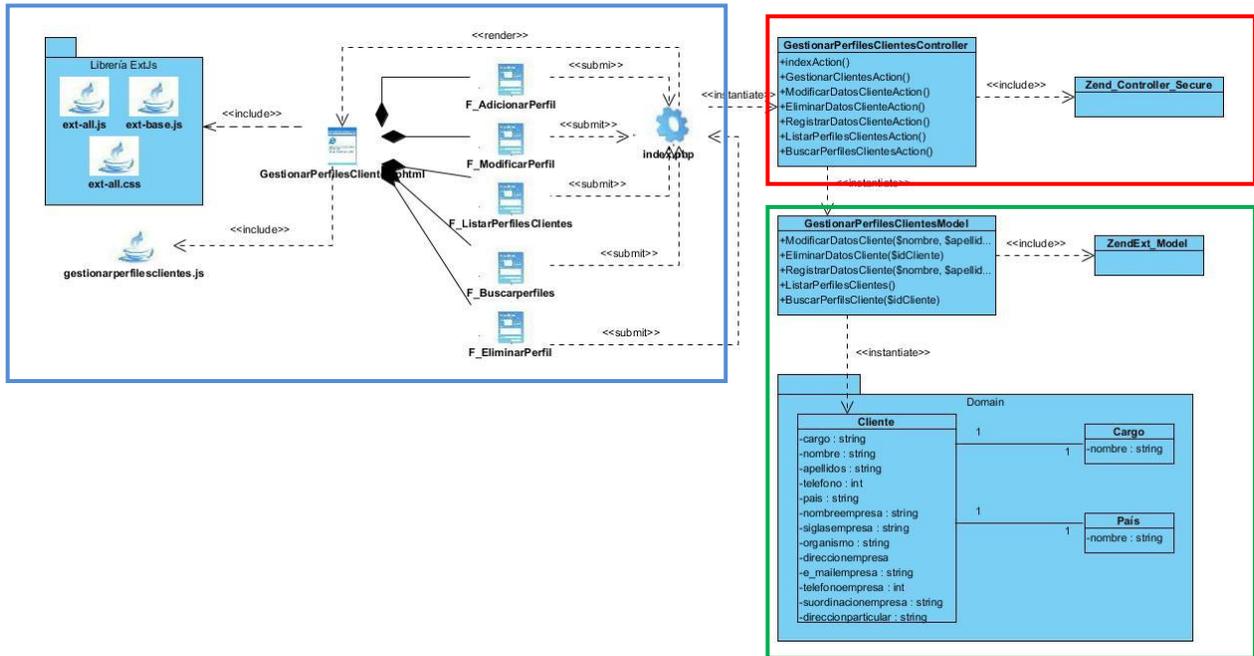


Figura 5. Diagrama de clases del diseño Gestionar perfiles de clientes

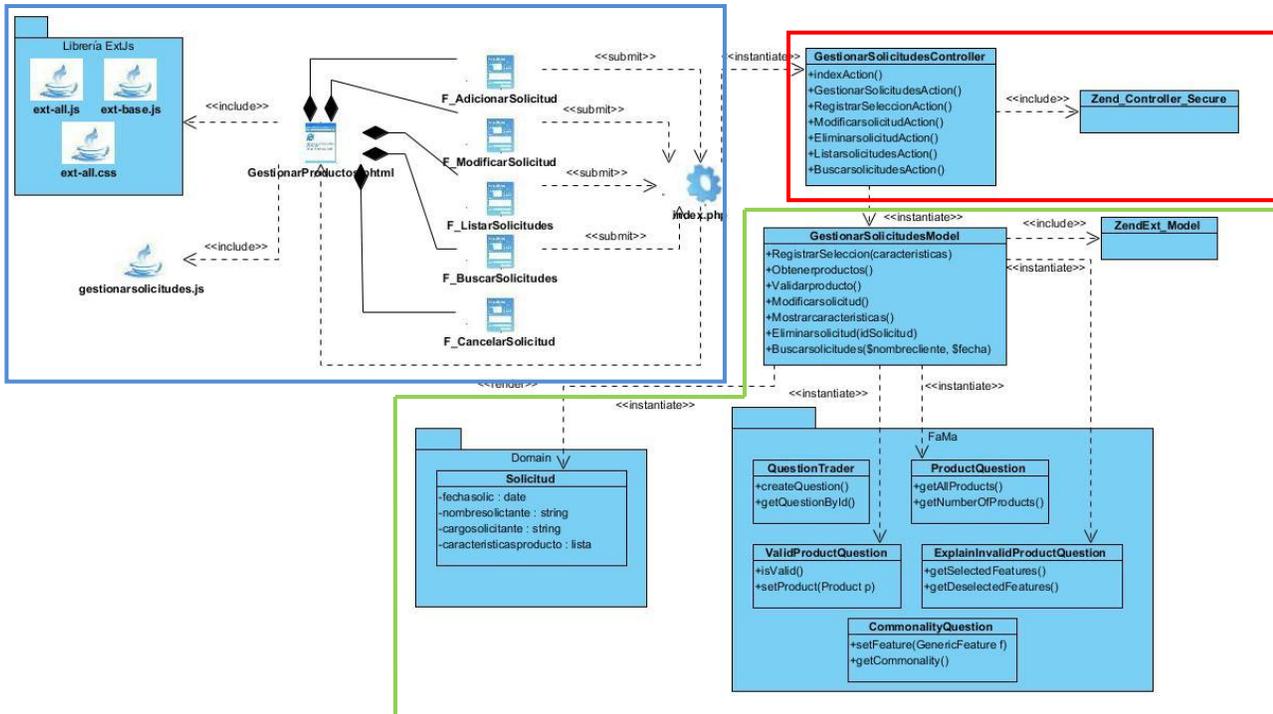


Figura 6. Diagrama de clases del diseño Gestionar solicitudes de productos

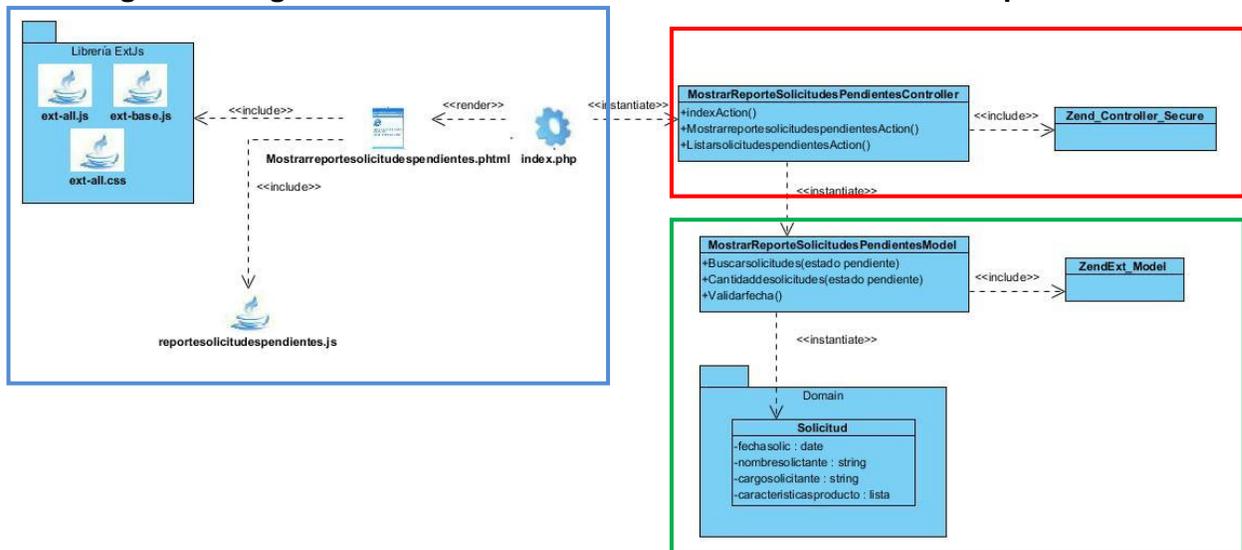


Figura 7. Diagrama de clases del diseño Realizar reporte de solicitudes pendientes

En los diagramas de clases del diseño mostrados anteriormente se evidencia el uso del patrón MVC donde el modelo es representado de color verde, la vista en azul y el controlador en rojo. Además están presentes el patrón experto en la definición de las clases GestionarClientesController, GestionarClientesModel, GestionarSolicitudesController, GestionarSolicitudesModel, MostarReporteProductosVendidosController y MostarReporteProductosVendidosModel, el patrón alta

cohesión se evidencia en el diseño de la solución donde se agruparon las clases en dependencia de los requerimientos funcionales, según la premisa de que cada clase debe implementar las operaciones que estén sobre la misma área funcional. En las clases GestionarClientesController, GestionarSolicitudesController y MostarReporteProductosVendidosController se evidencia el uso del patrón controlador y la misma tendrá a cargo la responsabilidad de manejar los eventos dentro del sistema.

Diagrama de clases del diseño Realizar reporte de características más comunes. (Ver Anexo 6)

Diagrama de clases del diseño Realizar reporte de productos vendidos. (Ver Anexo 7)

Diagrama de clases del diseño Realizar reporte de productos solicitados. (Ver Anexo 8)

Diagrama de clases del diseño Realizar reporte de mejores clientes. (Ver Anexo 9)

2.6. Modelo de datos

Un diagrama o modelo entidad-relación (DER), es un esquema gráfico para el modelado de datos de un sistema de información. Está compuesto por entidades que poseen características llamadas atributos y el enlace entre entidades se representa mediante relaciones. En el DER del sistema de gestión del portafolio de productos del CEIGE se evidencia el empleo del patrón bajo acoplamiento cuando se definen las clases persistentes, entre las cuales se establecen las relaciones necesarias de manera que sean independientes y reutilizables para reducir el impacto de los cambios y aumentar la productividad. El mismo se realizó basándose en el modelo conceptual previamente confeccionado, quedando un total de 5 tablas donde todas son propias del sistema.

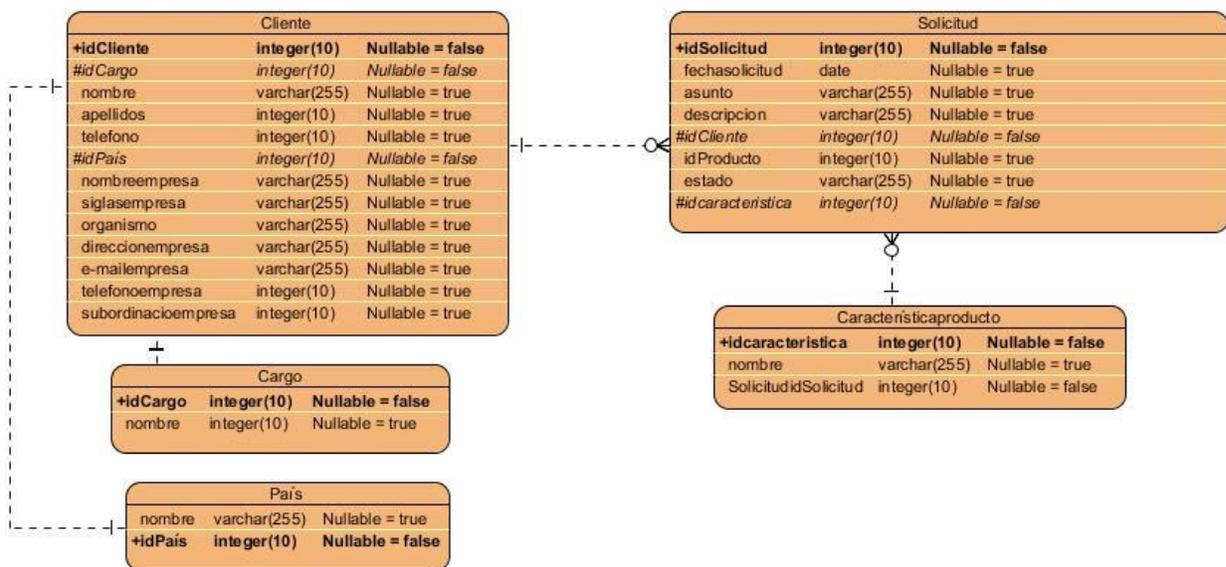


Figura 8. Modelo de datos

Conclusiones parciales

Luego de un análisis de lo expuesto en el capítulo, se arriban a las siguientes conclusiones:

- Se realizó el modelo conceptual del sistema para la gestión del portafolio de productos del CEIGE, lo cual permitió identificar los principales conceptos y relaciones que definen el dominio del problema.
- Con el objetivo de garantizar la correcta ejecución de la aplicación y mediante las técnicas de captura de requisitos se identificaron 16 requisitos funcionales, 6 de complejidad media y 10 de complejidad baja.
- Con la construcción de los prototipos de interfaz se facilita la comprensión del funcionamiento del sistema.
- Con el diagrama de clases del diseño se evidencia el uso de los patrones MVC, experto, alta cohesión, bajo acoplamiento y controlador.
- Los artefactos generados constituyen las bases para una futura implementación del sistema de gestión del portafolio de productos del CEIGE, a partir del análisis de las características de sus productos.

CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

Introducción

En el presente capítulo se realizará la validación de los requisitos funcionales, empleando las técnicas: revisiones técnicas formales y construcción de prototipos, permitiendo que la solución se ajuste a los intereses del cliente. La aplicación de las técnicas de validación evita la entrega de requisitos incompletos y el fracaso del proyecto debido a un mal manejo de los cambios. También se valida el diseño mediante la aplicación de las métricas Tamaño Operacional de Clase (TOC) y Relación entre Clases (RC) para estimar la efectividad del proceso y mejorar la calidad del trabajo.

3.1. Validación de requisitos

Para validar los requisitos funcionales se utilizó la técnica de construcción de prototipos realizando varias iteraciones para las revisiones de los mismos. En cada iteración se mostró cada una de las interfaces de usuario acompañada de una breve explicación verbal del funcionamiento de la misma. La primera iteración arrojó como resultado ajustes en los prototipos: crear perfil de cliente y definir producto, además se incluyeron nuevos requisitos: realizar reporte de solicitudes pendientes, realizar reporte de características más comunes, realizar reporte de productos vendidos, realizar reporte de productos solicitados y realizar reporte de mejores clientes. En una segunda iteración fue necesario realizar cambios mínimos a los prototipos correspondientes a los requisitos: listar perfiles de clientes, buscar productos y buscar perfil de cliente. Se realizó la revisión de la descripción de los requisitos poniendo en práctica la técnica revisiones técnicas formales con el objetivo de comprobar si el comportamiento del sistema descrito se ajusta a las necesidades de los usuarios. Además se aplicaron las métricas para evaluar la especificación de los requisitos obteniéndose los siguientes resultados:

Tabla XI. Aplicación de las métricas de especificación de requisitos

	Correctitud	Complejidad	Consistencia
Total de requisitos revisados	16	16	16
Total de observaciones de negocio	2	-	-
Total de observaciones de formato	-	0	-
Total de observaciones de consistencia	-	-	0
Resultados	0.0625	1	0

Donde:

Correctitud = $0.0625 < 0,10$

Compleitud = $1 > 0,90$

Consistencia = $0 < 0,20$

Por lo que los resultados en la aplicación de los criterios para la evaluación y aceptación de los requisitos se consideran satisfactorios.

3.2. Validación del diseño

3.3. Aplicación de la métrica TOC

Tabla XII. Aplicación de la métrica TOC

No.	Nombre de las clases	Cantidad de procedimientos
1	GestionarPerfilesClientesController	7
2	GestionarPerfilesClientesModel	5
3	GestionarSolicitudesController	7
4	GestionarSolicitudesModel	8
5	MostrarReporteSolicitudesPendientesController	3
6	MostrarReporteSolicitudesPendientesModel	3
7	MostarReporteCaracteristicasComunesController	3
8	MostarReporteCaracteristicasComunesModel	2
9	MostarReporteProductosVendidosController	3
10	MostarReporteProductosVendidosModel	1
11	MostarReporteProductosSolicitadosController	3
12	MostarReporteProductosSolicitadosModel	0
13	MostrarReporteMejoresClientesController	3
14	MostrarReporteMejoresClientesModel	2

Dado que el promedio es igual a 3.57 se obtuvieron los siguientes resultados:

Tabla XIII. Resultados del atributo Responsabilidad de la métrica TOC

Responsabilidad	Cantidad de clases	Por ciento
-----------------	--------------------	------------

Baja	10	71.43
Media	3	21.43
Alta	1	7.14

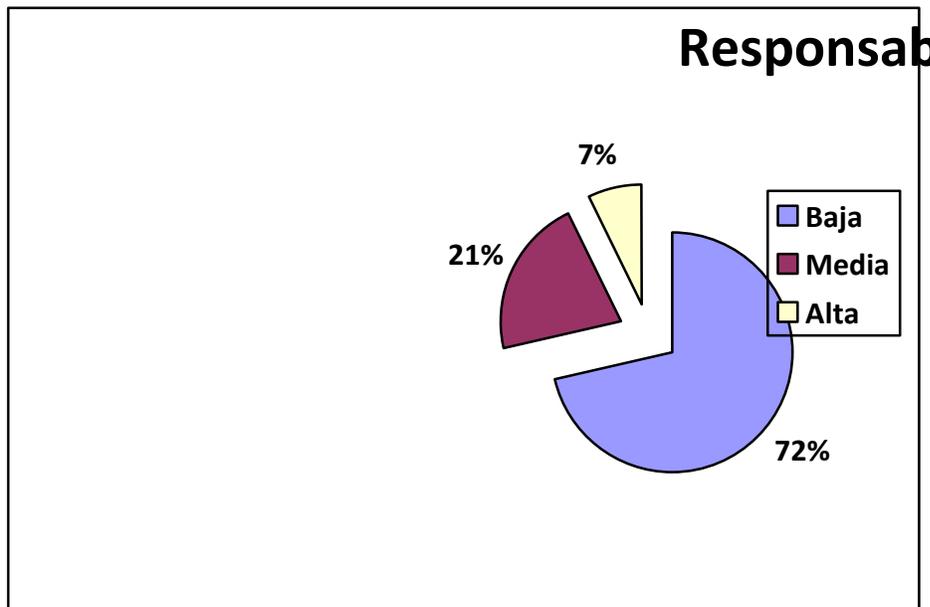


Figura 9. Gráfico del atributo Responsabilidad de la métrica TOC

Tabla XIV. Resultados del atributo Complejidad de implementación de la métrica TOC

Complejidad de implementación	Cantidad de clases	Porcentaje
Baja	10	71.43
Media	3	21.43
Alta	1	7.14

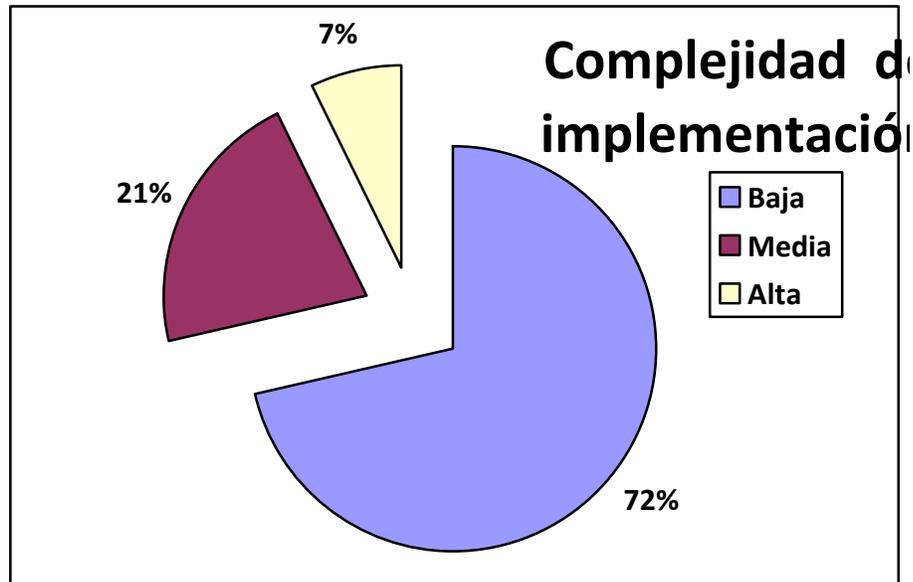


Figura 10. Gráfico del atributo Complejidad de implementación de la métrica TOC

Tabla XV. Resultados del atributo Reutilización de la métrica TOC

Reutilización	Cantidad de clases	Porcentaje
Baja	1	7.14
Media	3	21.43
Alta	10	71.43

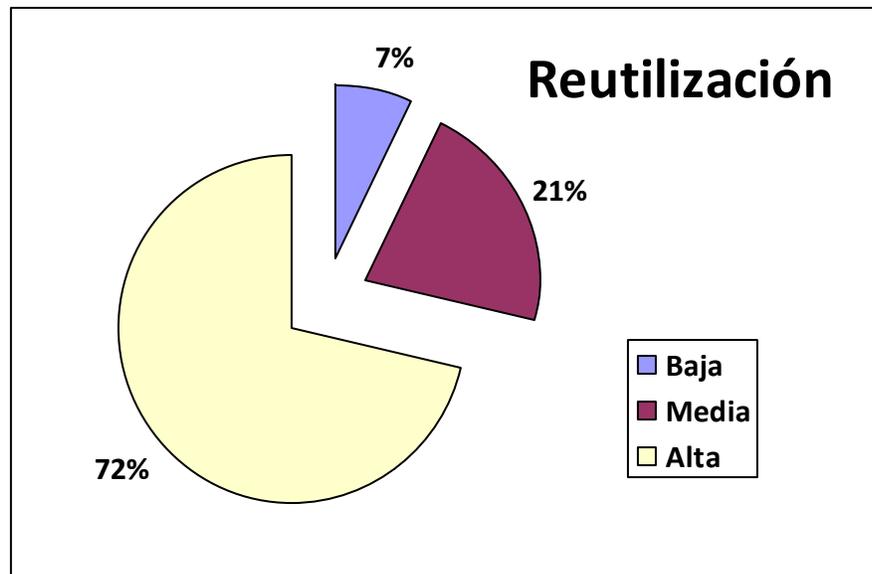


Figura 11. Gráfico del atributo Reutilización de la métrica TOC

3.4. Aplicación de la métrica RC

Tabla XVI. Aplicación de la métrica RC

No.	Clases	Cantidad de relaciones de uso
1	GestionarPerfilesClientesController	1
2	GestionarPerfilesClientesModel	1
3	GestionarSolicitudesController	1
4	GestionarSolicitudesModel	5
5	MostrarReporteSolicitudesPendientesController	1
6	MostrarReporteSolicitudesPendientesModel	1
7	MostarReporteCaracteristicasComunesController	2
8	MostarReporteCaracteristicasComunesModel	2

9	MostarReporteProductosVendidosController	1
10	MostarReporteProductosVendidosModel	1
11	MostarReporteProductosSolicitadosController	1
12	MostarReporteProductosSolicitadosModel	1
13	MostrarReporteMejoresClientesController	1
14	MostrarReporteMejoresClientesModel	1

Dado que el promedio es igual a 1.57 se obtuvieron los siguientes resultados:

Tabla XVII. Resultados del atributo Acoplamiento de la métrica RC

Acoplamiento	Cantidad de clases	Porcentaje
Ninguno	0	0
Bajo	11	78.57
Medio	2	14.28
Alto	1	7.14

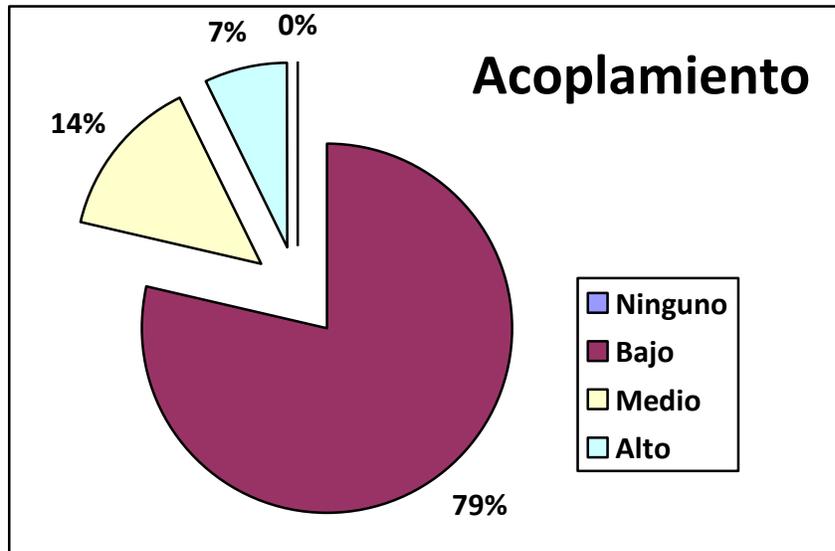


Figura 12. Gráfico del atributo Acoplamiento de la métrica RC

Tabla XVIII. Resultados del atributo Cantidad de pruebas de la métrica RC

Cantidad de pruebas	Cantidad de clases	Porcentaje
Baja	11	78.57
Media	2	14.28
Alta	1	7.14

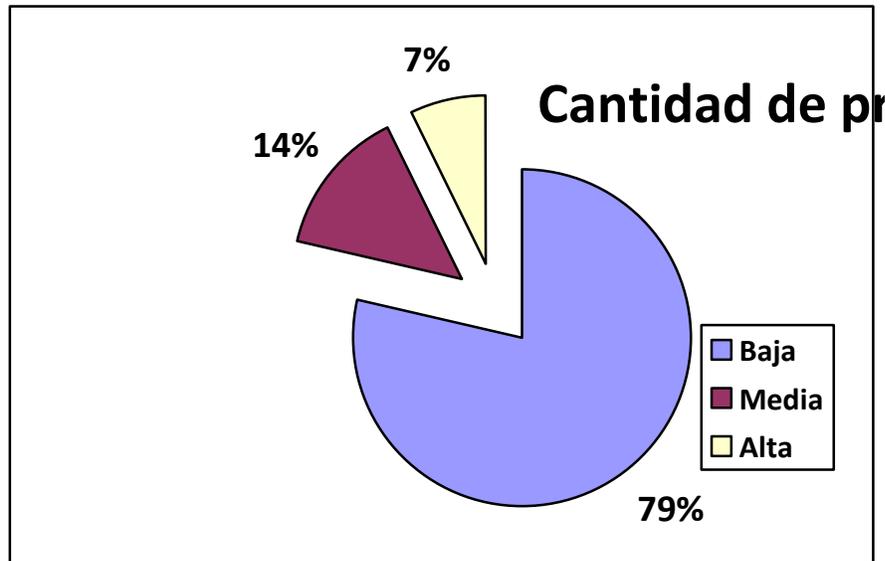


Figura 13. Gráfico del atributo Cantidad de pruebas de la métrica RC

Tabla XIX. Resultados del atributo Complejidad de implementación de la métrica RC

Complejidad de implementación	Cantidad de clases	Porcentaje
Baja	11	78.57
Media	2	14.28
Alta	1	7.14

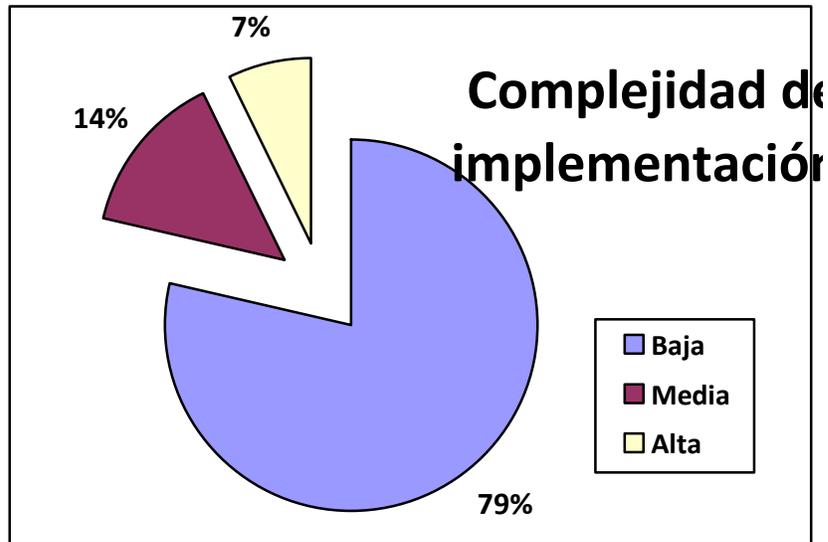


Figura 14. Gráfico del atributo Complejidad de implementación de la métrica RC

Tabla XX. Resultados del atributo Reutilización de la métrica RC

Reutilización	Cantidad de clases	Porcentaje
Baja	1	7.14
Media	2	14.28
Alta	11	78.57

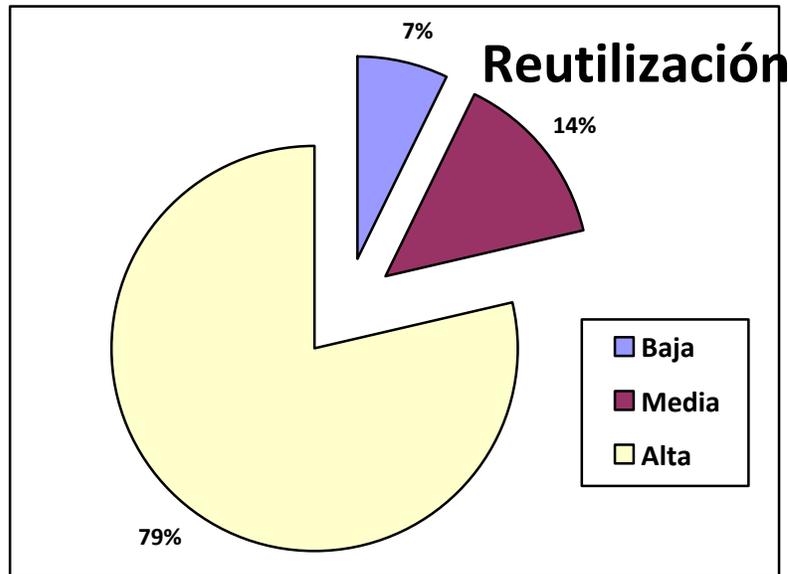


Figura 15. Gráfico del atributo Reutilización de la métrica RC

3.5. Resultado de la evaluación de las métricas

Analizando los resultados obtenidos en la evaluación de la métrica TOC se puede concluir que de un total de 14 clase, para un promedio de 3.57, el diseño de la solución tiene una calidad aceptable ya que el 71.43% de las clases poseen evaluaciones positivas en los atributos de calidad (Responsabilidad, Complejidad de implementación y Reutilización). La responsabilidad de las clases y la complejidad de implementación son bajas, lo que permite un alto nivel de reutilización.

Los resultados obtenidos al evaluar la métrica RC permiten concluir que, para un promedio de 1.57, el diseño de la solución tiene una calidad aceptable ya que el 78.57% de las clases poseen evaluaciones positivas en los atributos de calidad (Acoplamiento, Cantidad de pruebas, Complejidad de implementación y Reutilización), favoreciendo de esta manera la reutilización de las clases así como la modificación e implementación del diseño.

Conclusiones parciales

Validar la calidad del análisis y diseño del sistema para la gestión del portafolio de productos del CEIGE fue el elemento clave del capítulo que recién concluye. En ese sentido:

- ✓ Se aplicaron las técnicas construcción de prototipos y revisiones técnicas formales para validar los requisitos funcionales.
- ✓ Se aplicaron las métricas para validar el diseño: Relaciones entre Clases y Tamaño Operacional de la Clase, las cuales arrojaron valores satisfactorios para cada uno de los indicadores correspondientes.
- ✓ El análisis y diseño del sistema para la gestión del portafolio de productos del CEIGE desde el punto de vista funcional cumple con los requisitos capturados y especificados en las primeras etapas de desarrollo a partir de las expectativas del cliente.

CONCLUSIONES GENERALES

Con el presente trabajo de diploma fueron alcanzados los objetivos planteados, se logró realizar el análisis y diseño del sistema para la gestión del portafolio de productos del CEIGE sentando las bases para su posterior implementación:

- El análisis de los sistemas de gestión del portafolio de productos, permitió determinar que los mismos no serán utilizados para el desarrollo de la investigación pero se identificaron sus ventajas y funcionalidades comunes que sirven de soporte a la solución propuesta.
- El estudio de las técnicas de análisis de dominio basada en características se consideran de utilidad para la gestión dinámica de la identificación de productos.
- Se seleccionó FaMa como herramientas para el análisis automatizado de modelos de características por ser la única que emplea los tres razonadores: BDD, SAT y CPS
- Para guiar el proceso de desarrollo de software se seleccionó el Modelo de desarrollo orientado a componentes del CEIGE.
- Se definió el entorno de desarrollo a utilizar para la propuesta de solución, incluyendo las herramientas, lenguajes y tecnologías necesarias.
- Se realizó un estudio de los conceptos fundamentales vinculados a la gestión de portafolios de productos, las líneas de productos y los modelos de características. Además el análisis de los sistemas de gestión del portafolio de productos, teniendo en cuenta sus principales ventajas y funcionamiento, permitió una visión integral de la solución propuesta.
- Se realizó el modelo conceptual del sistema para la gestión del portafolio de productos del CEIGE, lo cual permitió identificar los principales conceptos y relaciones que definen el dominio del problema.
- Con el objetivo de garantizar la correcta ejecución de la aplicación y mediante las técnicas de captura de requisitos se identificaron 16 requisitos funcionales, 6 de complejidad media y 10 de complejidad baja.
- Con la construcción de los prototipos de interfaz se facilita la comprensión del funcionamiento del sistema.
- Con el diagrama de clases del diseño se evidencia el uso de los patrones MVC, experto, alta cohesión, bajo acoplamiento y controlador.

- Los artefactos generados constituyen las bases para una futura implementación del sistema de gestión del portafolio de productos del CEIGE, a partir del análisis de las características de sus productos.
- Se aplicaron las técnicas tormenta de ideas y entrevistas para la captura de requisitos, obteniendo los requisitos funcionales que responden a las necesidades del cliente. Se logró obtener el modelo de negocio, el modelo conceptual y el modelo de datos.
- Se emplearon los patrones de asignación de responsabilidades así como el arquitectónico modelo-vista-controlador para realizar el diseño de la solución propuesta.
- Se realiza la validación del diseño obteniendo resultados positivos, lo que disminuye las dificultades para la posterior implementación del sistema.

RECOMENDACIONES

Luego del trabajo realizado y los resultados obtenidos se recomienda:

- Realizar la implementación del sistema basándose, en el diseño propuesto.
- Consultar las referencias bibliográficas utilizadas en el presente trabajo para profundizar en el tema.

REFERENCIAS BIBLIOGRÁFICAS

1. Benavides, D. (2009). Automated Analysis of Feature Models: A Detailed Literature Review . Sevilla, España.
2. BigAgi. (2010). BPMN Business Process Modeling Notation.
3. Clements, P. (2001). Software Product Lines: Practices and Patterns. Addison-Wesley.
4. Corporation, C. (2009). "COMPUWARE CHANGEPOINT."
5. Cruz, M. Q. (2011). Análisis y diseño del componente Matriz y Proyección del subsistema Mantenimiento de Cedrux. La Habana: UCI.
6. David Benavides, P. T.-C. (2007). FAMA: Tooling a Framework for the Automated Analysis of Feature Models. Sevilla, Universidad de Sevilla.: Departamento de Lenguajes y Sistemas de Computadoras.
7. Deltek, I. (2012). "Deltek." . Recuperado el 2012, de <http://www.deltek.com/>.
8. Díaz, O. (2010). Lineas de productos de software. Facultad de Informatica, Universidad del Pais Vasco.
9. Expert, S. (2012). "SE Portfolio, Gestión de Portafolio."
10. ISA. (2001). Manual de usuario de FaMA.
11. JAZD Markets, I. (2012). "Tech Direct." . Obtenido de <http://www.jazdtech.com/techdirect/company/ProSight/ProSight-Portfolios-Project-Portfolio-Management-Software.htm?supplierId=60012311&productId=60016805>.
12. Lago, R. (2007). Patrones de diseño software. Obtenido de <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
13. Larman, G. (2004). UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.
14. Monte, E. M. (2011). Propuesta de un entorno tecnológico de descubrimiento y composición de servicios web semánticos para la plataforma de interoperabilidad del Órgano Justicia-MININT de Cuba. La Habana, Universida de las Ciencias Informáticas.
15. Obregón, I. W. (2012). Modelo de desarrollo de software.
16. Orta, L. V. (2012). Análisis y diseño de un Sistema para la Administración de un Repositorio de Activos de Software aplicable al proceso de desarrollo del CEIGE.
17. Pupo, Y. C. (2010). Libro de Ayuda del Marco de Trabajo Sauxe, En su versión 2.0.
18. Ramírez, J. R. (2009). HERRAMIENTA PARA EL MODELADO Y CONFIGURACIÓN DE MODELOS DE CARACTERÍSTICAS.
19. Ross, F. C. (2009). Análisis automático de líneas de producto software usando distintos modelos de variabilidad, supervisado por David Benavides y Antonio Ruis Cortés.
20. Sommerville, I. (2009). Ingeniería de Software. Séptima Edición.

21. Systemes, D. (2012). Dassault Systèmes. Obtenido de <http://www.3ds.com/>.
22. Trujillo, O. D. (2010). Fábricas de Software: experiencias, tecnologías y organización 2º edición .
23. Vázquez, M. A. (2011). Analisis y Diseño del proceso Competencias laborales del subsistema Capital Humano del sistema Cedrux.
24. Work, V. P. (2012). Visual Paradigm. Obtenido de <http://www.visual-paradigm.com/>.
25. Neo4j, Equipo. 2013. Manual de Neo4j. 2013.
26. HyperGraph. 2013. HyperGraph Database. [En línea] 2013. [Citado el: 7 de mayo de 2013.] <http://www.hypergraphdb.org/index>.
27. InfoGrid. 2013. InfoGrid. [En línea] 2013. <http://infogrid.org/trac/>.
28. Hernández, Odaisa Alfonso. 2011. Análisis y Diseño del módulo Bloqueo del subsistema de autoaprendizaje para el Laboratorio Virtual de Sistemas Operativos. 2011.
29. Evolution Solutions Co., Ltd. <http://www.evolus.vn>. [En línea] [Citado el: 15 de 05 de 2013.] <http://www.evolus.vn/Pencil/Screenshots.html>.
30. Pressman, Roger S. 2002. Ingeniería de Software. Un enfoque práctico. s. l.:McGraw-Hill. 2002.
31. Ivisate, Annerys Armenteros. 2011. Análisis y diseño del componente Ejecución del subsistema Mantenimiento de Cedrux. 2011.
32. Company, Big Lever. 2007. The Pragmatic 3-Tiered Software Product Line Methodology. 2007.
33. Marketing. 2012. marketing-xxi. [En línea] 2012. <http://www.marketing-xxi.com/analisis-de-la-cartera-producto-mercado-analisis-portfolio-o-bcg-20.htm>.
34. DRAE. 2013. Diccionario de la Real Academia. [En línea] 2013. <http://www.rae.es/>.
35. Corporation, Parametric Technology. 2006. *Gestión de la cartera* . 2006.
36. Zürich, P&P Software and ETH. 2004. FeatureIDE. Manual de usuario . 2004.

GLOSARIO DE TÉRMINOS

BPMN: (Business Process Modeling Notation) es una notación gráfica estandarizada que permite el modelado de procesos de negocio.

CASE: acrónimo de Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora.

Clase: es la unidad básica que encapsula toda la información de un objeto (un objeto es una instancia de una clase).

Cliente: persona que ya ha comprado productos o adquirido servicios; se diferencia de un “consumidor”, en que el consumidor no necesariamente nos ha comprado o adquirido nuestros productos o servicios.

Configuración de características: conjunto de características que describen a un miembro de la línea de productos de software, este contiene una característica si y solo si esa característica está en su configuración. Una configuración es permitida en un modelo de características si y solo si esta no viola las restricciones impuestas por el modelo.

Grafo: conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

Java: lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++.

Linux: es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo.

Metodología: conjunto de procedimientos racionales utilizados para alcanzar una gama de objetivos que rigen en una investigación científica, una exposición doctrinal o tareas que requieran habilidades, conocimientos o cuidados específicos. Alternativamente puede definirse la metodología como el estudio o elección de un método pertinente para un determinado objetivo.

Mercado: desde el punto de vista de la economía, mercado es el lugar donde se juntan compradores y vendedores para realizar transacciones de bienes y servicios, pero desde el punto de vista del marketing, mercado es el conjunto de personas u organismos con necesidades o deseos a satisfacer.

Mercadotecnia: conjunto de acciones que se pueden realizar para todo lo referente a la relación que existe entre el mercado (los consumidores) y un negocio o empresa, por ejemplo, acciones tales como la recopilación de información procedente del mercado (por ejemplo, las necesidades o gustos de los consumidores), el diseño de productos de acuerdo a dichas necesidades o gustos, la información sobre la existencia de dichos productos a los consumidores, y la distribución o venta de dichos productos.

Métrica: cualquier medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software o un sistema de información, generalmente para realizar comparativas o para la planificación de proyectos de desarrollo.

Negocio: cualquier ambiente o entorno en cual está enmarcado el problema.

Oferta: Oferta de mercancías o prestaciones de servicios por parte de una entidad a un cliente determinado.

Producto: bien o servicio que un negocio o empresa vende u ofrece a los consumidores. Un producto puede ser un bien o un servicio, pero usualmente se utiliza el término “productos” sólo para hacer referencia a los “bienes”.

Segmento de mercado: Segmento de mercado que la empresa decide captar, satisfacer y/o servir, dirigiendo hacia él su programa de marketing; con la finalidad, de obtener una determinada utilidad o beneficio.

Servicio: los servicios son actividades identificables, intangibles y perecederas que son el resultado de esfuerzos humanos o mecánicos que producen un hecho, un desempeño o un esfuerzo que implican generalmente la participación del cliente y que no es posible poseer físicamente, ni transportarlos o almacenarlos, pero que pueden ser ofrecidos en renta o a la venta; por tanto, pueden ser el objeto principal de una transacción ideada para satisfacer las necesidades o deseos de los clientes.

UNIX: es un sistema operativo multitarea, multiusuario que trabaja y funciona de manera similar a Linux, se caracteriza principalmente por ser muy robusto y correr en diversas arquitecturas y plataformas. El sistema operativo comenzó en los Laboratorios Bell, siendo la continuación de un proyecto para crear un sistema operativo multiusuario denominado MULTICS.

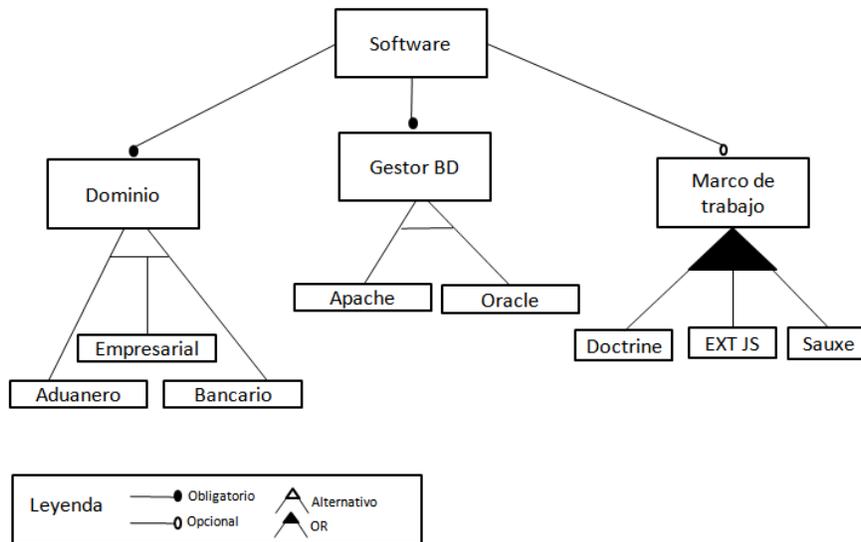
Validación: confirmación mediante examen y aportación de pruebas objetivas que se cumplen los requisitos concretos para su uso determinado.

Venta: Es la acción y efecto de vender. Contrato en virtud del cual se transfiere a dominio ajeno una cosa propia por el precio pactado.

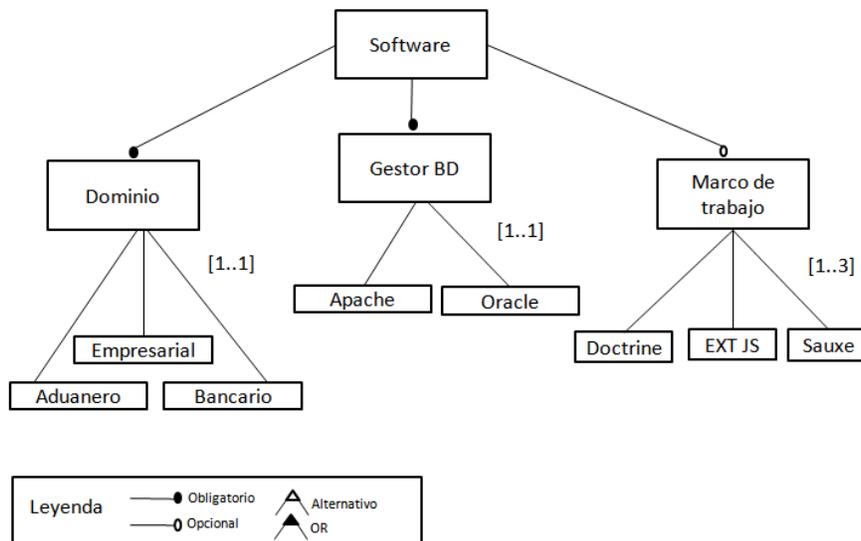
XML: siglas en inglés de Extensible Markup Language (lenguaje de marcas extensibles), es un metalenguaje extensible de etiquetas des arrollado por el World Wide Web Consortium (W3C).

ANEXOS

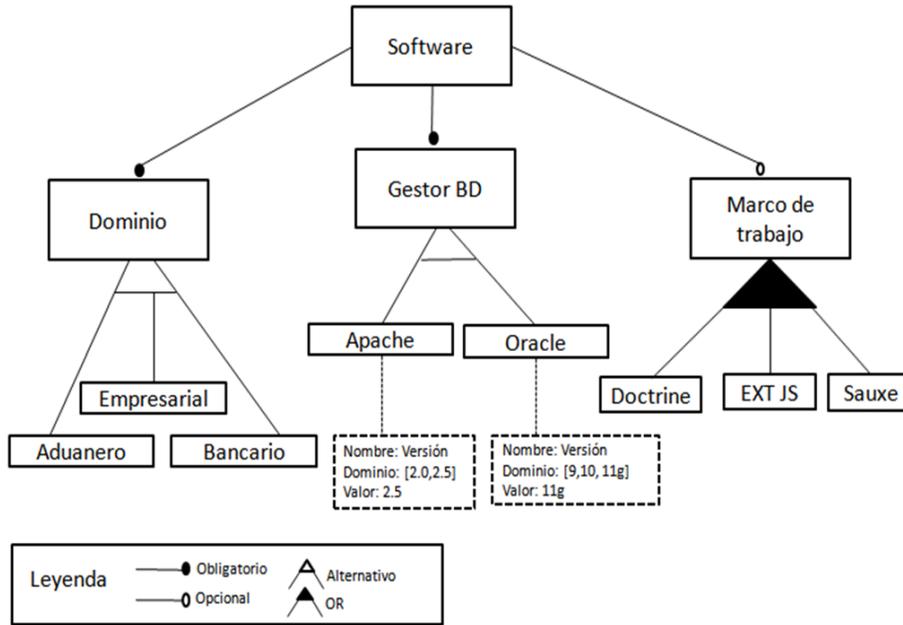
Anexo 1. Modelo de características básico



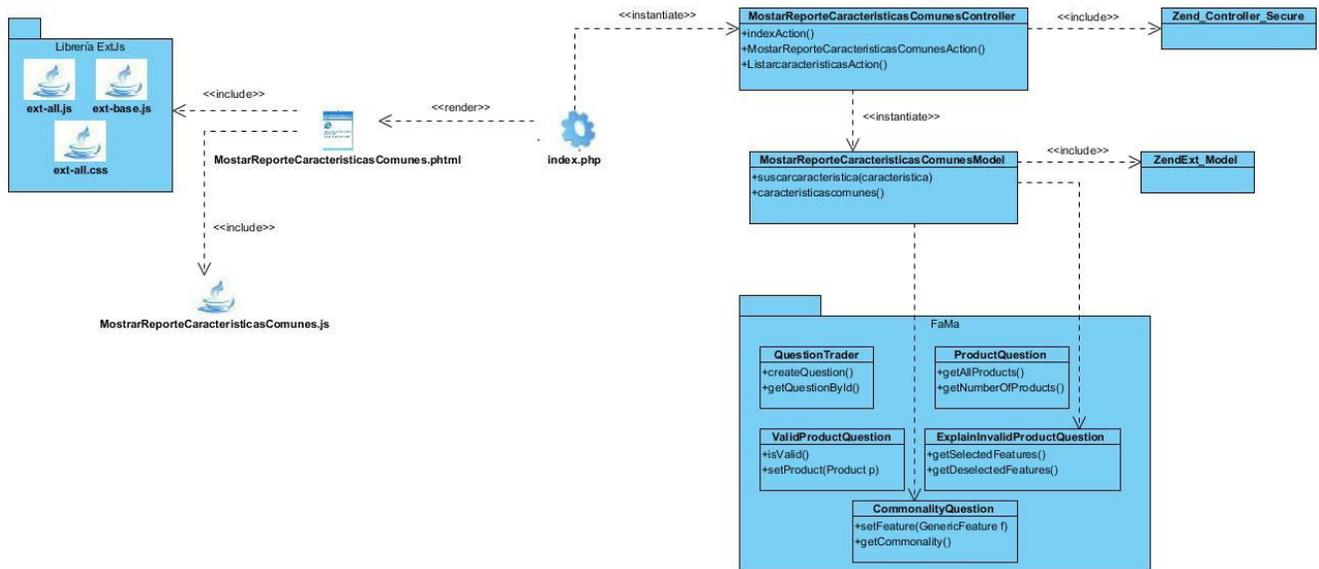
Anexo 2. Modelo de características con cardinalidad y multiplicidad



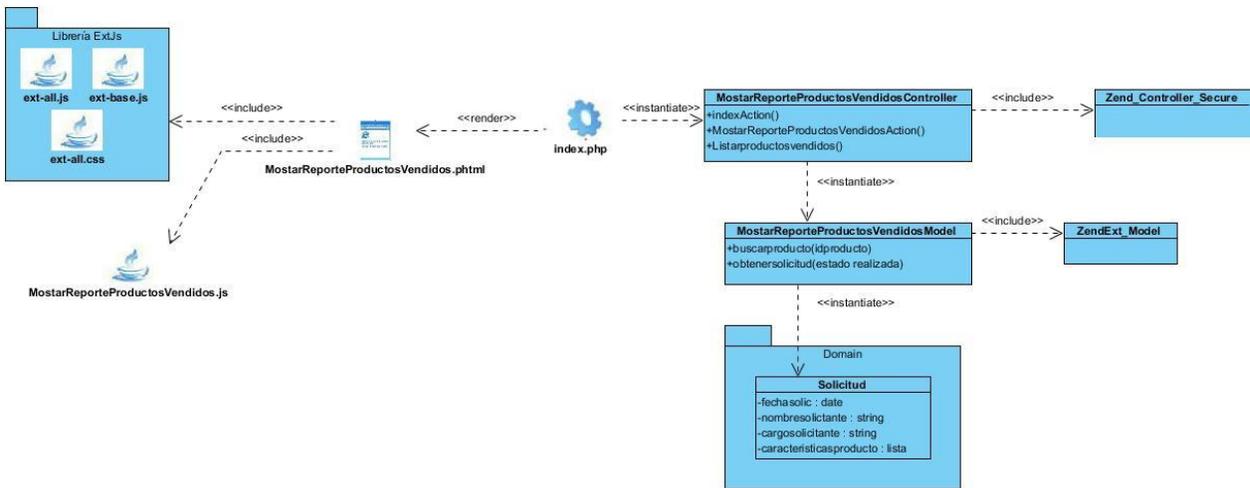
Anexo 3. Modelo de características extendido



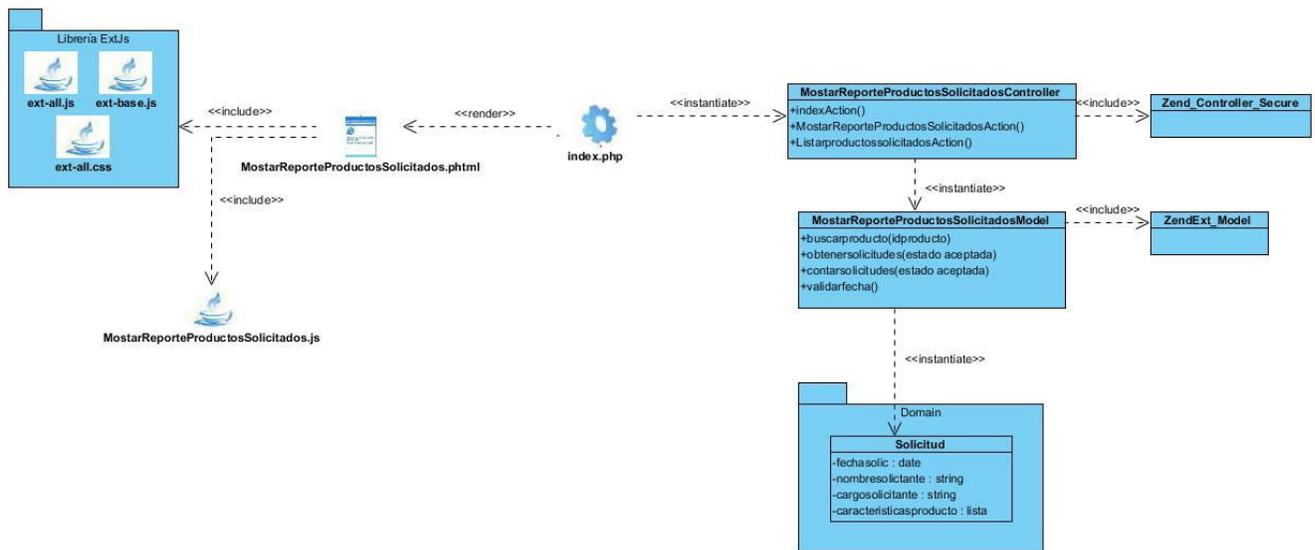
Anexo 6. Diagrama de clases Realizar reporte de características más comunes



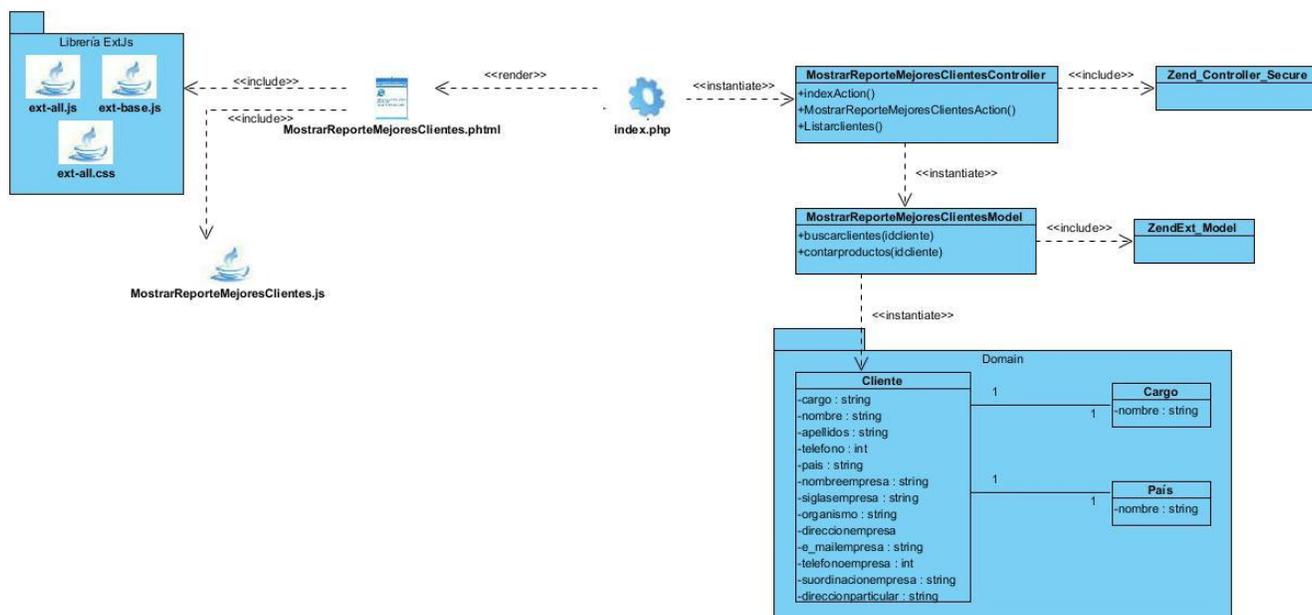
Anexo 7. Diagrama de clases Realizar reporte de productos vendidos



Anexo 8. Diagrama de clases Realizar reporte de productos solicitados



Anexo 9. Diagrama de clases Realizar reporte de mejores clientes



Anexo 10. Complejidad de los requisitos

Requisitos	Complejidad
RF1.1. Crear perfil de cliente	Media
RF1.2. Modificar perfil de cliente	Media
RF1.3. Eliminar perfil de cliente	Media
RF1.4. Listar perfiles de clientes	Media
RF1.5. Buscar perfiles de clientes	Media
RF2.1. Adicionar solicitud	Media
RF2.2. Modificar solicitud	Baja
RF2.3. Cancelar solicitud	Baja
RF2.4. Listar solicitudes	Baja
RF2.5. Buscar solicitudes	Baja
RF3. Realizar reporte de solicitudes pendientes	Baja
RF4. Realizar reporte de productos vendidos	Baja
RF5. Realizar reporte de productos solicitados	Baja
RF6. Realizar reporte de mejores clientes	Baja
RF7. Realizar reporte de características más comunes	Baja
RF8. Integrar FaMa	Baja