Universidad de las Ciencias Informáticas

Facultad 3

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

Diseño e Implementación de la versión 2.0 del instalador de Cedrux.

Autor: Rolando Cotilla Díaz

Tutores: Ing. Yoan Arlet Carrascoso Puebla

Ing. Omar Antonio Díaz Peña

Ciudad de la Habana, __ de _____ del 2011.

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

_____

Rolando Cotilla Díaz

Autor

_____

Ing. Yoan A. Carrascoso Puebla

Tutor

_____

Ing. Omar A. Díaz Peña

Tutor

Agradecimientos

A la Revolución que me dio la oportunidad de estudiar y superarme día a día en esta magnífica universidad.

A mi mamá por darme su incondicional cariño.

A mis hermanas por darme los mejores sobrinos del mundo.

A mi papá por estar ahí cuando lo necesité.

A Elisa que me soporto durante mucho tiempo y me regalo casi 5 años de su vida.

A mi familia por apoyarme y darme aliento para seguir.

A mis amigos Elton, Sandy, Driggs y Álvaro, por molestarme cada vez que tenían un chance.

A mis amigas Ariadna, Leydiana, Sisley, Danarys, Gretell y Aliankis, que me soportaron mis superpesadeses.

A todas las amistades que me impusieron durante estos 5 años que le agradezco a la vida habérmelas puesto en mi camino.

A mis tutores por ayudarme en todo lo que estaba a su alcance y sin los cuales no hubiera podido hacer mi tesis.

Al tribunal que desbarató este documento en innumerables ocasiones.

A mi oponente que me apoyó y asesoró.

A todos…

Dedicatoria

A mis abuelos Rolando y Amada que no me vieron graduado.

A mis padres por su guía constante en la vida.

A mis hermanas por ser mis más grandes amores.

A mi familia por su unión y armonía.

Resumen

En el mundo del software, existen sistemas que permiten el ahorro de tiempo y recursos en cuanto a la gestión de la información. Ninguno de los sistemas desarrollados en el mundo, se rige por los requerimientos necesarios al desempeño financiero en Cuba, por tanto a la Universidad de las Ciencias Informáticas, se le ha dado la tarea de desarrollar un Sistema de Gestión que cumpla con los requerimientos de la economía cubana, de ahí surge el Sistema Integral de Gestión Empresarial Cedrux, el cual cuenta actualmente con un instalador que no cumple con las necesidades actuales del sistema.

El instalador, el cual debe ejecutarse mediante la web, debe permitir llevar a cabo el proceso de instalación de forma rápida, sencilla, eficiente y personalizada. Este proceso se realizará de forma transparente para el usuario y de tal manera que este no necesite tener amplios conocimientos sobre informática para instalar el sistema. En el presente documento, se describen detalladamente los principales instaladores existentes, realizando comparaciones en busca de las mejores propuestas para desarrollar la herramienta con la mayor calidad posible, además se detallan las tecnologías y herramientas utilizadas para el desarrollo del nuevo instalador. Al final del documento se especifican las pruebas realizadas para validar la solución, así como las métricas aplicadas para saber si se llevó a cabo un buen diseño de la aplicación.

Palabras claves:

Software, Cedrux, Instalador web.

Tabla de Contenido

Tabla de Contenido

Tabla de Contenido

Introducción

En esta era de la información, las empresas manejan gran cantidad de flujo de datos que requieren un alto consumo de tiempo y recursos para su gestión. Dado este problema se han desarrollado técnicas y métodos para hacer más eficiente el flujo de dicha información. En el campo de las tecnologías se han creado herramientas que facilitan el manejo de datos que minimizan el tiempo de gestión y recursos, maximizando la cantidad de datos procesados. Los sistemas de Planeación de Recursos Empresariales (ERP por sus siglas en inglés) son sistemas de gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa (1). Estos sistemas integrales de gestión están compuestos por diferentes subsistemas o módulos integrados en una aplicación, entre ellas: logística, contabilidad, inventario, planificación y otros que tributan a la automatización y mejora de los procesos de una empresa según sea su objeto social. Los sistemas ERP brindan la solución práctica a estas necesidades existentes en el mundo de las entidades logrando unificar las áreas de la entidad en un solo sistema.

En Cuba se utilizan varios sistemas de gestión empresariales, los cuales no se rigen por los requerimientos necesarios al desempeño financiero del país e impiden la estandarización de la gestión de los procesos en todas las entidades. Por cuestiones estratégicas, la máxima dirección del país le ha dado la tarea a la Universidad de las Ciencias Informáticas (UCI por sus siglas), en coordinación con el Ministerio de Finanzas y Precios y en coordinación con las entidades TEICO Villa Clara y DESOFT, el desarrollo de un sistema de gestión empresarial llamado Cedrux, el cual se prevé que posteriormente se convierta en un ERP, que responda a las necesidades de las empresas cubanas, el cual se implementa sobre plataformas de Software Libre, lo que lo ubica a un paso por delante respecto a otros sistemas en explotación en el país permitiendo el ahorro de millones de dólares en concepto de licencia.

Esta herramienta se ajustaría a las necesidades de las empresas cubanas permitiendo lograr de forma parcial una solución en las áreas de las entidades, optimizando los procesos y ganando en eficiencia organizacional. En su versión inicial, se le están aplicando pruebas pilotos en empresas que tienen diferentes características para determinar su operatividad y confiabilidad.

Un software tiene varios requisitos para ser reconocido, tales como: la fiabilidad, rapidez, que sea robusto, donde muchos de estos requisitos pueden ser afectados en la instalación, siendo este el

momento crítico para su configuración. El instalador actual del sistema cuenta con un proceso poco personalizable que no tiene la máxima calidad en cuanto a las necesidades de configuración del mismo y no incluye las últimas modificaciones hechas por los desarrolladores. El mismo presenta varias dificultades como: no mostrar el progreso real de la instalación, no permite la selección de módulos a instalar de manera personalizada, no presenta implementado el Rollback, no reconoce roles ya creados con antelación en la base de datos, no prepara el mecanismo para la configuración posterior a la instalación, emplea el uso de las PDO (PHP Data Objects) como capa de acceso a datos lo cual pone en riesgo la seguridad de la información, necesita la selección del Sistema Operativo que es completamente innecesario ya que el sistema va a ser instalado únicamente en servidores basados en plataforma libre, específicamente Ubuntu 8.4 o superior y la agregación de un nuevo script implica realizar cambios significativos en el código de fuente.

A partir de esta situación problémica sale a relucir como **problema a resolver**: ¿Cómo mejorar el funcionamiento de la versión 1.0 del instalador del sistema Cedrux, de acuerdo a las no conformidades detectadas en las entidades en piloto para mejorar la instalación y configuración inicial del mismo?

El **objeto de estudio** se enmarcará en los diferentes instaladores de aplicaciones Web que existen, centrándose en el Diseño e implementación de un instalador que utilice tecnología Web para sistema Cedrux como **campo de acción**.

Para dar solución a lo planteado anteriormente se da como **objetivo general** la realización del diseño y la implementación de las nuevas funcionalidades del instalador del sistema Cedrux atendiendo a los nuevos requerimientos detectados en el proceso de despliegue en las entidades.

Definiendo como **objetivos específicos**:

- ✓ Realizar la fundamentación teórica de la investigación mediante la evaluación del Marco Teórico.
- ✓ Modelar una solución de diseño que cumpla con los requerimientos especificados en el análisis.
- ✓ Implementar los componentes del subsistema.
- ✓ Validar las funcionalidades.

Para dar cumplimiento a estos objetivos se establecieron las siguientes **tareas investigativas**:

- ✓ Análisis de la evolución de los diferentes instaladores.
- ✓ Sistematización del comportamiento de la versión anterior del instalador.
- ✓ Caracterización de los patrones de diseño, estándares de codificación, herramientas y tecnologías definidas por la Línea de Arquitectura.
- ✓ Diseño de la lógica de negocio de los componentes propuestos para el correcto funcionamiento del sistema.
- ✓ Implementación del diseño realizado.
- ✓ Resolución de no conformidades de la primera versión detectadas por el grupo de calidad.
- ✓ Implementación de pruebas para la evaluación de los componentes propuestos.

Con el diseño y la implementación de las nuevas funcionalidades en el instalador del sistema de gestión empresarial Cedrux, se obtendrán mejoras en el funcionamiento del mismo, siendo esta **la idea a defender** del presente proyecto.

Para poder dar cumplimiento a las distintas tareas investigativas se pusieron en práctica los siguientes **métodos de investigación**:

Métodos Teóricos

- ✓ **Histórico – Lógico:** Para realizar un estudio de las tendencias históricas y actuales en el desarrollo de los diferentes instaladores.
- ✓ **Analítico – Sintético:** Para realizar un análisis de la información empleada para la investigación.
- ✓ **Modelación:** Para modelar los nuevos requerimientos funcionales que se proponen en la solución, su relación con los procesos actuales y el diseño de los nuevos componentes visuales, las entidades relacionales y sus características.

Métodos Empíricos

- ✓ **Observación:** Para determinar cómo quedaría el nuevo desarrollo y su influencia en la integración con el actual sistema.
- ✓ **Experimento:** Para establecer las diferentes pruebas de calidad con el propósito de determinar la fiabilidad del sistema y el mejoramiento de la usabilidad del mismo, a través de la detección de errores.

El presente trabajo de diploma cuenta con Introducción, Tres Capítulos, Conclusiones, Recomendaciones, Bibliografía y Glosario de Términos.

Capítulo 1: "Fundamentación Teórica". Este capítulo aborda el estudio del estado del arte, el cual nos da un breve acercamiento a los principales conceptos asociados al dominio del problema y que son tratados a lo largo del documento. Además se describe el campo de acción, así como una breve descripción de los actuales instaladores que realizan el proceso de instalación de sistemas similares. Finalmente se justifica el uso de las diferentes tecnologías, herramientas y técnicas de modelado empleadas para el desarrollo de la aplicación.

Capítulo 2: "Diseño e Implementación de la solución". En este capítulo se reflejan los principales diagramas generados para el desarrollo eficiente del sistema. Se definen los nuevos componentes, estructuras de datos, se valoran los estándares de codificación establecidos y se exponen las nuevas clases implementadas.

Capítulo 3: "Validación de la Solución". Se evalúan los diseños propuestos a partir de las métricas de diseño y técnicas de prueba para la implementación. Se presentan los criterios de aplicación de las pruebas de calidad, así como un análisis de los métodos aplicados para la evaluación de satisfacción de los clientes.

Cada capítulo finaliza con conclusiones que determinan el conocimiento que se creó durante el desarrollo de estos.

Capítulo 1: Fundamentación Teórica

1. Introducción

Este capítulo se centra en el estudio de los distintos tipos de instaladores de aplicaciones Web que existen en el mundo para poder conocer las tendencias reales que hay en la actualidad referente al tema, aunque es aún novedosa la práctica de estos debido a que aún se mantiene un alto grado de preferencia a las aplicaciones de escritorio que ofrecen mayor velocidad y aceptación por parte de los clientes. Comenzará el capítulo dando un resumen de los principales conceptos que hay que dominar para poder darle seguimiento al documento. Se darán a conocer algunas de las características de los instaladores que se tomaron de muestra, como es el funcionamiento y las dependencias funcionales que exigen cada uno de estos. Tener conocimiento de los tipos de ficheros utilizados para la instalación en los Sistemas Operativos más populares es de vital importancia para poder emitir un mejor criterio respecto este tema. Se da un enfoque de las tecnologías utilizadas y las herramientas que sirvieron de apoyo durante el desarrollo del instalador. Se hace presentación de las diferentes metodologías existentes para determinar cuál es la que más se apega a las necesidades del proyecto y las tendencias de las tecnologías actuales para este entorno. Se describen las tecnologías enfocándose principalmente en las de desarrollo libre, analizando el Marco de Trabajo utilizado que es creado para el lenguaje de programación PHP. Para concluir se dan los resultados de la investigación dando una propuesta de solución coherente y acertada.

1.1. Definiciones.

1.1.1. Instalador.

Para definir que es un instalador, fue necesario el estudio de la documentación referente al tema proveniente de varias fuentes, entre ellas el libro Beyond Software Architecture del autor Luke Hohmann, resumiendo que un instalador es una herramienta que le permite al usuario final dar uso del software adicionándolo a la lista de programas que presenta el sistema operativo en el cual trabaje este. Generalmente estos programas están conformados por un conjunto de archivos que se encuentran en un directorio específico para darle funcionamiento al sistema completo. Los programas que se instalan en la plataforma deben cumplir con las reglas establecidas para que sea posible su funcionamiento dentro del sistema. Las dependencias se establecen según las necesidades que tenga

la herramienta para que sea posible su uso, estas pueden ser una o varias que a su vez pudieran estar relacionadas.

La inclusión de un programa específico se puede hacer a mano, pero al ser complejas y laboriosas conllevaría más tiempo y recursos lograr que el Sistema Operativo (SO) lo reconozca, señalando como el más complejo el SO Windows que para que cualquier programa trabaje en coordinación con todo el sistema es necesario que se encuentre en el Registro del SO. En estos momentos es cuando es notable la necesidad de un instalador, un programa que realiza esas tareas de manera automática y resulta ser totalmente transparente para el usuario.

Los pasos para la instalación deben ser sencillos y nada complicados. En el proceso, deben estar señaladas todas las direcciones que se establecen por defecto para la colocación de los archivos del software que se está instalando, para así no depender de un usuario que tenga pocos conocimiento de la estructura del sistema defina dicha ubicación, siempre contando con la posibilidad de que el usuario tenga la suficiente experiencia trabajando con su sistema y pueda cambiar el destino de la misma si es que se siente capacitado para esto. La secuencia de pasos debe ser clara y precisa, aclarando cualquier toma de decisiones que pudiera ocasionar confusión al cliente. Los instaladores generalmente implementan una selección típica o estándar de configuración para los usuarios que no suelen trabajar con el programa, también cuentan con la posibilidad de configuración avanzada para los usuarios de más experiencia.

Al terminar de instalarse el programa, de manera general, independientemente de la plataforma, se relacionan los archivos que pueden abrir dichas herramientas. Conservar los instaladores es más que una opción una necesidad ya que de esta manera es posible reinstalar el programa si en algún momento este se daña y es necesario reinstalarlo.

1.1.2. Tipos de ficheros de instalación más utilizados.

1.1.2.1. Bash.

Bash es un intérprete de órdenes para sistemas basados en Shell de Unix y compatible con POSIX (Portable Operating System Interface for Unix). Fue escrito para el proyecto GNU (GNU is Not Unix) y es el intérprete por defecto de la mayoría de las distribuciones GNU/Linux (2). Este tipo de archivo tiene varios propósitos en el sistema donde es interpretado, uno de ellos es la posibilidad de crear un script que permita instalar un programa determinado.

1.1.2.2. EXE.

Esta es la extensión de fichero que denota un fichero ejecutable. Los ficheros ejecutables o ficheros exe (de la abreviación del inglés executable) realizan una operación determinada en su ordenador cuando se activan. Estos ficheros pueden descomprimirse para abrir una serie de documentos, iniciar una aplicación o puede instalar un programa cuando se le da la orden para ello. Es útil conocer que los ficheros exe sólo funcionarán en los ordenadores equipados con DOS, Microsoft Windows, OS/2 y ReactOS; en cualquier otro sistema que no pertenezca a la familia de sistemas operativos de estos son totalmente inútiles. (3)

Hay cuatro formatos de archivo tipo exe principales:

- ✓ **Ejecutables en DOS:** son los menos complejos y pueden funcionar en todos los sistemas operativos DOS y Windows.
- ✓ **Ejecutables de 16-bit:** no pueden funcionar en DOS pero pueden funcionar en todos los sistemas operativos de Windows.
- ✓ **Ejecutables de 32-bit:** pueden funcionar solamente en Windows 95 y superior.
- ✓ **Ejecutables de 64-bit:** pueden funcionar solamente en las ediciones de 64-bit de Microsoft Windows, tales como la edición 64-Bit de Windows XP o la edición 64-Bit de Windows Server 2003.

1.1.2.3. MSI.

Un archivo msi es un archivo instalador que fue desarrollado por Microsoft Windows usado para instalar aplicaciones en sus sistemas operativos. El nombre de la extensión msi proviene de Microsoft Installer (Instalador de Microsoft). Al ejecutarse el archivo msi, éste se abre con la herramienta Windows Installer y no tiene un editor asociado (4). La principal desventaja de este tipo de instalador es que solo funciona en sistemas que sean desarrollados por la misma compañía.

1.2. Instaladores Web conocidos.

Los sistemas desarrollados en plataformas Web, tienen marcadas diferencias que lo hacen de mucho beneficio tanto para las empresas que lo utilizan, como para los usuarios finales del mismo. Hoy día las

empresas han evolucionado, desde el punto de vista informático, para hacer más fácil y eficiente tareas que antes llevaban mucho tiempo. Los sistemas web ascienden un escalón más en cuanto la administración de la información y la facilidad de acceso a la misma para los empleados de una empresa determinada por sus características propias. El acceso a este sistema se realiza por medio de un navegador, evitando instalar una aplicación en cada una de las estaciones de trabajo de la empresa siendo esta una de sus mayores ventajas, en algunos casos permite el acceso a la información en una estación de trabajo fuera del dominio laboral dependiendo de los intereses empresariales.

1.3.1. Instalador de Mantis.

Mantis es un sistema de registro y control de Bugs basado en tecnología Web. El acceso a la aplicación es mediante un navegador Web cualquiera, no presenta ningún tipo de dependencia en este aspecto. Este sistema crea y mantiene un sistema de control de Bugs, y se encuentra diseñada tal manera que se modifica con cierta facilidad, es personalizable y fácilmente actualizable. Mantis se despliega sobre el servidor Apache y utiliza diferentes gestores de base de datos como MySQL, MS SQL y PostgreSQL para la gestión de información, aunque presenta un mejor funcionamiento con el primero el cual es el recomendado por la empresa que lo desarrolló. Puede ser instalado en sistemas operativos Windows, MacOS o de tipo Unix. Este sistema ha sido desarrollado bajo licencia GNU General Public License (GPL) (5). Cuando se cumplen los requisitos funcionales, el sistema está listo para ser instalado y configurado mediante la web. Se accede mediante una URL la cual tiene la estructura siguiente: http://<dominio>:<puerto>/mantis/admin/install.php. En la página que se definen los siguientes datos:

- ✓ El tipo de base de datos.
- ✓ El nombre de la maquina que contiene la base de datos concatenado con el puerto (<dominio>:<puerto>).
- ✓ El nombre y contraseña del usuario de acceso a la base de datos que se ha creado en la fase de instalación de la base de datos.
- ✓ El nombre de la base de datos.
- ✓ El nombre y contraseña del usuario administrador que usara Mantis para acceder a la base de datos.

Después de pedido los datos se procede a realizar la instalación la cual es bastante rápida.

1.3.2. BitNami.

Este es un programa multiplataforma que crea un conjunto de paquetes que contienen todo lo necesario para darle inicio a la instalación de la aplicación Web. Este sistema está pensado para usuarios que no tienen mucho conocimiento de lo que son los programas de servicios tanto Web como de bases de datos. El mismo presenta licencia GPL y utiliza programas que se encuentran bajo esta misma licencia como Apache y la versión libre de MySQL. Los programas que instala son independientes a los que se encuentren instalados ya en el sistema, por lo tanto no interfiere en nada con estos. (6)

Con el BitNami el usuario no tiene que preocuparse por descargar ningún programa ni tener que aprender a configurar ninguno, todo esto viene incluido dentro de las funcionalidades del propio programa. El proceso de instalación realizado por esta herramienta se realiza de manera local.

1.3.3. Sistemas de Gestión de Contenidos.

Los Sistemas de Gestión de Contenidos (CMS por sus siglas en inglés) permiten la creación y administración de contenidos principalmente en páginas Web (7). Estos ejecutan el proceso de instalación mediante la web, permitiéndole al usuario realizar las configuraciones necesarias durante éste proceso. Para poder ejecutar este instalador es necesario tener montado un programa de servicios Web, en la mayoría de los casos preferentemente el Apache y un gestor de bases de datos, siendo el más utilizado MySQL. Después de esto se copian los archivos del instalador en la carpeta que indica la raíz del servidor Web y mediante el navegador, después de haber iniciado los servicios comienza la instalación. Los sistemas CMS que utilizan el gestor de base de datos PostgreSQL no permiten una configuración de interfaz tan personalizables como los que utilizan MySQL.

1.3.4. Microsoft Web Plataform Installer.

Microsoft Web Plataform Installer (Web PI) es un programa lanzado por la Microsoft en el 2008 dirigido a desarrolladores, que permite instalar aplicaciones Web en servidores de la misma marca (si se cuenta con Windows Vista RTM, Windows Vista SP1 o Windows Server 2008). Microsoft Web Plataform Installer 3.0 (Web PI) es una herramienta gratuita que facilita la descarga, instalación y la actualización de los últimos componentes de Microsoft Web Plataform, incluido Servicios de Internet

Information Server (IIS), SQL Server Express, .NET Framework y Visual Web Developer (8). Además, instala aplicaciones web muy utilizadas como ASP.NET y PHP que presentan código fuente abierto. El programa presta atención a aplicaciones libres que incluye DotNetNuke, Drupal, Gallery, Graffiti, osCommerce, phpBB, WordPress entre otros. Este programa realiza el proceso de instalación de manera local.

1.3.5. Cedrux 1.0.

Es el instalador anterior desarrollado para el sistema Cedrux, esta implementado en PHP y utiliza para la interfaz gráfica ExtJS en su versión 2.2. Se ejecuta mediante un explorador Web, solo es necesario tener instalados el servidor web Apache y el gestor de datos PostgreSQL, este último puede ser en una estación de trabajo diferente a la que se ejecuta el servidor Web. Cumple con múltiples aspectos para la instalación del sistema, ya que en el momento que surgió cumplía con las necesidades requeridas para el despliegue, tales como instalar el sistema independientemente del sistema operativo, se modificaron los scripts para que pudiera ser multiplataforma, muestra un ambiente amigable y configurable. Actualmente este instalador no cuenta con las últimas actualizaciones que se han realizado en el sistema y tiene algunas no conformidades que lo han hecho obsoleto dificultando que el proceso de instalación sea en un ambiente sencillo y lo más configurable posible.

1.4. Valoración sobre las soluciones existentes.

Después de haber realizado un estudio sobre los principales instaladores existentes y analizando cada una de sus funcionalidades, ventajas y desventajas, se observaron que estos no se acoplan a todas las necesidades que exige la instalación del sistema Cedrux en la actualidad. Los ficheros de instalación EXE, Bash y MSI realizan el proceso de instalación de forma local y cada uno de esos se ejecuta en sistemas operativos específicos, por lo que no cumplen con los aspectos necesarios para instalar el sistema Cedrux que consiste en que este se realice mediante la web y sea independiente a la plataforma. Los instaladores de la herramientas Mantis y los CMS, cumplen con la necesidad de ser instalados mediante la Web y el ser multiplataforma, pero ambos usan como gestor de base de datos principal MySQL, este gestor a pesar de tener una versión libre, la compañía que lo creó solo brinda soporte a aquellos que compren la versión comercial y teniendo en cuenta que Cedrux es un sistema que será desarrollado sobre tecnologías libres entonces no es posible utilizar esos sistemas como base del instalador, a todo esto se le agrega que estos instaladores tienen objetivos específicos, que en caso de sobrescribir su código, se estuvieran haciendo cambios significativos en su base y en algún

momento pudieran traer errores en las funciones del sistema Cedrux. La utilización del BitNami no es posible ya que incumple también con un requisito fundamental del sistema, que sea instalable mediante la Web. El instalador desarrollado por Microsoft, Microsoft Web Plataform Installer, no es tan siquiera una opción por el hecho de que solo es posible la instalación en un servidor que utilice Windows y además es necesario realizar la instalación de manera local. La versión inicial del sistema Cedrux es el instalador que más se acopla a las necesidades aunque le faltan funcionalidades que son necesarias para la nueva versión.

1.5. Modelo de desarrollo.

Para lograr el correcto desarrollo del instalador es necesario tener en cuenta las pautas arquitectónicas establecidas por la línea de arquitectura y la dirección del proyecto, las cuales proponen un modelo estandarizado, así como la definición clara de cada uno de los roles involucrados en el proceso de desarrollo de la solución. Este modelo está orientado a la reutilización de componentes parametrizables, donde se simplifican las actividades y se eleva el nivel de especialización de los involucrados. Proporciona una guía para regir el proceso de desarrollo de software tecnológico, centrado en la arquitectura.

Con el objetivo de producir software de alta calidad, basado en la participación directa del cliente en colaboración con el grupo de desarrollo para lograr la salida de un producto eficiente y que cumpla con los requerimientos funcionales y se adapte a las necesidades reales del sistema económico del país, el modelo implantado presenta las siguientes características:

Centrado en la arquitectura: La arquitectura guía los casos de uso y determina la línea base y los elementos del software. Debe estar estructurada para dar soporte a los requerimientos del sistema y permita nuevos cambios cada vez que se requiera. Diseña y controla la integración de los componentes, dando un nivel de independencia entre ellos para su implementación individual y acoplamiento posterior, además de determinar las prioridades en el desarrollo.

Orientado a componentes: Las iteraciones son realizadas dado el peso arquitectónico de los componentes en los cuales se descompone el sistema. Estos representan una abstracción de los procesos de la lógica del negocio y poseen interfaces que permiten la fácil integración y comunicación entre ellos.

Iterativo e incremental: El equipo de arquitectura, la dirección del proyecto y los clientes determinan las iteraciones así como los objetivos a cumplir por cada una de ellas. En cada iteración se determinan los riesgos, los componentes a desarrollar y su integración con el sistema, permitiendo el desarrollo incremental del producto.

Ágil y adaptable al cambio: El desarrollo de las partes formaliza las características principales de la solución, priorizando los talleres y las comunicaciones entre las personas. Los clientes y funcionales están involucrados en el proyecto y poseen parte de la responsabilidad del éxito del mismo. Los cambios son conciliados semanalmente, discutidos y aprobados.

1.6. Patrones de diseño.

Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. Representa un esquema o micro arquitectura que supone una solución a problemas (dominios de aplicación) semejantes; una estructura común que tienen aplicaciones semejantes (9).

Los patrones de diseño emplean un conjunto de buenas prácticas que facilitan el trabajo, definen una estructura de clases que da respuesta a uno o varios problemas en particular y presentan la ventaja de que son fáciles de comprender, además presentan características genéricas al no depender del lenguaje. Lo complejo es decidir cuál usar ya que presentan diferentes soluciones, ya sea a través del empleo de uno u otro, o la combinación de varios. De ahí la importancia de conocer y estudiar los diferentes patrones que existen para poder determinar su uso.

A continuación se explicarán los patrones de diseño utilizados para el desarrollo de esta herramienta.

1.6.1. Patrones GRASP

El patrón GRASP (General Responsability Assignment Software Patterns), no compite con los patrones de diseño, los patrones de GRASP, nos guían para ayudarnos a encontrar los patrones de diseño (que son más concretos). (10)

- **Experto:** Asigna responsabilidades a la clase que se encarga de gestionar la información referente a un objeto determinado y es capaz de cumplir con la actividad asignada conservando el encapsulamiento y promoviendo clases sencillas y fáciles de comprender y mantener.

- **Creador:** Tiene la responsabilidad de asignar la creación de una clase a otra. Esta asignación solo puede realizarse cuando agrega o contiene una instancia de ella, contiene instancias de sus objetos y contiene los datos de inicialización que serán transmitidos a la nueva clase.

- **Controlador:** Gestiona los eventos de entrada generado por actores externos, asignando responsabilidades a otras clases de manera independiente y que permita el bajo acoplamiento y la alta cohesión.

- **Bajo acoplamiento:** Es la medida en que cada una de las clases realiza actividades independientes, además de poseer un conocimiento de las actividades que realizan las otras clases del sistema permitiendo la reutilización.

- **Alta cohesión:** La cohesión es la relación que existe entre las clases y la medida en que estas realizan labores únicas dentro del sistema, pero que están estrechamente relacionadas entre ellas.

1.6.2. Patrones Gang of Four

Los patrones de diseño Gang of Four (GoF) que definen una estructura de clases que tratan una situación en particular. A continuación se describe el patrón referente a las interfaces.

- **Fachada:** Proporciona una interfaz única que simplifica los servicios generales del sistema, definiendo un comportamiento independiente de donde se vaya a utilizar, siendo esta más fácil de utilizar.

1.7. Tecnologías utilizadas.

Para desarrollar la solución se utilizó el Framework[1] Sauxe, que contiene un conjunto de componentes reutilizables y provee una estructura genérica logrando una mayor estandarización, flexibilidad, integración y agilidad den el proceso de desarrollo; integra un conjunto de tecnologías libres para cubrir un conjunto de escenarios arquitectónicos.

[1] Estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Este marco de trabajo hace uso del lenguaje de programación PHP versión 5.2.4 para la capa de negocio y en la capa de presentación utiliza ExtJS versión 2.2, una de las mejores librerías JavaScript que existen y HTML, finalmente en la capa de acceso a datos el lenguaje de consulta Doctrine Query Language (DQL) que implementa Doctrine versión 1.9.7. Además utiliza una arquitectura en capas aunque en una de sus capas contenga un modelo vista controlador. En la capa de negocio específicamente utiliza Zend Framework versión 1.2.1 por su flexibilidad en la integración con el resto de las capas de la arquitectura, otra de las ventajas que brinda es la sencillez a la hora de extender sus componentes, de estas extensiones surge ZendExt Framework desarrollada por el CEIGE y el Centro de Desarrollo y Asimilación de Tecnologías del Centro de Compatibilización Integración y Desarrollo del software para las FAR (UCID) con el objetivo de crear un marco de trabajo extensible y configurable centrando el desarrollo de las aplicaciones en la lógica del negocio y en las interfaces de usuario y alejando a los programadores de los detalles arquitectónicos, con soporte para entornos multientidad y para una arquitectura de sistema orientada a componentes.

Se decidió desarrollar para Sauxe el Sistema de Gestión Integral de Seguridad (Acaxia) que gestiona la seguridad en un entorno de varias aplicaciones. En todos los procesos que se gestionan dentro de una estructura se hace necesario que la información se registre y recupere por estructuras y dentro de ellas por usuarios, para lograr este objetivo surge la necesidad de desarrollar el Sistema de Estructura y Composición el cual permite gestionar las estructuras de un país, una entidad u otra estructura de forma dinámica. Sauxe cuenta con un componente llamado Traza que permite gestionar todas las acciones que se ejecutan dentro de un sistema. Con esta información no solo se puede identificar ataques, puesto que permite evaluar el rendimiento de cada acción para reimplementarla en caso que el tiempo de respuesta sea muy grande. En aras de estandarizar la interfaz de usuario, la gestión de la información y que todos los sistemas tuvieran un único punto de acceso se hizo necesario desarrollar un componente llamado Portal, este componente se encarga de de interactuar con Acaxia para realizar la autenticación de el usuario, luego busca los recursos a los cuales tiene acceso y se lo muestra en un menú. De esta forma los desarrolladores no tienen que preocuparse por cómo mostrarle la información al usuario ni por el qué se le va a mostrar, pues esta gestión queda a nivel central agilizando el proceso de desarrollo permitiendo que usuario solo tenga que autenticarse una vez para acceder a todos los sistemas a los cuales tiene acceso. (11) Para cubrir este grupo de requisitos tecnológicos se utilizaron las siguientes tecnologías:

1.7.1. ExtJS 2.2.

Este Framework provee a los programadores de una potente y amigable herramienta que da la posibilidad de manejar funcionalidades muy potentes sobre formularios, estilos, peticiones Ajax (Asynchronous JavaScript And XML), lectores de respuestas tipo JSON, creación de componentes propios, manejo del estándar DOM, DHTML entre otros con el lenguaje JavaScript. Una de las desventajas de este Framework es su peso, debido a que el cliente debe descargarla completamente si es que la adquiere por primera vez, pero en compensación a esto permite que el sistema sea diseñado utilizando llamadas Ajax que lo hacen más veloz en el momento en que este completamente descargado. Permite el uso de clases en JavaScript, una funcionalidad que no es posible con JavaScript tradicional, al ser un lenguaje funcional, pero ExtJS permite simular ciertas funcionalidades de la orientación a objetos como es la herencia. (12) Es soportado por los navegadores web más usados como Internet Explorer, Firefox, Safari y Opera y el uso de esta versión es totalmente libre, incluyendo su código.

1.7.2. Doctrine 1.2.1.

Es un potente y completo sistema ORM (Object Relational Mapper) para PHP 5.x con un DBAL (Database Abstraction Layer) incorporado. Una de sus principales características es la posibilidad de escribir consultas de base de datos en un dialecto orientado a objetos o DQL, inspirado en Hibernates HQL. Esto proporciona a los desarrolladores una poderosa alternativa al lenguaje de consulta estructurado (SQL por sus siglas en inglés) que mantiene la flexibilidad sin necesidad de duplicar código innecesario. (13)

Otra de las características de Doctrine es el bajo nivel de configuración que se necesita para comenzar un proyecto. Doctrine puede generar clases a partir de una base de datos creada, y el programador puede especificar relaciones y agregar funcionalidades comunes para las clases generadas. No hay necesidad de generar o mantener esquemas complejos en XML.

1.7.3. Lenguaje de programación PHP 5.2.4.

El PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características especificas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas web. (14)

Este lenguaje además de ser potente y rápido, posee una corta curva de aprendizaje. Soporta múltiples bases de datos como PostgreSQL, MySQL y Oracle. Es reconocido como software libre y es independiente a la plataforma donde se corra, más bien depende del servidor Web donde se encuentra corriendo los scripts.

1.7.4. Zend Framework 1.2.1.

Este es un framework para desarrollo de aplicaciones Web y servicios Web con PHP que brinda soluciones para construir sitios web robustos y seguros. Además es Open Source y trabaja con PHP 5. Zend Framework es desarrollado por la empresa Zend que es un conocedor y apoya comercialmente el lenguaje de programación PHP. (15)

Este framework trabaja utilizando el patrón de arquitectura MVC (Model View Controller) mejorando el desarrollo del software. El Marco de Zend también incluye objetos de diferentes bases de datos, por lo que es extremadamente simple para consultar una base de datos determinada sin tener que escribir complicadas consultas SQL. Posee una muy extensa y útil documentación y pruebas de alta calidad.

1.7.5. ZendExt Framework 1.2.1.

Está diseñado para PHP y utiliza el patrón MVC. Es fácilmente integrable a las aplicaciones, debido a que contiene diferentes clases de gran utilidad como la búsqueda dinámica de ficheros a incluir o utilizar.

ZendExt cuenta con un importante mecanismo de manejo de controladores y vistas, por lo que se propone tenerlo en cuenta para el diseño de estos dos componentes de la arquitectura. Posee un controlador vertical para el control de las acciones realizadas por las vistas hacia el controlador y un motor de reglas para las validaciones en el servidor. Tiene incorporado el ORM Doctrine Framework para trabajo en la capa de abstracción a base de datos y el ExtJS Framework para el desarrollo de las vistas.

El framework garantiza la comunicación entre diferentes módulos y componentes mediante un mecanismo que permite a otros módulos o componentes realizar acciones de control que se requieran para el conjunto de sucesos que tengan que ocurrir. Este mecanismo se basa en el patrón Inversión de Control.

1.7.6. Lenguaje de modelado UML 2.1.

UML son las siglas de Unified Modeling Language (Lenguaje Unificado de Modelado), notación con que se construyen sistemas por medio de conceptos orientados a objetos. (16) Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama. UML cuenta con varios tipos de diagramas que son la representación gráfica de un conjunto de elementos y sus relaciones que visualizan el sistema desde diferentes perspectivas.

1.8. Herramientas Utilizadas.

1.8.1. NetBeans IDE 6.9.

NetBeans es un entorno de desarrollo integrado (IDE por sus siglas en inglés), modular, de base estándar, escrito en el lenguaje de programación Java. Se puede instalar en Windows, MacOS, Linux y Solaris. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general para compilar cualquier tipo de aplicación. Con este IDE se pueden crear aplicaciones que sean soportadas por la plataforma Java como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++. (17)

Este proyecto cuenta con el apoyo de una gran comunidad de desarrolladores que brindan accesorios (plugins) que aumentan la cantidad de las funcionalidades y opciones de la herramienta.

1.8.2. Apache 2.2.

Apache es el servidor Web por excelencia, tiene buenas opciones de configuración, mantiene gran robustez y estabilidad hacen que cada vez millones de servicios aumenten su confianza en este programa. Apache es una muestra de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar. La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia permite darle cualquier uso al código fuente del programa, incluso para crear productos propietarios a partir de este, siempre que se les reconozcas el trabajo realizado por los desarrolladores de la entidad que generó el producto inicial.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. (18)

1.8.3. PostgreSQL 8.3.

PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad, integridad de datos y la corrección. Se ejecuta en todos los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. Es totalmente compatible con ACID (Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.), tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios idiomas). Se incluye la mayoría de SQL: múltiples tipos de datos, incluyendo INTEGER, numérico, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros y la documentación excepcional. (19)

1.8.3.1. Ventajas.

Instalación ilimitada: PostgreSQL permite la instalación en diferentes equipos sin faltar a alguna licencia que impida esto, se debe a que este potente gestor de bases de datos se encuentra bajo licencia GPL la cual permite el uso del programa por cualquier usuario. Con esta licencia no hay costo asociado alguno.

Mejor soporte que los proveedores comerciales: Además de las ofertas de soporte que brinda la compañía, se tiene una importante comunidad de profesionales seguidores y entusiastas de PostgreSQL de los se pueden obtener beneficios de sus contribuciones.

Ahorros considerables en costos de operación: El software está diseñado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento.

Estabilidad y confiabilidad: A diferencia de muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.

Extensible: El código fuente está disponible para todos sin costo. Es posible adaptar el PostgreSQL a las necesidades de cada usuario con un mínimo de esfuerzo, sin costo alguno. Esto es

complementado por la comunidad de profesionales seguidores de PostgreSQL en todo el mundo que también extienden PostgreSQL todos los días.

Multiplataforma: PostgreSQL está disponible para casi cualquier sistema Unix (34 plataformas en la última versión estable) y una versión nativa de Windows está actualmente en estado beta de pruebas.

Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

Herramientas gráficas de diseño y administración de bases de datos: Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin , pgAccess) y para hacer diseño de bases de datos (Tora , Data Architect, pgDesigner).

1.8.3.2. Características técnicas que PostgreSQL ofrece.

- ✓ Cumple completamente con ACID (Atomicity, Consistency, Isolation and Durability).
- ✓ Cumple con ANSI SQL (Instituto Nacional Estadounidense de Estándares).
- ✓ Integridad referencial.
- ✓ Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby
- ✓ Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL
- ✓ Sistema de tipos de datos extensible para proveer tipos de datos definidos por el usuario y rápido desarrollo de nuevos tipos.
- ✓ Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL.

1.8.4. Visual Paradigm 6.4.

Visual Paradigm proporciona un conjunto de productos que facilita a las organizaciones de diseño visual y esquemático, integra y despliega sus aplicaciones de misión crítica de la empresa y sistemas

gestores de datos subyacentes. Además ayuda al equipo de desarrollo de software en la generación del modelado del proceso de desarrollo del software maximizando y acelerando tanto las contribuciones del equipo como las individuales. (20)

Visual Paradigm para Lenguaje Unificado de Modelado (UML por sus siglas en inglés) es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software. El software de modelado UML aumente la rapidez de construcción de aplicaciones que tengan calidad con un menor coste. Permite crear todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar la documentación asociada a los mismos. La herramienta UML CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora en español) también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

1.8.4.1. Características.

- ✓ Soporte de UML versión 2.1
- ✓ Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento
- ✓ Modelado colaborativo con CVS (Concurrent Versioning System) y Subversión.
- ✓ Ingeniería de ida y vuelta
- ✓ Ingeniería inversa - Código a modelo, código a diagrama
- ✓ Ingeniería inversa Java, C++, Esquemas XML, YML,.NET exe/dll, CORBA IDL

1.9. Solución Propuesta.

Debido a la necesidad de darle solución a las no conformidades detectadas en las diferentes entidades pilotos donde se ha desplegado el sistema Cedrux en cuanto al proceso de instalación y el no existir un sistema que cumpla con las exigencias de este, se propone la siguiente solución: Crear un instalador sobre tecnologías libres el cual estará programado en su capa de negocio en PHP siendo este un lenguaje de programación libre y con altas prestaciones. Este constará con varias funcionalidades como instalar el sistema inicialmente, creando la base de datos sobre la cual trabajará Cedrux, ejecutar los scripts asociados a la misma, configurar los archivos de acceso, activar las trazas e iniciando la configuración inicial del sistema. También contará con un módulo que permita seleccionar los subsistemas que serán instalados dependiendo del objeto social de la empresa, para facilitar este

proceso se utilizará interfaz usando la librería de JavaScript ExtJS, la cual brinda un ambiente amigable y de fácil manipulación para el usuario que interactúa con la aplicación. La solución dará como resultado un proceso de instalación sencillo y fácil de adaptar a cada una de las necesidades de la empresa en la que se vaya a desarrollar y todo este proceso se realizará mediante la Web logrando mayor accesibilidad.

1.10. Conclusiones del capítulo.

Debido a la investigación realizada sobre los diferentes tipos de instaladores de sistemas que trabajen en la Web en la actualidad, se ha llegado a la conclusión de que ninguno de los mencionados cumple con los requerimientos del sistema Cedrux. Cada uno de estos instaladores está fuertemente ligado a su propósito y sería muy trabajoso adaptarlo a las exigencias de Cedrux. Por tanto es necesario desarrollar una herramienta, haciendo uso de algunas funcionalidades del instalador anterior, utilizando los modelos, estándares, tecnologías y herramientas propuestas por el Departamento de Tecnología del Centro de Informatización de la Gestión de Entidades (CEIGE) para así obtener la versión 2.0 del instalador del sistema.

Con el estudio realizado se lograron definir varios aspectos como:

- ✓ El proceso de desarrollo seleccionado fue el modelo de desarrollo de software tecnológico propuesto por el Departamento de Tecnología del CEIGE.

- ✓ Se propuso desarrollar la aplicación sobre el marco de trabajo Sauxe y para esto utilizar las herramientas que se mencionan anteriormente por las ventajas que estas brindan, basándose en la política de independencia tecnológica que lleva a cabo nuestro país.

- ✓ Se determinó mediante el estudio que es más factible construir un nuevo instalador que tratar de integrar uno existente con la solución.

Capítulo 2: Diseño e implementación de la solución

2. Introducción.

En el presente capítulo se describe la solución dada para la obtención de la nueva versión del instalador del sistema Cedrux, especificando las clases necesarias correspondientes al diseño. Se

realiza un análisis de los artefactos obtenidos durante el proceso de análisis a partir de los cuales se obtiene el diseño arquitectónico rigiéndose por los requisitos identificados por los analistas, patrones de diseño y definiciones arquitectónicas definidas por el departamento de Desarrollo del CEIGE. Se definen los componentes del sistema con las nuevas funcionalidades y se describen los estándares de codificación y se realizará la implementación del sistema.

2.1. Valoración del análisis

A partir de los procesos identificados, se realiza una valoración de las nuevas funcionalidades detectadas y los artefactos generados por el análisis. Esta valoración de las funcionalidades detectadas facilita que se admitan estos requerimientos funcionales como solución a las necesidades expresadas por los usuarios de la aplicación. Durante este análisis realizado de la versión anterior del instalador se detectaron los siguientes requisitos:

- ✓ Preparar el mecanismo para la configuración posterior a la instalación.
- ✓ Emplear como capa de acceso a datos Doctrine.
- ✓ Reconocer roles creados en la base de datos.
- ✓ Eliminar selección de sistema operativo.
- ✓ Implementar Rollback para una posible transacción fallida.
- ✓ Permitir selección de módulos a instalar.
- ✓ Minimizar el impacto en el código si se agrega un nuevo script.
- ✓ Mostrar progreso de instalación.

Cada uno de estos requisitos se evidencia durante el proceso de instalación y en el mantenimiento del código de fuente del instalador. Ninguno de los requerimientos anteriormente expuestos presenta ambigüedad o inconsistencia. Como no se detectaron errores o deficiencias significativas en las especificaciones de los requisitos fueron aprobadas dichas descripciones. Los requerimientos funcionales detectados a partir de las no conformidades detectadas durante el despliegue en las entidades en piloto fueron documentados y fundamentados con mucha precisión, mejorando el entendimiento de los mismos.

Capítulo 2: Diseño e implementación de la solución

Posteriormente de haber comprobado que las necesidades para mejorar la herramienta de instalación del Sistema de Gestión Empresarial Cedrux se encuentran cubiertas por los requerimientos descritos se da paso al desarrollo del diseño y la implementación de la nueva versión de la herramienta.

2.2. Descripción de la solución.

El instalador web para el Sistema de Gestión Empresarial Cedrux es un producto diseñado con el propósito de instalar la herramienta mediante la web. El producto ofrece funcionalidades de instalación avanzadas para instalar sólo los módulos que sean necesarios para la entidad dependiendo de las necesidades de la misma. La aplicación está pensada para llevar a cabo el proceso de instalación de forma sencilla y muy intuitiva, sin tener amplios conocimientos informáticos.

El desarrollo del producto buscará personalizar la instalación para cada una de las entidades del país perfeccionando la selección de los módulos necesarios para el buen funcionamiento de este en la empresa. Durante el proceso se creará la base de datos del sistema así como la ejecución de los script que contiene toda la información necesaria para que el sistema funcione en su estado inicial y además se crea la estructura para el posterior manejo del sistema. Para hacer posible lo mencionado el sistema permitirá al usuario seleccionar la localización de la base de datos y realizar una instalación personalizada donde instale los módulos que sean necesarios. Para dar solución a estas funcionalidades la primera interfaz para la instalación es la configuración de la base de datos introduciendo en el formulario el nombre de la base de datos, el servidor, el usuario y contraseña des gestor de datos y por ultimo especificar el puerto por donde se conectará Cedrux a dicho gestor. Dándole continuidad al proceso, se encuentra la interfaz de selección de módulos, donde el usuario selecciona los módulos que se desean instalar teniendo en cuenta las necesidades de la entidad que va a utilizar el producto. Después de que la instalación sea completada el proceso concluye con la configuración de los archivos de acceso y redirecciona a la página de configuración inicial.

2.3. Modelo del Dominio.

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan lo que existe o los eventos que suceden en el entorno en el que trabaja el sistema. El mismo se describe mediante diagramas de UML, específicamente mediante diagramas de clases. Estos diagramas muestran a los clientes, a los usuarios, revisores y a otros

desarrolladores las clases del dominio y cómo se relacionan unas con otras mediante asociaciones. (21)

A continuación se muestra el modelo del dominio del instalador del sistema Cedrux, cuyo objetivo es identificar y mostrar el comportamiento del negocio a través de conceptos relacionados entre sí.

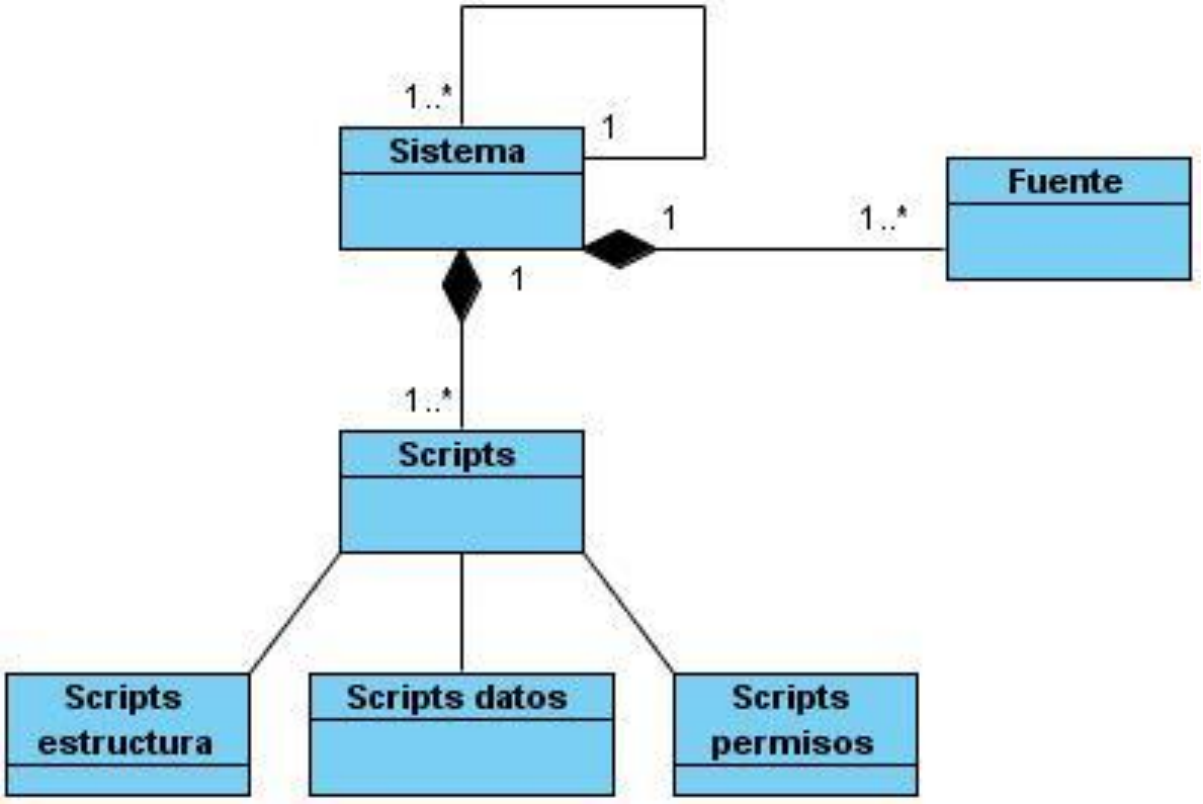

Figura 2.1 Diagrama del modelo de dominio.

2.3.1. Definición de las clases del modelo del dominio.

Sistema: Hace referencia al producto que se desea instalar el cual puede a su vez estar compuesto por otros subsistemas.

Script: De forma general, contienen todas las sentencias que deben ser ejecutadas sobre la base de datos para darle comienzo a la aplicación, estos tienen suma importancia ya que un mal funcionamiento de los mismos ocasionaría la inestabilidad del sistema e incluso el fallo en el proceso de instalación.

Script de Estructura: Son los encargados de formar la estructura necesaria para que pueda funcionar Cedrux. Estos forman la estructura de la base de datos junto a las relaciones que existen entre diferentes esquemas, tablas y otros componentes que la integran.

Script de Datos: Estos contienen los datos necesarios para el funcionamiento del sistema y son completamente necesarios para esto, no puede haber instalación sin llegar a correr un script de datos por lo que los hace imprescindibles a la hora del buen funcionamiento del sistema.

Script de Permisos: Otorgan los permisos a los usuarios del sistema teniendo en cuenta a qué módulo del mismo pueden acceder, son los últimos en correr durante la instalación ya que debe estar creada correctamente la base de datos con toda la estructura que se desea tener en la empresa, los roles deben de estar ya creados en el momento en que estos scripts se ejecutan.

Fuentes: Estas contienen toda la información, código para hacer funcionar la aplicación, sus instrucciones para llevar a cabo las tareas durante la instalación, son los ficheros PHP que componen Cedrux y lo hacen funcional.

2.4. Diagrama de clases.

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de inclusión. (22)

A continuación se muestra el diagrama de clases del diseño para una mejor comprensión del componente que se desarrolla.

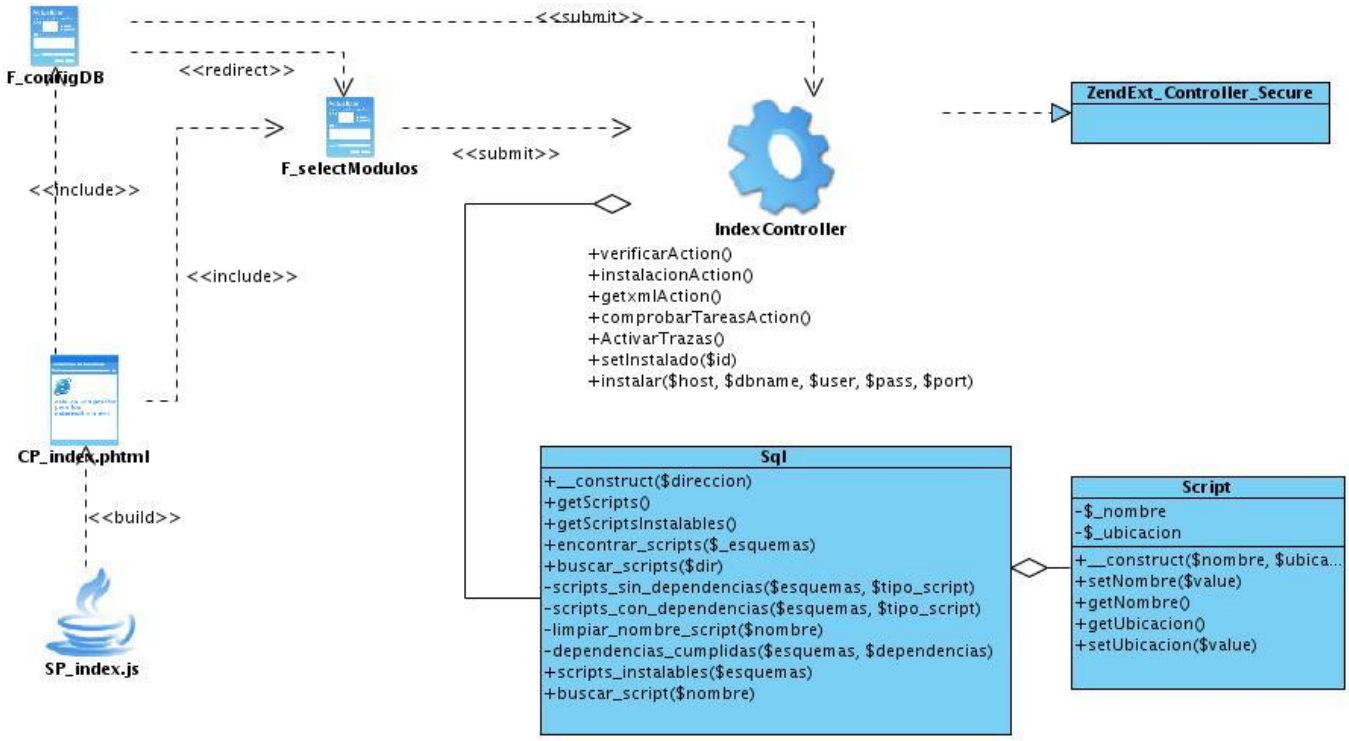

Figura 2.2 Diagrama de clases.

2.5. Prototipos de interfaz.

Para lograr un mejor entendimiento con el usuario del software se crean los prototipos de interfaz y con ellos llevar a cabo la mejora del mismo mediante la exposición del mismo con el cliente.

2.5.1. Configuración del servidor.

Esta interfaz es la primera con la que se encuentra el usuario. El objetivo de la misma es que el usuario inserte una serie de datos necesarios para la creación del servidor de datos.

- ✓ En el campo de texto con nombre Servidor se introduce el nombre del servidor donde se desea instalar el sistema.
- ✓ El campo de texto con nombre Base de datos es para introducir el nombre con el cual se quiere crear la base de datos.
- ✓ Los campos de texto Usuario y Contraseña son para el usuario y la contraseña de la base de datos.

- ✓ El campo de texto Puerto es para especificar el puerto por el cual se conectara a la base de datos en caso de que no se use el puerto por defecto del gestor utilizado.

Figura 2.3 Configuración de la base de datos.

2.5.2. Selección de subsistemas.

Esta interfaz le permite al usuario seleccionar cual o cuales subsistemas quiere instalar para darle utilidad en su entorno de trabajo.

Figura 2.4 Selección de módulos a instalar.

2.6. Modelo de implementación.

Los estándares de codificación comprenden todos los aspectos de la generación de código en el desarrollo de un proyecto determinado. Los programadores deben implementar rigiéndose por un estándar de forma prudente y debe estar creado de una para una buena comprensión y ser muy práctico. Un código de fuente completo debe reflejar un único estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada y el código de uno sea fácilmente entendible por otro.

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. El mantenimiento del código es la facilidad con que el sistema de software puede

modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores o bien para mejorar el rendimiento.

2.6.1. Diagrama de componentes.

Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes códigos de fuente, binario o ejecutable. Los componentes software tienen tipo, que indica si son útiles en tiempo de compilación, enlace o ejecución. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros. (23)

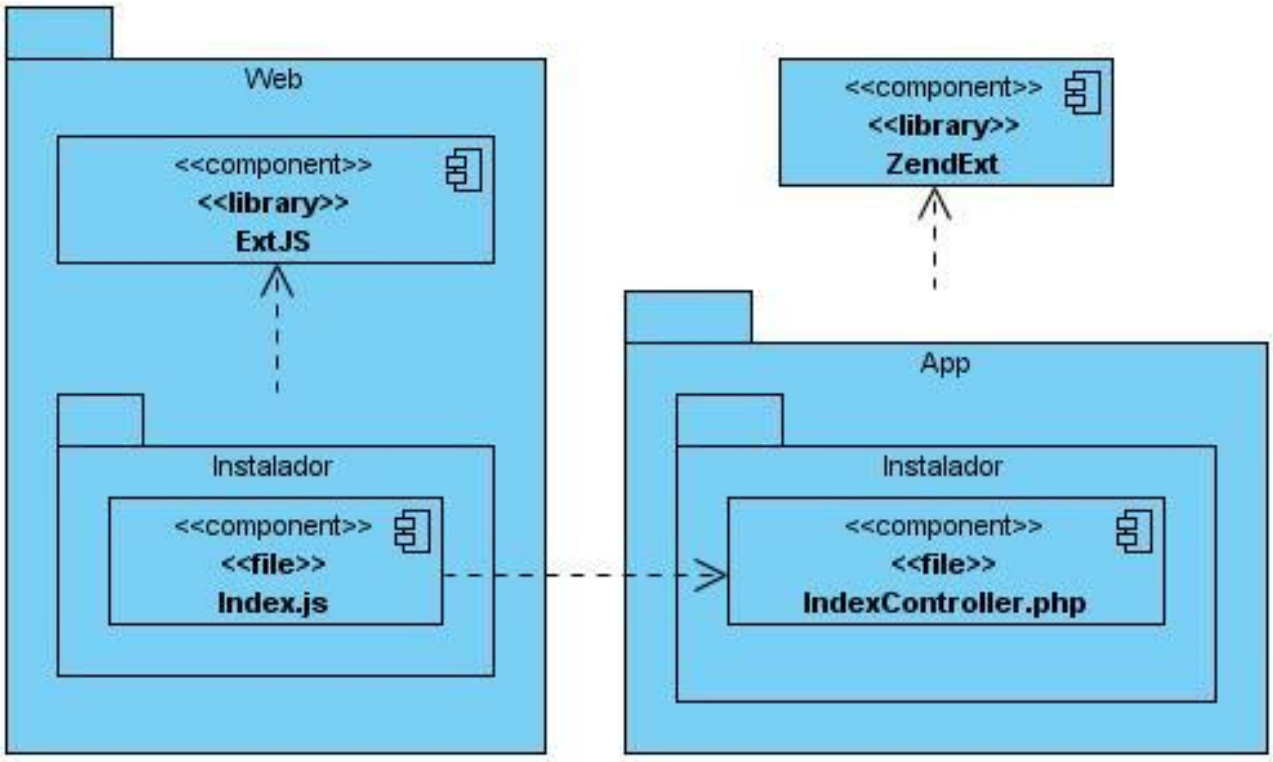

Figura 2.5 Diagrama de componentes del instalador.

La estructura del sistema se realiza atendiendo al empaquetamiento lógico de cada uno de los subsistemas de manera independiente. En la propia carpeta raíz se establece las subcarpetas apps, que contiene la lógica del negocio, y web, la cual contiene las interfaces de los módulos. Dentro de cada una de estas subcarpetas hay otras que tienen por nombre el subsistema el cual pertenecen. Los archivos del instalador se encuentran en cada una de estas carpetas.

Los recursos asociados a la aplicación, como los ficheros xml, se encuentran situados en el paquete *comun* que se encuentra en la carpeta *apps*.

Los scripts que serán ejecutados se encuentran dentro de la carpeta *web* dentro del propio subsistema del instalador.

2.6.2. Estándares de codificación.

Los estándares de codificación elevan la facilidad de mantenimiento del código, sirve como punto de referencia para los programadores, mantiene un estilo de programación, ayuda a mejorar el proceso de codificación, haciéndolo, entre otras cosas, más eficiente. (24)

2.6.2.1. PascalCasing.

PascalCasing establece que los nombres de los identificadores, las variables, métodos y clases están compuestos por una o más palabras juntas, iniciando cada palabra con letra mayúscula y el resto en minúscula.

Nomenclatura de las clases según su tipo.

Todas las clases están nombradas siguiendo el estándar PascalCasing, nombrándolas de acuerdo al propósito y la función de la misma.

- ✓ **Controllers:** Constituyen las clases controladoras del dominio y su identificador está estructurado por el nombre de la misma seguido por la palabra “Controller”. Ejemplo: IndexController.

2.6.2.2. CamelCasing.

CamelCasing es similar a PascalCasing con diferencia en la letra inicial del identificador que no comienza con mayúscula. Esta notación se utilizó para el nombre de funciones y atributos.

Nomenclatura de las funciones.

El identificativo de todas las funciones o métodos se escribe con la primera palabra en minúscula de acuerdo a la función que realizan. Ejemplo: instalarBasicos.

Los denominadores de las acciones de las clases controladoras tienen la peculiaridad de ir seguidos por la palabra “Action”. Ejemplo: verificarAction.

Nomenclatura de las variables.

El identificativo de los atributos se escribe atendiendo a su objetivo, con la primera letra en minúscula. Ejemplo: arrIdSelecc, usuario.

Nomenclatura de las constantes.

El identificativo de las constantes se realiza utilizando todas las letras en mayúscula. Ejemplo: ERROR.

2.7. Escenario arquitectónico.

Para poder tener un buen proceso de desarrollo del software, es necesario estudiar los requisitos funcionales que debe cumplir el sistema para así poder conocer cuáles serían los escenarios arquitectónicos lo cual es necesario para lograr garantizar una buena implementación del sistema. La relación de este escenario, los requisitos y la descripción de este, constituyen un factor clave en dicho proceso de desarrollo debido a que contribuyen a la medición de la asequibilidad, ajuste de la herramienta en desarrollo y las necesidades planteadas por el proyecto. A continuación se relacionan y describe dicho escenario.

Escenario:
Instalar uno o varios subsistemas.

Requisitos Asociados:

- ✓ Configuración del servidor donde se va a establecer el sistema.
- ✓ Selección de los subsistemas o componentes que se desean instalar.
- ✓ Configurar ficheros de conexión.
- ✓ Configurar trazas.

Descripción:
El sistema después de hacer las comprobaciones necesarias procede a la creación de la base de datos y las configuraciones necesarias para el funcionamiento de la aplicación.

2.7.1. Descripción de clases.

De acuerdo a la distribución de las clases y las funcionalidades que realizan, a continuación se realiza la descripción de cada una de ellas.

Clase Controladora.

Nombre: IndexController	
Tipo de clase: Controladora	
Para cada responsabilidad	
Nombre	**Descripción**
verificarAction()	Si es posible crear la base de datos la crea.
instalacionAction()	Inicia la ejecución de los scripts.
getxmlAction()	Devuelve los valores del XML subsistemas instalados.
comprobarTareasAction()	Verifica si sistema está instalado.
activarTrazas()	Activa las trazas de seguridad.
configurarFicheros($host, $dbname, $port)	Configura los ficheros pertinentes para el posterior acceso a la información desde la aplicación.
instalar($host, $dbname, $user, $pass, $port, $arr_selecc)	Ejecuta ordenadamente los scripts de la base de datos.
eliminardatabaseAction()	Elimina la base de datos creada por el usuario para realizar la instalación del sistema.
buscarDatosSubsistema($id)	Busca los datos de un subsistema dado.

Tabla 2.2: Descripción de la clase IndexController.

Nombre: Script	
Tipo de clase: Auxiliar	
Para cada responsabilidad	
Nombre	**Descripción**
__construct($nombre, $ubicacion)	Crea una instancia de la clase.
setNombre($value)	Modifica el valor de un atributo.
getNombre()	Devuelve el valor de un atributo.
getUbicacion()	Devuelve el valor de un atributo.
setUbicacion()	Devuelve el valor de un atributo.

Tabla 2.3: Descripción de la clase Script.

Nombre: Sql	
Tipo de clase: Auxiliar	
Para cada responsabilidad	
Nombre	**Descripción**
__construct()	Crea una instancia de la clase
+getScripts()	Devuelve un arreglo de tipo Script
+getScriptsInstalables()	Devuelve un arreglo con los scripts que serán instalados
+encontrar_scripts($direccion,$_esquemas)	Busca y retorna un arreglo de tipo Script con los scripts que respondan a los esquemas señalados

+buscar_scripts($dir)	Dada una dirección busca todos los scripts de tipo SQL que se encuentren en el árbol de carpetas señalado
+scripts_sin_dependencias($esquemas, $tipo_script)	Busca y retorna los scripts que no tengan dependencias de un tipo dado que respondan a los esquemas señalados
+scripts_con_dependencias($esquemas, $tipo_script)	Busca y retorna los scripts que tengan dependencias de un tipo dado que respondan a los esquemas señalados
+limpiar_nombre_script($nombre)	Devuelve un cadena de texto sin paréntesis ni la extensión del archivo(.sql)
+dependencias_cumplidas($esquemas, $dependencias)	Determina si se cumplen las dependencias para un script determinado
+scripts_instalables($esquemas)	Determina el orden de ejecución de los scripts
+buscar_script($nombre)	Busca la dirección de un script específico

Tabla 2.4: Descripción de la clase Sql.

2.8. Conclusiones del capítulo.

Con la elaboración de este capítulo fue posible constatar que el análisis propuesto por los analistas presenta una correcta descripción de las nuevas funcionalidades y cumplen con las necesidades que exige el instalador de Cedrux. La descripción del diseño arquitectónico permitió que la implementación se realizara de manera correcta respondiendo a la arquitectura de todo el sistema al cual se encuentra incluido. La descripción de las clases utilizadas permitió tener una idea de cómo funcionaría el sistema aumentando el entendimiento de la relaciones entre las clases y el código, facilitando el mantenimiento y la reutilización de sus funciones.

Capítulo 3: Validación de la solución

3. Introducción.

En el desarrollo de las aplicaciones, las pruebas aseguran la validación del buen funcionamiento del sistema, comprobando que se encuentra listo para su despliegue. En la actualidad, los software aumentan constantemente en tamaño y complejidad, estas características hace que presenten mayores probabilidades de poseer errores, dados por diferentes tipos de fallas en algunas funcionalidades poniendo en riesgo la utilidad de la herramienta. No es posible asegurar que un software no presente errores, pero es posible evitarlos haciendo estas pruebas las cuales no hacen más que detectar las fallas a tiempo para poder mitigarlas antes de que sea liberado el producto a los usuarios finales.

En este capítulo se realizarán pruebas tanto de diseño como funcionales a la solución desarrollada aplicando múltiples métricas garantizando la calidad y funcionalidad del sistema.

3.1. Métricas para la evaluación del modelo de diseño.

Para evaluar el modelo de diseño del instalador del sistema Cedrux se aplicaron múltiples métricas de las cuales a continuación se muestran los resultados de la misma.

3.1.1. Métrica Tamaño Operacional de Clase (TOC).

Esta métrica se basa en el total de funcionalidades contenidas en las clases, de esta forma determina la afectación que ejercen estas en el diseño.

Las afectaciones que se pueden observar al evaluar el diseño utilizando la métrica TOC son:

- ✓ Responsabilidad: El aumento del TOC provoca un aumento de la responsabilidad asignada a la clase.
- ✓ Complejidad de Implementación: El aumento del TOC provoca un aumento de la complejidad de implementación de la clase.
- ✓ Reutilización: Un aumento del TOC provoca una disminución en el grado de reutilización de la clase.

Para más detalles ver los instrumentos y la tabla de resultados para la métrica TOC en el Anexo 1.

Para cada atributo se establece un rango de valores que determinan la complejidad en la aplicación del TOC.

	Categoría	Criterio
Responsabilidad	Baja	< =promedio.
	Media	Entre promedio y 2* promedio.
	Alta	> 2* promedio.
Complejidad implementación	Baja	< = promedio.
	Media	Entre promedio y 2* promedio.
	Alta	> 2* promedio.
Reutilización	Baja	> 2* promedio.
	Media	Entre promedio y 2* promedio.
	Alta	<= promedio.

Tabla 3.1: Rango de valores de para la evaluación utilizando la métrica TOC.

Seguidamente se muestra la agrupación de las clases según los valores establecidos por la métrica TOC mostrado en %.

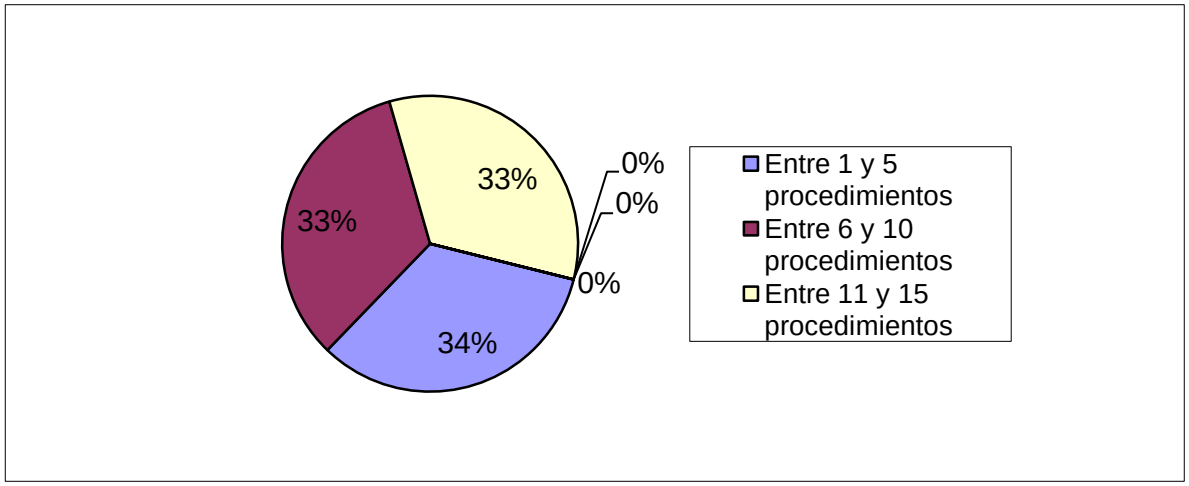

Figura 3.1: Agrupación de intervalos definidos según la métrica TOC.

La responsabilidad de las clases también es representada agrupada en %.

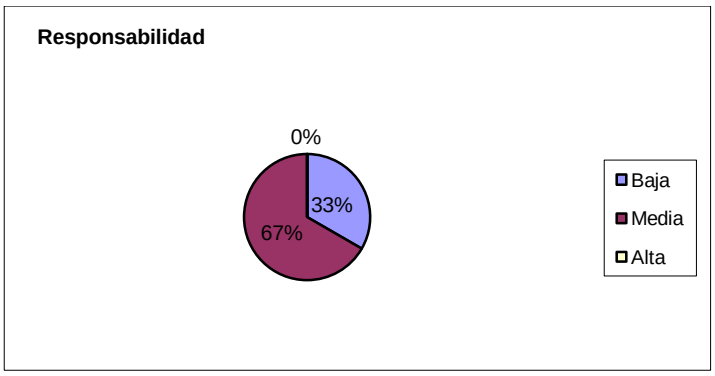

Figura 3.2: Agrupación según la responsabilidad.

La complejidad de las clases es un aspecto que también se mide en la métrica TOC.

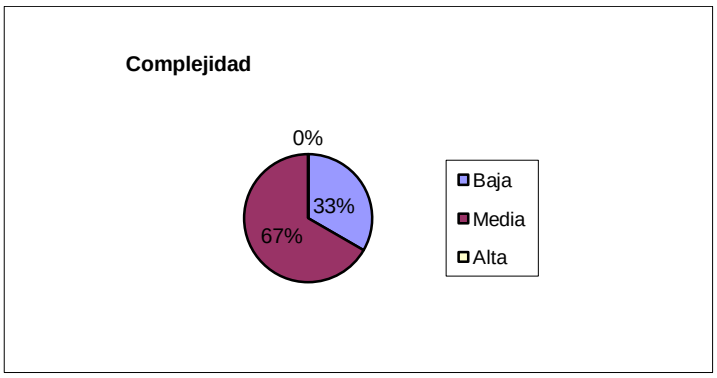

Figura 3.3: Agrupación según la complejidad.

Finalmente la reutilización es otro aspecto que mide la métrica TOC dando una idea de que % de código puede ser utilizado para otra aplicación.

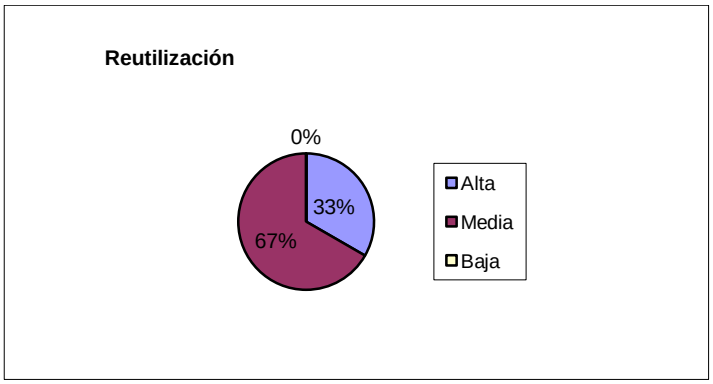

Figura 3.4: Agrupación según la reutilización.

Con la evaluación de la calidad de diseño utilizando la métrica TOC, se pudo constatar que la mayoría de las clases poseen de 1 a 10 procedimientos, siendo este un resultado que se encuentra dentro de los niveles aceptables de calidad. Como se muestra en la figura 3.2 y 3.3, los niveles de responsabilidad y complejidad de las clases no llegan en ninguno de los casos a ser alta, lo cual presenta un resultado positivo para nuestro diseño. En el caso de la reutilización, figura 3.4, muestra un nivel mayoritariamente medio, destacando que no presenta niveles bajos en este aspecto.

3.1.2. Métrica de Relaciones entre Clases (RC).

Esta métrica evalúa las relaciones de uso que existen en cada una de las clases que se encuentran en el diseño. A partir de este análisis se obtiene las dependencias de cada una de estas clases.

Las afectaciones que se pueden observar al evaluar el diseño utilizando la métrica RC son:

- ✓ Acoplamiento: El aumento del RC provoca un aumento del acoplamiento de la clase.
- ✓ Complejidad del mantenimiento: El aumento del RC provoca un aumento de la complejidad del mantenimiento de la clase.
- ✓ Reutilización: El aumento del RC provoca una disminución en el grado de reutilización de la clase.
- ✓ Cantidad de pruebas: El aumento del RC provoca un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Para más detalles ver los instrumentos y la tabla de resultados para la métrica RC en el Anexo 2.

Para cada atributo se establece un rango de valores que determinan la complejidad en la aplicación del RC.

	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2

Complejidad de Mantenimiento.	Baja	<= promedio.
	Media	Entre promedio. y 2* promedio.

	Alta	> 2* promedio.

Reutilización	Baja	>2* promedio.
	Media	Entre promedio. y 2* promedio.
	Alta	<= promedio.

Cantidad de Pruebas	Baja	<= promedio.
	Media	Entre promedio. y 2* promedio.
	Alta	> 2* promedio.

Tabla 3.2: Rango de valores de para la evaluación utilizando la métrica RC.

Seguidamente se muestra la agrupación de las clases según los valores establecidos por la métrica TOC mostrado en %.

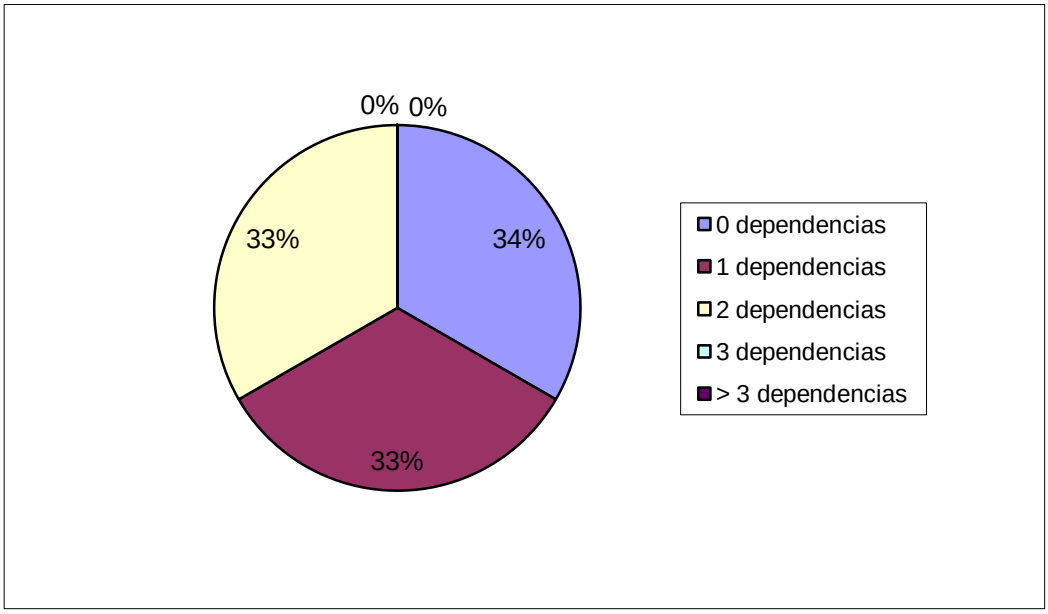

Figura 3.5: Agrupación de intervalos definidos según la métrica RC.

El acoplamiento entre las clases es uno de los valores que son medidos por esta métrica.

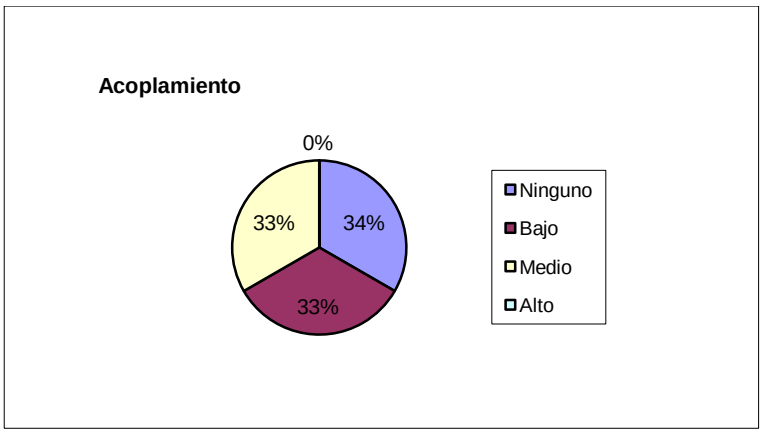

Figura 3.6: Agrupación de intervalos definidos según el acoplamiento.

La complejidad de mantenimiento es otro aspecto a tener en cuenta el cual también es medible y mostrado en %.

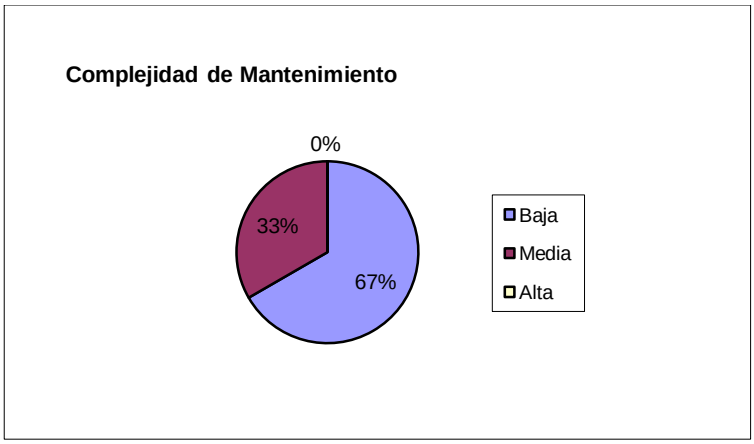

Figura 3.7: Agrupación de intervalos definidos según la complejidad de mantenimiento.

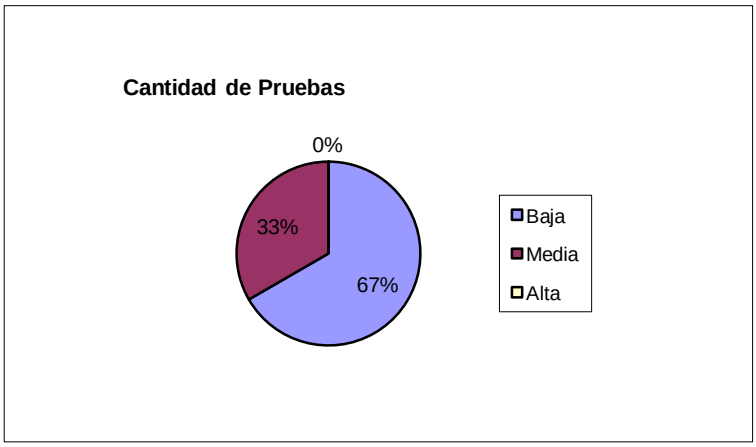

Figura 3.8: Agrupación de intervalos definidos según la cantidad de pruebas.

La posibilidad de reutilización del código es el último aspecto que mide esta métrica lo que no significa que sea el menos importante.

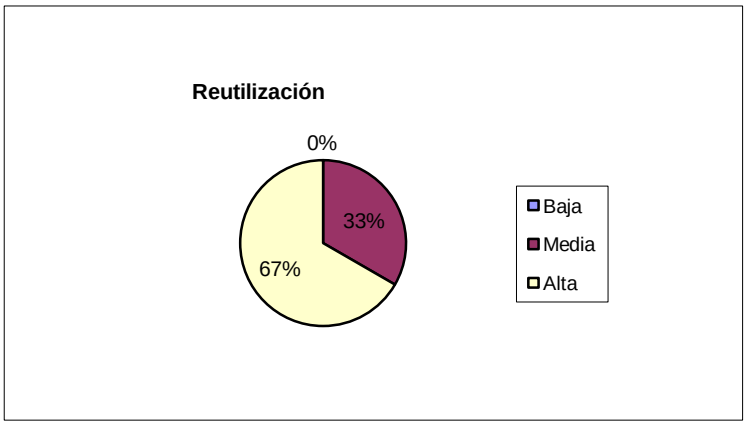

Figura 3.9: Agrupación de intervalos definidos según la cantidad de pruebas.

Con este resultado de evaluación de la métrica RC, se pudo constatar que la mayoría de las clases poseen de 0 a 2 dependencias, siendo este un resultado positivo. En la figura 3.6 se puede ver el nivel de acoplamiento entre las clases el cual se encuentra entre medio y ninguno siendo un nivel considerablemente bueno, ya que permite que la reutilización tenga valores positivos ya que la reutilización posible en un 67% de las clases en su totalidad (figura 3.9). El valor de la complejidad de mantenimiento es positivo al encontrarse entre bajo y medio, con un 67% y 33% respectivamente dejando un notable 0% el mantenimiento de alta complejidad. Además, la cantidad de pruebas que son necesarias hacerle al sistema son mayoritariamente bajas (figura 3.8). Estos valores obtenidos dan a demostrar que diseño de las clases se realizo de la manera más optima posible maximizando las ventajas de un buen diseño.

3.2. Diseño de casos de prueba.

Los casos de prueba se basan en las diferentes entradas que pueden ser introducidas en el sistema con sus correspondientes valores de salida para los que está previsto. Estos casos de prueba son utilizados para determinar si el requisito de una aplicación es parcial o completamente satisfactorio. Los casos de prueba dan a conocer si las funcionalidades para las que está destinado el producto son operativas, que los resultados esperados en cada una de las acciones que este implemente.

A continuación se especifica el caso de prueba “Realizar instalación” el cual completa la instalación del sistema.

Condiciones de ejecución: Debe existir un servidor con SO Ubuntu Server 8.4, PostgreSQL 8.3, Apache 2.2.9 y PHP 5.x junto con los módulos: php_gd, php_xsl y php_pgsql. En el puesto del cliente que va a realizar la instalación debe tener habilitado algún navegador web. El cliente debe conocer los datos necesarios para poderse conectar al servidor de datos.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Crear base de datos.	Se la base de datos del sistema.	EP 1.1: Se introducen datos válidos.	✓ Se introducen los datos necesarios para la creación de la base de datos. ✓ Se presiona el botón Aceptar.
		EP 1.2: Se introducen datos inválidos.	✓ Se introducen datos inválidos para la creación de la base de datos. ✓ Se presiona el botón
		EP 1.3: Se dejan campos obligatorios en blanco.	✓ Se introducen los datos para la creación de la base de datos dejando algún campo obligatorio en blanco. ✓ Se presiona el botón Aceptar.

Tabla 3.3: Descripción del caso de prueba para el requisito Crear base de datos.

Descripción de variable.

No.	Nombre de campo	Clasificación	Válido	Inválido
1	Dirección del servidor de datos	Campo de texto.	NA	NA
2	Nombre de la base de datos a crear	Campo de texto.	Cedrux1	1Cedrux
3	Usuario del servidor de datos	Campo de texto.	NA	NA
4	Contraseña de usuario del servidor de datos	Campo de texto.	NA	NA
5	Puerto por el cual se va a conectar al servidor de datos	Campo de texto.	5432	abcd

Tabla 3.4: Descripción de las variables del caso de prueba para el requisito Crear base de datos.

Juego de datos a probar.

Id del escenario	Escenario	Servidor de datos	Base de datos	Usuario	Contraseña	Puerto	Respuesta del Sistema	Resultado de la Prueba
EC 1.1	Se introducen datos válidos.	10.7.19.111	Cedrux1	Cliente	ClienteCedrux1	5432	Crea la base de datos en el servidor.	Crea la base de datos en el servidor.
EC 1.2	Se introducen datos inválidos.	localhost	1Cedrux	NA	NA	abcd	El sistema emite un mensaje para rectificar datos erróneos.	El sistema emite un mensaje para rectificar datos erróneos.
EC 1.3	Se dejan campos obligatorios en blanco.						El sistema muestra un mensaje que alerta de la falta de datos.	El sistema muestra un mensaje que alerta de la falta de datos.

Tabla 3.5: Juego de datos que serán probados.

3.2.1. Resultado de las pruebas.

Como medida para comprobar la calidad y funcionalidad del software desarrollado se realizaron 2 iteraciones de revisiones internamente, con el empleo de los diseños de casos de pruebas para la aplicación. Durante la aplicación de las pruebas en cada una de las iteraciones se detectaron un grupo de no conformidades las cuales se les dio solución.

Iteración 1: Durante esta iteración se encontraron 6 no conformidades, de ellas 4 significativas, representando un 67% del total. Estas no conformidades significativas le hacían referencia a la funcionalidad del sistema como tal. Todas las no conformidades fueron resueltas totalmente arrojando un resultado de un 100 % de solución.

Iteración 2: Durante esta iteración se encontraron 4 no conformidades, solamente 1 significativa, representando un 25% del total. Esta no conformidad significativa le hacían referencia a una funcionalidad del sistema. Todas las no conformidades fueron resueltas totalmente arrojando un resultado de un 100 % de solución.

Se puede apreciar que la cantidad de no conformidades de una iteración a otra disminuyeron un 33%, y las de tipo funcionales descendieron un 75%, lo que muestra el incremento en la calidad de la herramienta.

3.3. Conclusiones del capítulo.

Durante el desarrollo de este capítulo se pudo llegar la conclusión de que el diseño elaborado cumple con varios aspectos los cuales fueron medidos a partir de diferentes métricas de diseño, además de aplicar pruebas que permitieron evaluar cada uno de los elementos del producto. Se pudo constatar que la aplicación cuenta con funcionalidades bien implementadas que presentan mínimas posibilidad de que fallen dadas por una situación antes tratada, garantizando las necesidades de los clientes y evitando posibles caídas del sistema durante el proceso de instalación.

Conclusiones generales

Para la elaboración del presente trabajo de diploma fue necesario realizar un estudio de la problemática existente para identificar los principales objetivos y tareas a ejecutar en el desarrollo de la investigación. Entre los objetivos identificados se encuentra el estudio del estado del arte en la cual se estudiaron los principales conceptos relacionados con la investigación y los instaladores existentes para analizar la posibilidad de reutilizar de alguno de estos, lo cual no fue posible debido a las deficiencias detectadas en cada uno de ellos y se decidió implementar un nuevo instalador. Para el mismo se decidió usar el modelo de desarrollo, herramientas, tecnologías y estándares propuestos por el Departamento de Tecnología del CEIGE. El sistema se desarrolló haciendo uso de las mejores prácticas de diseño e implementación de las propuestas estudiadas. Se llevo a cabo la implementación cumpliendo con los estándares de interfaz de usuario, validación y de codificación garantizando la calidad del sistema. Como técnicas de validación se aplicaron métodos de validación por casos de prueba, brindándole al equipo de calidad una guía para ejecutar las pruebas y obtener la liberación del sistema.

Los resultados arrojados por estas permitieron concluir que el diseño presentaba valores positivos en indicadores de calidad tales como Reutilización, Facilidad de Mantenimiento, Acoplamiento, Cantidad de pruebas, entre otros. Actualmente el sistema se encuentra funcional, respondiendo al objetivo para la cual fue desarrollado.

Recomendaciones

El equipo de desarrollo que ha llevado a cabo esta investigación considera de suma importancia proponer las siguientes recomendaciones:

- ✓ Seguir el trabajo sobre la solución actual para el desarrollo de nuevas versiones en busca de mayor eficiencia en el proceso de instalación.
- ✓ Continuar con la realización de pruebas funcionales que permitan garantizar la calidad del producto.
- ✓ Agregar nuevas funcionalidades al producto obtenido.
- ✓ Que en el futuro el sistema incorpore la configuración no solo de la aplicación, también el servidor Web Apache.

Referencias bibliográficas

1. Aplicaciones Libres. [En línea] [Citado el: 5 de febrero de 2011.] http://www.aplicacioneslibres.com.ar/.

2. GNU Operating System. [En línea] 1 de 12 de 2010. [Citado el: 10 de 12 de 2011.] http://www.gnu.org/software/bash/bash.html.

3. masadelante.com. [En línea] 2010. [Citado el: 10 de 12 de 2010.] http://www.masadelante.com/faqs/exe.

4. Fermu Website. [En línea] 18 de 1 de 2006. http://www.fermu.com/es/articulos/guia-regedit/87/364.

5. Mantis Bug Tracker. [En línea] http://www.mantisbt.org/.

6. BitNami. [En línea] 2010. http://bitnami.org/.

7. Joomla. [En línea] 2010. http://www.joomla.cl/documentacion/ique-es-un-cms.

8. Microsoft. [En línea] http://www.microsoft.com/web/downloads/platform.aspx.

9. **Lago, Ramiro.** Patrones de diseño software. [En línea] Abril de 2007. http://www.proactiva-calidad.com/java/patrones/.

10. Adictos al trabajo. [En línea] 02 de 01 de 2006. http://www.adictosaltrabajo.com/tutoriales/pdfs/grasp.pdf.

11. **Piñeiro, Yadenis.** *Declaracion de autoria marco de trabajo Sauxe.* La Habana : s.n., 2009. s.n..

12. extjs. [En línea] [Citado el: 5 de 12 de 2010.] http://extjs.es.

13. tecnoretales. [En línea] [Citado el: 10 de 12 de 2010.] http://www.tecnoretales.com/tag/doctrine/.

14. universe xt. [En línea] [Citado el: 12 de 12 de 2010.] http://universe-xt.com.

15. Zend Framework. [En línea] 2011. http://framework.zend.com/.

16. **Larman, Craig.** *UML y Patrones.* 2001.

17. NetBeans. [En línea] [Citado el: 22 de 2 de 2011.] http://netbeans.org/.

18. Apache. [En línea] [Citado el: 12 de 12 de 2010.] http://www.apache.org/.

19. PostgreSQL. [En línea] [Citado el: 23 de 1 de 2011.] www.postgresql.org.

Referencias bibliográficas

20. Paradigma Visual. *freedownloadmanager.* [En línea] [Citado el: 13 de 12 de 2010.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/.

21. **Ivar Jacobson, Magnus Christerson, Patrik Jonsson, Gunnar Overgaard.** *Object-Oriented Software Engineering: A Use-Case-Driven Approach.* 1993. 84-7829-036-2.

22. Departamento de Ciencias de la Computacion. [En línea] Universidad de Chile. http://www.dcc.uchile.cl/~psalinas/uml/modelo.html.

23. Webdocs. [En línea] 2007. http://webdocs.cs.ualberta.ca/~pfiguero/soo/uml/implementacion01.html.

24. **Matas, Miguel.** http://www.miguelmatas.es/blog/2008/05/06/como-definir-un-estandar-de-codificacion-yo-trabajo/. [En línea] 5 de 6 de 2008. [Citado el: 4 de 3 de 2011.] http://www.miguelmatas.es/blog/2008/05/06/como-definir-un-estandar-de-codificacion-yo-trabajo/.

25. EcuRed. [En línea] [Citado el: 25 de 1 de 2011.] http://www.ecured.cu.

26. **Brito Acuña, Karenny.** http://www.eumed.net. [En línea] 2009. [Citado el: 12 de 3 de 2011.] http://www.eumed.net.

27. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de software.* s.l. : Addison Wesley Longman, 1999.

28. **Ciberaula.** Una Introducción a APACHE. [En línea] 2006. http://linux.ciberaula.com/articulo/linux_apache_intro/ .

29. **Consultora, DISAIC.Casa.** El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero. [En línea] 2010. http://www.disaic.cu/modules.php?name=Content&pa=showpage&pid=818..

30. **Corzo, Giancarlo.** Desarrollo en WEB. [En línea] 2008. http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/.

31. **Framework, Zend.** Introducción a Zend Framework. [En línea] 2010. http://manual.zfdes.com/.

32. **GBM.** SAP Enterprise Resource Planning (ERP). *Aplicaciones ERP: la nueva alternativa para la gestión de recursos.* [En línea] 2010. http://www.gbm.net/soluciones/enterprise_resource_planning.php.

33. **Gestión, Centro de Soluciones de.** *Definición del ciclo de vida de los proyectos de desarrollo de software v1.0.* 2009.

34. **Global, SAP.** SAP Solutions. [En línea] 2010. http://www.sap.com.

35. **Integral, ASSEST. Sistema de Gestión.** ¿Qué es lo nuevo en AssetsNS versión 2.0? [En línea] 2006. http://www.assets.co.cu/texto.asp?id=5.

Referencias bibliográficas

36. **Mariaelena.** *Entrevista con funcionales del Ministerio de Finanzas y Precios. Procesos bancarios de una entidad.* 2010.

37. **Mata, Manel Pérez.** TecnoRetales. *Qué es Doctrine ORM?* [En línea] 2009. http://www.tecnoretales.com/programacion/que-es-doctrine-orm/.

38. **Openbravo.** Openbravo. Manual de Usuario 1.1. [En línea] 2006. http://www.openbravo.com/docs/openbravo-manual-de-Usuario-v1.1.pdf.

39. **SISCONT.** SISCONT. Software Contable-Fianciero. [En línea] 2009. http://www.siscont.com/DESCRIPTIVO%20SISCONT.pdf.

40. **Ramos Arias, Taime y Torres Salas, Pedro Antonio.** *Diseño e implementación del módulo Banco del Sistema Integral de Gestión CEDRUX.* La Habana : s.n., 2010.

41. **Hilliard, Rich.** IEEE-Std-1471-2000. [En línea] http://www.enterprise-architecture.info/Images/Documents/IEEE%201471-2000.pdf.

42. **IBM.** Rational Host Integration Solution. [En línea] http://www-01.ibm.com/software/awdtools/hostintegration/features/?S_CMP=mav.

43. **EcuRed.** Ide de Programación. [En línea] 2011. http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.

44. **PostgreSQL.** PostgreSQL 8.3.15 Documentation. [En línea] http://www.postgresql.org/docs/8.3/static/release-8-3.html.

45. Kriptopolis. [En línea] 28 de 9 de 2007. http://www.kriptopolis.org/bitnami.

46. CMS en español. [En línea] 2010. http://www.cmsenespanol.com/.

47. Open Source CMS. [En línea] 2010. http://php.opensourcecms.com/.

Glosario de términos

Métricas: las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo, el producto se mide para intentar aumentar su calidad.

Plataformas libres: Plataformas de desarrollo de software desarrolladas sobre tecnologías libres.

Sistema de Gestión de Entidades: Sistema que se encarga de la gestión de los procesos dentro de una entidad.

PHP: Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas WEB dinámicas.

Librerías de enlace dinámico: Una librería de enlace dinámico, en adelante DLL (Dynamic Link Library), es un archivo que contiene funciones y/o recursos

MS-DOS: (Sistema Operativo de disco de Microsoft) es un Sistema Operativo perteneciente a la familia DOS.

Shell: En informática, el término **Shell** se emplea para referirse a programas que proveen una interfaz de usuario para acceder a los servicios del Sistema Operativo.

Plugins: Un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy especifica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como plug-in(del inglés "enchufable"), add-on(agregado), complemento, conector o extensión.

PHPDocumentor: Es un generador de documentación de código abierto escrito en PHP. Automáticamente analiza el código fuente PHP y produce la API de lectura.

Framework: La palabra inglesa framework define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

Glosario de términos

Objeto mapeador relacional: Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

XML: siglas En inglés de Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide WEB Consortium (W3C).

Debugger: Un depurador (en inglés, *debugger*), es un programa que permite depurar o limpiar los errores de otro programa informático.

AJAX: *Ajax*, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo WEB para crear aplicaciones interactivas.

WEB Services: Un **servicio WEB** (en inglés, *WEB service*) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Anexos

Anexo 1: Instrumento de medición de la métrica Tamaño Operacional de Clase (TOC).

No	Subsistema	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
1	Instalador	IndexController	13	Media	Media	Media
2	Instalador	Sql	9	Media	Media	Media
3	Instalador	Scripts	4	Baja	Baja	Alta

Tabla A1: Resultados de la evaluación de la métrica TOC.

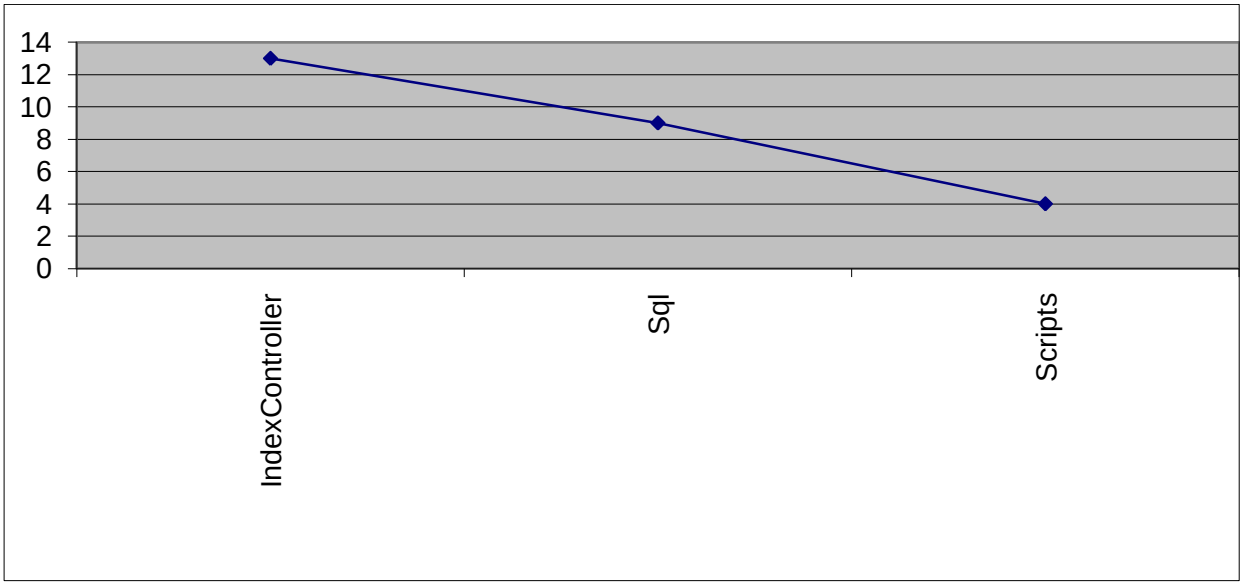

Figura A1: Representación de los resultados obtenidos agrupados en los intervalos definidos.

Anexo 2: Instrumento de medición de la métrica Relaciones entre Clases (RC).

No	Subsistema	Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización	Cantidad de Pruebas
1	Instalador	IndexController	2	Medio	Media	Media	Media
2	Instalador	Sql	1	Bajo	Baja	Alta	Baja
3	Instalador	Scripts	0	Ninguno	Baja	Alta	Baja

Tabla A2: Resultados de la evaluación de la métrica RC.

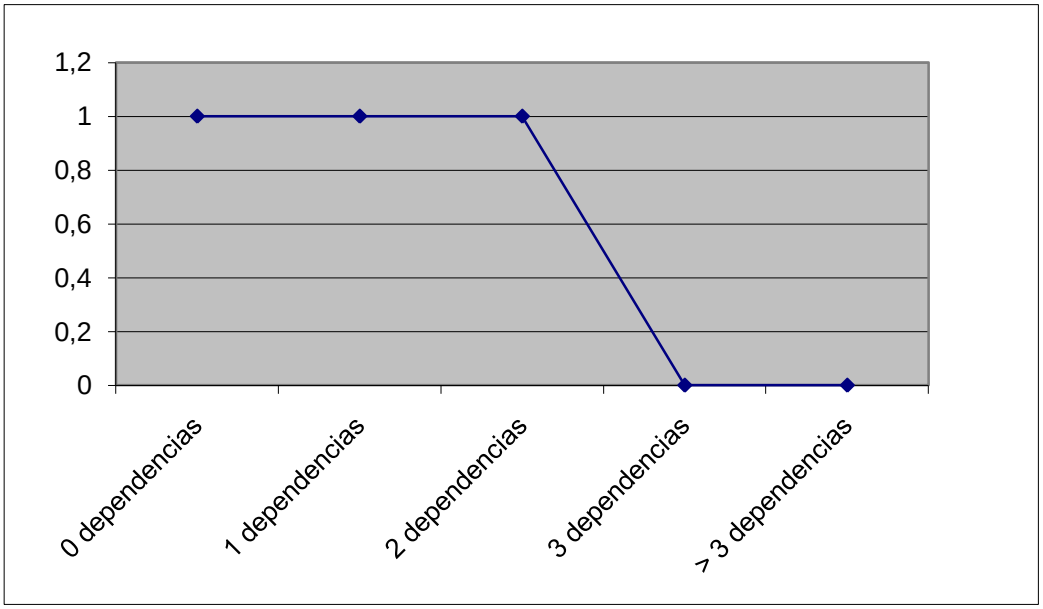

Figura A2: Representación de los resultados obtenidos agrupados en los intervalos definidos.