Universidad de las Ciencias Informáticas

Facultad 3

Implementación de los procesos Atención al cliente y Soporte dentro del Sistema de Administración de Relaciones con el Cliente.

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

Autor:

Antonio García López

Tutores:

Ing. Dayannis Estrada Duarte

Ing. Sonia Fernández Henríquez

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al Centro para la Informatización de Gestión de Entidades de la Universidad de las Ciencias Informáticas; a que haga el uso que estime pertinente con este trabajo.

Para que así conste firmo la presente declaración de autoría a los ____ días del mes de julio del año 2013.

_____

Antonio García López

Autor

_____

Ing. Dayannis Estrada Duarte.

Tutor

_____

Ing. Sonia Fernández Henríquez

Tutor

La revolución no se lleva en los labios para vivir de ella, se lleva en el corazón para morir por ella.

Ernesto Che Guevara.

Quiero agradecer:

A mis tutoras por la ayuda y los consejos que me han brindado.

A los demás profesores del proyecto que también me brindaron su ayuda.

Al tribunal y el oponente por los consejos y recomendaciones.

A mis amigos del aula, del edificio, todos los que han compartido siempre conmigo.

A todas las personas que de una forma u otra contribuyeron a la realización de este trabajo.

A la Revolución, Fidel y Raúl.

A todos muchas gracias.

Quiero dedicar este trabajo:

A mi mamá, mi papá, mi hermana, mi cuñado que ha sido como un hermano.

Gracias por su apoyo y confianza.

Resumen

En el mundo una gran cantidad de empresas diseñan estrategias centradas en servicios para sus clientes, para ello utilizan sistemas que le permiten gestionar las relaciones con sus clientes. Los sistemas de Administración de Relaciones con el Cliente (CRM por sus siglas en inglés) son soluciones tecnológicas que permiten fortalecer la comunicación entre la empresa y sus clientes para lograr la satisfacción de los mismos y garantizar su fidelidad con la empresa.

Actualmente la situación de los sistemas CRM en Cuba tiene un desarrollo muy pobre, es poco el conocimiento que se tiene sobre su utilización y en muchos casos los que son utilizados no integran funcionalidades específicas que requieren las empresas cubanas. El presente trabajo tiene como objetivo realizar la implementación de los procesos Atención al cliente y Soporte pertenecientes al módulo Servicios del sistema de Administración de Relaciones con el Cliente; desarrollado por un equipo de trabajo del departamento SOLEM en aras de darle cumplimiento a dichas particularidades.

La informatización de estos procesos permitirá gestionar con mayor agilidad los servicios solicitados por los clientes buscando mejorar el tiempo de respuesta de los servicios prestados por las empresas cubanas. Este sistema fue probado utilizando pruebas de caja blanca y caja negra; después de varias iteraciones se logra generar la primera propuesta del sistema que da cumplimientos a los requisitos planteados en el análisis y diseño de la aplicación.

Palabras claves: Administración de Relaciones con el Cliente, Atención al cliente, Soporte, implementación.

Tabla de Contenido

Índice de imágenes

Índice de tablas

Introducción

En las últimas décadas las Tecnologías de la Información y las Comunicaciones (TIC) han tenido un avance significativo, como consecuencia de esto se han creado nuevos canales de información accesibles tanto a las personas como a las organizaciones. Las nuevas tecnologías permiten a una empresa competir eficientemente y obtener la información disponible en el lugar y momento en el que se necesite por lo que estas se han convertido en arma esencial para ofrecer productos y servicios con mayor calidad. (1)

En el entorno actual las empresas no son viables sin una apropiada atención al cliente y sin brindar un soporte adecuado a los servicios que brinda, para lograr estar presente en un mercado cada vez más competitivo. Las empresas tienen que apostar por la calidad de los servicios, debido a que es muy importante la forma en que el cliente percibe la calidad y los medios que existen para mantenerlo satisfecho.

Atraer nuevos clientes sigue siendo una tarea de indiscutible importancia, sin embargo las empresas deben centrarse también en conservar los clientes que ya tienen y forjar relaciones rentables y duraderas con ellos. Actualmente el cliente no tiene que conformarse con lo que le ofrecen, si no se le da lo que necesita aparecerá otro suministrador que sí lo haga. Hoy existen un gran número de opciones y el cliente optará por el suministrador que mejor atienda sus necesidades y sobre todo lo entienda, conozca y cuide mejor sus intereses. (2)

Hoy en día existen nuevas formas de hacer negocio, entre ellas se encuentran los sistemas de Administración de Relaciones con los Clientes, que buscan fidelizar a los clientes conociendo su historia y preferencias. Un CRM consiste en construir relaciones duraderas mediante la comprensión de las necesidades y preferencias individuales y de este modo añadir valor a la empresa y al cliente, es conseguir que los clientes sean fieles; eso supone conocerlos, saber quiénes son, cuáles son sus gustos, sus preferencias así como ofrecerles un servicio de venta y post-venta óptimo. (2)

En el país existen organizaciones que utilizan sistemas informáticos que les permiten la gestión de los servicios a sus clientes. El conocimiento que existe sobre el trabajo con herramientas CRM aún es pobre y en muchos casos este proceso no se encuentra informatizado. El proceso de quejas y reclamaciones es

engorroso ya que no se cuenta con los medios necesarios para gestionar la recepción, tramitación y solución de las mismas, provocando conflictos a la hora de darle solución, teniendo como consecuencia que los clientes queden insatisfechos. En el caso del proceso de Soporte no son gestionadas las actividades necesarias para reparar, devolver o brindar asistencia técnica a los productos destinados a los clientes, por lo que la realización de estos trámites se pueden demorar largos periodos de tiempo, no garantizando la continuidad del servicio y la retención del cliente.

A partir de los principales lineamientos del nuevo modelo económico aprobado en el VI Congreso del Partido, Cuba se propone lograr un sistema empresarial constituido por empresas eficientes, bien organizadas y eficaces. Para contribuir con la implementación de estos lineamientos la Universidad de las Ciencias Informáticas (UCI) en colaboración con otras entidades tiene la tarea de la informatización de la sociedad cubana.

Dentro de la UCI se encuentra el Centro para la Informatización de la Gestión de Entidades (CEIGE) encargado de brindar soluciones empresariales, este propone la utilización de una herramienta que permita gestionar la forma en que se brindan los servicios, de forma tal que se logre mejorar la atención al cliente y así alcanzar mayor satisfacción en estos; la herramienta propuesta OpenERP es un sistema ERP integrado de código abierto actualmente producido por OpenERP S.A que permite la modificación directa del programa mientras que se respeten los términos de la licencia.

El OpenERP contiene un grupo de funcionalidades que gestionan los sistemas CRM relacionadas con los procesos Atención al cliente y Soporte, las cuales no cubren algunas requeridas por las empresas cubanas, por lo que el centro CEIGE se ha dado la tarea de extender las mismas para que este sistema se adapte a las particularidades del país. La implementación de estas nuevas funcionalidades partirá de un análisis y diseño previamente desarrollado por personal que radica en el centro CEIGE pertenecientes al Proyecto CRM.

Teniendo en cuenta lo expuesto anteriormente se tiene como **problema a resolver**: ¿Cómo contribuir a la gestión de los procesos Atención al cliente y Soporte de un Sistema de Administración de Relaciones con el Cliente en las empresas cubanas? En consecuencia se tiene como **objeto de estudio** los Sistemas de Administración de Relaciones con el Cliente.

Para ello se traza como **objetivo general**: Realizar la implementación de los procesos Atención al cliente y Soporte del Sistema de Administración de Relaciones con el Cliente. Se define como **campo de acción**: Sistemas de Administración de Relaciones con el Cliente que gestionen los procesos Atención al cliente y Soporte.

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

1. Realizar un estudio del estado del arte para la elaboración del marco teórico alrededor de los Sistemas de Administración de Relaciones con el Cliente.
2. Valorar los artefactos obtenidos en las fases de análisis y diseño relacionados con los procesos Atención al cliente y Soporte.
3. Implementar los procesos siguiendo las técnicas de programación estudiadas en la plataforma de desarrollo seleccionada.
4. Validar los requisitos implementados mediante la ejecución de pruebas.

Para satisfacer estos objetivos planteados quedan definidas las siguientes **tareas de la investigación**:

1. Definición del marco conceptual de la investigación.
2. Revisión de los procesos Atención al cliente y Soporte.
3. Análisis de los requisitos funcionales relacionados con los procesos Atención al cliente y Soporte.
4. Revisión de los patrones de diseño empleados.
5. Análisis de las herramientas y tecnologías a utilizar para el desarrollo de la propuesta de solución.
6. Realización del diagrama de componentes según los cambios ocurridos en el proyecto.
7. Obtención del código fuente de las funciones definidas.
8. Descripción de las principales clases implementadas.
9. Realización de pruebas de caja blanca sobre el código desarrollado.
10. Realización de pruebas de caja negra sobre las funcionalidades.

Además se plantea como **idea a defender**: Si se realiza la implementación de los procesos Atención al cliente y Soporte del Sistema de Administración de Relaciones con el Cliente entonces se podrá contribuir a la gestión de dichos procesos en las empresas cubanas.

La investigación para desarrollar el trabajo se sustenta en los siguientes **Métodos Científicos:**

Histórico-lógico: Permitió realizar un análisis documental de la información existente de los Sistemas de Administración de Relaciones con el Cliente siendo utilizado durante el proceso de revisión bibliográfica.

Analítico-Sintético: Permitió estudiar los mecanismos que se utilizan en los diferentes Sistemas de Administración de Relaciones con el Cliente haciendo uso de la documentación derivada de los flujos de análisis y diseño. Además posibilitó el análisis de documentos, materiales y temas relacionados con las mejores prácticas en el desarrollo de los sistemas CRM.

Modelación: Se empleó para crear abstracciones con vistas mediante las cuales se pueda explicar la realidad. Se utiliza en la modelación del diagrama de componentes, los diagramas de secuencia, diagramas de clases del diseño, modelo de datos y modelo conceptual.

La presente investigación está conformada por 3 capítulos cuya descripción es expuesta brevemente a continuación:

Capítulo 1: **"Fundamentos teóricos de la investigación"**. Se describen los conceptos relacionados con el dominio del problema. Se muestra el estado del arte de los distintos sistemas de Administración de Relaciones con el Cliente tanto en Cuba como en el mundo, de los cuales se analizan sus ventajas y desventajas. Se estudian las tecnologías, metodologías y herramientas definidas por el proyecto para dar solución al problema.

Capítulo 2: **"Implementación de los procesos Atención al Cliente y Soporte"**. En este capítulo se realiza una valoración de los artefactos del análisis y diseño propuesto por los analistas, como son los estándares de código a utilizar, el modelo de la Base de Datos, los diagramas de clases del diseño y la descripción de la propuesta inicial del sistema.

Capítulo 3: **"Validación de la solución"**. En este capítulo se realiza la validación de los resultados obtenidos a través de pruebas de caja blanca y caja negra con objetivo de garantizar la calidad de la solución propuesta.

Capítulo 1: “Fundamentos teóricos de la investigación”

Introducción

Realizar un producto informático que cumpla con lo esperado por el cliente necesita de un estudio profundo no solo de lo que el cliente desea informatizar, sino también de cuál es el mejor camino y cuáles son las herramientas necesarias para construir un software con la calidad requerida.

En este capítulo se hace un estudio de los sistemas de Administración de Relaciones con el Cliente, que guardan relación con el campo de acción de la investigación. Se aborda también sobre la metodología además de las tecnologías y herramientas utilizadas para dar cumplimento al objetivo trazado.

1.1 Marco conceptual

Atención al cliente: Se designa a la atención que prestan las empresas de servicios o que comercializan productos; en caso que sus clientes necesiten manifestar reclamos, sugerencias, plantear inquietudes sobre el producto o servicio en cuestión, así como solicitar información adicional. (3)

Soporte: El soporte es el servicio que brindan las empresas a los clientes para que puedan hacer uso de los productos o servicios que les son ofertados. La finalidad del soporte es ayudar a los usuarios a solucionar ciertos inconvenientes con los productos o servicios adquiridos. (4)

Administración de Relaciones con el Cliente: La definición de CRM engloba dos conceptos, hace referencia tanto a la estrategia de negocio focalizada hacia el cliente, como a las aplicaciones informáticas necesarias para procesar, analizar y exponer la información resultante, así como medir y retroalimentar la estrategia de negocio desarrollada.

Definiciones de CRM como estrategia de negocio:

El CRM consiste en una estrategia de la organización en 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 por los clientes desde los diferentes canales o medios de comunicación. (5)

Definiciones de CRM como aplicación informática:

Se refiere a aquellas aplicaciones que las empresas pueden utilizar para administrar todos los aspectos de sus encuentros con los clientes. Un sistema CRM puede incluir todo, desde tecnología para la recolección de datos en las llamadas telefónicas del área de ventas, hasta sitos web de autoservicio donde los clientes pueden aprender acerca de los productos y de su compra, o el análisis de los clientes y los sistemas de administración de campaña. El CRM es un concepto genérico en el que se denomina a las diversas soluciones de hardware y software que se estén ofreciendo hoy en el mercado y, se centra en lo que estas empresas llaman el "front office" que integra a las áreas de ventas, marketing, publicidad, Internet, canales, entre otras. (5)

1.2 Sistema informático de Administración de Relaciones con el Cliente OpenERP

Como se hizo mención anteriormente los arquitectos del proyecto en la fase análisis y diseño definieron que se utilizaría como base para la implementación la solución OpenERP debido a las siguientes características:

Libertad: OpenERP como producto no pertenece a ninguno de sus distribuidores, tiene libertad para elegir al proveedor que más le convenga según sus necesidades.

Código abierto: Al ser software libre, se podrá disponer del código para realizar cualquier mejora sobre los módulos ya existentes, o crear uno nuevo adaptado a sus necesidades.

Conectividad con otros productos: Visualización de informes en Adobe PDF, importación/exportación con Microsoft Office u OpenOffice, Google Maps, Mozilla Thunderbird, Magento, Joomla, y otros muchos, con la posibilidad de conexión con casi cualquier tecnología utilizando Jripple.

Flexibilidad: OpenERP dispone de más de 400 módulos, muchos de ellos específicos de determinados sectores. Se puede comenzar utilizando solamente el módulo de recursos humanos o la contabilidad, e ir integrando más módulos posteriormente.

Licencia: OpenERP es un producto que no tiene coste de licencias. No hay que pagar dinero por usarlo en más puestos de trabajo o renovar las costosas licencias anualmente.

Multiplataforma: Actualmente tiene clientes de escritorio funcionales para GNU/Linux, Mac OS X y Windows, e incluso interfaz web para poder trabajar en otros sistemas como tablets, pdas, smartphones. (6)

Complementando estas especificaciones se hace un estudio con mayor nivel de detalle para conocer qué elementos pueden resultar de interés para el trabajo a desarrollar. A continuación se especifica la síntesis de los elementos más importantes del estudio realizado:

OpenERP es un completo sistema de gestión de empresas/organizaciones (ERP) de licencia libre que cubre las necesidades de las áreas de contabilidad, ventas, compras, almacén, inventario, proyectos, CRM, recursos humanos, tiendas virtuales, entre otras. Además incorpora funcionalidades de gestión de documentos, conectores con otras aplicaciones, permite trabajar remotamente mediante una interfaz web o aplicación de escritorio multiplataforma (Windows, Linux y Mac) e incluye un entorno modular de programación/adaptación rápida de aplicaciones OpenObject. Se basa en tecnología Python/XML trabajando sobre una base de datos en PostgreSQL.

Entre sus funcionalidades se encuentra la Gestión de casos: el concepto de caso, permite gestionar diferentes comunicaciones de los clientes o proveedores que requieran una atención posterior por parte del personal de la empresa. Algunos de esos casos pueden ser: Reclamaciones de pedidos, problemas de calidad, gestión de llamadas, tickets de soporte y ofertas de trabajo. OpenERP asegura el correcto tratamiento de los casos por los usuarios del sistema, clientes y proveedores. Puede automáticamente reasignar un caso, enviar alarmas por e-mail y enlazar con otros documentos y procesos de OpenERP. Todas las operaciones son archivadas y existe una pasarela de e-mail donde puede actualizar un caso automáticamente desde los e-mails enviados y recibidos. Un sistema de reglas permite definir acciones que pueden automáticamente mejorar su proceso de calidad, asegurando que un caso abierto nunca se pierda. (7)

OpenERP permite registrar y realizar un seguimiento de las reclamaciones de los clientes. Una reclamación puede definirse en varios tipos, por el nombre del cliente, estado y nivel de prioridad. La

reclamación también puede ser una acción preventiva o una reparación. Puede estar vinculada a una referencia, como un pedido de cliente, o un número de lote del producto. Puede enviar correos electrónicos con archivos adjuntos directamente desde OpenERP y obtener el histórico del tratamiento de la reclamación mediante los correos electrónicos enviados, tipo de intervenciones realizadas, entre otros. (8)

Además del estudio realizado sobre OpenERP a continuación se estudiarán otros sistemas que de una forma u otra gestionan entre sus funcionalidades las relaciones con los clientes, en aras de obtener otros conocimientos que los mismos puedan aportar.

1.3 Sistemas informáticos de Administración de Relaciones con el Cliente utilizados en el mundo

Terrasoft CRM es una aplicación de escritorio desarrollado en Delphi**,** es compatible con los Sistemas Operativos Windows ME, Windows NT 4,0, Windows 2000, Windows XP y Windows 2003, para su operatividad se necesita un Servidor de base de datos MS SQL Server 2000. En lo relacionado al área de Asistencia Técnica; permite controlar las preferencias de productos y servicios de los clientes para pronosticar posibles interacciones en relación al soporte en el servicio o deficiencias en los productos.

Este software permite mejorar y automatizar el servicio de atención al cliente, administrando la documentación de cada uno, propuestas comerciales, acuerdos, especificaciones, encargos de producción y otros documentos relacionados con el producto o servicio, los cuales serán sistematizados según los proyectos y contratantes. (9)

SugarCRM es un software CRM completamente libre y abierto, funciona enteramente en el servidor web con soporte para PHP, tiene una base sólida y se pueden ampliar sus módulos. SugarCRM, al ser software libre, permite el acceso al código fuente de la aplicación, para poder modificarlo y adaptarlo a las necesidades de las empresas. De la misma manera, en Internet existen muchos módulos que pueden ampliar la funcionalidad de la aplicación, cartas, cuadernos, presupuesto, calendario, correo electrónico, y varios aspectos estéticos. Está basado en Linux-Apache-MySQL-PHP (LAMP), desarrollado por la empresa SugarCRM con sede en Cupertino, California. Tiene tres versiones, una libre y dos versiones con

componentes no libres. Está diseñada para ayudar a gestionar las ventas, oportunidades, contactos de negocios. (10)

La funcionalidad de SugarCRM para el área de atención al cliente centraliza toda la actividad de servicio al cliente, cada petición e incidencia. El carácter multicanal le permitirá recoger cualquier solicitud de los clientes, diagnosticar el problema y ofrecer una rápida solución, así como compartir el conocimiento con el resto del equipo de soporte, lo que les ayudará a resolver problemas cada vez con una mayor agilidad. (11)

1.4 Antecedentes de software que gestionan Administración de Relaciones con el Cliente utilizados en Cuba

SAP CRM es la solución de servicio al cliente que le proporciona la mayor flexibilidad de la gestión de clientes. Las soluciones de software de gestión de los clientes de SAP, están diseñadas para satisfacer las necesidades de la nueva economía al mismo tiempo que le proporcionan el mayor nivel de control de dicha relación. El manejo eficiente de los procesos de servicio al cliente generarán el mayor nivel de valor en torno a sus procesos. Para esto, SAP ha desarrollado soluciones CRM centradas en la optimización de la realización de procesos sectoriales integrales, para dar soporte a los departamentos de atención al cliente, tanto en marketing, como en ventas y servicios. (12)

Da soporte a los procesos relacionados con el cliente de principio a fin, organiza todas las tareas relacionadas con el servicio al cliente más allá de las fronteras que imponen los departamentos, incorporando a la perfección actividades como el suministro, la facturación y la contabilidad de deudores. Proporciona conocimiento sobre los clientes a toda la empresa: reúne todas las fuentes relevantes de datos de los clientes en toda la empresa para acelerar y mejorar la toma de decisiones. La aplicación SAP CRM está diseñada para proveer historiales de CRM en cualquier momento. Proporciona valor de forma inmediata: permite hacer frente a las prioridades estratégicas en primer lugar, cumpliendo rápidamente objetivos empresariales, y ampliar la solución de servicio al cliente CRM gradualmente, proporcionando un retorno de la inversión tangible en cada paso. (13)

SAP CRM para la gestión de reclamaciones primeramente hace una recepción de la reclamación a través de los diferentes canales teléfono, correo electrónico, fax, web y carta, después hace una investigación, se

propone una solución interna que posteriormente pasa al proceso de aprobación de la solución y así comunica la solución al cliente para luego pasar a la realización de la solución acordada con el cliente ya sea abono, devolución, entre otros. (14)

AvilaQuid es una aplicación WEB destinada a la gestión de incidencias y quejas. Permite la gestión de incidencia desde su registro, clasificación, asignación de responsabilidades y procesamiento, que posibilitan brindar la conclusión y la respuesta adecuada, para posteriormente supervisar y aplicar las medidas necesarias. Esta aplicación no es un CRM, pero es lo más cercano que se ha producido en Cuba. Permite una atención rápida y efectiva a las incidencias de los clientes y usuarios de la empresa, mejora la velocidad en la respuesta, diligencia en el servicio, rapidez en la contestación a los clientes.

Mejora las relaciones con los clientes, conociéndolos mejor y permitiendo disminuir los costos en la consecución de nuevos prospectos, ayuda a consolidar la fidelidad de los ya existentes, lo cual, en ambos casos, significa mayores ventas y más rentabilidad para la empresa. Además se pueden determinar los principales problemas que tiene la empresa, porque los errores pueden estar en la prestación del servicio, pero pueden provenir también del diseño del mismo o del planteamiento operativo.

Contribuye a que la empresa se oriente al cliente, apoya a las empresas a detectar e identificar problemas en el servicio y poder de esta manera, plantear propuestas de mejora continua, brindando elementos importantes para la toma de decisiones. El análisis de las incidencias, son indicativas de la existencia de defectos percibidos en los servicios por parte de los usuarios de los mismos o la necesidad de nuevas prestaciones por lo que según su comportamiento se puede determinar los problemas y decidir en función de su mejoramiento. (15)

Vtiger CRM es una solución CRM de fuente abierta, distribuida bajo los términos de la licencia Mozilla Public License (MPL). Está construida sobre las tecnologías de Apache, PHP y MySQL, todas también con equipamiento lógico de fuente abierta. El equipo de desarrollo de Vtiger CRM tiene su base de operaciones en Chennai, India y es financiado por AdventNet. Proporciona sistemas de gestión de entrada, sistemas de gestión del conocimiento, portales de autoservicio de clientes, informes, estadísticas y mucho más apoyo para que el equipo de su organización de soporte para satisfacer las necesidades de los clientes más exigentes. (16)

Da seguimiento a todas las incidencias relacionadas con los clientes, gestiona los problemas asociados con las cuentas, contactos, productos, y otros módulos para tener una mejor visibilidad sobre las incidencias. Proporciona soluciones a las incidencias presentadas a través del portal del cliente, actualiza automáticamente el estado de entradas a través de e-mail. Crea estadísticas de asistencia al cliente para ayudar a los gerentes para planificar un mejor proceso de atención al cliente y permite crear campos personalizados según las necesidades de su organización. (17)

El estudio de estos sistemas permitió tener en cuenta un grupo de sus funcionalidades relacionadas con los procesos Atención al cliente y Soporte, que contribuyeron al mejor entendimiento de las funcionalidades a implementar, así como otros indicadores de interés para la investigación. En la Tabla 1 se muestra la comparación de los sistemas estudiados partiendo de estos indicadores:

Tabla 1: Tabla comparativa de los sistemas estudiados

Características/Sistemas	Terrasoft CRM	Open ERP	Sugar CRM	SAP CRM	AvilaQuid	Vtiger CRM
Incidencias.	Si	Si	Si	Si	Si	Si
Reparaciones y devoluciones.	Si	No	No	Si	No	No
Asistencia técnica.	Si	No	No	Si	No	No
Multiplataforma.	No	Si	Si	No	No	Si
Costo (Bajo).	No	Si	Si	No	Si	Si
Privativo.	Si	No	No	Si	No	No

1.5 Modelo de desarrollo

Para el ciclo de vida de los proyectos del CEIGE y posterior especificación se tienen en cuenta las fases y actividades por áreas de procesos que plantea el nivel dos de CMMI establecido en la UCI. En la figura que se muestra a continuación se puede apreciar el total de fases por las que pueden transitar los proyectos del centro, pudiendo realizar iteraciones a partir de la fase de Modelado del negocio. El conjunto de fases propuesta abarca el total de acciones que se realizan en las distintas líneas de desarrollo para la elaboración del servicio o producto final, sin embargo, se debe adaptar a las características particulares

del proyecto que puede que no realice determinada fase, así como la elaboración de determinados artefactos definidos (18).

Figura 1: Ciclo de vida de proyectos del CEIGE

Para la implementación de los procesos a desarrollar se utilizó el Modelo de desarrollo del centro CEIGE versión 1.1 dentro de la cual se propone que la Fase de Implementación se lleve a cabo de la siguiente forma.

Descripción general de la Fase Implementación

A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes de software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes. (18)

Tabla 2: Descripción de las actividades de la fase Implementación

Actividades	Descripción	Roles	Artefactos de salida
Analizar artefactos generados en la fase de diseño	Hacer un análisis de los diagramas de clases para conocer las interrelaciones y responsabilidades de las mismas. A partir de este estudio entonces el programador debe ser capaz de desarrollar cada una de las funcionalidades descritas. En esta actividad se determinan además las adecuaciones de ser necesarias a componentes que se vayan a reutilizar y/o nuevos desarrollos e integraciones.	Programador	Solicitud de cambio
Implementar funcionalidades en términos de componentes	Se deben generar los ficheros de código que respondan a las funcionalidades de sistema solicitadas mediante la lógica de negocio diseñada. Se realizan los desarrollos necesarios para su adecuación en los componentes que se deseen reutilizar. Se realiza la integración de los componentes y por tanto la implementación que también esto requiera. Todo el código es debidamente comentado y se especifica en cada fichero los requisitos a los que se le da cumplimiento para llevar control de la trazabilidad.	Programador	Ficheros de código. Algoritmos de Implementación
Probar componentes implementados su integración	Realizar pruebas a las implementaciones para garantizar su pleno funcionamiento operativo, para ello auxiliarse de las descripciones de requisitos y de los diagramas de secuencia. Esto incluye las pruebas individuales a los componentes y a las integraciones de estos. En caso de que se encuentren errores o inconsistencias en el código, en esta actividad se llevaría a cabo su refinamiento y corrección.	Programador	

1.6 Tecnologías y herramientas para el desarrollo

A continuación se describen las tecnologías y herramientas para el desarrollo de la aplicación, las cuales fueron definidas por los arquitectos del proyecto, por tanto el estudio será más preciso y enfocado en esa selección.

1.6.1 Lenguaje de modelado

El lenguaje que se utilizará para el modelado es el **Lenguaje Unificado de Modelado 2.1** (UML, por sus siglas en inglés), es un lenguaje de modelado visual para especificar, visualizar, construir y documentar artefactos de un sistema de software. UML, proporciona una forma estándar de representar los planos de un sistema. Comprende tanto elementos conceptuales, como los procesos de negocio y las funciones del sistema. Incluye además, elementos concretos como: las clases escritas de un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables. (19)

Las ventajas de este lenguaje de modelado son:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).Permite especificar todas las decisiones de análisis, diseño e implementación, construyendo así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo.
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- UML es independiente del proceso, aunque para utilizarlo óptimamente, se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. (20)

1.6.2 Lenguajes de programación

Python 2.7 es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis muy limpia lo cual favorece un código legible**.** Se trata de un lenguaje de programación multiparadigma ya que

soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemente tipado y multiplataforma. (21)

Características:

- Propósito general: se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.
- Multiplataforma: hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- Interpretado: quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí se realiza una compilación, pero esta se lleva a cabo de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.
- Interactivo: Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- Orientado a Objetos: la programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- Funciones y librerías: dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, entre otros. Además, existen muchas librerías que se pueden importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.
- Sintaxis clara: Python tiene una sintaxis muy visual, gracias a una notación identada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de

una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar (22).

La sintaxis de Python es sencilla y cercana al lenguaje natural. Python se enfoca en la legibilidad, coherencia, y la calidad del software en general. Al ofrecer un diseño legible, el proceso de mantenimiento se hace más fácil y rápido. El código en Python regularmente es de 1/3 a 1/5 el tamaño de un código equivalente escrito en C++ o Java. Esto quiere decir que el desarrollador tendrá menos que escribir, menos que depurar, y menos que mantener. Además, los programas en Python pueden ejecutarse inmediatamente, sin tener que recurrir a los largos pasos de compilación y enlazado. (23)

Extensible Markup Language 1.1 (XML) es un formato simple, texto muy flexible derivado de Estándar de Lenguaje de Marcado Generalizado (SGML, por sus siglas en inglés). Originalmente diseñado para afrontar los retos de la gran edición electrónica, XML también está desempeñando un papel cada vez más importante en el intercambio de una amplia variedad de datos en la Web y en otros lugares. (24)

Características:

- XML provee un conjunto de reglas, normas y convenciones para diseñar formatos de texto para datos estructurados que van desde las hojas de cálculo, o las libretas de direcciones de Internet, hasta parámetros de configuración, transacciones financieras o dibujos técnicos.
- Los programas que los generan, utilizan normalmente formatos binarios o de texto. XML permite resolver problemas comunes, como la falta de extensibilidad, carencias de soporte debido a características de internacionalización, o problemas asociados a plataformas específicas.
- El formato texto puede ser usado en cualquier plataforma, esto le da innumerables ventajas de portabilidad, depuración, independencia de plataforma, e incluso de edición, pero su sintaxis es más estricta que la de HTML, una marca olvidada o un valor de atributo sin comillas convierten el documento en inutilizable. No hay permisividad en la construcción de documentos, ya que esa es la única forma de protegerse contra problemas más graves.
- XML no requiere licencia es un estándar abierto independiente de la plataforma, y tiene un amplio soporte extendido a un sin número herramientas y desarrolladores. (25)

XML puede ofrecer más facilidades para la representación en los navegadores ya que elimina muchas de las complejidades de SGML en aras de una mayor flexibilidad del modelo, con lo que la escritura de programas para manejar XML es mucho más sencilla. La información será más accesible y reutilizable porque la flexibilidad de las etiquetas de XML puede utilizarse sin tener que amoldarse a las reglas específicas de un fabricante. Los archivos XML válidos son válidos también en SGML, por lo que también pueden utilizarse fuera de la Web en un entorno SGML. (26)

1.6.3 Herramienta para el modelado

La herramienta de ingeniería de software asistida por computadora (CASE, por sus siglas en inglés) denominada como **Visual Paradigm 8.0**, utiliza UML como lenguaje de modelado. Esta herramienta permite modelar el ciclo de vida completo del desarrollo de software, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML, ayuda a una más rápida construcción de aplicaciones de calidad, más óptimas y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, obtener código inverso, generar código desde diagramas y generar documentación. (27)

1.6.4 Framework de desarrollo

OpenObject 6.3 es una plataforma de Desarrollo Rápido de Aplicaciones (RAD, por sus siglas en inglés) de OpenERP, basado en el patrón Modelo Vista Controlador (MVC), utiliza PostgreSQL como base de datos y como lenguajes de programación Python y XML.

Características:

- El mapeo objeto-relacional (ORM, por sus siglas en inglés), proporciona una capa de abstracción sobre la base de datos, gestiona los derechos de acceso y evita tener que escribir código SQL.
- Acceso generalizado a través de servicios web XML-RPC. Todos los objetos de OpenERP se puede acceder a través de servicios web, ya sea leer, escribir, crear y borrar.
- El motor de flujo de trabajo (workflow) en OpenERP es sencillo, pero suficiente para las necesidades de las pequeñas y medianas empresas. El motor de flujo de trabajo es específico de OpenERP.

- OpenObject tiene una gestión integrada de módulos y puede trabajar por herencia. Los módulos tienen dependencias entre ellos, y, al instalar un nuevo módulo, OpenERP maneja la instalación de las dependencias necesarias. Pero lo más importante es que se puede cambiar el comportamiento de los módulos nativos de OpenERP o añadir funcionalidad de módulos independientes que trabajan a través de herencia. (28)

1.6.5 IDE de desarrollo

La plataforma **Eclipse 5.1** consiste en un Entorno de Desarrollo Integrado (IDE, Integrated Development Environment) abierto y extensible. Un IDE es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Como elementos básicos, un IDE cuenta con en un editor de código, un compilador/intérprete y un depurador. Eclipse sirve como IDE Java y cuenta con numerosas herramientas de desarrollo de software. También da soporte a otros lenguajes de programación, como son C/C++, Cobol, Fortran, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (plugins) para extender la funcionalidad. (29)

PyDev es un entorno de programación Python para Eclipse**,** que puede ser utilizado en Python**,** Jython y IronPython desarrollo. (30)

Características:

- Código de terminación: Establece terminaciones sensibles al contexto**.**
- El resaltado de sintaxis: PyDev puede editar los ajustes de la ficha y los colores utilizados en la apariencia PyDev, como el código de colores, los decoradores, los números, las cadenas, los comentarios, entre otros.
- Código análisis: El análisis de código proporciona error en la búsqueda de programas Python. Se encuentra errores comunes, tales como tokens definidos, firmas duplicadas y advierte acerca de variables no utilizadas o las importaciones no utilizados.
- Ir a la definición: El ir a la definición de acciones permite llegar a una definición dada.
- Navegador Tokens: Le permite ver rápidamente todas las definiciones disponibles, que incluye:
  - ✓ Las definiciones de clases.
  - ✓ Definiciones de método.

- ✓ Las variables globales.
- ✓ Clase y la instancia de atributos.

1.6.6 Control de versiones

El sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros y mantiene un historial de cambios realizados. Es útil para guardar cualquier documento que se cambie con frecuencia, como el código fuente de un programa. Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local lo cual permite compartir los cambios sobre un mismo conjunto de ficheros. Además, el repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad. (31)

Es necesario el uso de este sistema de control de versiones dado las ventajas que brinda el mismo:

1. Actualización de ficheros modificados.
2. Copias de seguridad centralizadas.
3. Historial de cambios.
4. Brinda acceso remoto.
5. Provee seguridad al sistema.

El **Subversion 1.4.5** posee un excelente dominio de las versiones y un mayor control del trabajo que se está realizando sobre cada uno de los elementos de configuración brindando soporte a todas las operaciones que se realizan sobre los mismos haciendo el entorno de desarrollo del proyecto libre a errores fatales. Su característica de software libre hace que su selección sea una opción de prioridad para la investigación.

1.6.7 Gestor de base de datos

PostgreSQL 9.1 es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y su código fuente está disponible libremente. 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. (32)

Características:

- Múltiples lenguajes de procedimientos, ya que los disparadores y otros procedimientos pueden ser escritos en varios lenguajes de procedimientos. El código del lado del servidor es comúnmente escrito en PL / PostgreSQL, un lenguaje de procedimiento similar al de Oracle PL / SQL.
- Múltiples-cliente de Interfaz de Programación de Aplicaciones (API, por sus siglas en inglés), debido a que soporta el desarrollo de aplicaciones cliente en varios lenguajes, interfaz para PostgreSQL desde C, C + +, ODBC, Perl, PHP, Tcl / Tk, y Python.
- Diferentes tipos de datos como integer, string, numeric, boolean, char, varchar, date, int erval, y timestamp, tipos geométrica, tipo de datos booleanos y tipos de datos diseñados específicamente para hacer frente a las direcciones de red.
- La extensibilidad es una de las características más importantes de PostgreSQL ya que puede ser ampliado, se pueden añadir nuevos tipos de datos, nuevas funciones y operadores, e incluso nuevos lenguajes de procedimiento y de cliente.
- Posee integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. (32)

1.6.8 Servidor web

OpenERP proporciona un servidor de aplicaciones en el que se pueden construir aplicaciones de negocio específicas. También es un marco de desarrollo completo, que ofrece una amplia gama de funciones para escribir esas aplicaciones. El servidor OpenERP también cuenta con una capa específica diseñada para comunicarse con el navegador web basado en el cliente denominado Servidor-Web, esta capa web ofrece una interfaz para comunicarse con los navegadores estándares.

Desde la perspectiva del desarrollador, el servidor actúa como una biblioteca, al tiempo que oculta los detalles de bajo nivel, el servidor de OpenERP es sencillo de instalar, configurar y ejecutar las aplicaciones; también otros servicios, como los modelos de datos extensibles y la vista, el motor de flujo de trabajo o el motor de informes.

El Mapeo Objeto Relacional (ORM por sus siglas en inglés) es una de las características más destacadas del servidor OpenERP. Proporciona funcionalidades adicionales y esenciales en la cima de servidor

PostgreSQL. Los modelos de datos se describen en Python y OpenERP crea las tablas de base de datos subyacente que utilizan este ORM. (33)

1.6.9 Navegador web

Firefox 16.0 posee entre las características más destacadas de esta versión encontramos soporte inicial para aplicaciones web, además de mejoras en la sensibilidad de JavaScript gracias a la mejora de su recolector de basura. Esta nueva versión incluye una barra de herramientas para desarrolladores, que se sitúa en la parte inferior del navegador y ofrece un rápido acceso a las Developer Tools de Firefox. Una línea de comando único con la que controlar rápidamente las herramientas para desarrolladores sin quitar las manos del teclado. Se ha mejorado el soporte del estándar CSS3 (Animaciones, Transiciones, Transformaciones y Gradientes CSS3) convirtiendo esta nueva versión en algo más que interesante para desarrolladores web. (34)

Conclusiones parciales

En este capítulo se estudió el sistema OpenERP seleccionado por el equipo de arquitectos del proyecto como base para el de desarrollo, lo cual permitió obtener conocimientos sobre su funcionamiento; elemento necesario para la continuidad del trabajo. Se estudiaron también un conjunto de sistemas que de una forma u otra gestionan diferentes servicios prestados a sus clientes; de estos se determinaron las principales características relacionadas con los procesos a implementar contribuyendo de esta forma al mejor entendimiento de los requisitos que debe cumplir el sistema.

Se determinó guiarse por el Modelo de desarrollo del centro CEIGE versión 1.1 para la realización de las actividades necesarias dentro de la fase de implementación, así como los artefactos que deben generarse de dichas actividades. Además se introduce la información necesaria para el estudio de las tecnologías y herramientas a utilizar, quedando de forma clara los elementos esenciales para el uso de la infraestructura definida por el equipo de proyecto. Tecnologías que posibilitarán transitar por las fases de implementación y pruebas con el fin de dar solución al problema a resolver y cumplir de esta forma con el objetivo trazado.

Capítulo 2: “Implementación de los procesos Atención al cliente y Soporte”

Introducción

En este capítulo se expone una valoración del análisis y diseño realizado por los analistas, el cual supone el punto de partida a la hora de obtener un desarrollo apropiado de la aplicación. Se realiza una descripción de los elementos más importantes en la implementación, además se muestra cómo queda concebida la propuesta inicial del sistema según las posibilidades que proporciona el marco de trabajo utilizado.

2.1 Valoración del análisis y diseño

Del análisis y diseño propuesto por los analistas se pudieron extraer las clases fundamentales que deben ser definidas para que el sistema funcione satisfactoriamente. Permitió además una mejor comprensión de los aspectos relacionados con los requisitos funcionales y los no funcionales. La obtención del análisis y diseño, resultó de gran importancia, pues crea una entrada apropiada para que las actividades de implementación fluyan con la calidad requerida. A continuación, en la Tabla 3, se exponen un grupo de cambios que fueron necesarios realizar en el análisis y diseño para lograr el resultado esperado teniendo en cuenta las nuevas estrategias trazadas en el proyecto. Los artefactos implicados se pueden encontrar en el Expediente de proyecto, los mismos fueron aceptados por el cliente para los cuales se emitió un Acta de aceptación que se puede consultar en el Anexo 1.

Tabla 3: Listado de cambios realizados al análisis y diseño

Artefacto	Cambios
CIG-CRM-N-i2620	Buscar asistencia técnica: se modifica para obtener la funcionalidad de Búsqueda avanzada de asistencia técnica. Se cambian atributos visibles en la interfaz. Listar asistencias técnicas: se cambian atributos visibles en la interfaz. Adicionar asistencia técnica: se cambian datos de entrada para crear una nueva asistencia técnica. Modificar asistencia técnica: se cambian datos requeridos para modificar una

	asistencia técnica. Consultar asistencia técnica: se cambian atributos visibles en la interfaz.
IG-CRM-N-i2618	Buscar devolución: se modifica para obtener la funcionalidad de Búsqueda avanzada de devoluciones. Se cambian atributos visibles en la interfaz. Listar devoluciones: se cambian atributos visibles en la interfaz. Adicionar devolución: se cambian datos de entrada para crear una nueva devolución. Modificar devolución: se cambian datos requeridos para modificar una devolución. Consultar devolución: Se agrega este requisito.
CIG-CRM-N-i2617	Buscar incidencia: se modifica para obtener la funcionalidad de Búsqueda avanzada de asistencia técnica. Se cambian atributos visibles en la interfaz. Listar incidencias: se cambian atributos visibles en la interfaz. Adicionar incidencia: se cambian datos de entrada para crear una nueva incidencia. Modificar incidencia: se cambian datos requeridos para modificar una incidencia. Consultar incidencia: se cambian atributos visibles en la interfaz.
CIG-CRM-N-i2619	Buscar reparación: se modifica para obtener la funcionalidad de Búsqueda avanzada de reparaciones. Se cambian atributos visibles en la interfaz. Listar reparación: se cambian atributos visibles en la interfaz. Adicionar reparación: se cambian datos de entrada para crear una nueva reparación. Modificar reparación: se cambian datos requeridos para modificar una reparación Consultar: Se agrega este requisito.
CIG-CRM-N-i2621	Se eliminó el campo visible en la interfaz sobrante “Descripción”.
CIG-CRM-N-i2616	Se eliminaron campos de entrada sobrantes.
CIG-CRM-N-i2622	Se agrega la agrupación de requisitos Gestionar tipo de reparación.
CIG-CRM-N-i2623	Se agrega la agrupación de requisitos Gestionar Motivos de la devolución.
CIG-CRM-N-i2624	Se agrega la agrupación de requisitos Gestionar Motivos de asistencia técnica.
CIG-CRM-N-i2625	Se agrega la agrupación de requisitos Gestionar Cargo de empleado.
CIG-CRM-N-i2626	Se agrega la agrupación de requisitos Gestionar Departamento de empleado.

Diseño de casos de prueba	Fueron creados porque no existían en el Expediente de proyecto.
Diagrama de componentes	Se modifica teniendo en cuenta que no se integrará con el sistema Cedrux.
Diagrama de clases	Se modifica teniendo en cuenta que por la no integración con el sistema Cedrux deben aparecer otras clases para tomar los datos necesarios como la clase Empleado; así como el cambio de nombres de atributos, inclusión de algunos nuevos y eliminación de otros. Se redefine uno para cada agrupación de requisito.
Diagramas de Secuencia	Fueron creados porque no existían en el Expediente de proyecto. Se obtiene un Diagrama de Secuencia por cada requisito descrito.
Modelo conceptual	Se modifica teniendo en cuenta que aparecen nuevos conceptos.
Modelo de datos	Se modifica teniendo en cuenta que aparecen nuevas tablas.

2.1.1 Requisitos funcionales

Los requisitos funcionales se definen como las capacidades o condiciones que el sistema debe cumplir. En la Tabla 4 se muestran los requisitos definidos en la etapa de análisis y diseño agrupados por categorías, cada una de estas agrupaciones cuenta con un adicionar, eliminar, modificar, buscar, listar y consultar.

Tabla 4: Listado de requisitos funcionales

Requisitos funcionales
Gestionar incidencia.
Gestionar tipo incidencia.
Gestionar asistencia técnica.
Gestionar tipo asistencia técnica.
Gestionar reparación.
Gestionar devolución.

Luego de la valoración del análisis y diseño fue necesario hacer algunos cambios en los requisitos y agregar otros nuevos para ello se aplica una de las técnicas existentes para la captura de requisitos:

Tormenta de ideas: Se realizaron diferentes encuentros para debatir sobre los cambios ocurridos en el proyecto con objetivo de que se obtuvieran ideas para enfrentarlos y lograr cumplir las especificaciones definidas. Como resultado se identificaron los nuevos requisitos funcionales que a continuación se muestran en la Tabla 5.

Tabla 5: Listado de nuevos requisitos funcionales

Nuevos Requisitos funcionales
Gestionar motivos de la asistencia técnica.
Gestionar tipo reparación.
Gestionar motivos de la devolución.
Gestionar empleado.
Gestionar cargo.
Gestionar departamento.

Para cada nuevo requisito identificado se realizó su respectiva especificación. En la Tabla 6 se muestra la descripción del requisito Adicionar Empleado. Las demás especificaciones se encuentran en el Expediente de proyecto.

Tabla 6: Listado de nuevos requisitos funcionales

Precondiciones	Debe existir una estructura creada en la empresa de departamentos y de cargos para asignarles a los empleados.
Flujo de eventos	
Flujo básico Adicionar empleado de forma correcta	
1	Se introducen los datos del empleado:
2	El sistema valida (ver validación 1) los datos introducidos.

3	Si los datos son correctos el sistema los registra.
5	El sistema confirma el registro de los datos.
5	Concluye el requisito.
Pos-condiciones	
1	Se registró en el sistema un nuevo empleado.
Flujos alternativos	
Flujo alternativo 3.a Información errónea	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.
Pos-condiciones	
1	No se registran los datos.
Flujo alternativo 3.b Información incompleta	
1 El sistema señala los datos vacíos y permite corregirlos.	
2 El usuario corrige los datos.	
3 Volver al paso 2 del flujo básico.	
Pos-condiciones	
1 No se registran los datos.	
Flujo alternativo *.a El usuario cancela la acción	
1 Concluye el requisito.	
Pos-condiciones	
1 No se registran los datos.	
Validaciones	
1	Se validan los datos según lo establecido en CIG-CRM-N-i2201 Modelo conceptual.
Conceptos	**Empleado** **Visibles en la interfaz:** Nombre, Departamento, Cargo, Dirección, Teléfono, Email, CI

	Utilizados internamente: N/A
Requisitos especiales	N/A
Asuntos pendientes	N/A

Prototipo elemental de interfaz gráfica de usuario

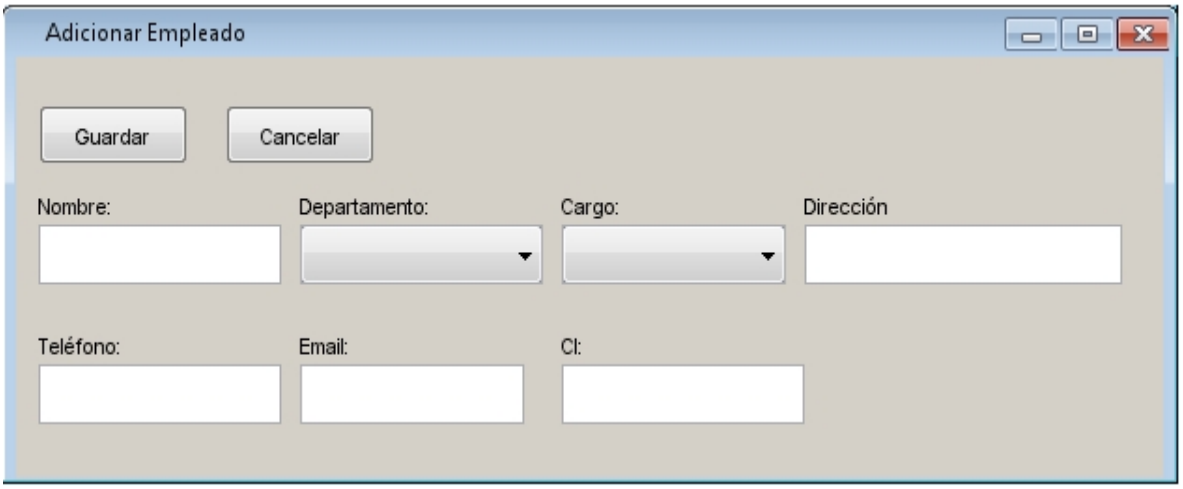

Figura 2: Prototipo para adicionar empleado

Validación de los nuevos requisitos

Prototipos: A partir de las especificaciones de requisitos fueron conformados prototipos de interfaz de usuario. El empleo de esta técnica ofreció como resultado que el equipo de proyecto tuviera una idea de la estructura de la interfaz de usuario y se favoreciera la comunicación entre los mismos, ya que tenía una visión inicial del módulo. Con lo cual se validó que los requisitos estaban en concordancia con las necesidades requeridas.

2.1.2 Diagrama de clases del diseño

Como resultado de la valoración del análisis y diseño se hizo necesario hacer modificaciones en los diagramas de clases del diseño existentes y agregar otros según las nuevas necesidades del proyecto. A

continuación se presenta en la Figura 3 el diagrama de clases del diseño para la agrupación del requisito Gestionar empleado, de forma tal que se facilite la comprensión de las relaciones entre los distintos componentes. Para ver el resto de los diagramas consultar el Expediente de proyecto.

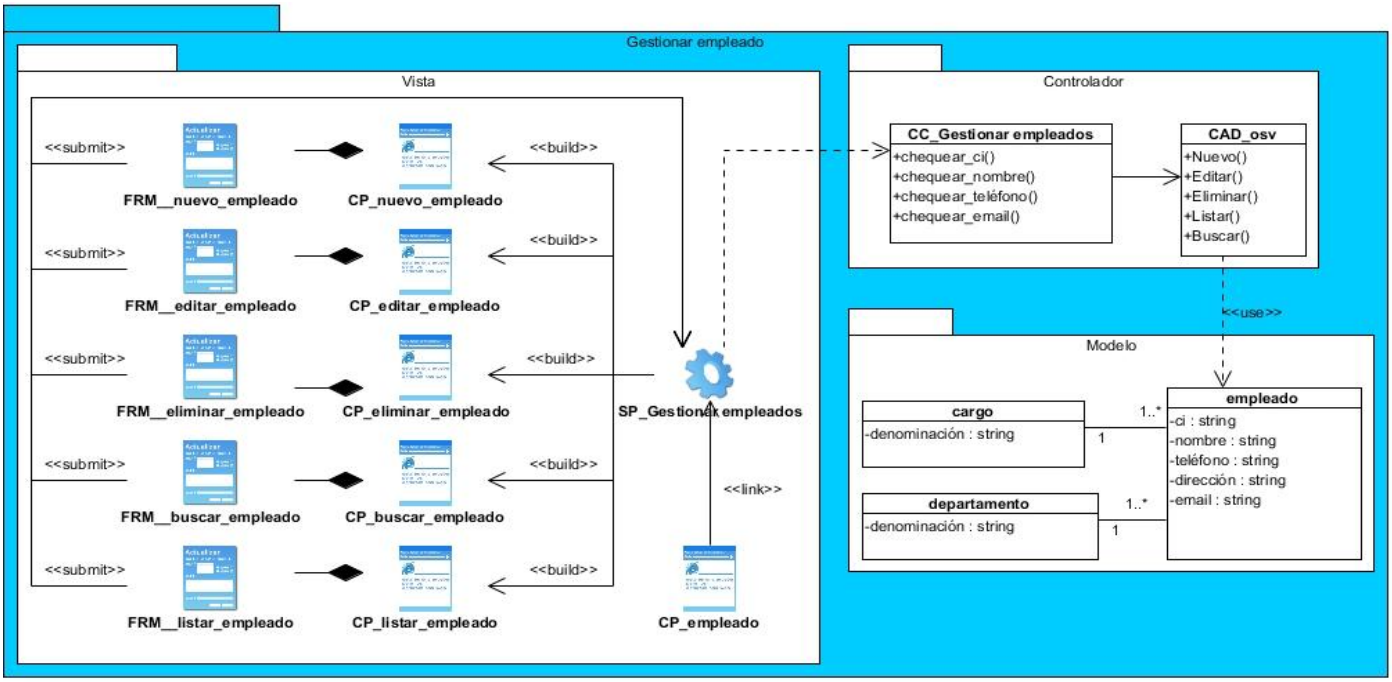

Figura 3: Diagrama de clases del diseño agrupación del requisito Gestionar empleado

Para la elaboración de los diagramas de clases del diseño y modificación de los ya existentes se empleó el patrón arquitectónico Modelo Vista Controlador (MVC), este divide el sistema en tres capas separando la lógica de negocio, los datos y la interfaz de usuario, este patrón permite la reutilización de código y la separación de conceptos, de esta forma busca facilitar el desarrollo de aplicaciones y su posterior mantenimiento; en OpenERP el modelo está compuesto por una base de datos, pero a nivel de desarrollo, solo se trabaja a través del ORM ya que en OpenERP todas las clases derivan de la clase osv.osv y esta implementa el ORM, la parte del controlador es completamente código python y la vista es definida en XML.

Además se utilizaron los patrones generales de software para asignación de responsabilidades (GRASP). Con la utilización de estos patrones se distribuyen las responsabilidades entre las clases de forma tal que

no existan un exceso de relaciones y que no se sobrecargue de métodos una clase en específico, además de mantener la complejidad dentro de límites manejables, permitiendo de esta forma obtener clases que pueden ser reutilizadas y poco vulnerables a los cambios que puedan surgir.

Controlador: Este patrón se puede ver con la creación de las clases controladoras teniendo en cuenta las diferentes funcionalidades que se implementan; para la gestión de las Incidencias, Reparaciones, Devoluciones, Asistencia Técnica se crearon las clases del mismo nombre encargadas de atender los pedidos que llegan desde la vista.

Experto: Este patrón delega las responsabilidades a quién contiene la información necesaria para cumplirlas. Fue utilizado en todas las clases definidas.

Alta cohesión: Su empleo está dado en que fueron asignadas responsabilidades a las clases de forma tal que la cohesión siguiera siendo alta, o sea, cada clase se encargará de realizar solamente las funciones que estén en correspondencia con la responsabilidad que posea. Este patrón se evidencia en la clase empleado, esta contiene varias funcionalidades con un propósito único, no desempeñado por el resto de los elementos, siendo estas funcionalidades las encargadas de controlar las acciones de las vistas.

Bajo acoplamiento: El objetivo de este patrón es tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. Este patrón se encuentra presente en todo el sistema en general.

2.1.3 Diagrama de secuencia

Los diagramas de secuencia muestran la forma en que los objetos se comunican entre sí al transcurrir el tiempo. Luego de la valoración del análisis y diseño se determinó que era necesaria la realización de estos diagramas debido a que no existían en el Expediente de proyecto y son de gran apoyo para que los programadores tengan una visión más clara del flujo del requisito a implementar. En la Figura 4 se muestra el diagrama de secuencia para el requisito Adicionar empleado, el resto de los diagramas se encuentran en el Expediente de proyecto.

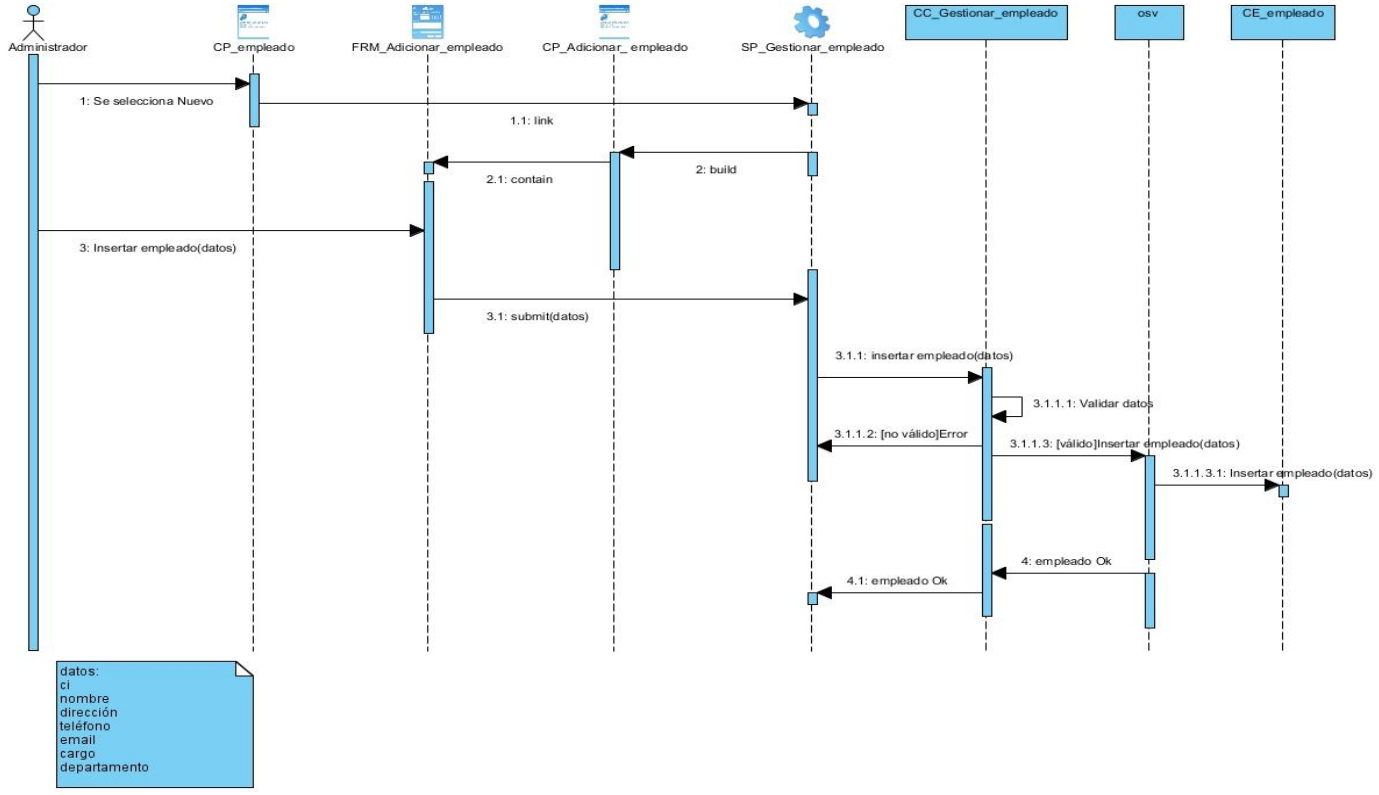

Figura 4: Diagrama de secuencia Adicionar empleado

2.1.4 Modelo conceptual

El modelo conceptual de los procesos Atención al cliente y Soporte fue redefinido partiendo de las clases conceptuales, atributos y relaciones existentes entre las mismas; teniendo en cuenta las modificaciones que surgieron en estos procesos según las nuevas estrategias trazadas en el proyecto. Los nuevos conceptos son: Empleado, Cargo, Departamento, Tipo de asistencia técnica, Tipo de reparación, Motivo de la asistencia técnica, Motivo de la devolución. En la Figura 5 se muestra cómo quedó finalmente el nuevo Modelo conceptual.

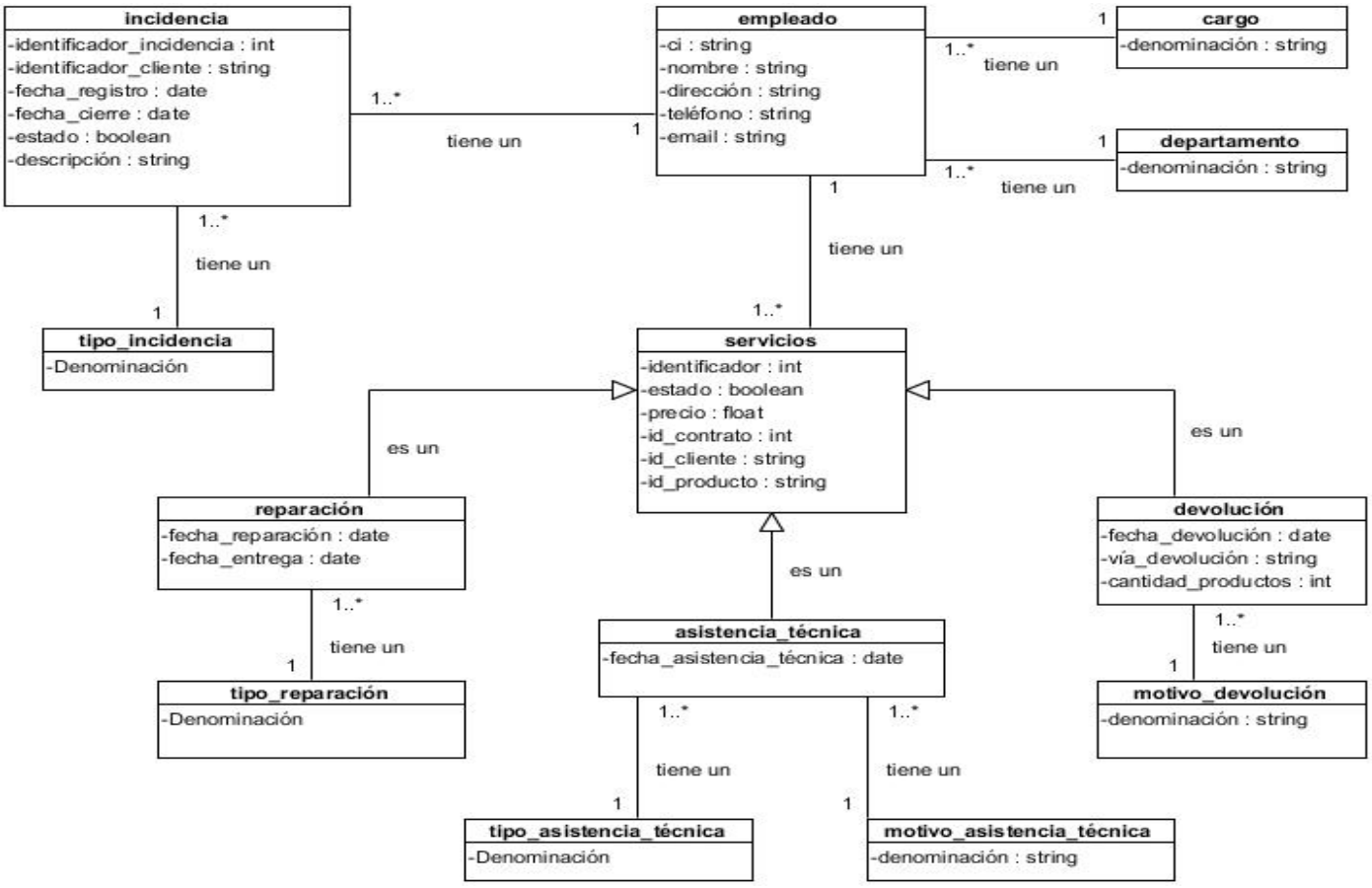

Figura 5: Modelo conceptual de los procesos Atención al cliente y Soporte

2.1.5 Modelo de datos

En el modelo de datos elaborado para los procesos de Atención al cliente y Soporte se visualizan las entidades con sus atributos, así como las relaciones entre ellas, estas son las encargadas de almacenar todos los datos necesarios para que el sistema pueda ofrecer las funcionalidades que fueron identificadas durante la captura de requisitos. A continuación se presenta en la Figura 6 el Modelo de datos con los cambios necesarios para lograr darle cumplimiento a los requisitos planteados:

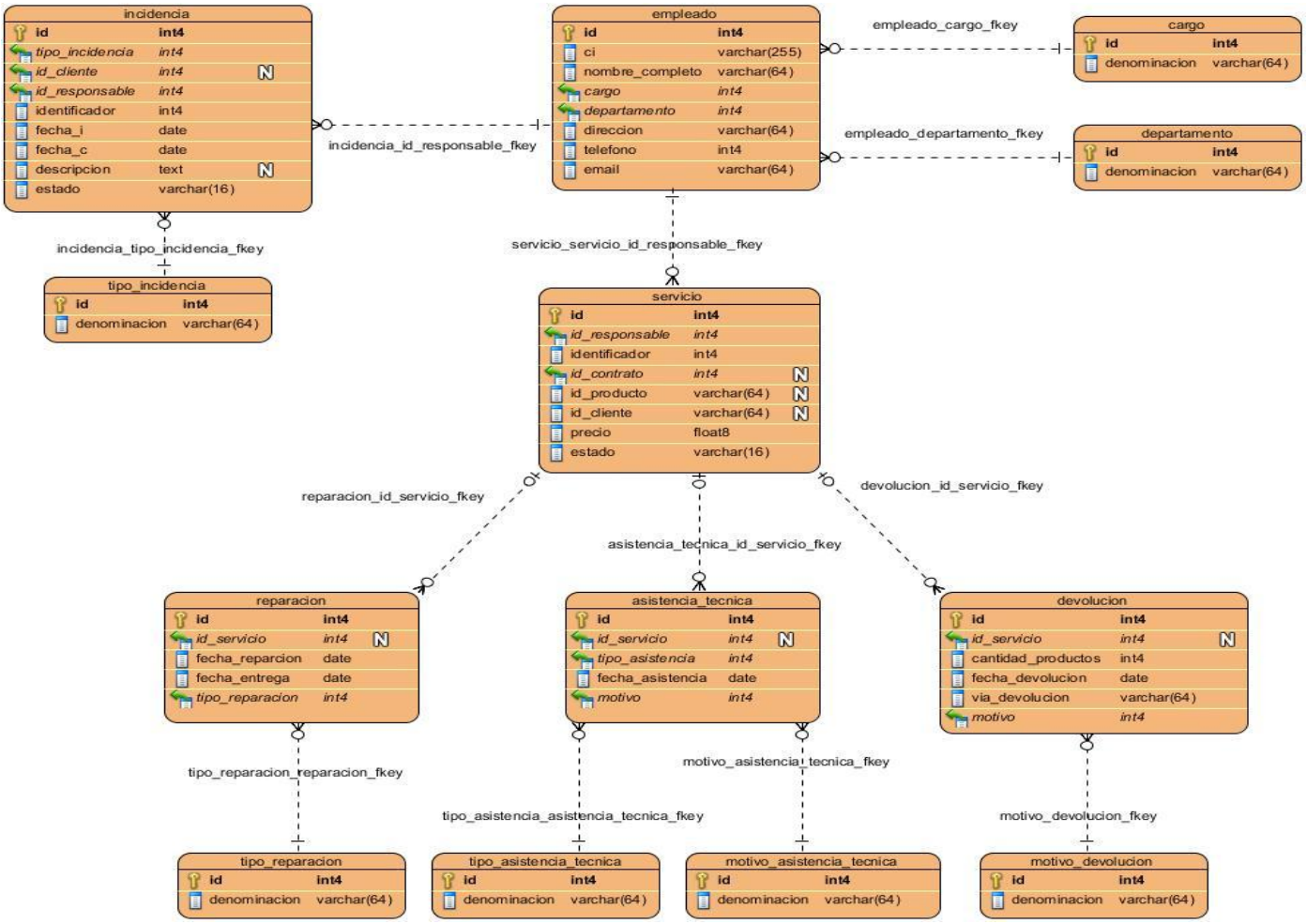

Figura 6: Modelo de datos de los procesos Atención al cliente y Soporte

2.2 Diagrama de componentes

El diagrama de componentes representa los elementos de diseño de un sistema de software, permiten visualizar con mayor facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. A continuación se presenta en la Figura 7 el Diagrama de componentes propuesto para el desarrollo de la aplicación, este recibió un grupo de cambios necesarios para lograr alcanzar el resultado deseado luego de las nuevas definiciones en el proyecto.

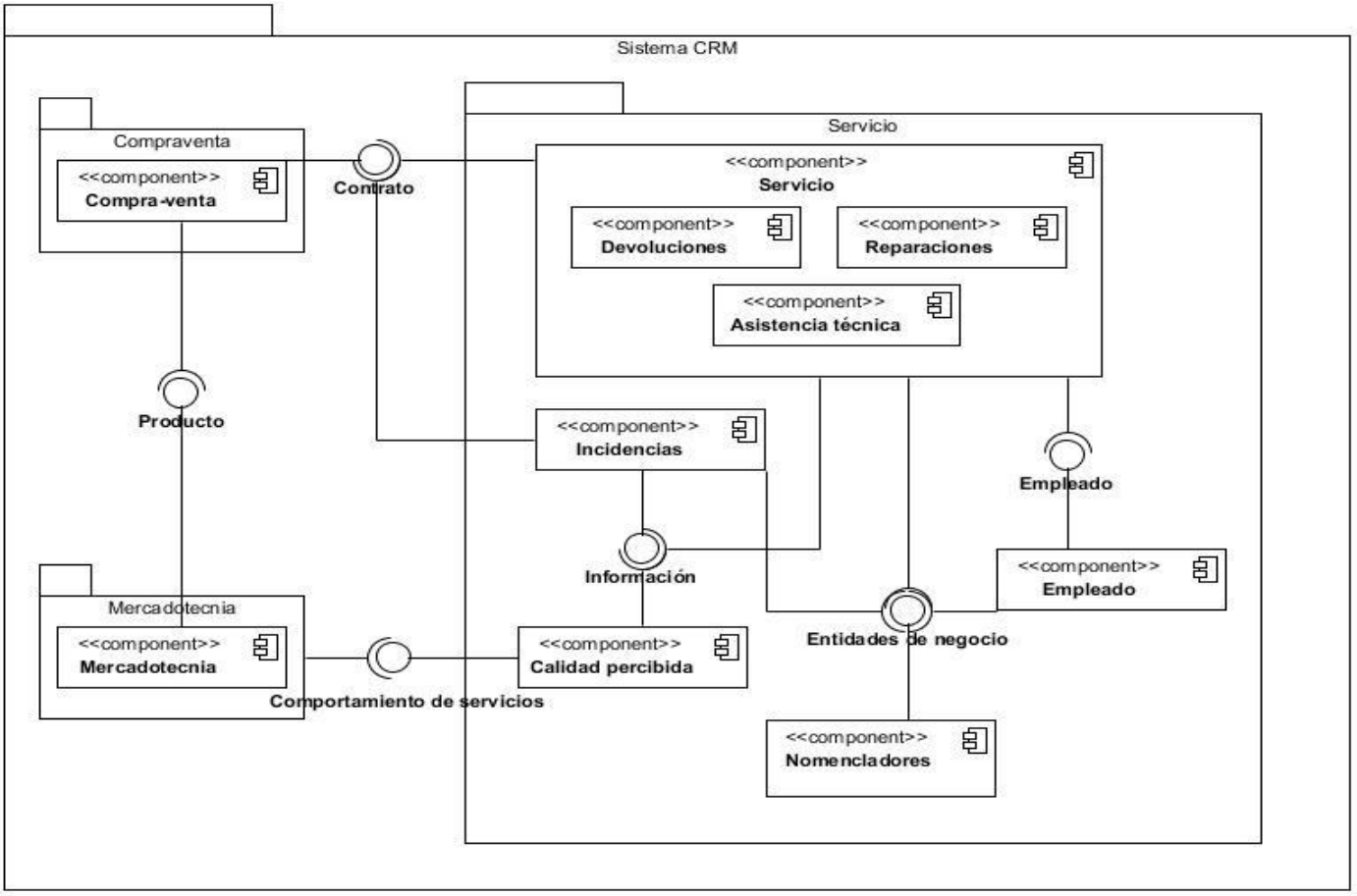

Figura 7: Diagrama de componentes

Seguidamente se presenta una explicación más detallada de la funcionalidad de cada uno de estos componentes a partir de los servicios que brindan y reciben:

Mercadotecnia: Se encarga de realizar un estudio del mercado, además de ejecutar actividades que ayudarán a gestionar las relaciones con las empresas. El módulo Servicios le brinda información referente al comportamiento de los clientes según la satisfacción que muestren por los servicios recibidos.

Compra/Venta: Se encarga de registrar las solicitudes de compra, los pedidos de compra, las solicitudes de presupuestos, solicitudes de compra, las solicitudes de presupuesto y los pedidos de compra, así como de gestionar los clientes de la empresa. Este componente le brinda información necesaria a Servicios para

su correcto funcionamiento como por ejemplo quiénes son los clientes de la empresa y sus productos adquiridos; información que se envía a través del contrato.

Servicio: En este componente se les suministran diferentes prestaciones a los clientes como devoluciones, reparaciones, asistencia técnica, incidencias que pueden ser quejas o reclamaciones, también permite gestionar tipos de incidencia y tipos de asistencia técnica, para de este modo lograr la satisfacción de los mismos, además se encargará de gestionar los empleados de la empresa.

Incidencias: En este componente se gestionan las incidencias que pueden surgir luego de realizado un servicio a un cliente, las mismas pueden tener diferentes clasificaciones y requerirán de una persona encargada de darle respuesta en la fecha establecida, para ello el componente Empleado le brinda acceso al empleado que se desea seleccionar para atender la incidencia.

Calidad percibida: Este componente es el encargado de gestionar una serie de cuestionarios para investigar sobre la satisfacción de los clientes con relación a los servicios recibidos para finalmente brindarle información a Mercadotecnia sobre el comportamiento de los servicios realizados. En esta fase no se implementará dicho componente.

Empleado: El componente empleado es el encargado de gestionar las personas que trabajarán en la empresa, las cuales estarán encargadas de prestar los diferentes tipos de servicios y resolver las incidencias.

Nomencladores: Este componente gestiona las entidades de negocio de los componentes Servicios, Incidencias y Empleado.

2.3 Estándares de codificación

Nomenclatura de las clases, funciones y variables: El nombre de la clase lleva una separación “_”, con el propósito de no poner letras en mayúscula, lo mismo sucede con la funciones y las variables, estos serán escritos de forma tal que sean lo suficientemente claros y precisos y que con sólo leerlo se reconozca el propósito de los mismos.

Ejemplo de nombre de una clase: servicios_incidencia.

Ejemplo de nombre de una función: chequear_fechas_devolucion.

Ejemplo de nombre de una variable: id_incidencia.

Normas de comentarios: Es necesario comentar todo lo que se haga dentro del desarrollo de los procesos, para de esta forma se puedan establecer normas que conduzcan a lograr un código más legible y que pueda ser reutilizado.

Nomenclatura de los comentarios: Los comentarios deben ser lo suficientemente claros y precisos de tal forma que se comprenda que se está desarrollando. Antes de declarar una clase o una función se escribe una breve descripción que exponga el propósito de la misma.

En la Figura 8 se muestra un ejemplo de un comentario de una clase:

```
#########################################################################
# Clase: servicio_incidencia                                            #
# Descripción: Gestiona los departamentos de una empresa                #
# Autor: Antonio García López                                           #
# Versión: 1.0                                                          #
#########################################################################
```

Figura 8: Comentarios de la clase incidencia

En la Figura 9 se muestra el ejemplo de un comentario de la función get_contrato:

```
#función para determinar el cliente y el producto por el contrato
```

Figura 9: Comentarios de la función get_contrato

2.4 Descripción de las principales clases

Seguidamente se muestra una breve descripción de las principales clases que forman parte de la solución, sus atributos y los datos que admiten sus campos. Las Tablas de la 7 a la 12 muestran dichos datos.

Tabla 7: Descripción de la clase incidencia

Descripción	Es la queja, reclamación, duda, entre otras., que tiene un cliente en relación con algún producto o servicio adquirido.					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	Clases No válidas
Id	Identificador de la incidencia.	int	No	Si	Números	Letras y caracteres especiales
Identificador del cliente	Identificador del cliente.	string	No	Si	Letras, números y caracteres especiales	N/A
Identificador del responsable	Identificador del responsable.	string	No	Si	Letras, números y caracteres especiales	N/A
Fecha de la incidencia	Fecha en que se registra la incidencia.	datetime	No	No	Números y el carácter especial / (dd/mm/aaaa)	Letras y caracteres especiales
Fecha de cierre	Fecha en que culmina la solicitud.	datetime	No	No	Números y el carácter especial / (dd/mm/aaaa)	Letras y caracteres especiales
Estado	Estado de la incidencia.	boolean	No	No	Abierta/ Cerrada	N/A
Detalles	Detalles de la incidencia.	string	No	No	Letras	Números y caracteres

						especiales

Tabla 8: Descripción de la clase servicio

Descripción	Servicios de soporte que brinda la empresa; los cuales pueden ser Asistencias técnicas, devoluciones y reparaciones.					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	Clases No válidas
Identificador	Identificador del servicio.	int	No	Si	Números	Letras y Caracteres especiales
Identificador del contrato	Identificador del contrato que tiene el cliente de la empresa.	int	No	Si	Números	Letras y Caracteres especiales
Identificador del cliente	Identificador del cliente.	string	No	Si	Números	Letras y Caracteres especiales
Identificador del responsable	Identificador del responsable.	string	No	Si	Números	Letras y Caracteres especiales
Identificador del producto	Identificador del producto.	string	No	Si	Letras, números y caracteres especiales	N/A
Precio	Precio por la prestación de dicho servicio.	float	No	No	Números	Letras y Caracteres especiales
Estado	Estado del servicio.	boolean	No	No	Abierta/ Cerrada	N/A

Tabla 9: Descripción de la clase asistencia técnica

Descripción	Asistencia que se les brinda a los clientes en caso de presentar algún problema con el recurso adquirido.					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	Clases No válidas
Fecha de la asistencia técnica	Fecha en que se brinda la asistencia técnica.	datetime	No	No	Números y el carácter especial / (dd/mm/aaaa)	Letras y caracteres especiales
Tipo de la asistencia técnica.	Tipos de asistencia técnica que brinda la empresa.	string	No	Si	Letras, números y caracteres especiales	N/A
Motivo de la asistencia técnica	Motivo por el cual se presta la asistencia técnica.	string	No	Si	Letras, números y caracteres especiales	N/A

Tabla 10: Descripción de la clase devoluciones

Descripción	Son las devoluciones realizadas de un producto adquirido.					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	Clases No válidas
Motivo de la devolución	Motivo por el cual se devuelve el producto.	string	No	Si	Letras y números	Caracteres especiales

Fecha de la devolución	Fecha en la que fue devuelto el recurso.	datetime	No	No	Números y el carácter especial / (dd/mm/aaaa)	Letras y caracteres especiales
Cantidad	Cantidad de recursos devueltos.	int	No	No	Números	Letras y caracteres especiales
Vía de devolución	Vía por la cual se realiza la devolución.	string	No	No	Letras y números	Caracteres especiales

Tabla 11: Descripción de la clase reparaciones

Descripción	Es la acción que se realiza para reparar un producto defectuoso					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	Clases No válidas
Fecha de reparación	Fecha en que se repara el producto.	datetime	No	No	Números y el carácter especial / (dd/mm/aaaa).	Letras y caracteres especiales.
Fecha de entrega	Fecha en que se entrega el producto.	datetime	No	No	Números y el carácter especial / (dd/mm/aaaa).	Letras y caracteres especiales.
Tipo reparación	Tipo de reparaciones que ofrece la empresa.	string	No	Si	Letras y números.	Caracteres especiales.

Tabla 12: Descripción de la clase empleado

Descripción	Personal que trabaja en la empresa, al cual atiende los servicios de soporte y resolución de incidencias.					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	Clases No válidas
Nombre completo	Nombre completo de la persona.	string	No	No	Letras	Números y caracteres especiales.
CI	Número de carné de identidad.	string	No	Si	Números	Letras y caracteres especiales.
Cargo	Es el cargo que ocupa en la empresa.	string	No	Si	Letras	Número y caracteres especiales.
Departamento	Departamento al que pertenece.	string	No	Si	Letras	Número y caracteres especiales.
Teléfono	Teléfono del recurso humano.	string	Si	No	Números	Letras y caracteres especiales.
Dirección particular	Dirección particular donde radica la persona.	string	No	No	Letras, números y caracteres especiales.	N/A
Email	Correo electrónico de la persona.	string	Si	No	Letras, números y caracteres especiales.	N/A

2.5 Tratamiento de errores

Para garantizar el correcto funcionamiento del sistema se debe tener en cuenta el tratamiento a los errores, este es uno de los procesos más importantes en la creación de cualquier sistema informático. Son capturadas todas aquellas excepciones que son lanzadas y se facilita un tratamiento adecuado de las mismas para que el sistema no colapse, para de esta forma contribuir a la satisfacción de los usuarios.

Las validaciones garantizan que se realice un correcto manejo de los errores en la aplicación, de esta manera las peticiones del usuario son capturadas y verificadas para así dar tratamiento a los diferentes tipos de errores lanzados. Permite mostrarle de manera visible al usuario mensajes de confirmación, esto le dará una idea al usuario de qué es lo que está incorrecto y como solucionarlo. La información que requiera ser adicionada por el usuario se validará garantizando que sea correcta y que los campos obligatorios no estén vacíos. De haber algún error en la información introducida por el mismo, se mostrará un mensaje en pantalla informando el error y señalando los campos que contengan estos errores, al oprimir el botón Aceptar el mensaje desaparecerá y el usuario podrá corregir los datos en el formulario.

Además serán validadas las opciones que correspondan a la modificación de datos que provienen del servidor de base datos, si se desea eliminar algún elemento de este se le pregunta al usuario si está seguro de realizar dicha acción. Si se quiere eliminar un nomenclador y este está asociado a otra clase o un empleado asociado a un servicio o incidencia se mostrará un mensaje de error. También se validará que no se permita insertar una devolución si el producto no tiene garantía o una reparación que el precio sea mayor que cero si el producto aún está en garantía. En la figura 10 se muestra cómo el sistema señala los campos incorrectos.

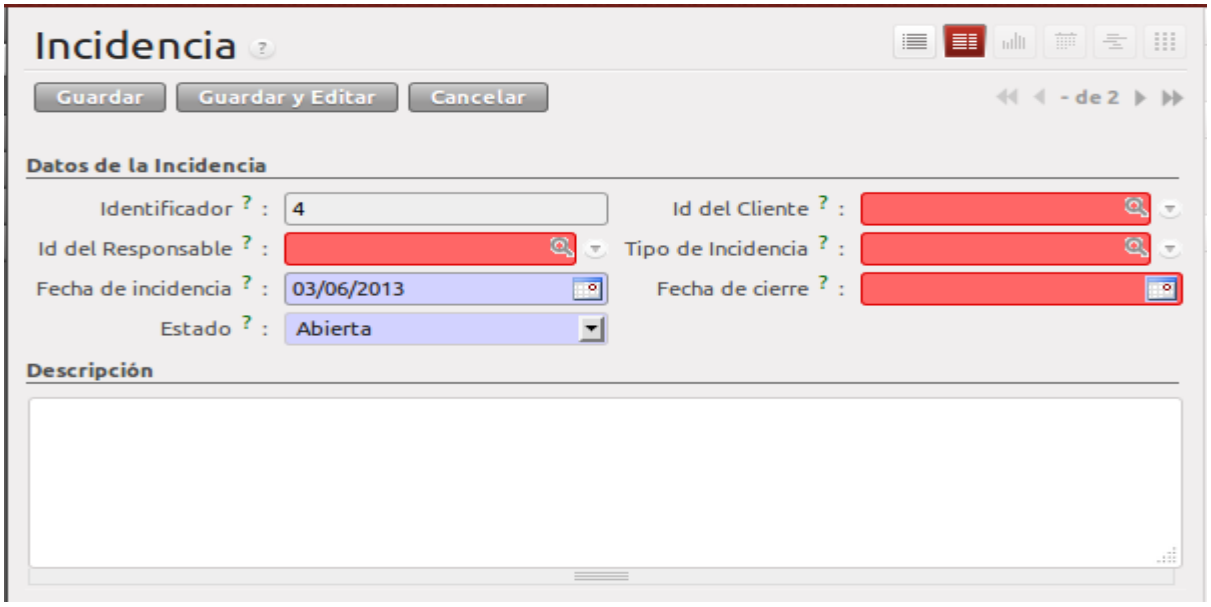

Figura 10: Tratamiento errores

2.6 Descripción de la aplicación

La interfaz principal muestra todos los módulos instalados en el sistema, ver Figura 11, seguidamente se puede seleccionar el módulo Servicios y acceder a las funcionalidades implementadas de los procesos Atención al cliente y Soporte, permitiéndole al usuario interactuar con el sistema según los privilegios otorgados.

En la interfaz principal del módulo Servicios, se encuentra a la izquierda un menú donde el usuario puede seleccionar la funcionalidad que desee utilizar de los procesos implementados. En la parte derecha se encuentra un tablero donde se pueden observar los datos que se han adicionado a la aplicación, además se puede acceder también directamente a la información dando un clic sobre esta y desde ahí adicionar, modificar, consultar o eliminar dicha información (ver Figura 12).

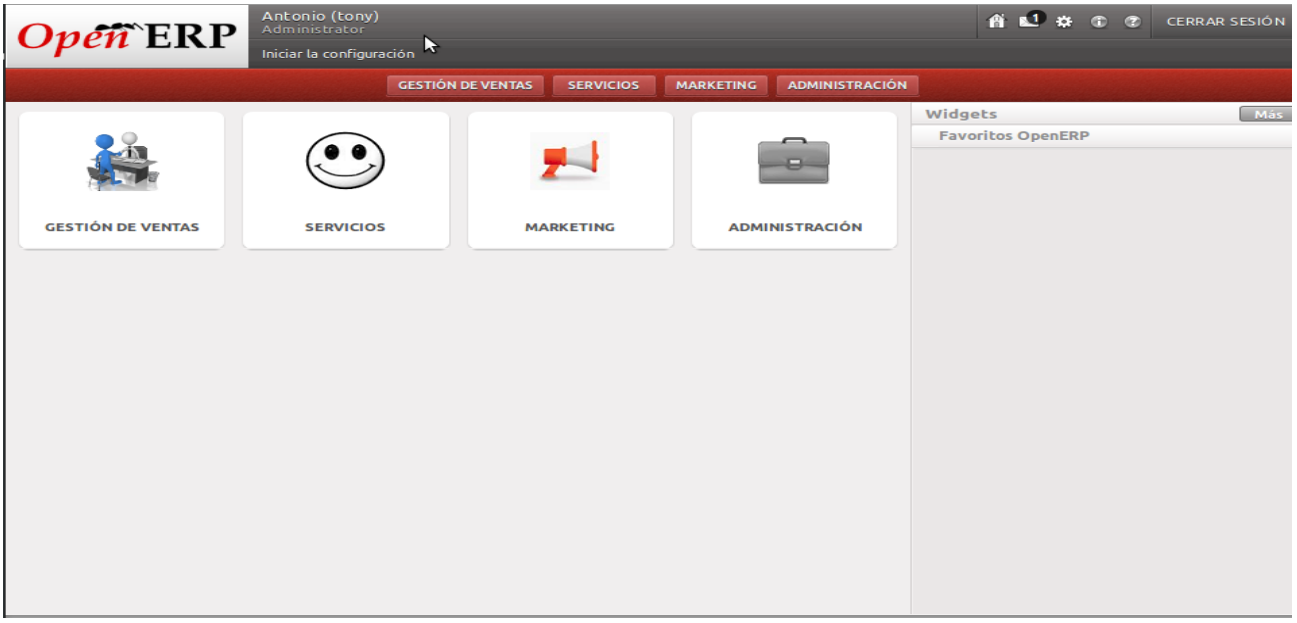

Figura 11: Interfaz principal del OpenERP

Figura 12: Interfaz principal del módulo Servicios

Este sistema permite mejorar la gestión de los procesos Atención al cliente y Soporte dentro de la empresa. Todo el personal de un departamento o sección de la misma dispone en tiempo real de la información de interés para su área de influencia, al centralizar la información y hacer que el personal trabaje con el mismo sistema se logra que todas las áreas de la empresa tengan la información sobre las incidencias, reparaciones, devoluciones, asistencia técnica. Mejora la imagen de la empresa de cara a los clientes, ya que se pierde menos tiempo buscando información y por tanto se agiliza el trato con el cliente.

Conclusiones parciales

En este capítulo se hizo una valoración del análisis y diseño propuesto para el desarrollo de la aplicación, en la cual se definieron un grupo de cambios que permitieron lograr la implementación de los procesos con la calidad necesaria. Se definió el estándar de código por el cual se regirán los programadores del proyecto, logrando de esta forma hacer más legible el código fuente, así como posibilitar un mejor entendimiento del mismo, permitiendo una codificación homogénea y evitando las redundancias. Además se tuvieron en cuenta elementos muy importantes como el tratamiento de errores y la descripción de las clases implementadas. De forma general se logró implementar las funcionalidades asociadas a los procesos Atención al cliente y Soporte correspondientes al módulo Servicios haciendo uso de la plataforma de desarrollo seleccionada, quedando listo el sistema para el posterior desarrollo de las pruebas.

Capítulo 3: “Validación de la solución”

Introducción

El desarrollo de un software es algo complicado y existen numerosas posibilidades de cometer errores, motivo por el que este proceso ha de ir conducido por actividades que garanticen la calidad; las pruebas de caja blanca y caja negra son un elemento que ayudan a garantizar la calidad del software. En este capítulo se valida la solución propuesta, de manera que se puede comprobar que los procesos implementados cumplan con los requisitos planteados en el análisis y diseño de la aplicación. Para esto se presenta la descripción de casos de prueba que verifican la correspondencia con las funcionalidades definidas y las pruebas de caja blanca al código obtenido.

3.1 Pruebas de Caja Blanca

Pruebas de caja blanca son mucho más amplias que las de caja negra, normalmente se denominan pruebas de cobertura o pruebas de caja transparente, al total de pruebas de caja blanca se le llama cobertura, la cobertura es un número porcentual que indica cuanto código del programa se ha probado.

Básicamente la idea de pruebas de cobertura consiste en diseñar un plan de pruebas en las que se vaya ejecutando sistemáticamente el código hasta que haya corrido todo o la gran mayoría de él. Cuando el programa contiene código de difícil alcance, como por ejemplo los manejadores de errores o código muerto. Entiéndase por código muerto a aquellas funciones y/o procedimientos que se han incluido por encontrarse en recopilaciones pero que nunca son ejecutadas por el programa, estas funciones no necesariamente deberán ser removidas pero sí probadas por si algún día en revisiones futuras son incluidas. (35)

Uno de los tipos de pruebas de caja blanca es la Prueba del Camino Básico, a continuación se especifica en qué consiste la misma:

Permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. (36)

Para realizar la prueba de caja blanca específicamente la prueba del camino básico es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar. A continuación, en la Figura 13, se enumeran las sentencias de código del procedimiento realizado sobre el método get_contrato (self, cr, uid, ids, id_contrato) el cual se encarga de devolver el cliente y el producto asociado a un contrato que se pasa por parámetro.

```
def get_contrato(self, cr, uid, ids,id_contrato):
    if not id_contrato:                                                  ->1
        return {'value': {'id_cliente': False, 'id_producto': False}}   ->6
    cr.execute('SELECT producto_type.name                                ->2
FROM public.contrato_type, public.producto_type
WHERE   contrato_type.id_producto = producto_type.id
AND contrato_type.id = %d' % (id_contrato,))
    produto = cr.fetchone()[0]                                           ->3
    cr.execute('SELECT contrato_type.nombre_cliente                      ->4
FROM public.contrato_type, public.producto_type
WHERE contrato_type.id_producto = producto_type.id
AND contrato_type.id = %d' %(id_contrato,))
    cliente = cr.fetchone()[0]                                           ->5
    return {'value': {'id_cliente': cliente, 'id_producto': produto}}   ->6
```

Figura 13: Código del algoritmo get_contrato (self, cr, uid, ids, id_contrato)

Seguidamente, en la Figura 14, se muestra el grafo que representa los posibles caminos a seguir del método get_contrato.

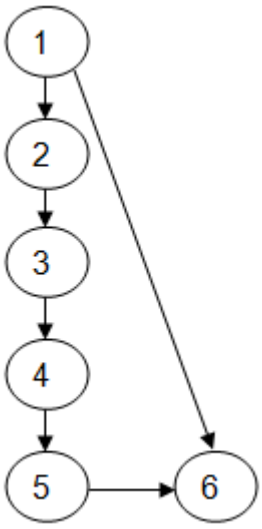

Figura 14: Grafo de flujo asociado al algoritmo get_contrato (self, cr, uid, ids, id_contrato)

Fórmulas para calcular complejidad ciclomática:

1. V (G) = (A – N) + 2. Siendo “A” la cantidad total de aristas y “N” la cantidad total de nodos. Para cada formula “V (G)” representa el valor del cálculo.

V (G) = (6– 6) + 2 = 2

2. V (G) = P + 1. Siendo “P” la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

V (G) = 1 + 1 = 2

3. V (G) = R. Siendo “R” la cantidad total de regiones.

V (G) = 2

El cálculo efectuado mediante las tres fórmulas arrojaron el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es dos, lo que significa que existen dos caminos posibles por donde puede circular el flujo, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado. Seguidamente se representan en la Tabla 13 los caminos básicos por los que puede recorrer el flujo.

Tabla 13: Caminos básicos del flujo asociado al algoritmo get_contrato (self, cr, uid, ids, id_contrato)

Número	Camino
1	1-2
2	1-2-3-4-5-6

Luego de haber extraído los caminos básicos del flujo, se pasa a la ejecución de los casos de pruebas para este procedimiento, debe realizarse al menos un caso de prueba por cada camino básico. Para realizarlos es preciso cumplir con las siguientes exigencias:

- Descripción: Se hace la entrada de datos necesaria y se valida que ningún parámetro obligatorio pase nulo al procedimiento o se entre algún dato errado.
- Condición de ejecución: Se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.
- Entrada: Se muestran los parámetros que entran al procedimiento.
- Resultados Esperados: Se expone el resultado que se espera devuelva el procedimiento.

Caso de prueba para el camino básico 1:

Camino 1: [1-2]

- Descripción: No se pasa la variable id_contrato y se crea una tupla vacía.
- Condición de ejecución: No se pasa la variable id_contrato.
- Entrada: vacío

- Resultados esperados: No se completan los campos identificador del cliente e identificador del producto.

Caso de prueba para el camino básico 2:

Camino 1: [1-2-3- 4-5-6]

- Descripción: Se pasa la variable id_contrato, se realizan las consultas a la base de datos y se obtienen los valores de identificador cliente e identificador producto.
- Condición de ejecución: Se pasa la variable id_contrato.
- Entrada: 1
- Resultados esperados: Se completan los campos correspondientes al identificador del cliente y el identificador del producto.

A continuación, en la Figura 15, se enumeran las sentencias de código del procedimiento realizado sobre el método chequear_tiene_garantia (self, cr, uid, ids, context=None) el cual se encarga de chequear si un producto tiene garantía.

```
def checkear_tiene_garantia(self, cr, uid, ids, context=None):
    result = True                                              ->1
    record = self.browse(cr, uid, ids, context=context)        ->1
    for data in record:                                        ->2
        id = data.id_contrato                                  ->3
    cr.execute('SELECT contrato_type.devolver_producto         ->4
FROM public.contrato_type, public.producto_type
WHERE contrato_type.id_producto = producto_type.id
AND contrato_type.id = %d' %(id,))
    devolver = cr.fetchone()[0]                                ->4
    if not devolver:                                           ->5
        result = False                                         ->6
    return result                                              ->7
```

Figura 15: Código del algoritmo chequear_tiene_garantia (self, cr, uid, ids, context=None)

Seguidamente, en la Figura 16, se muestra el grafo que representa los posibles caminos a seguir del método chequear_tiene_garantia (self, cr, uid, ids, context=None).

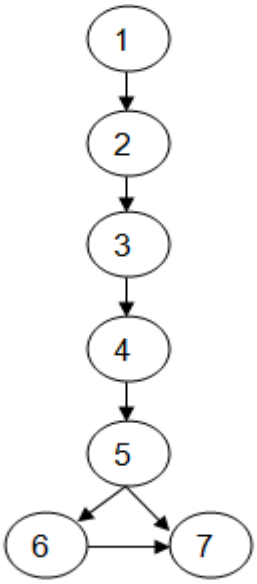

Figura 16: Grafo de flujo asociado al algoritmo chequear_tiene_garantia (self, cr, uid, ids, context=None)

Complejidad ciclomática por las tres fórmulas:

V (G) = (A – N) + 2

V (G) = (7– 7) + 2 = 2

V (G) = P + 1

V (G) = 1 + 1 = 2

V (G) = R**.**

V (G) = 2

Seguidamente se representan en la Tabla 14 los caminos básicos por los que puede recorrer el flujo.

Tabla 14: Caminos básicos del flujo asociado al algoritmo chequear_tiene_garantia (self, cr, uid, ids, context=None)

Número	Camino
1	1-2-3-4-5-7
2	1-2-3-4-5-6-7

Caso de prueba para el camino básico 1:

Camino 1: [1-2-3-4-5-7]

- Descripción: Se selecciona de la lista un producto asociado a un cliente mediante el contrato.
- Condición de ejecución: Debe existir un contrato.
- Entrada: Se selecciona un producto que no tenga devolución.
- Resultados esperados: Se muestra un error informando que el producto no tiene devolución.

Caso de prueba para el camino básico 2:

Camino 1: [1-2-3-4-5-6-7]

- Descripción: Se selecciona de la lista un producto asociado a un cliente mediante el contrato.
- Condición de ejecución: Debe existir un contrato.
- Entrada: Se selecciona un producto que tenga devolución.
- Resultados esperados: Se adiciona correctamente la devolución.

Luego de aplicar los distintos casos de pruebas para las funcionalidades más críticas implementadas, se pudo comprobar que el flujo de trabajo de las mismas es correcto. Además se probó que cada sentencia es ejecutada al menos una vez, cumpliéndose así las condiciones de la prueba.

3.2 Pruebas de Caja Negra

Pruebas de caja negra: Son aquellas que se enfocan directamente en el exterior del módulo, sin importar el código, son pruebas funcionales en las que se trata de encontrar fallas en la interfaz de usuario, apariencia de los menús, control de las teclas, entre otros. Este tipo de pruebas no es aplicable a los módulos que trabajan en forma transparente al usuario.

Para realizar estas pruebas existe una técnica algebraica llamada "clases de equivalencia", consiste en tratar a todos las posibles entradas y parámetros como un modelo algebraico, y utilizar las clases de este modelo para probar un amplio rango de posibilidades. (35)

Las pruebas de caja negra se realizan durante el proceso de revisión del software, en estas se pretenden encontrar a través de los casos de prueba estos tipos de errores:

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

Fue necesario desarrollar los diseños de casos de prueba asociados a cada una de las funcionalidades implementadas teniendo en cuenta que dichos artefactos no existían en el Expediente de proyecto. En el siguiente epígrafe en las tablas de la 15 a la 17 se expone una muestra de los casos de prueba realizados, el mismo corresponde al requisito adicionar empleado para la agrupación de requisitos Gestionar empleado. Los demás casos de prueba de las 12 agrupaciones de requisitos se pueden encontrar en el Expediente de proyecto.

Descripción del Caso de Prueba “Adicionar empleado”

Condiciones de ejecución:

1. Se debe identificar y autenticar en el sistema, además de tener los permisos para ejecutar esta acción.
2. Debe existir un cargo y un departamento para asignar al empleado.
3. Se debe seleccionar la opción del menú: Servicios/Empleado/Empleado/Nuevo.

Requisito a probar

Tabla 15: Escenarios de prueba del requisito adicionar empleado

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Adicionar empleado.	Debe permitir adicionar empleados en el sistema.	EP 1.1: Adicionar empleado introduciendo datos válidos.	➢ Se introducen los datos del empleado correctamente. ➢ Se presiona el botón **Aceptar**. ➢ Se muestra el empleado adicionado. ➢ Se presiona el botón **Guardar**.
		EP 1.2: Adicionar un empleado introduciendo datos inválidos.	➢ Se introducen los datos inválidos del empleado. ➢ Se presiona el botón **Aceptar**. ➢ Se muestra un mensaje de error, el

			sistema señala los campos erróneos y permite corregirlos. ➢ Se presiona el botón **Aceptar**.
		EP 1.3: Adicionar empleado que ya se encuentra en el sistema.	➢ Se introducen los datos de un empleado que ya se encuentra en el sistema. ➢ Se presiona el botón **Aceptar**. ➢ Se muestra un mensaje informando del error. ➢ Se presiona el botón **Aceptar**.
		EP 1.4: Adicionar empleado dejando campos vacíos.	➢ Se introducen los datos dejando algún campo en blanco. ➢ Se presiona el botón **Aceptar**. ➢ Se muestra un mensaje informando del error, el sistema te muestra en rojo los campos vacíos y permite corregirlos.

			➢ Se presiona el botón **Aceptar**.
		EP 1.5: Cancelar.	➢ Se introducen o no los datos del empleado. ➢ Se presiona el botón **Cancelar**.

Descripción de variable

Tabla 16: Descripción de las variables del requisito adicionar empleado

No	Nombre de campo	Detalles	Tipo	Válido	Inválido
1	CI	Carnet de identidad del empleado.	string	Números	Letras y caracteres especiales.
2	Nombre	Nombre del empleado.	string	Letras.	Números y caracteres especiales.
3	Dirección	Dirección del empleado.	string	Números, letras y caracteres especiales.	N/A
4	Teléfono	Teléfono del empleado.	int	Números	Letras y caracteres especiales.

5	Email	Correo electrónico del empleado.	string	Letras y caracteres especiales.	Números.
6	Cargo	Cargo del empleado.	string	Letras.	Números y caracteres especiales.
7	Departamento	Departamento del empleado.	string	Letras.	Números y caracteres especiales.

Juegos de datos a probar

Tabla 17: Juegos de datos a probar requisito adicionar empleado

Id del escenario	Escenario	Ci.	Nombre	Dirección	Teléfono	Email	Cargo	Departamento	Respuesta del sistema	Resultado

EP 1.1	Adicionar empleado introduciendo datos válidos.	V(89090734327)	V(Cary)	V(Calle A)	V(58076222)	V(cary@uci.cu)	V(admin)	V(Dirección)	El sistema adiciona el empleado y mantiene la interfaz abierta. Se presiona el botón **Guardar** y se muestra una lista con los empleados adicionadas en el sistema.	NA
EP 1.2	Adicionar empleado introduciendo datos inválidos.	I(¡4xcf)	I(asd87)	N/A	I(580a)	I(cary@uci.)	I(asd	I(asd87)	El sistema muestra el mensaje de error: “Datos incorrectos”, el sistema te señala los campos incorrectos y	NA
		I(¡4xcf)	I(asd87)	N/A	I(580a)	V(cary@uci.cu	I(asd	I(asd87)		
		I(¡4xcf)	I(asd87)	N/A	V(58076222)	I(cary@uci.)	V(admin)	V(Dirección)		
		I(¡4xcf)	V(Cary)	N/A	I(580a)	I(cary@uci.)	V(admin)	I(asd87)		

		V(890 90734 327)	V(Cary)	N/A	I(580a)	I(cary@u ci.)	I(asd 87)	V(Direc ción)	permite corregirlos. El sistema mantiene la interfaz abierta	
		I(¡4xcf)	V(Cary)	N/A	V(5807 6222)	I(cary@u ci.)	I(asd 87)	I(asd 87)		
		I(¡4xcf)	I(asd 87)	N/A	V(5807 6222)	V(cary @uci.cu)	I(asd 87)	V(Direc ción)		
		V(890 90734 327)	V(Cary)	N/A	I(580a)	V(cary @uci.cu)	V(admi n)	I(asd 87)		
		I(¡4xcf)	V(Cary)	N/A	V(5807 6222)	V(cary @uci.cu)	V(admi n)	V(Direc ción)		
		V(890 90734 327)	V(Cary)	N/A	I(580a)	I(cary@ uci.)	V(admi n)	V(Direc ción)		

EP 1.3	Adicionar un empleado que ya se encuentre registrada en el sistema.	V(89090734327)	V(Cary)	V(Calle A)	V(58076222)	V(cary@uci.cu)	V(admin)	V(Dirección)	El sistema muestra el mensaje de error: “Ya existe una empleado con el mismo identificador.”. El sistema mantiene la interfaz abierta.	NA
EP 1.4	Adicionar empleado dejando campos vacíos.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	El sistema muestra el mensaje de error: “Por favor verifique los campos incorrectos.”. El sistema señala los	NA
		V(89090734327)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	V(admin)	I(Vacío)		
		I(Vacío)	V(Cary)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	V(Dirección)		
		I(Vacío)	I(Vacío)	V(Calle A)	I(Vacío)	I(Vacío)	V(admin)	I(Vacío)		
		I(Vacío)	I(Vacío)	I(Vacío)	V(58076222)	I(Vacío)	I(Vacío)	V(Dirección)		

		I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	V(cary@uci.cu)	V(admin)	I(Vacío)	datos erróneos y permite corregirlos. El sistema mantiene la interfaz abierta.	
		V(89090734327)	V(Cary)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)		
		I(Vacío)	V(Cary)	V(Calle A)	I(Vacío)	I(Vacío)	V(admin)	V(Dirección)		
		V(89090734327)	I(Vacío)	V(Calle A)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)		
		I(Vacío)	I(Vacío)	V(Calle A)	V(58076222)	V(cary@uci.cu)	V(admin)	I(Vacío)		
		V(89090734327)	V(Cary)	V(Calle A)	I(Vacío)	I(Vacío)	V(admin)	V(Dirección)		
		I(Vacío)	V(Cary)	V(Calle A)	V(58076222)	I(Vacío)	I(Vacío)	I(Vacío)		
		V(89090734327)	V(Cary)	V(Calle A)	V(58076222)	I(Vacío)	I(Vacío)	I(Vacío)		
		I(Vacío)	V(Cary)	V(Calle A)	V(58076222)	V(cary@uci.cu)	V(admin)	V(Dirección)		

EP 1.5	Cancelar	NA	NA	NA	NA	NA	NA	NA	El sistema cierra la interfaz sin realizar ninguna operación.	NA

Utilizando la descripción de los casos de prueba se realiza el flujo varias veces recreando posibles escenarios. Para todos los casos la aplicación debe ser capaz de mostrar el error e informar apropiadamente al usuario para su posterior corrección, de no ser así se toma como una no conformidad y se tendrá que rectificar previo a la siguiente iteración de las pruebas. Este proceso se repite para todos los casos de prueba de la aplicación garantizando la cobertura total de las funcionalidades.

Luego de ejecutar todos los casos de pruebas desarrollados por los requisitos se obtuvieron las siguientes No conformidades, ver Tabla 18.

Tabla 18: No conformidades detectadas por iteraciones

Iteración	No conformidades
Iteración 1	Ortografía 5, Validación 8
Iteración 2	Validación 2
Iteración 3	-

Como se muestra en la tabla anterior en la primera iteración se encontraron cinco No conformidades relacionadas con faltas de ortografía, todas estas fueron omisiones de tildes. Se encontraron además ocho problemas de validaciones de las cuales seis eran que el sistema permitía eliminar un nomenclador mientras este estaba siendo utilizado por otra funcionalidad; en los otros dos casos el sistema permitía

adicionar una devolución aunque el producto no estuviera en garantía, además permitía insertar una reparación donde el precio fuera mayor que cero cuando el producto aún estaba en garantía.

En la segunda iteración se encontraron dos No conformidades de validación, la primera era que se permitía modificar el identificador de un servicio, el cual era auto incrementado por el sistema de forma automática y de esta forma se podían introducir datos erróneos innecesariamente. La segunda no conformidad, el sistema cuando se insertaba un servicio la fecha de inserción del mismo no tomaba la fecha actual como predeterminada. Ya en la tercera iteración no se encontraron no conformidades, quedando listo el sistema para su utilización.

Conclusiones parciales

En este capítulo se aplicó la prueba de caja blanca a los principales algoritmos arrojando como resultado la aceptación de los parámetros del código implementado. Se desarrollaron los diseños de casos de prueba para cada una de las funcionalidades implementadas, los cuales fueron aplicados, buscando posibles errores para darle la solución requerida, comprobando que dichas funcionalidades cumplieran con las condiciones necesarias que se habían planteado. La ejecución de los diseños de casos de prueba arrojó 15 No conformidades, la cuales fueron solucionadas de forma inmediata; logrando así la aceptación de la solución requerida.

Conclusiones generales

A partir del estudio y refinamiento de los artefactos generados en el análisis y el diseño de la investigación sobre los diferentes sistemas de Administración de Relaciones con el Cliente, además del estudio de las herramientas, lenguajes de programación y metodología de desarrollo, se logró realizar de forma satisfactoria la implementación de los procesos Atención al cliente y Soporte logrando adaptar OpenERP a las funcionalidades requeridas. La realización de las pruebas de caja blanca y caja negra demostró que el sistema cumple con los requerimientos planteados para el buen funcionamiento de la aplicación.

Con la implementación de estos procesos, las incidencias que se reportan por parte de los clientes de la empresa pueden ser atendidas con mayor facilidad, tramitando la información en el menor tiempo posible. Además permite gestionar un grupo de servicios como son asistencia técnica, devoluciones y reparaciones los cuales buscan disminuir el tiempo de espera de un cliente y de esta forma garantizar la continuidad del servicio y la retención de los mismos. Por lo tanto se le da cumplimiento al objetivo de este trabajo, el mismo sirve de apoyo para lograr fidelidad de los clientes a las empresas y contribuir de este modo al proceso de informatización que se realiza en Cuba.

Recomendaciones

Las recomendaciones propuestas para la continuidad del presente trabajo son:

- Ampliar las funcionalidades del módulo con los requisitos asociados al componente Calidad percibida y además la implementación de la notificación de la cercanía de vencimiento de una incidencia.

- Poner al alcance de todos, este documento, como material de estudio, guía y apoyo para su posterior continuación.

Referencias bibliográficas

1. Cedeño Domínguez, Vicente. Diseño y Desarrollo de un sistema web que soporte el concepto CRM. 2007. págs. 3-8.
2. García Valcárcel, Ignacio. CRM: gestión de la relación con los clientes. 2012. págs. 4-6.
3. Definición ABC. [En línea] [Citado el: 25 de octubre de 2012.]
http://www.definicionabc.com/economia/atencion-al-cliente.php
4. Definición De [En línea] [Citado el: 25 de octubre de 2012.]
http://definicion.de
5. Definición CRM. [En línea] [Citado el: 25 de octubre de 2012.]
http://www.webandmacros.com/crm.htm
6. OpenERP. [En línea] [Citado el: 13 de noviembre de 2012.]
http://www.cybercia.com/index.php?option=com_content&view=article&id=52&Itemid=71#2.3
7. OpenERP. [En línea] [Citado el: 25 de octubre de 2012.]
http://www.openerpsite.com/erp-openerp-modulos
8. OpenERP-Spain. [En línea] [Citado el: 13 de noviembre de 2012.]
http://www.openerpspain.com/crm-srm
9. Aplicaciones Empresariales. [En línea] [Citado el: 25 de octubre de 2012.]
http://www.aplicacionesempresariales.com/terra-soft-crm-version-28-interactua-con-el-cliente-ahora.html
10. Complusoft. [En línea] [Citado el: 27 de octubre de 2012.]
http://www.complusoft.es/en/soluciones/sugarcrm
11. Redk. [En línea] [Citado el: 27 de octubre de 2012.]
http://www.redk.net/tecnologias/sugar-crm/funcionalidad/automatizacion-del-sac.html
12. SAP. [En línea] [Citado el: 27 de octubre de 2012.]
http://www.sap.com/
13. SAP CRM. [En línea] [Citado el: 29 de octubre de 2012.]
http://www.sap.com/spain/solutions/business-suite/crm/index.epx
14. Zelecto. [En línea] [Citado el: 13 de noviembre de 2012.]
http://www.zelecto.net/?p=379
15. Castellón García, Yudelsy. Daquinta Gómez, Edson. Rivero Martínez, Orlenys.

Avilaquid V4.0, Sistema de Gestión de Incidencias.
16. Vtiger [En línea] [Citado el: 29 de octubre de 2012.]
www.vtiger.com/es/
17. Vtiger [En línea] [Citado el: 29 de octubre de 2012.]
https://www.vtiger.com/es/crm/crm-features-pricing/customer-support-service/
18. González Obregón, William. Modelo de desarrollo de software. 2012. págs. 8-36.
19. Booch, G., Rumbaugh, J. y Jacobson I. El Lenguaje Unificado de Modelado. 2009.
págs. 50-68.
20. UML. [En línea] [Citado el: 2 de noviembre de 2012.]
http://www.gratisblog.com/index.php?itemid=130280.
21. Python. [En línea] [Citado el: 2 de noviembre de 2012.]
http://www.python.org/about/
22. Desarrollo web. [En línea] [Citado el: 2 de noviembre de 2012.]
http://www.desarrolloweb.com/articulos/1325.php
23. Calidad del software. [En línea] [Citado el: 3 de noviembre de 2012.]
http://www.slideshare.net/doknos/por-qu-usar-python
24. W3. [En línea] [Citado el: 2 de noviembre de 2012.]
http://www.w3.org/XML/
25. Conocimiento y Sistemas. [En línea] [Citado el: 5 de noviembre de 2012.]
http://conocimientoysistemas.wordpress.com/tag/caracteristicas-xml/
26. XML. [En línea] [Citado el: 5 de noviembre de 2012.]
http://www.hipertexto.info/documentos/xml.html
27. Visual Paradigm. [En línea] [Citado el: 7 de noviembre de 2012.]
(http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8%29_14720_p)
28. OpenObject. [En línea] [Citado el: 6 de noviembre de 2012.]
http://www.opentia.es/index.php/es/blogs/47-open-company/96-experiencia-openerp-francia.html#framework-qualites
29. Eclipse. [En línea] [Citado el: 6 de noviembre de 2012.]
http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.html/eclipse.html

30. Pydev. [En línea] [Citado el: 7 de noviembre de 2012.]
http://pydev.org/
31. Subversion [En línea] [Citado el: 7 de noviembre de 2012.]
http://www.osmosislatina.com/subversion/basico.htm.
32. PostgresSQL. [En línea] [Citado el: 7 de noviembre de 2012.]
http://danielpecos.com/docs/mysql_postgres/x15.html.
33. OpenERP-Aarchitecture. [En línea] [Citado el: 11 de noviembre de 2012.]
http://doc.openerp.com/trunk/developers/server/02_architecture/
34. Desarrollo Web. [En línea] [Citado el: 5 de noviembre de 2012.]
http://www.desarrolloweb.com/actualidad/firefox-16-llega-beta-novedades-7401.html
35. Pruebas de software. [En línea] [Citado el: 11 de noviembre de 2012.]
(http://www.elguille.info/Clipper/probando.htm)
36. Márquez Alpízar, Yaimí. Valdés Echavarría, Yenni. 2008. Procedimiento general de pruebas de Caja Blanca aplicando la técnica del Camino Básico. 2008.

Bibliografía

1. García Valcárcel, Ignacio. CRM: gestión de la relación con los clientes. 2012. págs. 19-50.
2. CRM. [En línea] [Citado el: 25 de octubre de 2012.]
http://www.crm.es/
3. OpenERP. [En línea] [Citado el: 25 de octubre de 2012.]
http://www.openerp.com/
4. OpenERP Spain. [En línea] [Citado el: 25 de octubre de 2012.]
http://www.openerpspain.com/
5. SAP [En línea] [Citado el: 28 de octubre de 2012.]
http://www.sap.com/spain/solutions/business-suite/crm/index.epx
6. Terrasoft. [En línea] [Citado el: 28 de octubre de 2012.]
http://www.terrasoft.es/platform/
7. Complusoft. [En línea] [Citado el: 7 de noviembre de 2012.]
http://www.complusoft.es/en/soluciones/open-erp
8. Subversion. [En línea] [Citado el: 11 de noviembre de 2012.]
http://subversion.apache.org/
9. Scrib [Online] 2012. [En línea] [Citado el: 11 de noviembre de 2012.]
http://es.scribd.com/doc/86921799/XML
10. Msdn [Online] 2012. [En línea] [Citado el: 2 de diciembre de 2012.]
http://msdn.microsoft.com/es-es/library/bb972240.aspx
11. Patrones-GRASP [Online] 2012. [En línea] [Citado el: 2 de diciembre de 2012.]
http://www.practicasdesoftware.com.ar/2011/03/patrones-grasp/
12. Sparx Systems [Online] 2012. [En línea] [Citado el: 5 de diciembre de 2012.]
http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html
13. Programación [Online] 2012. [En línea] [Citado el: 5 de diciembre de 2012.]
http://www.programacion.com/articulo/introduccion_a_uml_181/7
14. Msdn [Online] 2012. [En línea] [Citado el: 5 de diciembre de 2012.]
http://msdn.microsoft.com/es-es/library/dd409390.aspx

Glosario de términos

Bytecodes: Es un código intermedio más abstracto que el código máquina.

Framework: Significado en español marco de trabajo.

GRASP: Patrones generales de software para asignación de responsabilidades por sus siglas en inglés General Responsibility Assignment Software Patterns.

IDE: Entorno de desarrollo integrado, acrónimo en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación.

Licencia BSD: es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution)

MVC: Patrón arquitectónico Modelo Vista Controlador.

Notación identada: Significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores para separarlo del texto adyacente.

Prospecto: Es aquel consumidor o empresa que tiene interés en comprar un producto o servicio.

TIC: Tecnologías de la Información y las Comunicaciones.

Tipado dinámico: Se utiliza este término para las variables que pueden tomar valores de distinto tipo en distintos momentos.

Anexo 1

Universidad de las Ciencias Informáticas

SISTEMA DE ADMINISTRACIÓN DE RELACIONES CON EL CLIENTE

Por el presente damos constancia de que todas las funcionalidades de los procesos Atención al cliente y Soporte para el Sistema de Administración de Relaciones con el Cliente fueron implementadas cumpliendo todas las necesidades plasmadas en los requisitos funcionales.

Estos procesos pertenecientes al componente Servicios concebido para garantizar la gestión de los servicios, está desarrollado sobre tecnologías libres, por tanto responde a las políticas de software libre dictadas en la Universidad de las Ciencias Informáticas. Durante la ejecución de la tesis con título "Implementación de los procesos Atención al cliente y Soporte dentro del Sistema de Administración de Relaciones con el Cliente" se actualizaron los artefactos ya realizados que sufrieron algún tipo de modificación.

La utilización de este componente permite la gestión de las incidencias, de los servicios que pueden ser reparaciones, devoluciones y asistencias técnicas; además permite gestionar los empleados de la empresa y un grupo de entidades del negocio que pueden ser cargo, departamento y tipo de incidencia.

Y para constancia de ello firman el presente aval a los 4 días del mes de Junio del 2013:

Centro de Informatización de la Gestión de Entidades
CEIGE

Ing. Olga Yansbel Rojas Grass
Jefe de línea

Ing. Erich Mario Gómez Pérez
Jefe de departamento

Figura 17: Carta de aceptación