

Universidad de las Ciencias Informáticas
Facultad 1



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

Sistema de gestión de relaciones con clientes “COMSYS”

Autor:

Carlos Alberto Castillo Pérez

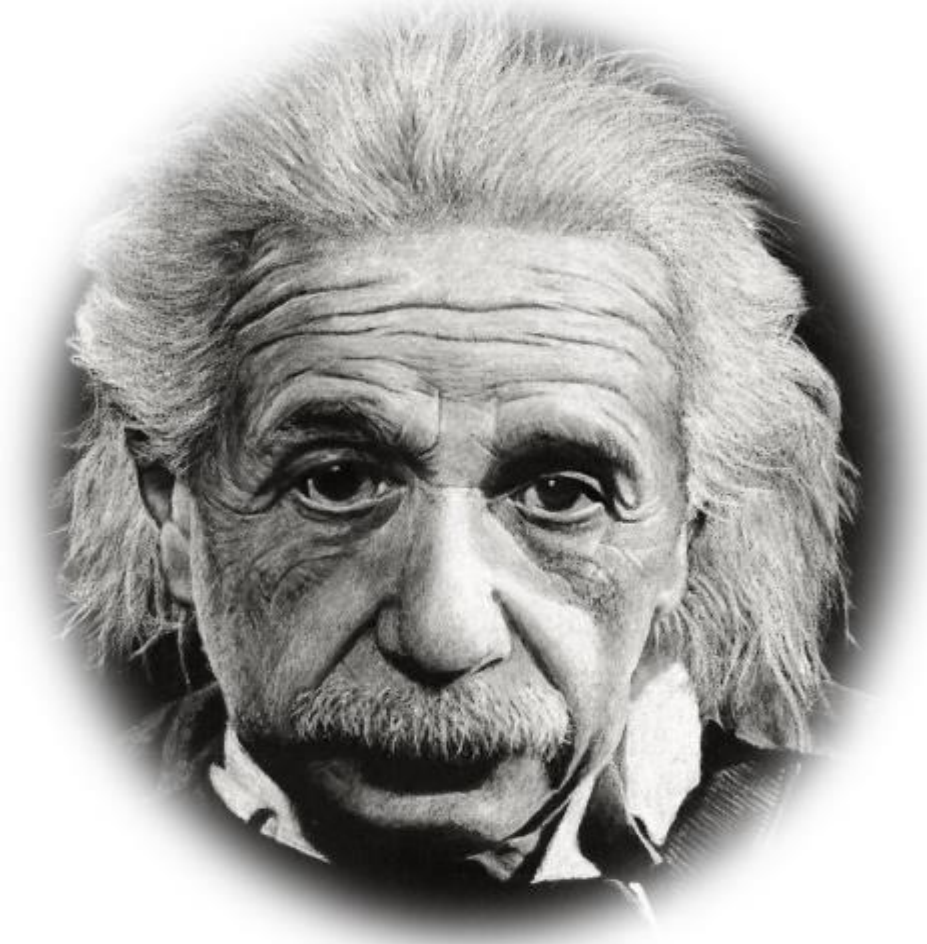
Tutores:

Ing. Yasmani J. Alvarez Gómez

Ing. Elieyis Gómez Padrón

Ing. Rita Milena Hernández Díaz

La Habana, 2015



Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del presente trabajo y autorizo exclusivamente a la Universidad de las Ciencias Informáticas para su uso a conveniencia.

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

Carlos Alberto Castillo Pérez

Firma del Autor

Ing. Yasmani J. Alvarez Gómez

Ing. Elieyis Gómez Padrón

Ing. Rita Milena Hernández Díaz

Firma del Tutor

Firma del Tutor

Firma del Tutor

AGRADECIMIENTOS

En primer lugar a mis abuelos (Florentino y Juanita) y a mi mamá por su dedicación, esfuerzo y siempre confiar en mí, todo se lo debo a ustedes. Por siempre apoyarme en todo, respetar mis decisiones. Por ustedes hoy soy ingeniero.

A mi familia por siempre estar preocupados por mis estudios y formación como profesional en especial a mi papá y hermano.

A mis tutores Elieyis, Rita y Yasmari, gracias a ellos hoy este sueño se hace realidad, cuando más necesitaba ayuda ahí estaban ellos, al igual que a todos los profesores que me ayudaron de una forma u otra.

A Sachie, Leonardo y Victor por ser como hermanos para mí y siempre poder contar con ellos. A todos mis amigos...los de antes...ahora...y siempre.

A todos ustedes...Muchas Gracias. En general a todos aquellos compañeros que me brindaron apoyo...

DEDICATORIA

A mis abuelos (Guanita y Florentino) por ser todo para mí y darme amor, a quienes debo todo lo que soy.

A mi mamá y papá, a mi hermano y a toda la familia.

A mis amigos, los que vienen conmigo desde hace mucho y los de ahora.

A todos gracias por formar parte de este momento.

Resumen

Mantener un cliente satisfecho es lo primordial para cualquier empresa, pues estos son los que adquieren, consumen y demandan los productos que son fabricados por ellas. La presente investigación comprende el desarrollo del Sistema de gestión de relaciones con clientes “COMSYS” para el Centro de Ideoinformática de la Universidad de las Ciencias Informáticas. Esta solución fue desarrollada con el propósito de gestionar las relaciones del Centro con sus clientes para facilitar el análisis de la información comercial y apoyar la toma de decisiones. El proceso de desarrollo estuvo guiado por la metodología OpenUP, para el modelado se utilizó la herramienta Visual Paradigm 8.0 y el lenguaje de modelado UML. La implementación fue realizada sobre los marcos de trabajo Symfony 2.3 y Bootstrap 3.0. Se emplearon como lenguajes de programación PHP 5.4, HTML5 y CSS3. Para graficar se utilizó la biblioteca Highcharts y como sistema gestor de bases de datos PostgreSQL 9.3. Además, se realizaron pruebas funcionales y de seguridad para comprobar el correcto funcionamiento de la aplicación. Las funcionalidades del sistema desarrollado permiten mostrar, añadir, modificar, eliminar y buscar los datos de los clientes, contactos, productos, oportunidades de negocio, así como programar tareas que notifiquen cuando hay un evento importante con un cliente. Como resultado, también se almacena toda la documentación de los clientes y se brindan informes actualizados para facilitar la toma de decisiones a la alta gerencia con el objetivo de lograr la satisfacción del cliente.

Palabras claves: administración, cliente, relaciones, oportunidades, venta.

Índice:

DECLARACIÓN DE AUTORÍA	II
AGRADECIMIENTOS	III
DEDICATORIA	IV
Resumen	V
Introducción	1
Capítulo 1. Fundamentación teórica del Sistema de gestión de relaciones con clientes	5
1.1 Sistemas de información gerencial	5
1.2 Sistema de gestión de relaciones con clientes (CRM)	5
1.3 Valoración de CRM existentes	6
1.5 Tecnologías, herramientas y metodologías	12
1.5.1 Frameworks de desarrollo	12
1.5.2 Framework de diseño	14
1.5.3 Lenguajes de programación.....	14
1.5.4 Bibliotecas	17
1.5.5 Sistemas de gestión de bases de datos.....	19
1.5.6 Servidores web	20
1.5.7 Entorno de desarrollo Integrado (IDE)	20
1.5.8 Herramientas CASE	21
1.5.9 Herramienta para las pruebas de seguridad	22
1.5.10 Metodología de desarrollo de software.....	22
1.6 Conclusiones parciales	24
Capítulo 2. Características, análisis y diseño del Sistema de gestión de relaciones con clientes	25
2.1 Propuesta de sistema.....	25
2.2 Modelo de dominio	25
2.3 Análisis y diseño del sistema.....	26
2.3.1 Requisitos funcionales (RF)	27
2.3.2 Requisitos no funcionales (RNF)	28
2.4 Diagrama de casos de uso (CU) del sistema.....	29
2.4.1 Especificación de casos de uso (CU).....	30

2.5	Descripción del sistema propuesto.....	36
2.6	Patrones de diseño implementados	36
2.7	Patrón arquitectónico implementado por <i>Symfony2</i>	38
2.8	Estructura en bundle de la aplicación	39
2.9	Seguridad en el sistema.....	40
2.10	Diagrama de clases del diseño.....	40
2.11	Modelo de datos.....	41
2.12	Conclusiones parciales	43
Capítulo 3. Implementación y pruebas del Sistema de gestión de relaciones con clientes.....		44
3.1	Modelo de implementación.....	44
3.1.1	Diagrama de componentes.....	44
3.1.2	Diagrama de despliegue.....	45
3.2	Estándares de codificación.....	46
3.3	Pruebas funcionales.....	48
3.4	Pruebas de seguridad	50
3.5	Conclusiones parciales	55
Conclusiones		56
Recomendaciones		57
Glosario de términos.....		58
Referencias.....		59
Bibliografía consultada.....		61

Índice de figuras:

Figura 1	Modelo de dominio.....	26
Figura 2	Diagrama de casos de uso del sistema.....	30
Figura 3	Diagrama del patrón arquitectónico MVC.....	39
Figura 4	DCDEW de gestionar cliente.....	41
Figura 5	Modelo de datos.....	42
Figura 6	Diagrama de componentes general.....	45
Figura 7	Diagrama de despliegue	46

Figura 8 Nombre de las clases, namespace y use	47
Figura 9 Nombre las funciones en Lower Camel Case.....	47
Figura 10 Atributos en Lower Camel Case	48
Figura 11 Resultados de las pruebas funcionales	50
Figura 12 Resultados de las pruebas de seguridad con Acunetix	55

Índice de Tablas:

Tabla 1 Comparación de sistemas existentes.....	10
Tabla 2 Descripción de actores	29
Tabla 3 CU Autenticar usuario.....	30
Tabla 4 CU Gestionar cliente.....	32
Tabla 5 Descripción de las variables	48
Tabla 6 Prueba funcional para el CU Gestionar oportunidad, Sección: Crear oportunidad	49
Tabla 7 Lista de chequeo prueba de autorización.....	51
Tabla 8 Lista de chequeo prueba de autorización.....	51
Tabla 9 Lista de chequeo comprobación del sistema de autenticación.....	52
Tabla 10 Lista de chequeo validación de datos	53

Introducción

La gestión comercial es uno de los eslabones de relevancia en el desarrollo de las empresas en el mundo. Durante años se ha buscado la forma de analizar comportamientos, principalmente los relacionados con los clientes y las relaciones comerciales con ellos. Actualmente las compañías invierten en tecnologías para aplicar estrategias comerciales adecuadas que les permitan trazar políticas innovadoras.

Uno de los principales elementos de las organizaciones empresariales del mundo es el *marketing*; como respuesta al entorno cada vez más turbulento, competitivo y exigente, tanto del mercado como en tecnología; la orientación al mercado y el *marketing* relacional convergen en la estrategia de negocio basada en la gestión de relaciones con clientes [1].

Un elemento clave en el éxito de una empresa u organización es la relación con el cliente. La misma aumenta los ingresos, la calidad de los servicios y permite tener un manejo centralizado de la información de contactos. Por lo que se hace necesario la utilización de herramientas de software que permitan simplificar y optimizar el proceso. Para ello se han desarrollado sistemas de Gestión de Relaciones con Clientes, CRM (por sus siglas en inglés, *Customer Relationship Managment*) los cuales facilitan la adopción de estrategias de fidelización de clientes a una institución o empresa.

La notable evolución de las nuevas tecnologías de la información y las comunicaciones ha sido un factor clave para el desarrollo del CRM. Las tecnologías de la información (bases de datos relacionales, software de análisis, multimedia, etc.) y los desarrollos en inteligencia artificial han abierto importantes expectativas que simplemente no existían en el pasado [1].

Para poder aplicar una estrategia de fidelización a clientes, resulta imprescindible llevar un control estricto de sus oportunidades, ingresos y demás elementos comerciales. La alta gerencia está consciente de que debe tener un conocimiento de su estado comercial para la correcta toma de decisiones.

En la actualidad muchas empresas usan sistemas para gestionar las relaciones con clientes y oportunidades de negocio. Con este propósito, las herramientas más utilizadas en el ámbito empresarial son privativas, por las facilidades que brindan. En consecuencia, las entidades deben pagar por los servicios prestados, estando fuera del alcance de la mayoría de las empresas en Cuba. Además, la información recolectada es almacenada en servidores fuera del país, atentando contra la soberanía tecnológica, propiciando que esta

pueda ser utilizada con fines ajenos a los intereses de la empresa propietaria. Unido a esto se tiene como inconveniente que para acceder a los servicios se debe poseer una conexión a internet, limitando el alcance para organizaciones que cuentan solamente con red nacional.

El Centro de Ideoinformática (CIDI) es uno de los centros productivos de la Universidad de las Ciencias Informáticas que desarrolla un conjunto de productos y servicios para la comercialización. En el mismo se gestiona la información de clientes, productos, servicios, oportunidades de negocios, ingresos y otros elementos comerciales que influyen en su desarrollo. El manejo eficiente de estos elementos es de vital importancia pues de ello depende que CIDI cumpla con los objetivos planteados por la facultad.

Actualmente la información de los clientes y las relaciones con estos, se gestiona a través de varios documentos digitales que se actualizan manualmente y por separado, duplicando, en algunos casos la información. Estos documentos son gestionados y almacenados localmente por el grupo comercial provocando la dependencia de información de esta área. Como consecuencia la información comercial de CIDI puede no estar disponible en todo momento para la alta gerencia, de manera que cuando se hace necesario consultarla de forma inmediata, se deba solicitar al grupo comercial, quiénes pueden estar realizando otras tareas.

Cada cierto período de tiempo se realiza un análisis estadístico del comportamiento de los negocios para medir el avance comercial y de esta forma tomar decisiones oportunamente. Actualmente, para este análisis resulta necesario consultar información en más de un documento a la vez y es importante ser cuidadoso para no cometer errores. Esto trae consigo que los procesos de gestión de la información comercial sean sensibles, por el alto nivel de detalle que se debe tener al procesar la información y poco óptimos para el desenvolvimiento con que se espera que se manejen los negocios en el centro.

Por lo antes planteado se identifica el siguiente **problema de investigación**: ¿Cómo gestionar las relaciones del centro CIDI con sus clientes para facilitar el análisis de la información comercial y apoyar la toma de decisiones?

Se plantea como **objeto de estudio** los sistemas de información gerencial.

Para darle solución al problema de la investigación se plantea como **objetivo general** de este trabajo de diploma: Desarrollar una aplicación que permita gestionar las relaciones del centro CIDI con sus clientes para facilitar el análisis de la información comercial y apoyar la toma de decisiones.

Desglosando del objetivo general los siguientes **objetivos específicos**:

1. Valorar el marco teórico conceptual y el estado del arte respecto a las tecnologías actuales para el desarrollo de herramientas de gestión.
2. Definir la metodología, tecnologías y herramientas para la implementación de la aplicación.
3. Diseñar una aplicación que permita la gestión de las relaciones del centro CIDI con sus clientes.
4. Implementar una aplicación que permita la gestión de las relaciones del centro CIDI con sus clientes.
5. Validar el correcto funcionamiento de la aplicación.

Se selecciona como **campo de acción** los sistemas de gestión de relaciones con clientes.

Para el cumplimiento de los objetivos específicos se han trazado las siguientes **tareas de la investigación**:

1. Valoración del estado del arte de los sistemas de gestión de relaciones con clientes.
2. Definición de la metodología a utilizar en el desarrollo de la aplicación.
3. Selección de las tecnologías de desarrollo.
4. Identificación de los requisitos funcionales y no funcionales.
5. Diseño e implementación de las funcionalidades identificadas.
6. Aplicación de las pruebas funcionales y de seguridad a la aplicación.

Se propone como **idea a defender** que el Sistema de gestión de relaciones con clientes (COMSYS) del centro CIDI facilita el análisis de la información comercial y tributa a una mejor toma de decisiones.

Para profundizar en el objeto de estudio, durante el proceso de desarrollo del Sistema de gestión de relaciones con clientes se realizan investigaciones, las cuales se guían por **métodos teóricos** de investigación como:

El Analítico-Sintético, que permite investigar acerca de los aspectos relacionados con los sistemas de gestión comercial y su implementación, mediante el análisis de la documentación existente. Además, este método facilitará la extracción de los elementos más importantes relacionados con el campo de acción.

La Modelación, permite realizar los modelos correspondientes al ciclo de vida del desarrollo del sistema de gestión propuesto mediante el lenguaje de modelado UML.

Los **métodos empíricos** permiten, como parte del procedimiento trazado para la investigación, determinar el método de recolección de datos y tipo de instrumento que será utilizado, por tanto los métodos empíricos que se utilizan en el presente trabajo de diploma son:

Entrevista: posibilita obtener información, experiencias, ideas, puntos de vistas, que contribuyan al desarrollo de la investigación y aporten conocimientos específicos del tema.

Observación: se realiza un estudio de diferentes arquitecturas y sistemas, obteniendo un mejor entendimiento de las funcionalidades comunes en los sistemas de gestión de relaciones con clientes.

El presente trabajo consta de tres capítulos, estructurados de la siguiente forma:

Capítulo 1. Fundamentación teórica del Sistema de gestión de relaciones con clientes: se abordan los conceptos fundamentales que permiten entender el mundo de los sistemas de gestión de relaciones con clientes y además se fundamentan las herramientas y tecnologías que se emplean en el desarrollo del trabajo de diploma.

Capítulo 2. Características, análisis y diseño del Sistema de gestión de relaciones con clientes: se delimitan las funcionalidades del sistema a través de requisitos funcionales y no funcionales. Se especifican a través de ejemplos concretos los patrones de diseño, los estilos arquitectónicos utilizados y la arquitectura definida para la implementación de la solución informática.

Capítulo 3. Implementación y pruebas del Sistema de gestión de relaciones con clientes: se describe la etapa de implementación. Se elaboran y documentan las pruebas realizadas a la solución propuesta para demostrar el correcto funcionamiento de la misma. Se realiza un análisis de los resultados de la aplicación en un entorno real.

1 Capítulo 1. Fundamentación teórica del Sistema de gestión de relaciones con clientes

En el presente capítulo se hace un estudio acerca del dominio del problema, así como el análisis de los conceptos fundamentales relacionados con la gestión de relaciones con clientes. Se profundiza en la búsqueda de soluciones que presenten características similares en Cuba y en el mundo. Otro aspecto importante, es el estudio de las tecnologías y herramientas de desarrollo de software en la actualidad y sus tendencias.

1.1 Sistemas de información gerencial

La informática ha propiciado el desarrollo de nuevas tecnologías y filosofías de gestión, que han permitido a muchas empresas convertir sus sistemas empresariales en ventajas competitivas y, por tanto, mejorar su posición respecto a sus competidores. Esto ha conllevado a una vertiginosa carrera en el desarrollo de sistemas tales como: DRP¹, ECR², ERP³, SCOR⁴ y CRM [2].

Estos sistemas de información gerencial permiten a los directivos tener un control de la empresa en tiempo real, manejar los datos y la información, controlándolos, centralizándolos y compartiéndolos entre los diferentes módulos del sistema, de manera que se evita que los datos se corrompan, dañen o dupliquen.

1.2 Sistema de gestión de relaciones con clientes (CRM)

El CRM consiste en una estrategia de la organización la cual centra sus esfuerzos en el conocimiento de sus clientes, detectando sus necesidades, aumentando su grado de satisfacción, incrementando su fidelidad a la empresa e incrementando la rentabilidad o beneficios del cliente a la empresa, mediante el análisis de las informaciones extraídas de los clientes desde los diferentes canales o medios de comunicación [3].

Es una herramienta de negocios, que a partir de soluciones tecnológicas, puede mejorar las relaciones con los clientes. Para esto el sistema brinda un grupo de datos acerca de los clientes que son usados para

¹ Planificación de los Recursos de Distribución del inglés *Distribution Resource Planning*.

² Respuesta Eficiente al Consumidor del inglés *Efficient Consumer Response*.

³ Planificación de Recursos Empresariales del inglés *Enterprise Resource Planning*.

⁴ Referencia de Suministro de Operaciones en Cadena del inglés *Supply Chain Operations Reference*.

realizar un estudio previo del mercado para conocer los gustos de los clientes, adaptando los productos a sus necesidades y lograr que queden totalmente satisfechos.

Lo anterior lleva a analizar la utilización de un CRM en los procesos de gestión comercial en el Centro CIDI teniendo en cuenta los procesos particulares de la gestión en la Universidad.

1.3 Valoración de CRM existentes

Existen hoy varias aplicaciones CRM, entre las que se pueden mencionar Sugar CRM, GoldMine CRM, SAP CRM, Oracle CRM, Oracle Fusion CRM, Microsoft Dynamics CRM de Microsoft Corporation y alternativas Open Source como OpenERP o Vtiger. A continuación se relacionan algunos de los CRM más utilizados en el mundo empresarial.

Microsoft Dynamics

Microsoft Dynamics CRM permite definir una estrategia de negocio centrada en anticipar, conocer, satisfacer las necesidades y los deseos de sus clientes, incrementando la efectividad de los empleados de ventas y servicios [4]. Microsoft Dynamics CRM permite que pequeñas y medianas empresas, y áreas específicas de grandes organizaciones, incrementen el éxito de sus ventas, entreguen un mejor servicio al cliente y tomen mejores decisiones, en base al manejo de información de valor [5].

Acciones de ventas:

- Gestiona tipo de operación.
- Genera reportes personalizados para cada etapa.
- Gestiona las liquidaciones de comisiones.
- Gestiona los pedidos de venta.
- Gestiona los descuentos de ventas.
- Gestiona las solicitudes de ventas.

Acciones de compras:

- Gestiona presupuestos.
- Define la cantidad de autorizaciones para generar órdenes de compras.
- Gestiona pedidos de compras.
- Gestiona las solicitudes de compras.

Funcionalidades:

- Administrar sus datos de manera eficaz.
- Automatizar los flujos de trabajo.
- Obtener conocimientos claves para la toma de decisiones.
- Simplificar el seguimiento de los clientes potenciales.

SugarCRM

SugarCRM es un sistema para la administración de la relación con los clientes (CRM) basado en LAMP (Linux-Apache-MySQL-PHP), desarrollado por la empresa SugarCRM, Inc. ubicada en Cupertino, California [6].

Tiene cinco ediciones, una de ellas libre y otras cuatro ediciones con componentes no-libres y con un costo por usuario [7]. SugarCRM es una aplicación CRM muy completa para negocios de distinto tamaño. Está diseñada para facilitar la gestión de ventas, oportunidades, contactos de negocios y más. A partir de la versión 4.5 permite utilizar SQL Server como base de datos; y la empresa ha firmado acuerdos con Microsoft para poder expandir su mercado sobre servidores con Windows [6].

SugarCRM desarrolla software CRM en cinco ediciones:[7]

- Sugar Community Edition (anteriormente conocido como Sugar Open Source)
- Sugar Professional
- Sugar Corporate
- Sugar Enterprise
- Sugar Ultimate

Cada producto se deriva de la misma Base de Código, con la Sugar Community Edition que no contiene muchas de las funcionalidades contenidas en Sugar Professional y Sugar Enterprise [7].

SugarCRM hace que Sugar Community Edition esté disponible de forma gratuita, aunque se están eliminando gradualmente para empujar a los clientes basados a las ediciones de suscripción: Profesional, Empresarial, Enterprise, Ultimate [7].

Acciones de ventas:

- Gestión de oportunidades.
- Gestión de clientes.
- Gestión de clientes potenciales.
- Gestión de pronósticos de ventas.
- Gestión de contratos.

OpenERP (CRM)

OpenERP es un sistema de gestión empresarial (ERP) y de relación con el cliente (CRM) de código abierto. Este sistema cubre las necesidades de las áreas de contabilidad, ventas, compras, y almacén e inventario, entre otras. También soporta múltiples monedas, múltiples compañías y múltiples contabilidades; además incorpora funcionalidades de gestión de documentos para agilizar la colaboración entre departamentos y equipos en la empresa; y permite trabajar remotamente mediante una interfaz web desde una computadora conectada a internet [8].

Características:

- La arquitectura del sistema es cliente-servidor, lo que permite que todos los usuarios trabajen sobre el mismo repositorio de datos. Esto tiene la ventaja de que toda la información está disponible y sincronizada en todo momento además de que descarga la mayor parte del trabajo de procesamiento de datos de las máquinas clientes (donde trabajan efectivamente los usuarios). Dispone de interfaces XML-RPC⁵ y SOAP⁶.
- Dentro de la construcción misma del software se hace un uso intensivo de flujos de trabajo (modelo *workflow*) que se pueden integrar con sus distintos módulos [9].

Acciones de ventas:

- Gestionar iniciativa de venta.
- Gestión de pedido de venta.
- Gestión de cliente.

⁵ XML-RPC es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes

⁶ SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

- Convertir iniciativa en oportunidad.

Acciones de compras:

- Gestión de solicitud de compra.
- Gestión de presupuesto.
- Gestión de pedido de compra.

VtigerCRM

VtigerCRM es una empresa de software CRM de código abierto principalmente para pequeñas y medianas empresas. VtigerCRM está construido sobre los probados, rápidos, fiables LAMP / WAMP (Linux / Windows, Apache, MySQL y PHP) y tecnologías de otros proyectos de código abierto [10].

Con VTiger se puede administrar completamente una empresa, a través de los diversos módulos que incluye, con los que se puede manejar las actividades de inventario, incluyendo fuerza de ventas, mercadotecnia, clientes y contactos, productos, presupuestos, facturación y pedidos, soporte y servicio al cliente. Actualmente existen muchas herramientas y *plugins* para VTigerCRM que asistirán en mejorar la productividad de una organización. No existe límite de usuarios ni tiene costo por licencia y se pueden integrar nuevos módulos de acuerdo a los requerimientos de una empresa [11].

Acciones de ventas:

- Gestión de contactos.
- Gestión de oportunidad de venta.
- Gestión de pedido de venta.
- Gestión de facturas.
- Gestión de tarifas.

Ventajas

- Es un programa muy completo, flexible, adaptable.
- Es un programa de código abierto.
- Excelente herramienta de gestión para empresas que está en español.

Desventajas

- Es un programa muy cargado, aunque es intuitivo.

- No es fácil de manejar porque tiene demasiadas variables.

SAP CRM

El software SAP CRM ofrece, en su conjunto de aplicaciones buenas funcionalidades para *marketing*, ventas y servicios. Soporta procesos de negocios para tratar directamente con los clientes en múltiples canales de interacción, permite que las organizaciones mantengan el foco en estrategias de crecimiento centrado en el cliente y que se diferencien en el mercado por ofrecer una experiencia superior para el cliente. SAP CRM es capaz de:

Gestionar procesos relacionados con los clientes: asegura la orquestación de todas las tareas relacionadas con los clientes, de un departamento a otro, incorporando, de forma transparente, actividades tales como distribución, facturación y cuentas por cobrar.

Suministrar a toda la organización información de clientes: reúne todas las fuentes relevantes de datos de clientes, distribuidas por todas las empresas, contribuyendo para lograr un mejor proceso de toma de decisiones.

Ofrecer beneficios inmediatos: permite que las empresas resuelvan, en primer lugar, las prioridades estratégicas y cumplan con los objetivos más rápidamente. La solución puede expandirse gradualmente y cada etapa traerá, de una manera tangible, el correspondiente retorno sobre la inversión [12].

Acciones de ventas:

- Gestión de cuentas y contactos.
- Gestión de actividades.
- Gestión de oportunidades.
- Gestión de territorios.
- Gestión de pedido de venta.
- Gestión de contratos.

Tabla 1 Comparación de sistemas existentes

CRM	Plataforma	Costo	Facilidad de uso	Soporte
Microsoft Dynamic	Windows	Alto	Buena	No hay soporte de una empresa que certifique o de seguridad del producto.

SugarCRM	Windows, Linux, Mac OS X	Alto	Regular	Administra asuntos de soporte desde el contacto inicial hasta la resolución final.
OpenERP	Windows, Linux	Bajo	Buena	No hay soporte de una empresa que certifique o de seguridad del producto.
Vtiger	Linux	Bajo	Regular	No hay soporte de una empresa que certifique o de seguridad del producto.
SAP	Windows	Alto	Buena	Administra asuntos de soporte desde el contacto inicial hasta la resolución final.

A partir de la investigación realizada se observó que los CRM existentes a pesar de propiciar grandes ventajas en el área de mercadotecnia, no cumplen totalmente con las necesidades del CIDI. Este planteamiento está fundamentado en que muchos de estos sistemas fueron desarrollados bajo herramientas y software propietarios; ejemplo claro de esto lo constituyen Microsoft Dynamic y SAP, sistemas que a pesar de administrar asuntos de soporte desde el contacto inicial hasta la resolución final y poseer funcionalidades para el manejo de las relaciones con los clientes, tienen un alto costo inicial, y de utilizar alguno, el país incurriría en grandes gastos de licencia y mantenimiento.

Los demás sistemas que existen y responden a la familia de los OpenSource, a pesar de tener un costo inicial bastante bajo, a diferencia de los propietarios, no tienen soporte de una empresa que certifique o garantice seguridad del producto. Dentro de estos sistemas el que más se adecua a las políticas y modo de trabajo de las entidades cubanas es el OpenERP, esta es una herramienta altamente configurable, propiciando de esta manera seleccionar los módulos y con ellos los procesos; asociados a estos procesos se encuentran los componentes que implementan las funcionalidades, permitiendo que se pueda extender fácilmente el sistema.

Cedrux

En Cuba no se contaba con un sistema ERP completo que integrara todos los módulos necesarios. Para dar respuesta a esta necesidad, un equipo multidisciplinario compuesto por especialistas de diferentes entidades del país y la Universidad de las Ciencias Informáticas, se dio a la tarea de desarrollar un sistema completo, el Sistema Integral de Gestión Empresarial Cedrux, más conocido por ERP cubano. Como parte

de esta solución aparecen varios sistemas integrados: configuración, contabilidad, planificación, recursos humanos y logística [13].

Los sistemas de planificación de recursos empresariales (*ERP*) son sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una empresa pero no está enfocado al cliente, ni a sus necesidades.

Por todos estos argumentos se hace necesario el desarrollo de una herramienta CRM, dirigida específicamente a satisfacer las necesidades de los clientes y que se ajuste al nuevo sistema económico cubano.

1.5 Tecnologías, herramientas y metodologías

En el desarrollo de software influyen un conjunto de elementos que permiten una correcta implementación de dicho proceso: las herramientas, tecnologías y la metodología para el desarrollo. La acertada selección de estos elementos garantiza la calidad del producto final.

1.5.1 Frameworks de desarrollo

Un *framework* es un término muy manejado en el campo de la informática, que se utiliza para referirse a un conjunto de bibliotecas, que son usadas para implementar la estructura de un modelo para una aplicación. Esto se realiza con el objetivo de promover la reutilización de código, posibilita que no sea necesario perder tiempo en reinventar la rueda. Existen diferentes tipos de *frameworks*, para diferentes propósitos, algunos orientados al desarrollo de aplicaciones web, o para un determinado sistema operativo o lenguaje.

Zend Framework (ZF)

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. ZF es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de este framework es algo única; cada componente está construido con una baja dependencia de otros componentes [14].

Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "*use-at-will*" (uso a voluntad). Los componentes de la biblioteca estándar de Zend Framework conforman un *framework* de aplicaciones web al combinarse. ZF

ofrece una implementación MVC⁷, una abstracción de base de datos, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos [14].

Symfony 2.3

Symfony es un proyecto PHP de software libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional [15].

Sus principales características son:

- Su código, y el de todos los componentes y librerías que incluye, se publican bajo la licencia MIT de software libre.
- La documentación del proyecto también es libre e incluye varios libros y decenas de tutoriales específicos.
- Aprender a programar con Symfony te permite acceder a una gran variedad de proyectos: el *framework* Symfony2 para crear aplicaciones complejas, el *micro framework* Silex para sitios web sencillos y los componentes Symfony para otras aplicaciones PHP.
- Aunque en su desarrollo participan cientos de programadores de todo el mundo, las decisiones técnicas importantes siempre las toma Fabien Potencier, líder del proyecto. Esto evita el peligro de que surjan *forks* absurdos y la comunidad se fragmente.
- Los componentes de Symfony son tan útiles y están tan probados, que proyectos tan gigantescos como Drupal 8 están contruidos con ellos [15].

Symfony2 es un framework fácil de instalar y configurar en la mayoría de las plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix estándares). Propone diferentes patrones de diseño para la web. Puede integrarse con diferentes gestores de bases de datos. Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo. Es extensible, flexible como para adaptarse a los casos más complejos. Es orientado a objetos. Se escogió Symfony2 por todas las ventajas anteriormente señaladas.

⁷ MVC arquitectura Modelo Vista Controlador

1.5.2 Framework de diseño

Bootstrap

Bootstrap es un *framework* que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ha sido desarrollado por *Twitter* que recientemente liberó su versión 3.x. La mayor ventaja es que se pueden crear interfaces que se adapten a los distintos navegadores (*responsive design*) apoyándose en un *framework* potente con numerosos componentes web que ahorrarán mucho esfuerzo y tiempo [16].

Características principales:

- *Bootstrap* ofrece una serie de plantillas CSS y ficheros *JavaScript* que permiten integrar el *framework* de forma sencilla y potente en proyectos webs.
- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como *tablets* y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías *JavaScript*, por ejemplo *JQuery*.
- Ofrece un diseño sólido usando LESS y estándares como CSS3/HTML5.
- Es un *framework* ligero que se integra de forma limpia en nuestro proyecto actual.
- Funciona con todos los navegadores, incluido *Internet Explorer* usando *HTML Shim* para que reconozca los *tags* HTML5.
- Dispone de distintos *layout* predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos [16].

Se incluirá Bootstrap en la solución propuesta para el desarrollo de las interfaces de usuario, ya que brinda enormes facilidades en este tema.

1.5.3 Lenguajes de programación

PHP

PHP es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el *script*, aunque

no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP [17].

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales [17]. Una de las más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos (p.ej., para MySQL), o utilizar una capa de abstracción como PDO, o conectarse a cualquier base de datos que admita el estándar de Conexión Abierta a Bases de Datos por medio de la extensión ODBC. Otras bases de datos podrían utilizar cURL o sockets, como lo hace CouchDB [18].

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets de red puros e interactuar usando cualquier otro protocolo. PHP tiene soporte para el intercambio de datos complejos de WDDX entre virtualmente todos los lenguajes de programación web. Y hablando de interconexión, PHP tiene soporte para la instalación de objetos de Java y emplearlos de forma transparente como objetos de PHP [18].

PHP tiene útiles características de procesamiento de texto, las cuales incluyen las expresiones regulares compatibles con Perl (PCRE), y muchas extensiones y herramientas para el acceso y análisis de documentos XML. PHP estandariza todas las extensiones XML sobre el fundamento sólido de libxml2, y amplía este conjunto de características añadiendo soporte para SimpleXML, XMLReader y XMLWriter [18].

La selección del framework de desarrollo, su poder y sencillez, su versatilidad al momento de escribir código, su sencillez en la sintaxis, e inclusive su seguridad justifican la utilización de PHP como lenguaje de desarrollo.

JavaScript

JavaScript es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto podríamos decir que nació como un *"lenguaje de scripting"* del lado del cliente, sin embargo, hoy es mucho más. Las necesidades de las aplicaciones web modernas y el

HTML5 han provocado que el uso de JavaScript que encontramos hoy haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel [19].

Pero además, en los últimos años JavaScript se está convirtiendo también en el lenguaje "integrador". Lo encontramos en muchos ámbitos, ya no solo en internet y la web, también es nativo en sistemas operativos para ordenadores y dispositivos, del lado del servidor y del cliente [19].

En el contexto de un sitio web, con este lenguaje puedes hacer todo tipo de acciones e interacción. Antes se utilizaba para validar formularios, mostrar cajas de diálogo y poco más. Hoy es el motor de las aplicaciones más conocidas en el ámbito de internet: Google, Facebook, Twitter, Outlook... absolutamente todas las aplicaciones que disfrutas en tu día a día en la Web tienen su núcleo realizado en toneladas de JavaScript. La Web 2.0 se basa en el uso de este lenguaje para implementar aplicaciones enriquecidas que son capaces de realizar todo tipo de efectos, interfaces de usuario y comunicación asíncrona con el servidor por medio de Ajax [19].

JavaScript permite:

- Máxima interactividad entre el usuario y la página.
- Verificación de los datos introducidos por el usuario, antes de enviar el formulario al servidor.
- Ejecución de carritos de compra en el navegador.
- Ejecución de pequeñas cantidades de información al igual que en una base de datos.
- Manejo de *applets* y *plugins* dentro de múltiples marcos de HTML.
- Pre-procesado de información antes de enviarla al servidor.

Se selecciona este lenguaje por su sencillez y su compatibilidad con la mayoría de los navegadores. Además tiene la ventaja que mediante él se puede realizar gran parte de las funciones del cliente, lo que reduce la carga del servidor. JavaScript también es un lenguaje muy potente de alto nivel, es dinámico y responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página, puede cambiar totalmente el aspecto de la misma. Permite realizar cálculos en base a variables cuyo valor es determinado por el usuario.

HTML5

HTML5 es un lenguaje de marcado (de hecho, las siglas de HTML significan *Hyper Text Markup Language*) usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es de hecho la quinta revisión del estándar que fue

creado en 1990. A fines del año pasado, la W3C la recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos venideros. Por así decirlo, qué es HTML5 está relacionado también con la entrada en decadencia del viejo estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que podemos ver hoy en día. Con HTML5, tenemos otras posibilidades para explotar usando menos recursos. Con HTML5, también entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación [20].

CSS3

Mientras que HTML permite definir la estructura de una página web, las hojas de estilo en cascada (*Cascading Style Sheets* o CSS) ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web [21].

Las hojas de estilo permiten definir de manera eficiente la representación de páginas y es uno de los conocimientos fundamentales que todo diseñador web debe manejar a la perfección para realizar su trabajo.

A partir del año 2005 se comenzó a definir el sucesor de esta versión, al cual se le conoce como CSS3 o *Cascading Style Sheets Level 3*. Actualmente en definición, esta versión ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, CSS3 es el estándar que dominará la web por los siguientes años [21].

1.5.4 Bibliotecas

JQuery

JQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. JQuery es la biblioteca de JavaScript más utilizada [22].

JQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados [23]. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían

de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Sus principales características son:

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.
- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+ [24].

Se utilizará jQuery 1.9.1 ya que utiliza sintaxis muy parecida a CSS, permite manipular series de elementos y modificarlas con una simple línea de código. Encadenamiento de enunciados, te ayuda a concentrarte en el resultado final. JQuery es muy fácil de expandir, ya que cuenta con gran cantidad de *plugins* que se pueden utilizar o hasta crear uno propio, también es compatible con todos los navegadores modernos.

Highcharts

En ocasiones es necesario presentar información en otros medios que no sean textuales, de modo que el usuario asimile los datos con mayor facilidad. Por eso las infografías se han vuelto populares últimamente. Aparte de las infografías también se puede necesitar presentar datos numéricos, en tal caso es mejor usar gráficos estadísticos. Ahora se puede crear fácilmente estos gráficos mediante el uso de bibliotecas JavaScript.

Highcharts es una librería escrita en JavaScript que permite la creación de gráficos. La librería ofrece un método fácil e interactivo para insertar gráficos en su sitio web o aplicación web.

Características:

- La librería es compatible con todos los navegadores modernos incluyendo iPhone/iPad e Internet Explorer desde su versión 6.
- No es comercial, no se necesita el permiso de los autores para su implementación en sitios web personales o sin fines de lucro.
- Es abierto, todas las características pueden ser personalizadas permitiendo una gran flexibilidad, además Highcharts está escrito solamente con código JavaScript, sólo se requiere incluir el archivo highcharts.js y cualquiera de los tres frameworks más populares de JavaScript (jQuery, MooTools o Prototype).

El motivo de la selección de esta librería es la fuerte integración que tiene con el framework de desarrollo Symfony 2, lo hace mediante un bundle llamado op/highchart.

1.5.5 Sistemas de gestión de bases de datos

Los sistemas de gestión de bases de datos son un tipo de software dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El propósito general de estos sistemas es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización [25].

PostgreSQL 9.3

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando [25].

Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema [25].

1.5.6 Servidores web

El servidor Apache es el servicio que se encarga de resolver las peticiones de página de los clientes utilizando el protocolo de internet HTTP. Es un software que se estructura por módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Apache se adapta a una gran variedad de entornos a través de su diseño modular.

Los módulos de Apache se pueden clasificar en tres categorías [26]:

- Módulos Base. Módulo con las funcionalidades básicas de Apache.
- Módulos Multiproceso. Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender las peticiones.
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Características de Apache [26]:

- Flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos.
- Multiplataforma.
- Tecnología gratuita de código abierto.
- Servidor altamente configurable de diseño modular.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP.

Además de las características antes mencionadas, se selecciona Apache como servidor web porque es uno de los servidores web más utilizados, además es una tecnología gratuita de código fuente abierto que ofrece instalaciones sencillas para sitios pequeños como es el caso y si se requiere se puede expandir. Además, permite la creación de sitios web dinámicos utilizando PHP que es el lenguaje de programación seleccionado.

1.5.7 Entorno de desarrollo Integrado (IDE)

Netbeans 7.4 es un entorno de desarrollo integrado de código abierto para desarrolladores de *software*. Cuenta con todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java, así como con PHP y JavaScript. Netbeans IDE 7.4 amplía el soporte avanzado de desarrollo HTML5, ofrece un nuevo soporte para el desarrollo web móvil. Está disponible en inglés, chino, portugués, japonés y ruso [27].

Se selecciona Netbeans 7.4 como herramienta IDE, por su soporte a tecnologías, su flexibilidad, y su potencia para el desarrollo de aplicaciones web y su compatibilidad con los frameworks más populares.

1.5.8 Herramientas CASE

Las herramientas de ingeniería de software asistida por computación, también conocidas por sus siglas en inglés CASE (*Computer Aided Software Engineering*) facilitan el trabajo de ingeniería de un proyecto, aportando funcionalidades para la organización, el modelado y el seguimiento de un *software*.

Visual Paradigm es una herramienta CASE. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Visual Paradigm ha sido concebido para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de *software* libre de probada utilidad para el analista que fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque orientado a objetos [28].

Entre sus principales características se encuentran:

- Entorno de creación de diagramas para UML.
- Diseño centrado en casos de uso y enfocado al negocio que genera un *software* de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo. Disponibilidad de múltiples versiones, para cada necesidad.
- Generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos [28].

Se selecciona *Visual Paradigm 8.0* como herramienta CASE. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas.

1.5.9 Herramienta para las pruebas de seguridad

Acunetix Web Vulnerability Scanner es una herramienta que es capaz de escanear sitios web en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en internet. Esta aplicación ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor [29].

1.5.10 Metodología de desarrollo de software

Las metodologías son un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayudan a los desarrolladores a realizar software. Estas indican cómo hay que obtener los distintos productos parciales y finales. Además, se pueden clasificar en dos tipos: pesadas y ágiles [30].

Las metodologías pesadas están guiadas por una fuerte planificación durante todo el proceso de desarrollo. El proceso de desarrollo de software llevado a cabo a partir de este tipo de metodología es mucho más controlado, con cierta resistencia a posibles cambios, además de poseer numerosas políticas o normas. Existe un contrato prefijado y el cliente interactúa con el equipo de desarrollo mediante reuniones o entrevistas, los grupos de trabajos son grandes y posiblemente distribuidos por roles que posibilitan un mejor trabajo. La más conocida y usada de estas metodologías es el Proceso Racional Unificado (RUP) [30].

Sin embargo este enfoque no resulta ser el más adecuado para proyectos donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad. Como solución a esta dificultad surgen las metodologías ágiles [31].

Las metodologías ágiles se han convertido en herramientas muy utilizadas por las empresas desarrolladoras de software debido al bajo costo en la producción de software y a la reducción en el tiempo de entrega del producto, sin descuidar la calidad y fiabilidad que debe mantener un producto de software que fue desarrollado por un método tradicional [31].

Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Está especialmente indicada para proyectos con un rápido cambio de requisitos.

Algunas de sus principales características son: el desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de treinta días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. Otra característica importante es el intercambio constante con el cliente, donde destaca la reunión diaria del equipo de desarrollo para coordinación e integración. Scrum es un complemento a las metodologías ágiles para el control, seguimiento y corrección de errores, cuyo objetivo primario es que toda la organización esté alineada en un mismo sentido ante este escenario [32].

eXtreme Programming (XP)

XP está guiada por una rápida programación y se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y en la reutilización de código. También se enfoca en la realización de pruebas a los principales procesos con el objetivo de tratar de obtener los posibles errores futuros, esto conocido como pruebas unitarias, así como en la simplicidad de las soluciones implementadas y preparación para enfrentar los cambios. Impone un alto nivel de disciplina entre los programadores, lo cual permite mantener un mínimo nivel de documentación que a su vez se traduce en una gran velocidad de desarrollo. XP se define como especialmente adecuada para proyectos de corto plazo con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. Está formada por cuatro partes fundamentales, las cuales son: historias de usuario, roles, procesos y prácticas.

OpenUP

OpenUP es un proceso de desarrollo de software mínimamente suficiente, esto quiere decir que incluye solo el contenido fundamental, esto es que no provee orientación sobre temas en los que el proyecto tiene que lidiar, como son: el tamaño del equipo, el cumplimiento, seguridad, orientación tecnológica entre otras. Sin embargo OpenUP es completa en el sentido de que manifiesta por completo el proceso de construir un sistema. Para atender las necesidades que no están cubiertas en su contenido, OpenUP es extensible a ser utilizado como base sobre la cual se pueden añadir o adaptarse contenidos de otros procesos si es necesario.

Esta metodología está compuesta por cuatro fases:

1. **Concepción:** se enfoca en el entendimiento del propósito y objetivos del proyecto que se desarrollará, obteniendo suficiente información para confirmar que se debe hacer. En la misma se debe entender qué construir, identificar los actores y quién está interesado en este sistema.

2. **Elaboración:** se tratan los riesgos significativos para la arquitectura. El propósito de esta fase es establecer la base de elaboración de la arquitectura del sistema. Además obtener una comprensión más detallada de los requerimientos que debe cumplir el mismo.
3. **Construcción:** se centra en el diseño, implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de la misma es completar el desarrollo del sistema basado en la arquitectura definida.
4. **Transición:** su propósito es asegurar que el sistema es entregado a los usuarios, y evalúa la funcionalidad del último entregable de la fase de construcción.

Beneficios de esta metodología:

- Es apropiada para proyectos pequeños y de bajos recursos, permite disminuir las probabilidades de fracaso en los proyectos pequeños e incrementar las probabilidades de éxito.
- Permite detectar errores tempranos a través de un ciclo iterativo.
- Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.
- Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.

Teniendo en cuenta que no se necesita una metodología robusta, pero si generar suficiente documentación, se define OpenUP como metodología pues esta se basa en RUP y utiliza además enfoques ágiles. Es un proceso extensible, dirigido a la gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental, apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo.

1.6 Conclusiones parciales

En el presente capítulo se trataron conceptos generales acerca de los sistemas de gestión de relaciones con clientes. Como resultado del estudio realizado se seleccionó la metodología de desarrollo de software *OpenUP* para la planeación y elaboración del sistema. Se eligió *Symfony 2.3* como *framework* para el desarrollo del sistema y *Bootstrap 3* para el diseño, combinación que permitirá la realización de un sistema con calidad y agilizar el proceso de implementación por las enormes facilidades que brindan. Se escogió *Apache 2* como servidor web, se optó por *PostgreSQL 9.3* como sistema gestor de base de datos. Para el entorno de desarrollo integrado se decidió utilizar *NetBeans 7.4*, para el modelado de los diagramas propuestos se definió el *Visual Paradigm 8.0* y como biblioteca para graficar *Highchart*.

2 Capítulo 2. Características, análisis y diseño del Sistema de gestión de relaciones con clientes

Dadas las necesidades planteadas en la situación problemática del presente trabajo de diploma, se determina desarrollar una aplicación web, esta aplicación debe contar con las funcionalidades necesarias para el correcto funcionamiento y satisfacer al cliente final.

En el presente capítulo se identifican las clases del dominio y la relación entre ellas, los requerimientos funcionales y no funcionales que debe cumplir la aplicación. Se realiza el modelado de casos de uso del sistema y las descripciones textuales de estos. Este capítulo tiene como objetivo realizar el diseño de la solución a implementar y que esta contribuya a satisfacer las necesidades que dieron su origen.

2.1 Propuesta de sistema

Se propone una aplicación web que permita la gestión de las relaciones con los clientes en los diferentes elementos de la comercialización de productos y servicios del centro CIDI, que brinde además informes actualizados que le facilite tomar decisiones a la alta gerencia con el objetivo de lograr la satisfacción del cliente. La aplicación debe permitir a los especialistas encargados de realizar los estudios añadir, modificar, eliminar y buscar los datos de los clientes, contactos, productos, oportunidades de negocio, así como programar tareas que notifiquen cuando hay un evento importante con un cliente. Contará con una funcionalidad para almacenar toda la documentación para una mejor comunicación por parte del equipo de trabajo.

2.2 Modelo de dominio

Debido a la falta de un proceso de negocio bien definido se decidió desarrollar un modelo de dominio (MD) que representa una parte visual de las clases y sus interacciones con el entorno real del proyecto. El diagrama que se elabora en el MD representa los objetos relacionados con los principales conceptos que se trabajarán en el desarrollo de este trabajo.

Este diagrama permite a los usuarios, desarrolladores e interesados lograr una mayor comprensión de los conceptos que se manejan, permitiéndoles utilizar un vocabulario común para comprender el contexto en el que se encuentra el sistema. Contribuirá, además, a identificar personas, eventos, transacciones y objetos involucrados en el sistema [33].

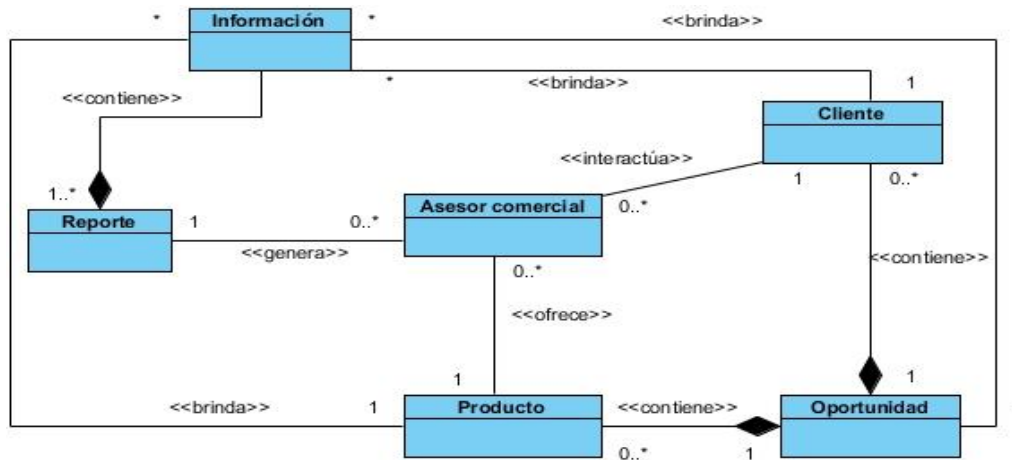


Figura 1 Modelo de dominio

Descripción de clases del modelo de dominio

Información: Datos que se guardan en el negocio.

Cliente: Persona interesada en algún producto que se brinde.

Reporte: Análisis estadístico que se genera para valorar el estado de comercialización.

Asesor comercial: Persona encargada del proceso de venta y compra de un producto.

Producto: Software con que cuenta el centro para su venta.

Oportunidad: Se refiere al interés que puede tener un cliente en un determinado producto, para su posterior compra.

2.3 Análisis y diseño del sistema

Para la realización de este trabajo se emplearon diferentes técnicas en la obtención de los requisitos, los cuales posibilitaron comprender el dominio del sistema, buscar y recolectar información para definir sus límites y restricciones, así como identificar a las personas involucradas en el mismo.

Las técnicas realizadas fueron: la entrevista y sistemas existentes; a continuación se describirá brevemente como se utilizaron:

Entrevista: se realizaron reuniones entre el equipo de desarrollo y el cliente posibilitando entender el dominio del problema y sus necesidades. Éstas se basaron en un conjunto de preguntas y respuestas, con el objetivo de comprender de manera general lo que el cliente necesitaba y como lo deseaba.

Sistemas existentes: esta técnica se empleó durante el análisis de varios sistemas existentes, ya fueran libres o propietarios. Igualmente se analizaron las distintas salidas de estos sistemas, Posibilitando el surgimiento de nuevas ideas sobre la base de estas.

2.3.1 Requisitos funcionales (RF)

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen y muestran cómo los casos de uso serán llevados a la práctica [34]. Se identificaron un total de 43 requisitos funcionales, que a continuación se presentan:

- | | |
|--|--|
| RF1. Autenticar usuario | RF2. Cerrar sección de usuario |
| RF3. Editar perfil | RF4. Recuperar contraseña |
| RF5. Mostrar lista de documentos | RF5. Subir documento al sistema |
| RF6. Eliminar documento | RF7. Mostrar usuarios |
| RF8. Añadir usuario | RF9. Modificar usuario |
| RF10. Eliminar usuario | RF11. Mostrar contactos |
| RF12. Añadir contacto | RF13. Modificar contacto |
| RF14. Eliminar contacto | RF15. Mostrar clientes |
| RF16. Añadir cliente | RF17. Modificar cliente |
| RF18. Eliminar cliente | RF19. Mostrar productos |
| RF20. Añadir producto | RF21. Modificar producto |
| RF22. Eliminar producto | RF23. Mostrar actividades |
| RF24. Añadir actividad | RF25. Modificar actividad |
| RF26. Eliminar actividad | RF27. Mostrar oportunidades |
| RF28. Añadir oportunidad | RF29. Modificar oportunidad |
| RF30. Eliminar oportunidad | RF31. Mostrar oportunidades archivadas |
| RF32. Añadir oportunidad archivada | RF33. Modificar oportunidad archivada |
| RF34. Eliminar oportunidad archivada | RF35. Mostrar notificaciones |
| RF36. Crear la notificación | RF37. Marcar como leída |
| RF38. Eliminar notificaciones | RF39. Mostrar calendario con las actividades |
| RF40. Enviar correos a los contactos | RF41. Generar reporte de contactos por mes |
| RF42. Generar reporte de clientes por sector | RF43. Generar reporte de clientes por país |

2.3.2 Requisitos no funcionales (RNF)

Los requerimientos no funcionales son cualidades o propiedades que el producto debe poseer, o restricciones en el producto que está siendo desarrollado. Estos no describen lo que el software hará, sino cómo lo hará. Estos requisitos son de gran significación en la aceptación del software, debido a que representan las ventajas más visibles al usuario y repercuten en el óptimo funcionamiento y mantenimiento del sistema. Son aquellos requisitos que especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar [34]. Como parte de la captura de requisitos no funcionales se identificaron un total de 12, agrupados en 7 categorías:

Confiabilidad:

RNF1. La aplicación podrá ser utilizada sólo por los usuarios que estén registrados en el sistema.

RNF2. La información no podrá ser modificada por usuarios no autorizados, protegiendo así la integridad de los datos.

Licencia:

RNF3. Se requiere el uso de herramientas y tecnologías de software libre, las cuales se podrán usar, modificar y distribuir libremente.

Diseño e Implementación:

RNF4. Se requiere el uso de PHP 5.4 como lenguaje de programación del lado del servidor.

RNF5. Se requiere el uso de Symfony 2.3 como framework para el desarrollo y Bootstrap para el diseño de la interfaz de usuario.

RNF6. Se requiere PostgreSQL 9.3 como gestor de bases de datos.

RNF7. Se requiere el uso de Apache2 como servidor web.

Soporte:

RNF8. El sistema debe dar la posibilidad de ser actualizado, además se le podrán incorporar nuevas funcionalidades en caso de ser necesarias.

Hardware:

RNF9. Para garantizar un buen funcionamiento del sistema, el servidor donde estará desplegado deberá cumplir como mínimo las siguientes condiciones: ser Pentium IV o sus equivalentes de otras compañías, con una velocidad de 3.00 GHz, 2 GB de RAM y 160 GB de disco duro.

Software:

RNF10. Para el cliente como versión más antigua del navegador: Mozilla Firefox 25, Google Chrome 15, Opera 9 e Internet Explorer 11.

RNF11. Para el servidor web Apache 2.2 o superior con módulo PHP 5.x disponible.

Interfaz:

RNF12. La interfaz de la aplicación debe ser adaptable a todos los dispositivos y resoluciones.

Tabla 2 Descripción de actores

Actor	Objetivos
Usuario	Persona encargada de consultar la información, genera reportes, no tiene acceso a modificar ni eliminar.
Moderador	Persona encargada de trabajar con el sistema completo, agregarle información, modificarla y eliminarla, no tiene acceso a la gestión de usuarios.
Administrador	Acceso total al sistema.

2.4 Diagrama de casos de uso (CU) del sistema

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. Estos muestran la relación entre los actores y los casos de uso en un sistema. Se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo. A continuación se presenta el diagrama de casos de uso de la aplicación.

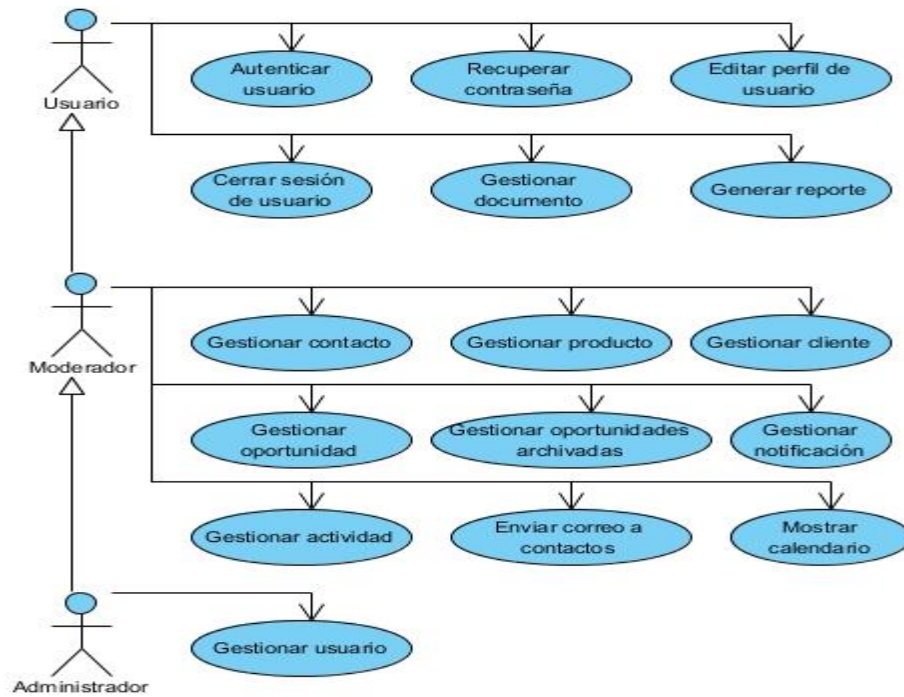


Figura 2 Diagrama de casos de uso del sistema

2.4.1 Especificación de casos de uso (CU)

Tabla 3 CU Autenticar usuario

Objetivo	Autenticar usuario
Actores	Usuario: (Inicia) Autentica usuario ante el sistema.
Resumen	El CU se inicia cuando un usuario desea autenticarse e introduce los parámetros necesarios, el sistema realiza la operación solicitada y finaliza así el CU.
Complejidad	Baja
Prioridad	Crítico.
Precondiciones	Que se introduzcan los parámetros obligatorios.
Postcondiciones	Se autentica un usuario.
Prototipo	



Flujo de eventos

Sección 1: “Autenticar usuario”

Flujo básico Autenticar usuario

	Actor	Sistema
1.	El usuario introduce los valores correspondientes en los campos Nombre de usuario y Contraseña y presiona el botón Entrar.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
2.		El sistema autentica al usuario, le concede los permisos, muestra la página principal y finaliza el CU.

Flujos alternos

1a Autenticar usuario con datos incorrectos

	Actor	Sistema
--	--------------	----------------

1.	El usuario introduce valores incorrectos en los campos Nombre de usuario y Contraseña y presiona el botón Entrar.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
2.		El sistema no autentica al usuario y muestra el mensaje: "Nombre de usuario o Contraseña inválidos"
1b Autenticar usuario dejando campos obligatorios vacíos		
	Actor	Sistema
1.	El usuario no introduce valores en los campos Nombre de usuario o Contraseña y presiona el botón Entrar.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
2.		El sistema no autentica al usuario y muestra el mensaje: "Este campo es requerido".
	CU extendidos	
Requisitos no funcionales	RNF12. La interfaz de la aplicación debe ser adaptable a todos los dispositivos y resoluciones.	
Asuntos pendientes		

Tabla 4 CU Gestionar cliente

Objetivo	Crear, editar, eliminar o ver información referente a los clientes en el sistema.
Actores	Moderador, Administrador
Resumen	El CU se inicia cuando el actor, desea crear, editar, eliminar o ver información referente a los clientes e introduce los parámetros necesarios en cada caso, el sistema realiza la operación solicitada y finaliza así el CU.
Complejidad	Alta
Prioridad	Crítico.
Precondiciones	Que el usuario se encuentre autenticado.
Postcondiciones	Se crea, edita, elimina o muestran los clientes.
Flujo de eventos	

Prototipo																																					
<p>Cientes + Nuevo</p> <p>10 <input type="text"/> Buscar: <input type="text"/></p> <table border="1"> <thead> <tr> <th>Nombre ▲</th> <th>Sector ◆</th> <th>País ◆</th> <th>Web</th> <th>Fecha ◆</th> <th>Clasificación</th> <th></th> </tr> </thead> <tbody> <tr> <td>cast</td> <td>as</td> <td>Cuba</td> <td>http:// lol.cu</td> <td>2015-05-20</td> <td>as</td> <td> </td> </tr> <tr> <td>sds</td> <td>sd</td> <td>Cuba</td> <td>http:// cashh.vv</td> <td>2015-05-20</td> <td>sd</td> <td> </td> </tr> <tr> <td>Sebastianas</td> <td>as</td> <td>Cuba</td> <td>http:// castlol.cu</td> <td>2015-05-20</td> <td>asa</td> <td> </td> </tr> <tr> <td>uuuu</td> <td>uuu</td> <td>Albania</td> <td>http:// 10.53.6.253/cliente/new</td> <td>2015-05-20</td> <td>yyur</td> <td> </td> </tr> </tbody> </table> <p>Mostrando de 1 a 4 < 1 ></p>			Nombre ▲	Sector ◆	País ◆	Web	Fecha ◆	Clasificación		cast	as	Cuba	http:// lol.cu	2015-05-20	as		sds	sd	Cuba	http:// cashh.vv	2015-05-20	sd		Sebastianas	as	Cuba	http:// castlol.cu	2015-05-20	asa		uuuu	uuu	Albania	http:// 10.53.6.253/cliente/new	2015-05-20	yyur	
Nombre ▲	Sector ◆	País ◆	Web	Fecha ◆	Clasificación																																
cast	as	Cuba	http:// lol.cu	2015-05-20	as																																
sds	sd	Cuba	http:// cashh.vv	2015-05-20	sd																																
Sebastianas	as	Cuba	http:// castlol.cu	2015-05-20	asa																																
uuuu	uuu	Albania	http:// 10.53.6.253/cliente/new	2015-05-20	yyur																																
Sección 1: “Listar clientes”																																					
Flujo básico																																					
1 Listar los clientes																																					
	Actor	Sistema																																			
1.	En el menú principal hace clic en Clientes, y selecciona la opción Listar del submenú.	Muestra el listado de los clientes y finaliza el CU.																																			
Flujos alternos																																					
1a Listar los clientes cuando no existe ninguno registrado																																					
1		En caso de que no exista ningún cliente registrado aparece el siguiente mensaje: “No hay datos disponibles en la tabla”.																																			
Sección 2: “Crear cliente”																																					
Flujo básico																																					
2 Crear un cliente correctamente																																					

	Actor	Sistema
1.	En el menú principal hace clic en Clientes, y selecciona la opción Nuevo cliente del submenú.	Muestra el formulario para crear el cliente con los siguientes datos: Nombre* Sector* País* Dirección Descripción Sitio web Clasificación
2.	Introduce los datos y da clic en el botón Crear.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
3.		Crea el cliente y muestra una vista con el nuevo cliente y un mensaje "Un nuevo cliente ha sido creado" y finaliza el CU.
Flujos alternos		
2a Crear un cliente con datos incorrectos		
	Actor	Sistema
2.	Introduce datos incorrectos y da clic en el botón Crear.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
3.		Muestra un mensaje de error indicando que se han introducido datos incorrectos y marca el o los campos erróneos.
2b Crear un cliente dejando campos obligatorios vacíos		
	Actor	Sistema
2.	Introduce los datos dejando campos obligatorios vacíos y da clic en el botón Crear.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
3.		Muestra un mensaje de error debajo de cada campo obligatorio que se encuentre vacío, indicando que el campo es requerido.
Sección 3: "Editar cliente"		
Flujo básico		
3 Editar un cliente correctamente		

	Actor	Sistema
1.	En el menú principal hace clic en Clientes, y selecciona la opción Listar del submenú.	Muestra el listado de los clientes.
2.	Solicita editar un cliente haciendo clic en el cliente que desee editar.	Muestra el formulario para editar el cliente con los siguientes campos: Nombre* Sector* País* Dirección Descripción Sitio web
3.	Modifica los datos a editar y da clic en el botón Editar.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
4.		Edita el cliente y muestra un mensaje “El Cliente ha sido editado” y finaliza el CU.
Flujos alternos		
3a Editar un cliente con datos incorrectos		
	Actor	Sistema
3.	Modifica los datos a editar y da clic en el botón Editar.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
4.		Muestra un mensaje de error indicando que se han introducido datos incorrectos y marca el o los campos erróneos.
3b Editar un cliente dejando campos obligatorios vacíos		
	Actor	Sistema
3.	Introduce los datos dejando campos obligatorios vacíos y da clic en el botón Editar.	El sistema verifica que los campos obligatorios estén llenos y que los datos sean correctos.
4.		Muestra un mensaje de error debajo de cada campo obligatorio que se encuentre vacío, indicando que el campo es requerido
Sección 4: “Eliminar cliente”		
Flujo básico		
4 Eliminar un cliente correctamente		

	Actor	Sistema
1.	En el menú principal hace clic en Clientes, y selecciona la opción Listar del submenú.	Muestra el listado de los clientes.
2.	Solicita mostrar un cliente haciendo clic en el cliente que desee eliminar.	Muestra una vista con los datos del cliente y la opción de eliminarlo.
4.	Solicita eliminar un cliente haciendo clic en el botón eliminar.	Muestra el mensaje:” ¿Desea eliminar el cliente Nombre del cliente?”
5.	Si el usuario selecciona la opción Eliminar	Elimina el cliente, muestra el listado de los clientes y un mensaje “El cliente ha sido eliminado” y finaliza el CU.
Flujos alternos		
5a Cancelar la operación		
	Actor	Sistema
5.	Si el usuario selecciona la opción Cancelar.	Muestra el listado de los clientes.
	CU incluidos	
	CU extendidos	
Requisitos no funcionales	RNF12. La interfaz de la aplicación debe ser adaptable a todos los dispositivos y resoluciones.	
Asuntos pendientes		

2.5 Descripción del sistema propuesto

El sistema propuesto tiene como objetivo fundamental mejorar las relaciones con los clientes para tomar decisiones a la alta gerencia. Para esto el sistema brinda un grupo de datos acerca de los clientes que son usados para realizar un estudio previo del mercado para conocer los gustos de los clientes, adaptar los productos a sus necesidades y lograr que queden totalmente satisfechos.

2.6 Patrones de diseño implementados

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de *software*. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. Se deben tener presente los siguientes elementos de un

patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) [35].

Los Patrones Generales de *Software* para Asignar Responsabilidades (GRASP del inglés *General Responsibility Assignment Software Patterns*) tienen una importante utilidad en el diseño de una aplicación al igual que los *Gang-of-Four* o Pandilla de los Cuatro (GoF por sus siglas en inglés). Los empleados en la solución de la problemática planteada, se exponen a continuación:

Patrones GRASP:

Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento [35].

Este patrón es utilizado principalmente para el trabajo con las entidades que se almacenan en la base de datos, un ejemplo de su utilización es cuando se registra una cuenta, la clase *UsuarioController* es la encargada de instanciar los métodos de la entidad Usuario, que luego se almacenará en la base de datos, debido a que obtiene todos los datos necesarios para la inicialización de los mismos.

Bajo Acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados negativos que puede producir un acoplamiento alto. No soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. El mismo no se puede considerar de manera aislada a otros patrones como el Experto o el de Alta Cohesión, sino que necesita incluirse como uno de los diferentes principios de diseño que influyen en una elección, al asignar una responsabilidad [35].

El bajo acoplamiento se evidencia en el proyecto completo, ya que todas las clases controladoras heredan únicamente de la clase *Controller* para alcanzar un bajo acoplamiento de clases y no existe otra dependencia o herencia entre clases en la implementación.

Alta Cohesión este patrón es un principio a tener en mente durante todas las decisiones de diseño. Incrementa la claridad y facilita la comprensión del diseño. Se simplifican el mantenimiento, las mejoras y soporta a menudo bajo acoplamiento [35].

Se puede observar claramente en el sistema ya que cada clase controladora se ajusta a manejar solo las responsabilidades correspondientes a las entidades con las que se relaciona, además para cada página

cliente existe una página servidora encargada de manejar sus solicitudes poniéndose de manifiesto el patrón de diseño alta cohesión.

Controlador es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. La mayor parte de los sistemas reciben eventos de entrada externa, en cualquiera de los casos que puedan existir, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada [35].

Se pone de manifiesto en todo el sistema ya que cada uno de los eventos generados por el usuario es redirigido a una clase controladora para que realice las operaciones solicitadas.

Patrones GoF:

Observador este patrón define, en sentido general, una dependencia de tipo uno-a-muchos entre objetos, de forma tal que cuando uno de los objetos cambia su estado, notifica su cambio a todos sus dependientes. La solución muestra la utilización de este patrón en los objetos de tipo Usuario, ya que la clase entidad Usuario, presenta una relación uno-a-mucho con la entidad Documento y de esta forma si se produce algún cambio en un objeto de esta última, se notificará a los objetos de tipo Documento con los cuales tiene relación.

Decorador *Symfony2* utiliza este patrón en la creación de plantillas generales para que otras las utilicen. Este patrón se utiliza en la solución con la creación de la plantilla `base.html.twig`, la cual posee en su contenido el código que será general para todas las interfaces de la solución, evitando de esta forma, tener que copiar este código en cada plantilla de la misma [35].

2.7 Patrón arquitectónico implementado por *Symfony2*

Symfony2 es un *framework* diseñado para optimizar, proporcionando varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web. *Symfony2* está fundamentado en un patrón clásico del diseño web conocido como arquitectura modelo vista controlador (MVC). El modelo representa la lógica de negocio, la vista transforma el modelo en una página web que permite al usuario interactuar con ella y el controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. En el sistema, el modelo es la representación específica de la información y gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, el

controlador responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información.

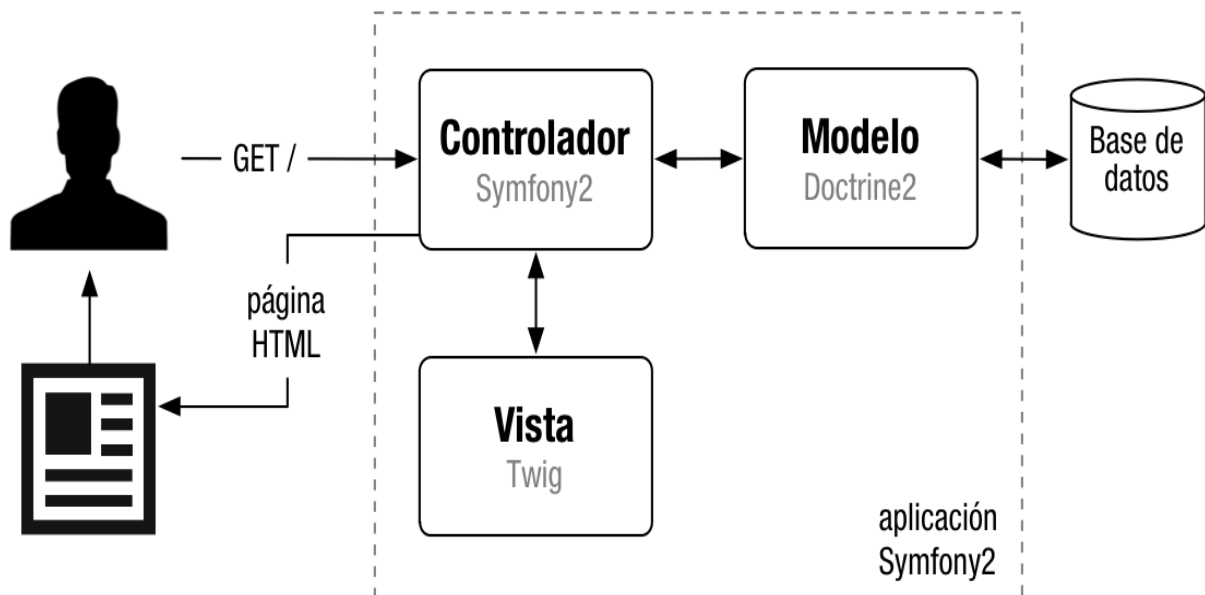


Figura 3⁸ Diagrama del patrón arquitectónico MVC

"*Symfony2* no es un *framework* MVC. *Symfony2* sólo proporciona herramientas para la parte del Controlador y de la Vista. La parte del Modelo es responsabilidad del desarrollador". La separación de la vista y el modelo trae ventajas; es posible tener diferentes representaciones de la misma información, haciendo uso del mismo código dentro del modelo. Es posible además programar el código del modelo, abstrayéndose de la representación visual que se le dará a la información [36].

2.8 Estructura en bundle de la aplicación

Los *bundles* son la base de la nueva filosofía de trabajo de *Symfony2*. El código de las aplicaciones y el propio código fuente de *Symfony2* se estructura mediante *bundles*. Técnicamente, un *bundle* es un directorio que contiene todo tipo de archivos dentro de una estructura jerarquizada de directorios. Los *bundles* de las

⁸ Tomado de: "Desarrollo ágil con *Symfony2*" por Javier Eguiluz publicado el 7-11-2013.

aplicaciones *Symfony2* suelen contener clases PHP y archivos web (*JavaScript*, CSS e imágenes). No obstante, no existe ninguna restricción sobre lo que puedes incluir dentro de un *bundle*. Tampoco existen límites técnicos sobre el tamaño que puede llegar a tener un *bundle* [37].

Para el desarrollo de la solución al problema planteado se decide crear un proyecto de *Symfony2* que consta de tres *bundles*:

UsuarioBundle: es el encargado de la gestión de usuarios de la aplicación.

FileManagerBundle: es el encargado de guardar los documentos en el servidor y de la gestión de estos.

AppBundle: es el encargado de toda la lógica de la aplicación.

2.9 Seguridad en el sistema

En el desarrollo de una aplicación web, la seguridad del sistema es uno de los aspectos más importantes, ya que es la garantía de la confidencialidad, integridad y disponibilidad de la información. Muchas aplicaciones web se ven afectadas en numerosas ocasiones por diferentes vulnerabilidades del propio código fuente o por descuido de los desarrolladores. En este sentido *Symfony2* implementa una serie de mecanismos para garantizar una aplicación segura, definiendo una lista de control de acceso, en el fichero *app/config/security.yml*.

Aparte de la seguridad que proporciona el *framework* utilizado, se estableció el mecanismo de autenticación al sistema basado en usuario y contraseña, además, para esta última se empleó un algoritmo de cifrado para almacenarla en la base de datos y se utilizó el modelo de control de acceso basado en roles, con estos procedimientos se asegura que sólo un usuario correctamente identificado pueda acceder al sistema, garantizando la confidencialidad e integridad de los datos, y estos usuarios solo podrán accionar sobre la información precisa que por el rol establecido en el sistema pueden controlar, también se estableció el protocolo *HTTPS* para asegurar el flujo de información entre el usuario y la aplicación.

2.10 Diagrama de clases del diseño

Los diagramas de clases del diseño permiten describir gráficamente las especificaciones de las clases del *software*. Muestran las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí [34].

Las siguientes figuras muestran los diagramas de clases del diseño con estereotipos web (DCDEW) del CU Gestionar Cliente.

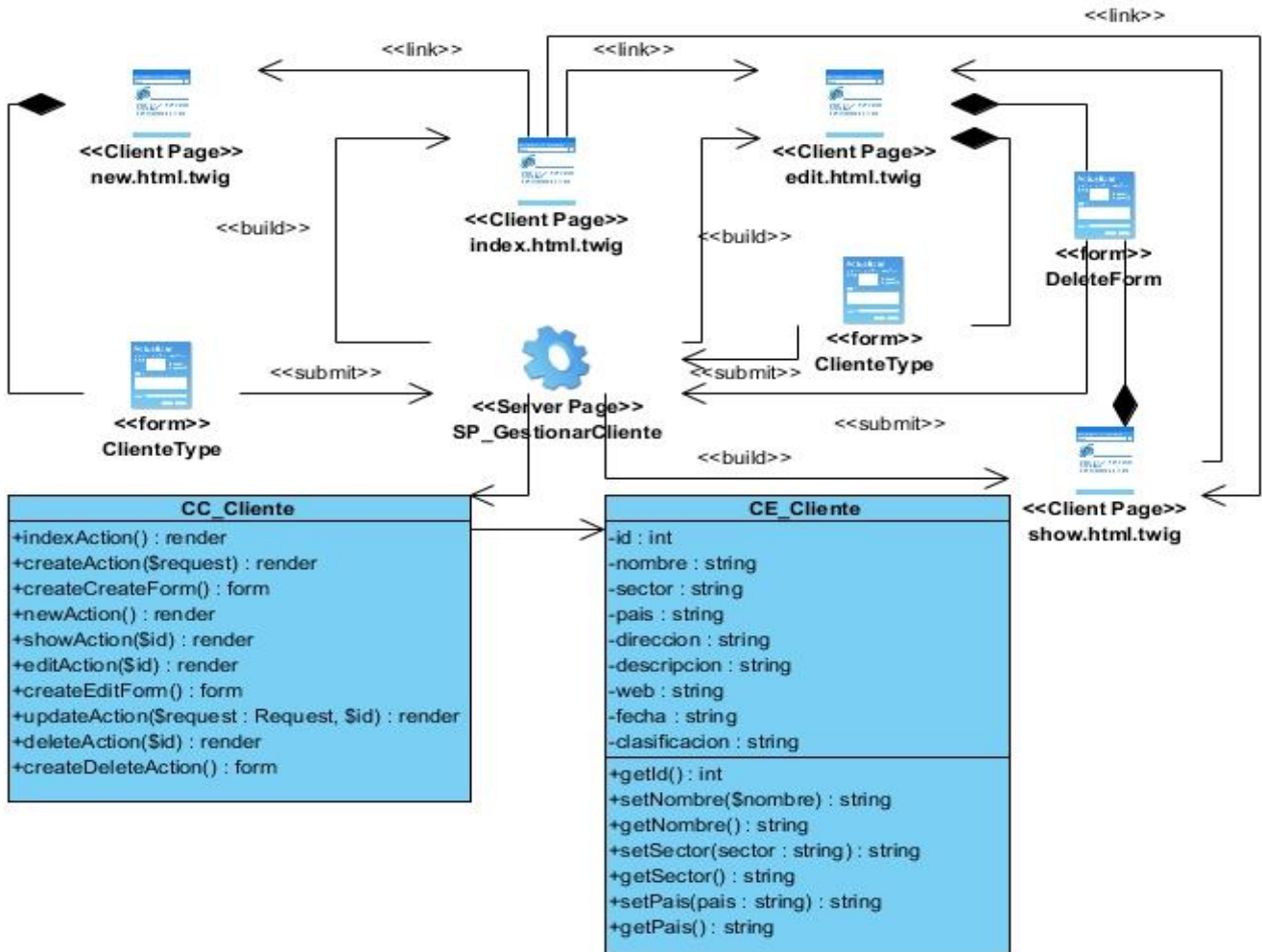


Figura 4 DCDEW de gestionar cliente

2.11 Modelo de datos

Un modelo de base de datos es un tipo de modelo de datos que determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos [38].

A continuación se expone el modelo de datos de la solución propuesta.

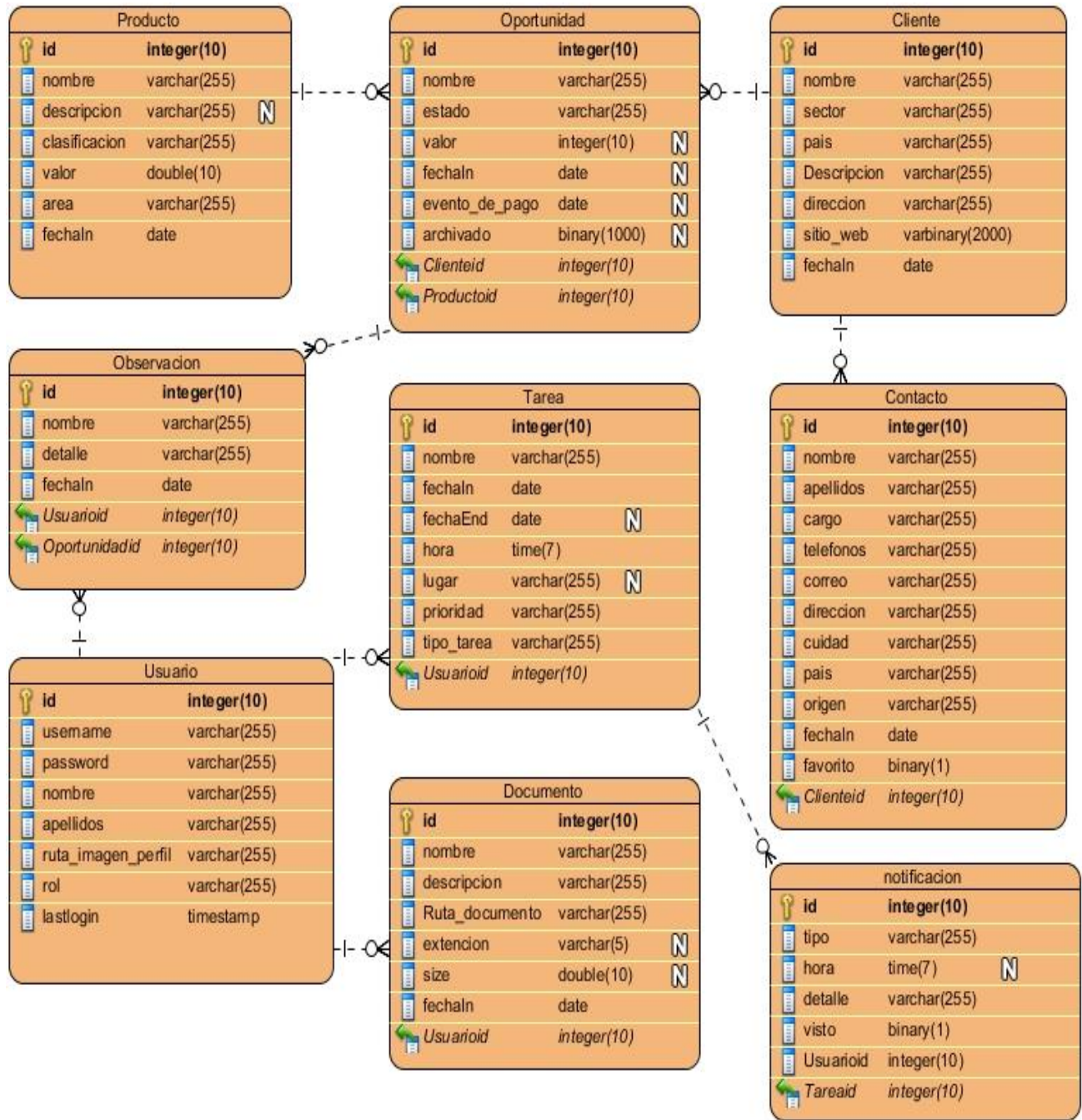


Figura 5 Modelo de datos

2.12 Conclusiones parciales

La modelación de los diferentes diagramas de clases del diseño, modelo de la base de datos y casos de uso, junto con el levantamiento de los requisitos, ayudaron al equipo de desarrollo a entender con más facilidad el problema planteado. Los patrones de diseño escogidos, propiciaron una arquitectura sólida y robusta al sistema. Se aplicaron diferentes mecanismos de seguridad con el fin de garantizar confiabilidad e integridad del sistema.

3 Capítulo 3. Implementación y pruebas del Sistema de gestión de relaciones con clientes

El momento cumbre del desarrollo de un proyecto es la implementación de cada una de las funcionalidades requeridas. En el presente capítulo se detalla la fase de construcción de la metodología OpenUP que se centra en la implementación y prueba de las funcionalidades para desarrollar un sistema completo. El propósito de la misma es completar el desarrollo del sistema basado en la arquitectura definida.

Por último, se le realizan las pruebas pertinentes al producto final, verificando las funcionalidades de la aplicación, y validando junto al usuario final la solución desarrollada a fin de contar con todas las evidencias que garanticen el cumplimiento en alcance, funcionalidad y calidad que el cliente espera.

3.1 Modelo de implementación

Este modelo representa la composición física de la implementación en términos de subsistemas de implementación. Describe cómo los elementos de diseño se implementan en componentes.

Se considera el artefacto más significativo del flujo de trabajo de Implementación, debido a la importancia que tiene para los desarrolladores comprender el funcionamiento del sistema desde el punto de vista de componentes y sus relaciones. Este modelo está conformado por los diagramas de componente y el de despliegue.

3.1.1 Diagrama de componentes

Los diagramas de componente describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos *software* que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente [28]. A continuación se muestra el diagrama de componentes general del sistema:

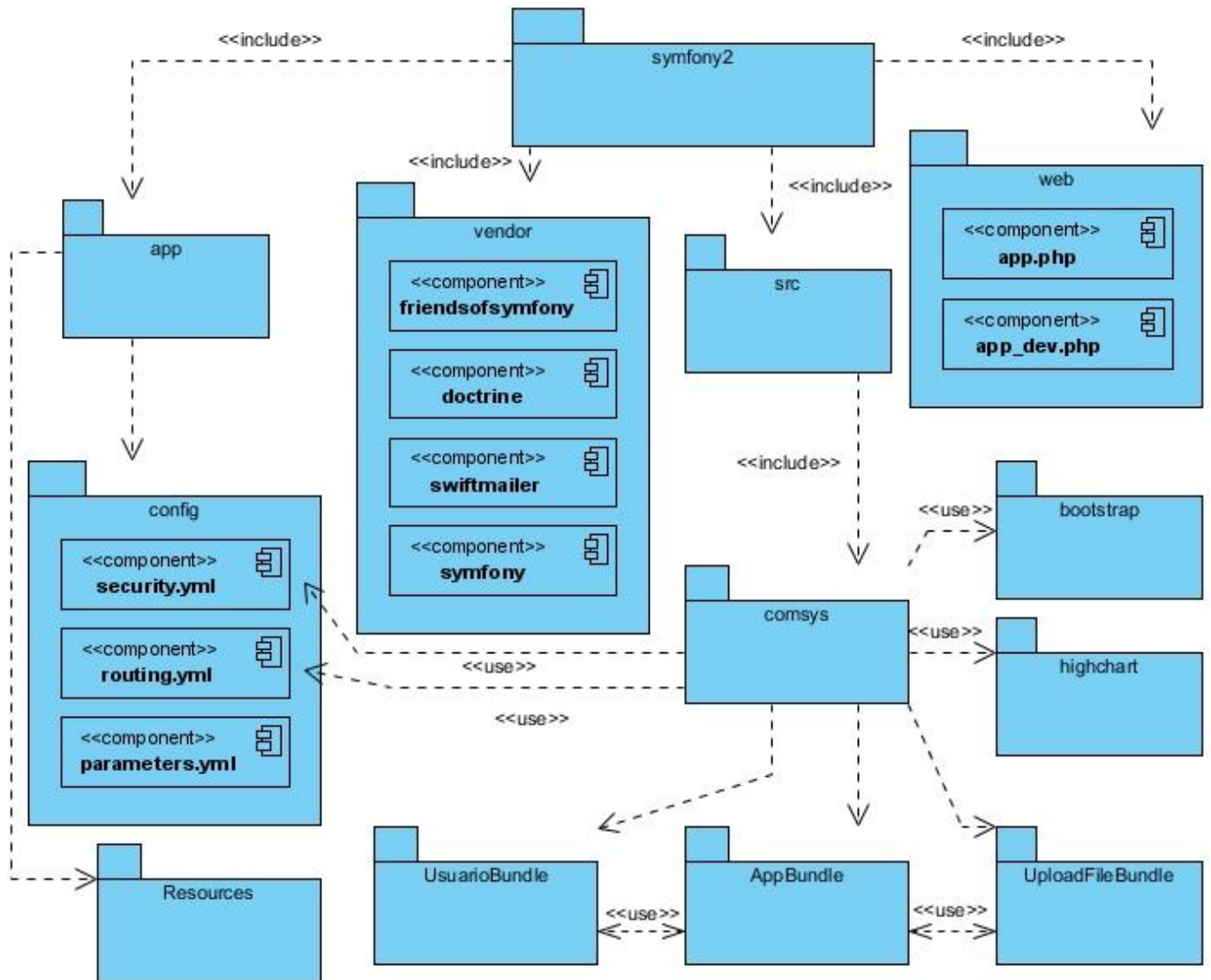


Figura 6 Diagrama de componentes general

3.1.2 Diagrama de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene memoria y a menudo, capacidad de procesamiento. Los nodos se utilizan para modelar la topología del *hardware* sobre el que se ejecuta el sistema. Representan un procesador o un dispositivo sobre el que se pueden desplegar

los componentes. La relación entre un nodo y el componente que despliega puede mostrarse con una relación de dependencia [35].

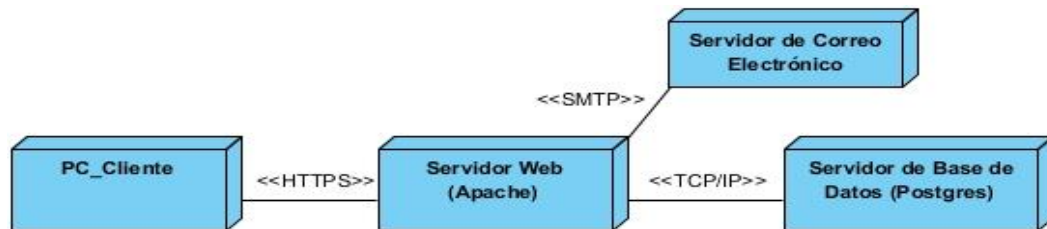


Figura 7 Diagrama de despliegue

3.2 Estándares de codificación

Un estándar es un modelo, norma, patrón, referencia o la especificación que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad. En la implementación de la aplicación web a desarrollar, se utilizarán varios estándares de codificación, que certificarán legibilidad y organización al código de la misma, simplificando esfuerzos a la hora de darle mantenimiento y seguimiento a la aplicación.

A continuación se especifican los estándares de codificación a utilizados en la construcción de la solución:

Camel Case: la notación "Camel Case" consiste en escribir frases o palabras compuestas eliminando los espacios intermedios y poniendo en mayúscula la primera letra de cada palabra [39].

Lower Camel Case: similar al Camel Case sólo que la primera letra de la primera palabra es también en minúscula.

Upper Camel Case: similar al Camel Case siendo la primera letra de la primera palabra en mayúscula.

El nombre de las clases debe ser declarado utilizando el estilo *Upper Camel Case* y cuando en un fichero se define el espacio de nombre (*namespace*) debe existir una línea en blanco a continuación de la declaración.

Además cuando están presentes todas las declaraciones de uso se escriben a continuación de la definición del *namespace* y se debe utilizar la palabra reservada *use* una vez por cada declaración, se debe concluir con una línea en blanco.

Se debe añadir un solo espacio después de cada delimitador coma y debe existir un solo espacio alrededor de los operadores (`==`, `&&`). Se coloca una coma después de cada elemento del arreglo en un arreglo multilínea, incluso después del último. Se añade una línea en blanco antes de las declaraciones *return*,

excepto en los casos que el valor devuelto solo sea dentro de un grupo de declaraciones (tal como una declaración if). Para indicar la estructura del cuerpo de control se usan llaves, independientemente del número de declaraciones que contenga. Deben declararse las propiedades de clase antes que los métodos. Se declaran primero los métodos públicos, luego los protegidos y finalmente los privados.

```
<?php

namespace comsys\AppBundle\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use comsys\AppBundle\Entity\Producto;
use comsys\AppBundle\Form\ProductoType;

class ProductoController extends Controller
{
    public function indexAction()
    {
        $em = $this->getDoctrine()->getManager();

        $entities = $em->getRepository('AppBundle:Producto')->findAll();

        return $this->render('AppBundle:Producto:index.html.twig', array(
            'entities' => $entities,
        ));
    }
}
```

Figura 8 Nombre de las clases, namespace y use

El nombre de las funciones debe ser declarado utilizando el estilo *Lower Camel Case*.

```
class ProductoController extends Controller
{
    public function indexAction() {...}

    public function createAction(Request $request){...}

    public function showAction($id)
    {
        $em = $this->getDoctrine()->getManager();

        $entity = $em->getRepository('AppBundle:Producto')->find($id);

        if (!$entity) {
            throw $this->createNotFoundException('Unable to find Producto entity.');
```

Figura 9 Nombre las funciones en Lower Camel Case

Los atributos de las clases deben ser privados, además, no debe existir más de un atributo por línea y deben utilizar el estilo *Lower Camel Case*.

```
class Producto
{
+ /** @var integer ...*/
private $id;
+ /** @var string ...*/
private $nombre;
- /**
* @var string
*
* @ORM\Column(name="descripcion", type="text")
*/
private $descripcion;
```

Figura 10 Atributos en Lower Camel Case

3.3 Pruebas funcionales

Las pruebas funcionales están basadas en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el *software*. Las mismas se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático. Son pruebas específicas, concretas y exhaustivas para comprobar y validar que el *software* hace lo que debe y sobre todo, lo que se ha especificado [40].

Se ejecutaron este tipo de prueba para todos los casos de uso con diferentes entradas del usuario, con el objetivo de determinar que los resultados obtenidos fueran los esperados bajo cualquier situación, y así, dar por cumplidos los requerimientos del cliente. A continuación se presenta una muestra del caso de prueba Gestionar Cliente:

Tabla 5 Descripción de las variables

No.	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Nombre	Campo de texto	No	Este campo admite caracteres alfanuméricos.
2	Sector	Campo de texto	No	Muestra una lista desplegable con los clientes existentes.
3	País	Lista desplegable	No	Muestra una lista desplegable con los países.

4	Dirección	Área de texto	Sí	Este campo admite todos los caracteres.
5	Descripción	Área de texto	Sí	Este campo admite todos los caracteres.
6	Sitio web	Campo de texto	No	Este campo admite solo direcciones web reales.

Tabla 6 Prueba funcional para el CU Gestionar oportunidad, Sección: Crear oportunidad

Escenario	Descripción	Variables	Respuesta del sistema	Flujo central
EC 1.1 Crear un cliente correctamente	El sistema permite adicionar un cliente	Nombre (V): Cliente1 Sector (V): Salud Cuba (V): Cuba Dirección (NA) Descripción (NA) Sitio web (V) http://cubadebate.cu	El sistema muestra el siguiente mensaje: "Un nuevo cliente ha sido creado"	El usuario selecciona en el menú principal la opción Cliente/ Nuevo Cliente. Inserta la información requerida y selecciona el botón "Crear".
EC 1.2 Introduce datos incorrectos y da clic en el botón Crear.	El sistema no permite adicionar un cliente porque existen datos incorrectos	Nombre (I): ++ Sector (V): Salud Cuba (V): Cuba Dirección (NA) Descripción (NA) Sitio web (I) web.23	El sistema muestra un mensaje de error indicando que se han introducido datos incorrectos y marca el o los campos erróneos.	
EC 1.3 Introduce los datos dejando campos obligatorios vacíos y da clic en el botón Crear.	El sistema no permite crear el cliente pues el usuario deja los campos obligatorios vacíos.	Nombre (I): Sector (I): Cuba (V): Cuba Dirección (NA) Descripción (NA) Sitio web (NA)	El sistema muestra un mensaje de error debajo de cada campo obligatorio que se encuentre vacío, indicando que el campo es requerido.	

Las celdas de la tabla contienen V, I, o N/A; V indica válido, I indica inválido, y N/A que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Estas pruebas fueron realizadas a la aplicación, arrojando como resultado un total de 17 no conformidades agrupadas en 5 tipos: errores de idioma, validación, ortografía, funcionalidad y redacción. La ejecución de estas pruebas se distribuyó en 3 iteraciones, en una primera iteración se encontraron 12 no conformidades que fueron resueltas de manera satisfactoria, en la segunda se detectaron 5 no conformidades resueltas también satisfactoriamente y en la tercera iteración no se encontraron errores ([Ver figura 10](#)).

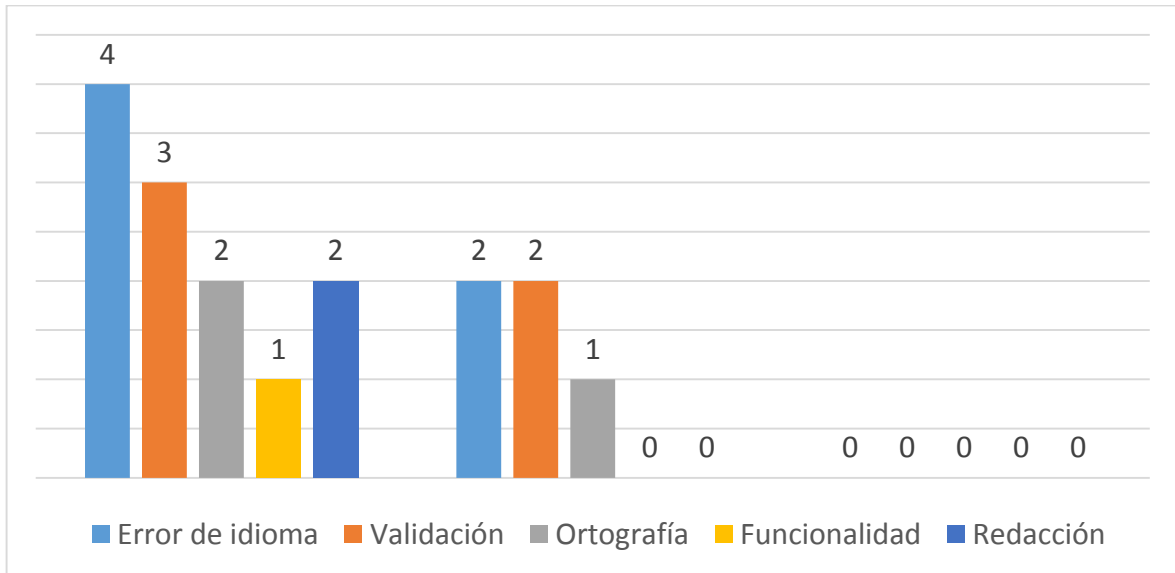


Figura 11 Resultados de las pruebas funcionales

3.4 Pruebas de seguridad

Las pruebas de seguridad comprueban que los mecanismos de protección integrados en el sistema realmente lo protejan de irrupciones inapropiadas. En ellas se debe intentar conseguir las claves de acceso por cualquier medio, deben producir a propósito errores del sistema para ser corregidos luego por los desarrolladores del sistema [41].

Para garantizar la seguridad del sistema se realizaron pruebas de seguridad mediante una lista de chequeo ajustable a aplicaciones web. El objetivo general de esta lista es evaluar a través de indicadores la seguridad de las aplicaciones en un primer nivel, establecido por las especialistas del grupo de seguridad del Departamento de Pruebas de Software (DEPSW) de la UCI, dichas pruebas cuentan con 15 indicadores separados en 4 tipos de pruebas. Los resultados de las mismas se exponen a continuación.

Forma de uso de la lista de chequeo:

Peso: Define si el indicador a evaluar es crítico o no.

Evaluación (Eval): Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal y 0 en caso de que el elemento revisado no presente errores.

Cantidad de elementos afectados: Especifica la cantidad de errores encontrados sobre el mismo indicador.

Comentarios: Especifica los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo.

Indicadores a Evaluar: Descripción del parámetro a evaluar.

N.P. (No Procede): Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

Estructura de la lista de chequeo:

Tabla 7 Lista de chequeo prueba de autorización

Pruebas de Autorización					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Puede un usuario estándar (no administrador) modificar sus privilegios en la aplicación.	0		0	
Crítico	Puede un usuario estándar (no administrador) modificar los privilegios de otro usuario.	0		0	

Tabla 8 Lista de chequeo prueba de autorización

Pruebas de Gestión de Sesiones					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Copiar la URL de la aplicación estando autenticado, cerrar el navegador y volver a abrirlo, pegar la dirección copiada anteriormente, la aplicación permite que el usuario conserve su estado de autenticado.	0		0	

Crítico	Al cerrar la sesión de un usuario y dar clic en el botón del navegador "Atrás" la aplicación vuelve a entrar a la sesión autenticada.	0		0	
---------	---	---	--	---	--

Tabla 9 Lista de chequeo comprobación del sistema de autenticación

Comprobación del sistema de Autenticación					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Se cierra la sesión del usuario después de un tiempo X (establecido por la aplicación) estando sin usar la misma.	0		0	
Crítico	Se bloquea la cuenta del usuario después de un número X (establecido por la aplicación) de intento de login fallidos por el usuario. De ser así definir la cantidad de intentos en la columna Comentarios.	0		0	
Medio	Se muestran diferentes mensajes de error al usuario "intenta autenticarse con un usuario correcto y una contraseña incorrecta" y "al autenticarse con un usuario incorrecto y la misma contraseña del intento anterior".	0		0	
Crítico	El mensaje devuelto al intentar autenticarse con un usuario correcto y contraseña incorrecta, es el mismo al	0		0	

	intento con usuario y contraseña incorrecta, si es el mismo intento verificar el tiempo de respuesta, pues podría apreciarse una media ligeramente diferente en el tiempo de respuesta, que igualmente puede usarse esta información para descubrir cuentas válidas.				
Crítico	El campo usuario de la autenticación al sistema tiene el auto completamiento activado (guarda los usuarios que se autentican). Para ver esto se debe autenticar al sistema, luego salir y poner la primera letra del usuario autenticado para ver si muestra el nombre de usuario completo.	0		0	
Crítico	El sistema protege el envío de los datos mediante protocolo seguro (<i>https</i>).	0		0	
Crítico	El sistema usa algún certificado.	0		0	

Tabla 10 Lista de chequeo validación de datos

Validación de Datos					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	Se enmascaran datos confiables cuando se visualicen en la aplicación (Por ejemplo: Contraseñas).	0		0	

Medio	La aplicación solamente permite contraseñas alfanuméricas, que incluyan caracteres especiales y que tengan seis caracteres mínimos de longitud.	0		0	
Crítico	La aplicación permite la funcionalidad de cambio de contraseña únicamente a usuarios autenticados validando la antigua contraseña, la nueva contraseña y la respuesta a la pregunta de seguridad (opcional en dependencia del proyecto).	0		0	
Crítico	El sistema muestra algún mensaje indebido, al colocar en la barra de dirección o en campos de entrada los caracteres: comillas simples.	0		0	

La aplicación de la lista de chequeo arrojó un total de tres no conformidades: una dentro de la clasificación “Comprobación del Sistema de Autenticación” y dos dentro de “Validación de Datos”; determinándose que la relacionada con el tipo de contraseñas permitidas por la aplicación y los diferentes mensajes de error mostrados por la aplicación en respuesta a una autenticación fallida no proceden.

También, para evaluar la seguridad del sistema se utilizó la herramienta *Acunetix Web Vulnerability Scanner 8.0*. Esta herramienta agrupa las vulnerabilidades encontradas en cuatro categorías: Alta, Media, Baja e Informativa. Se realizaron un total de dos iteraciones de pruebas, la primera arrojó un total de 3 vulnerabilidades de tipo Media, 2 de tipo Baja y 5 de tipo Informativa, estas últimas son referentes al autocompletamiento de los campos “Usuario” y “Contraseña” en los formularios de *login* con que cuenta la

aplicación, para resolver este tema se eliminó esta opción para los campos especificados. En una segunda iteración no se detectaron vulnerabilidades. ([Ver Figura 11](#)).

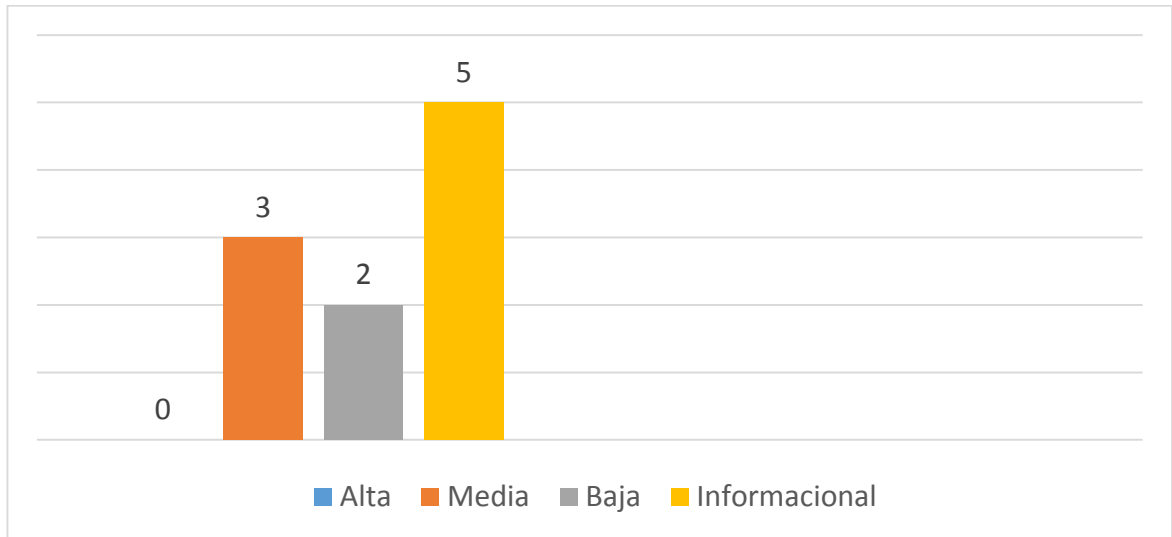


Figura 12 Resultados de las pruebas de seguridad con Acunetix

3.5 Conclusiones parciales

La elaboración del diagrama de componentes permitió una mejor comprensión de la estructura del sistema de gestión de relaciones con clientes. Para lograr un código estándar, legible y fácil de comprender se aplicaron estándares de codificación. La realización de pruebas funcionales y de seguridad permitió detectar y corregir las no conformidades, así como validar el correcto funcionamiento de la solución desarrollada.

Conclusiones

El presente trabajo dio cumplimiento a cada uno de los objetivos trazados, pudiéndose destacar de manera general, las conclusiones siguientes:

- El estudio de las herramientas de gestión de relaciones con clientes en el ámbito nacional y extranjero, las tendencias actuales de las mismas y las exigencias del cliente, permitió elegir las herramientas, tecnologías, los lenguajes y metodología para guiar el proceso de desarrollo.
- En la investigación realizada se observó que los CRM existentes no cumplen totalmente con las necesidades del centro CIDI; pero propician grandes ventajas en el área de mercadotecnia y aportaron un gran valor en la identificación de requisitos funcionales para el desarrollo de la aplicación.
- La identificación y especificación de los requisitos funcionales, así como la elaboración de los correspondientes diagramas de clases permitió diseñar la propuesta de solución y sentar las bases para la correcta implementación del sistema.
- Como resultado de la implementación y dándole cumplimiento al objetivo general de la investigación se obtuvo un Sistema de gestión de relaciones con clientes (COMSYS), capaz de gestionar toda la información del centro CIDI relacionada con sus clientes, para apoyar la toma de decisiones.
- La aplicación de las pruebas funcionales y de seguridad permitió validar el correcto funcionamiento de la aplicación desarrollada.

Recomendaciones

Con la culminación del presente trabajo, se dio cumplimiento a los objetivos propuestos, dotando al centro CIDI de una aplicación para la gestión de relaciones con clientes que cumple con todas las funcionalidades requeridas por el mismo. Sin embargo, se realizan una serie de recomendaciones para enfocar el trabajo futuro en el mejoramiento de la propuesta realizada:

- Incluir funcionalidades de envío automático de reportes por correo.
- Implementar funcionalidades para detectar clientes potenciales.
- Incluir funcionalidades para realizar campañas de marketing automáticas.

Glosario de términos

ERP (Enterprise Resource Planning) El sistema de planificación de recursos empresariales, más conocido como ERP (Enterprise Resource Planning) es un sistema estructurado que busca satisfacer la demanda de soluciones de gestión empresarial. Se basa en el concepto de una solución completa que permita a las empresas unificar las diferentes áreas de productividad de la misma [42]. Los ERP son sistemas que le permiten a las empresas gestionar toda la información en tiempo real, permitiendo una mejor planificación, análisis y gestión de sus recursos. En ocasiones estos son complejos y difíciles de implantar debido a que necesitan un desarrollo personalizado para cada empresa partiendo de la parametrización inicial de la aplicación que es común para todas.

Clientes En toda estrategia empresarial, el cliente es una figura que siempre está presente, es lógico, es quien demanda los productos y servicios que las empresas ofrecen, por lo que consiguen consolidarse en el mercado para obtener rendimientos e ingresos para posicionarse y sobrevivir [43]. Los clientes son aquellos que le hacen compras a la empresa de forma periódica o que lo hicieron en una fecha reciente.

Cliente Potencial Es aquella (persona, empresa u organización) que no le realiza compras a la empresa en la actualidad pero que son visualizados como posibles clientes en el futuro porque tienen la disposición necesaria, el poder de compra y la autoridad para comprar. Este tipo de clientes es el que podría dar lugar a un determinado volumen de ventas en el futuro (a corto, mediano o largo plazo) y por tanto, se les puede considerar como la fuente de ingresos futuros [44]. Cliente potencial es una persona, empresa u organización que aún no se relaciona con la empresa pero pueden ser un posible cliente en el futuro.

Marketing es el proceso social y administrativo por el que los grupos e individuos satisfacen sus necesidades al crear e intercambiar bienes y servicios [45].

Marketing relacional es la actividad del *marketing* que tiene el fin de generar relaciones rentables con los clientes. Esto parte del estudio de comportamiento de los compradores con base en el diseño de estrategias y acciones destinadas a facilitar la interacción con los mismos y brindarles una experiencia memorable [46].

Referencias

1. *Evolución del marketing hacia la gestión orientada al valor del cliente.* **Fernández, Leslier Maureen.** 2, s.l. : Theoria, 2006, Vol. 15.
2. **MCLEOD, Raymond.** *Sistemas de información gerencial.* s.l. : Pearson educación, 2000.
3. **Peppers, Don y Rogers, Martha.** *Managing Customer Relationships: A Strategic Framework (Second Edition).* s.l. : wiley, 2011.
4. Microsoft Dynamics. [En línea] 2011. [Citado el: 10 de 3 de 2015.] <http://www.microsoft.com/latam/dynamics/crm/introduccion/default.aspx..>
5. Microsoft Dynamics. [En línea] 2011. [Citado el: 10 de 3 de 2015.] <http://www.microsoft.com/latam/dynamics/crm/datasheet/default.aspx..>
6. SugarCrmhispano.com. [En línea] 2007. [Citado el: 10 de 3 de 2015.] http://www.sugarcrmhispano.com/crm/vp-tid:7-pid:2-que_es_sugarcrm.html..
7. sugarCRM. [En línea] [Citado el: 28 de 5 de 2015.] <http://www.sugarcrm.com/products/editions-and-pricing>.
8. excale Solutions. [En línea] 2011. [Citado el: 11 de 3 de 2015.] <http://www.excalesolutions.com/servicios/desarrollo-implementacion-formacion-openerp..>
9. Cybercia. [En línea] 2012. [Citado el: 11 de 3 de 2015.] http://www.cybercia.com/index.php?option=com_content&view=article&id=52&Itemid=71#1.2..
10. Tecnologías información. [En línea] 2012. [Citado el: 11 de 3 de 2015.] <http://www.tecnologias-informacion.com/crm.html..>
11. OaxRom. [En línea] 2010. [Citado el: 11 de 3 de 2015.] http://www.oaxrom.com/index.php?option=com_content&view=article&id=29&Itemid=56..
12. INFORMATICA HOY. [En línea] 2014. [Citado el: 11 de 3 de 2015.] <http://www.informatica-hoy.com.ar/software-crm/SAP-CRM.php..>
13. *CEDRUX. Solución para sistemas de control logístico. CUJAE: 15 Convención Científica de Ingeniería y Arquitectura.* **Quevedo Campins, Virgen Damaris.** 2010.
14. **Zend Technologies Ltd.** Zend Framework Manual. [En línea] 2011. [Citado el: 4 de 12 de 2014.] <http://framework.zend.com/manual/en/introduction.overview.html>.
15. symfony.es. [En línea] [Citado el: 1 de 12 de 2014.] <http://symfony.es/que-es-symfony>.
16. Genveta:dev. [En línea] 2012. [Citado el: 5 de 12 de 2014.] <http://www.genbetadev.com/frameworks/bootstrap>.
17. [En línea] 2014. <http://php.net/manual/es/intro-what-is.php>.

18. php.net. [En línea] 2014. <http://php.net/manual/es/intro-what-is.php>.
19. desarrolloweb. [En línea] 2013. <http://www.desarrolloweb.com/javascript/#quees>.
20. Bitelia. [En línea] 05 de 2013. <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>.
21. CSS3 HTML5. [En línea] 23 de noviembre de 2010. <http://html5.dwebapps.com/que-es-css3/>.
22. **Arias, Angel.** *Aprende a Programar Ajax y jQuery*. 2014.
23. jQuery.org. [En línea] 2014. [Citado el: 2 de 12 de 2014.] <https://jquery.org/license/>.
24. jQuery.com. [En línea] 2014. [Citado el: 2 de 12 de 2014.] <http://jquery.com/browser-support/>.
25. **Martínez, Rafael.** PostgreSQL. [En línea] 2 de 10 de 2010. http://www.postgresql.org.es/sobre_postgresql.
26. **Bowen, Rich.** What's New in Apache Web Server 2.2? [En línea] 2007.
27. Netbeans. [En línea] Oracle, 2014. [Citado el: 4 de 11 de 2014.] <https://netbeans.org/features/php>.
28. **Sommervil, Ian.** Ingeniería de Software. 2005.
29. Marco de Desarrollo de la Junta de Andalucía. [En línea] [Citado el: 15 de 5 de 2015.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/656>.
30. **PRESSMAN, Roger S.** *Ingeniería de Software: Un enfoque práctico*. s.l. : Mikel Angoar, 1997.
31. *Metodologías Ágiles en el desarrollo de Software*. **Canós, J. H., Letelier, P., Pennadés, M. C.** VIII Jornadas de Ingeniería de Software y Bases de Datos : s.n., 2003.
32. **Brito Acuña, Kareenny.** Por qué utilizar Scrum para desarrollar aplicaciones web. [En línea] 2009. [Citado el: 4 de 12 de 2014.] <http://www.eumed.net/libros/2009c/584/Por%20que%20utilizar%20Scrum%20para%20desarrollar%20aplicaciones%20web.htm>.
33. **IBM.** Rational Unified Process. [En línea] IBM, 2011. [Citado el: 5 de 11 de 2014.]
34. **Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico. Sexta Edición*. s.l. : INTERAMERICANA DE MEXICO, 2005.
35. **Larman, Craig.** *UML y patrones Tomo II*. s.l. : PEARSON, 2003.
36. **Sebastián, J.** *Modelo Vista Controlador-Definición y características*. 2010.
37. **Eguiluz, Javier.** *Desarrollo web ágil con Symfony2*. Madrid. 2013.
38. **Olga Pons, Nicolas Marin.** *Introducción a las base de datos: modelo relacional*. s.l. : Paraninfo, 2005.
39. **Lago, Diego.** Guía de estilode programación. [En línea] 2008. <http://did.cexto.com>.

40. **Adame, Javier Venegas.** AUTOMATIZACIÓN DE TESTING. [En línea] Barcelona.
41. **Quality, V&V.** Pruebas de seguridad. [En línea] 2011. [Citado el: 20 de 04 de 2015.]
http://www.vvquality.com/w1/index.php?option=com_content&view=article&id=79&Itemid=151.
42. GestioPolis.com. [En línea] 2004. [Citado el: 1 de 6 de 2015.]
<http://www.gestiopolis.com/recursos2/documentos/fulldocs/ger/erpjoel.htm..>
43. Microsoft Pymes y autónomos. [En línea] 1 de 6 de 2015. <http://www.microsoft.com/business/es-es/Content/Paginas/article.aspx?cbcid=475..>
44. Clientes-Potenciales. [En línea] [Citado el: 1 de 6 de 2015.] <http://clientes-potenciales.blogspot.com/..>
45. **KOTLER, Philip y ARMSTRONG, Gary.** *Fundamentos de Marketing*. 2003. págs. 2-41.
46. **LARA, Pedro J. Reinares y CASADO, José Manuel Ponzoa.** *Marketing relacional*. s.l. : Financial Times/Prentice Hall, 2004.
47. ieb.school. [En línea] 7 de febrero de 2014. <http://comunidad.iebschool.com/iebs/agile-scrum/que-es-agile/>.
48. scribd. [En línea] 2013. <http://es.scribd.com/doc/37116717/Open-Up>.
49. **Vea.** [En línea] 2013.
50. **Jeffries, R, Anderson, A y Hendrickson, C.** *Extreme Programming Installed*. s.l. : Addison-Wesley, 2001.
51. SIESA. [En línea] 2011. [Citado el: 10 de 3 de 2015.] <http://www.siesa.com/oferta-de-valor/94.html..>

Bibliografía consultada

- Ana P. Rodríguez, Josué Polanco, Darwin Hernández.** 2013. *Open UP*. 2013.
- ASSETS. [En línea] <http://www.assets.co.cu/assets.asp>.
- Bowen, Rich.** 2007. *What's New in Apache Web Server 2.2?* 2007.
- Byteflair.** 2013. Byteflair. [En línea] 2013. [Citado el: 15 de Mayo de 2015.]
<http://byteflair.com/2013/07/comenzando-con-elasticsearch/>.
- Carrasco, Nicolas.** 2014. Prezi. [En línea] 2014. [Citado el: 22 de Febrero de 2015.]
<http://prezi.com/eunnrsm1pfba/copy-of-informatica-forense/>.
- Celaya, Javier.** 2008. *La empresa en la web 2.0*. s.l. : Gestión 2000, 2008.
- Chavez, Miguel Angel.** 2012. *AIRESWEB: SISTEMA DE REPORTES Y ANÁLISIS INTELIGENTE DE REGISTROS WEB*. 2012.

- Cuervo, Víctor. 2012.** aulambra.com. *Tecnologías de Internet*. [En línea] 28 de febrero de 2012. [Citado el: 16 de enero de 2015.] <http://www.aulambra.com/novedades/programacion/dojo-1-7/>.
- Eguiluz, Javier. 2013.** *Desarrollo web ágil con Symfony2*. Madrid : s.n., 2013.
- . **2008.** *Introducción a JavaScript*. 2008.
- Eguiluz, Javier y Potencier, Fabian. 2011.** *Desarrollo web ágil con Symfony 2*. 2011.
- Fabien Potencier, Ryan Weaver. 2014.** *Symfony 2.3, el libro oficial*. 2014.
- Fowler, M. 2003.** *Web Server Patterns. Model View Controller, Patterns of Enterprise Application Architecture*. 2003.
- G. Figeroa, Robert, J. Solis, Camilo y A. Cabrera, Armando. 2008.** *Metodologías tradicionales vs. metodologías ágiles*. 2008.
- Ganazhapa, Ing. Jenny Alexandra Pujos. 2012.** *EL SITIO WEB Y SU INCIDENCIA EN LA DIFUSIÓN Y PROMOCIÓN DEL CENTRO EDUCATIVO ALBERT EINSTEIN UBICADA EN LA PARROQUIA CIUDAD NUEVA DEL CANTÓN PÍLLARO PROVINCIA DE TUNGURAHUA*. 2012.
- Garcia, Gines Bravo. 2010.** *PHP 5 A traves de ejemplos*. 2010.
- Gauita, Alvaro Martínez. 2009.** desarrolloweb.com. [En línea] 2009. [Citado el: 22 de Marzo de 2015.] http://www.desarrolloweb.com/de_interes/chartbeat-traffic-web-tiempo-real-3176.html.
- Genveta:dev. 2012.** Bootstrap, framework de twitter. [En línea] 16 de Junio de 2012. [Citado el: 4 de Mayo de 2015.] <http://www.genbetadev.com/frameworks/bootstrap>.
- Godoy Jiménez, José Manuel. 2002.** Diseño de proyectos de software en código abierto. [En línea] 2002. <http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-dise%F1o-software/doc-dise%F1o-software-parte-1.pdf>.
- González, Eduardo Sánchez. 2014.** EstrategasWeb. [En línea] Marzo de 2014. [Citado el: 23 de Mayo de 2015.] <http://estrategasweb.com/wp-content/uploads/2014/03/Introducci%C3%B3n-Anal%C3%ADtica-Web-Parla2.pdf>.
- Google.** Sistemas viables. [En línea] [Citado el: 3 de Marzo de 2015.] <https://www.google.com.cu/#q=Google+Analytics+ayuda+a+dise%C3%B1ar+anuncios+m%C3%A1s+orientados%2C+a+mejorar+las+iniciativas+de+marketing+y+a+crear+sitios+web+que+generen+m%C3%A1s+conversiones>.
- Groussard, Thierry. 2010.** *Recursos Informáticos Java Enterprise Edition*. Barcelona : s.n., 2010.

- Gutiérrez, Javier J.** Departamento de Lenguajes y Sistemas Informáticos. *Departamento de Lenguajes y Sistemas Informáticos*. [En línea] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
- Herrera, Daniel Ernesto Navarro. 2011.** LaWebera.es. ¿Cómo darle un uso apropiado a CSS3? [En línea] 31 de octubre de 2011. [Citado el: 2014 de diciembre de 19.] <http://www.lawebera.es/maquetacion-web/darle-uso-apropiado-css3.php#>.
- . **2012.** LaWebera.es. *MooTools vs otros frameworks. Comparativa*. [En línea] 14 de febrero de 2012. [Citado el: 15 de enero de 2015.] <http://www.lawebera.es/mootools/mootools-vs-otros-frameworks-comparativa.php>.
- Hipertextual. 2013.** Bitelia. [En línea] 28 de Mayo de 2013. [Citado el: 10 de Febrero de 2015.] <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>.
- Jacobson, Booch y J, Rumbaugh. 2000.** *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.
- 2014.** Junta de Andalucía. [En línea] 2014. [Citado el: 11 de Mayo de 2015.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/656>.
- Kaushik, Avinash. 2011.** *Analítica Web 2.0*. s.l. : Gestión 2000, 2011.
- . **2011.** *Analítica Web 2.0*. s.l. : Gestión 2000, 2011. ISBN: 9788498750959.
- Kotler, Philip. 2008.** *Principios de Marketing*. s.l. : Pearson Educación, 2008.
- Lago, Diego. 2008.** Guía de estilode programación. [En línea] 2008. [Citado el: 4 de Mayo de 2015.] <http://did.cexto.com>.
- Larman, Craig. 2003.** *UML y patrones Tomo II*. s.l. : PEARSON, 2003.
- . **2003.** *UML y Patrones. 2da Edición*. s.l. : Prentice Hall, 2003. ISBN: 84-205-3438-2.
- LibrosWeb. 2014.** LibrosWeb. [En línea] 2014. [Citado el: 16 de Febrero de 2015.] http://librosweb.es/JavaScript/capitulo_1/breve_historia.html.
- Lima, Fernando. 2014.** SlidePlayer. [En línea] 2014. [Citado el: 7 de Febrero de 2015.] <http://slideplayer.es/slide/143422/>.
- Marco de desarrollo de la Junta de Andalucía. 2014.** Acunetix Web Vulnerability Scanner. [En línea] 2014. [Citado el: 24 de enero de 2015.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/656>.
- Marley, Jimi. 2011.** Programacion. [En línea] 2011. http://www.programacion.com/articulo/%20por_que_elegir_php_143.

- 2013.** Mootools. *a compact JavaScript framework*. [En línea] 2013. [Citado el: 15 de enero de 2015.]
<http://mootools.net/>.
- Morgado, Jose Maria Acuña. 2013.** Reeditor.com. *Librerías JavaScript de gran utilidad para tu desarrollo web*. [En línea] 9 de octubre de 2013. [Citado el: 13 de enero de 2015.]
<http://www.reeditor.com/columna/11084/17/informatica/librerias/JavaScript/gran/utilidad/tu/desarrollo/web>.
- Ojeda, Antonio Navajas. 2012.** *Guía Completa de CSS3*. s.l. : Autoedición, 2012.
- Olga Pons, Nicolas Marin. 2005.** *Introducción a las bases de datos: el modelo relacional*. s.l. : Paraninfo, 2005.
- Oracle. 2013.** NetBeans. [En línea] Oracle, 2013. [Citado el: 6 de Mayo de 2015.]
https://netbeans.org/index_es.html.
- Peter Eisentraut, Bernd Helmle. 2008.** *PostgreSQL-Administration*. 2008.
- 2009.** Posicionamiento y Marketing en Buscadores. [En línea] 2009. [Citado el: 4 de Febrero de 2014.]
http://pos-y-marketing-en-buscadores-2013.blogspot.com/2013_04_01_archive.html.
- Potencier, Fabien.** Symfony. *Oficial Site*. [En línea] <http://symfony.com/>.
- Pressman, Roger. 2005.** *Ingeniería del Software. Un enfoque práctico. Sexta Edición*. s.l. : INTERAMERICANA DE MEXICO, 2005. ISBN 9789701054734.
- . **2005.** . *Un enfoque práctico a la Ingeniería de Software*. 2005. .
Proceso de Desarrollo de Software Metodología DAC. **Mendez, Ing.Aleli Sanchez.**
- Quality, V&V. 2011.** Pruebas de seguridad. [En línea] 2011. [Citado el: 24 de Mayo de 2014.]
http://www.vvquality.com/w1/index.php?option=com_content&view=article&id=79&Itemid=151.
- Rafal Kuc, Marek Rogozinski. 2013.** . *Mastering Elasticsearch server*. s.l. : PACKT PUBLISHING,, 2013.
. ISBN-10: 178328143X..
- 2007.** RibosoMatic. [En línea] 17 de noviembre de 2007. [Citado el: 16 de enero de 2014.]
<http://www.ribosomatic.com/articulos/top-librerias-ajax-dhtml-y-JavaScript/>.
- Salinas Caro, Patricio.** Tutorial de UML. [En línea] <http://www.dcc.uchile.cl/~psalinas/uml/>.
- Sebastián, J. 2010.** *Modelo Vista Controlador- Definición y Características*. 2010.
- Silva, Maurício Samy.** *JQuery. A Biblioteca do Programador JavaScript*. Brasil : s.n.
Sistema de Gestión de la Nueva Universidad. *Sistema de Informacion Docente de la Educacion Cubana*.
[En línea] <http://sigenu.mes.edu.cu:8080/dmmes/pages/info/aboutUsWelcome.faces>.
- Sommerville, Ian. 2005.** *Ingeniería de Software*. 2005.

Suárez, Pablo Javier Kabytes. 2010. Excelentes plugins para crear gráficas con jquery. [En línea] 8 de Octubre de 2010. [Citado el: 30 de enero de 2014.] <http://www.kabytes.com/programacion/4-excelentes-plugins-para-crear-graficas-con-jquery/>.

The Apache Software Foundation. Apache. [En línea] <http://httpd.apache.org/>.

The jQuery Foundation. JQuery. *write less, do more*. [En línea] [Citado el: 15 de enero de 2015.] <http://jquery.com/>.

Toma de decisiones empresariales. **Fincowsky, Enrique Benjamín Franklin. 2011.** 11, 2011, Vol. 6.

Vea, Andreu. 2013. *Cómo creamos Internet*. s.l. : Península, 2013.

Visual Paradigm. *Oficial Site*. [En línea] <http://www.visual-paradigm.com/product/vpuml>.

Zambrano, Renán Alexi Briones. 2013. *Metodología de Investigación*. 2013.