Universidad de Cienfuegos “Carlos Rafael Rodríguez”
Facultad de Ingeniería
Carrera de Ingeniería Informática

SISTEMA INFORMÁTICO PARA LA GESTIÓN DE REPARACIÓN DE LAS PIEZAS DE TELEFONÍA PÚBLICA EN ETECSA.

Trabajo de diploma para optar por el título de Ingeniero Informático.

Autor:
Sergio E. Suárez Calañas.

Tutores:
Ing. Jorge Luis de Armas Díaz.
Msc. Rewer Canosa Rodríguez.

Cienfuegos, Cuba
Curso 2013 – 2014

Yo Sergio Ernesto Suárez Calañas declaro que soy el único autor de este trabajo y autorizo al Taller de Reparaciones de ETECSA y al Departamento de Informática de la Facultad de Ingeniería en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo (firmamos) la presente a los _____ días del mes de _____ de _____”.

_____

Autor: Sergio Ernesto Suárez Calañas

_____ _____

Firma Tutor: Ing. Jorge Luis de Armas Díaz Firma Tutor: Msc. Rewer Canosa Rodríguez

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

---------- ----------

Firma Tutor Firma Tutor

---------- ----------

Firma ICT Firma Vicedecano

Agradecimientos

- A mis padres, por brindarme todo el apoyo y amor del mundo en el transcurso de la vida.

- A mi hermano, por estar siempre a mi lado y compartir los momentos tristes y buenos de mi vida.

- A mis abuelos, gracias por estar presentes y formar parte de mi vida.

- A mis tutores que me han ayudado en todo momento, sin importar el horario o situación en que se encontraran, teniendo la paciencia y dedicación para hacer posible este trabajo.

- A mis compañeros de grupo por todo su apoyo y ayuda en estos años compartidos.

- A cada persona que de una forma u otra contribuyó con su esfuerzo al éxito de este trabajo.

Resumen

La presente investigación tiene como finalidad la elaboración de un sistema informático capaz de gestionar los datos relacionados con las piezas de telefonía pública en el taller de reparaciones de Empresa de Telecomunicaciones de Cuba S.A. (ETECSA), el cual fundamenta la idea de alcanzar una mejor gestión de la información y elevar la rapidez del proceso.

El sistema propuesto tiene como título “Sistema de Gestión de Piezas del Taller de Telefonía Pública en Cienfuegos”. El mismo permite la gestión de los datos de entrega y recepción de todas las piezas y los modelos pertenecientes. Además de las piezas que sean particularmente Microteléfono, se identifican las roturas que pueden ser de cuatro formas. También facilita que el acceso y manipulación de los datos sean fáciles y seguros.

A través del documento de la investigación quedan descritos los elementos que conforman el análisis, diseño e implementación del sistema propuesto, siguiendo lo establecido por la metodología de Desarrollo Basado en Funcionalidades, por sus siglas en inglés FDD, además utilizando el Lenguaje Unificado de Modelado (UML) por sus siglas en inglés. Para la implementación del mismo se utilizó PostgreSQL como Sistema Gestor de Bases de Datos, Apache como servidor Web, PHP y JavaScript como lenguajes de programación y el framework CodeIgniter.

Palabras claves: División Territorial de Cienfuegos (D.T.CF), telefonía, pública, piezas, taller.

Abstract

The present investigation has like purpose the elaboration of an information-technology capable system to negotiate the data related with the pieces of public telephony in the repair shop of Cuba's Telecommunications Company S.A. (ETECSA), which bases the idea of attaining a better information science and raising the speed of the process.

The proposed system has like title "System of Step of Pieces of the Workshop of Public Telephony in Cienfuegos". This allows the step of the data of delivery and reception of all the pieces, the respective fashion models. Besides, to the pieces that are particularly handset, identify the breakings that can be of four ways. Also it facilitates that the access and manipulation of the data are themselves easy and safe.

Through the document of investigation there are described the elements that conform the analysis, design and implementation of the proposed system, following what's established around the methodology of development based in functionalities, for its initial letters in English FDD, besides using the Unified Modeling Language (UML). For the implementation of the same CodeIgniter used PostgreSQL like Sistema Managing of Bases of Datos, Apache like Web server, PHP and JavaScript like programming languages and the framework himself.

Key words: Territorial Division of Cienfuegos (D.T.CF), telephony, public, pieces, workshop.

Índice

Introducción

La situación del servicio telefónico a inicio de los 90s era muy desfavorable. Problemas organizativos y de financiamiento ocasionaron un serio perjuicio a la telefonía, que no estaba a la altura de las exigencias del país para su desarrollo. Es por ello que se decide la formación de una empresa que integrara todas las actividades de telecomunicaciones, frenara el deterioro e impulsara este sector.

Antes de la creación de la Empresa de Telecomunicaciones de Cuba S.A (ETECSA) existían 14 Empresas Integrales de Comunicaciones que abarcaban las especialidades de telefonía, radio, correos y prensa, además de otras empresas nacionales especializadas. En esta categoría estaban las empresas de Proyectos, Construcción y Montaje, Cable Coaxial y Larga Distancia.

A partir de la fusión de una parte de las empresas integrales provinciales de comunicaciones y de las especializadas, comienza el funcionamiento de ETECSA; en un proceso que se extendió desde inicios de 1994 hasta febrero de 1995, siendo este el año de comienzo de la operación de ETECSA. En el mes de febrero del propio año se realiza la contratación de todos sus trabajadores. Desde ese momento, la empresa ha atravesado distintos períodos de cambios tecnológicos, de estructura, de sistemas gerenciales, de orientación estratégica, de desarrollo de nuevos servicios, etc.

El 16 de diciembre del 2003, mediante el Acuerdo 4996 del Comité Ejecutivo del Consejo de Ministros y el Decreto 275, se amplía la concesión de ETECSA como operador unificado de telecomunicaciones, a través de la fusión de Cubacel y C_COM en ETECSA, con el propósito fundamental de integrar en una sola empresa mixta todas las actividades relacionadas con la telefonía fija y celular, así como de otros servicios de telecomunicaciones, para asegurar el proceso de investigación, inversión, producción, prestación de servicios y su comercialización en Cuba.

Esta entidad proporciona a los usuarios y a toda la población, servicios que garanticen la satisfacción de sus necesidades en materia de

telecomunicaciones, respaldando los planes de desarrollo social y económico que lleva a cabo el país, las tareas de la defensa y garantizando los resultados económicos planeados.

Actualmente existen Divisiones Territoriales (D.T) en cada una de las Provincia a lo largo del Territorio del País.

La División Territorial en la provincia de Cienfuegos (D.T.CF) está repartida en cuatros Centros Telefónicos Principal (CT.P) y cuatros Centros Telefónicos Asociado (CT.A). Está Aguada como CT.P y tiene dos CT.A que son Abreus y Rodas, Cruces como CT.P y sus dos CT.A Lajas y Palmira. Cumanayagua y Cienfuegos los CT.P restantes.

Dentro de la división territorial a nivel de Empresa está el departamento de Desarrollo y Operaciones de la Red que tiene además de dos grupos de planta una interior y otra exterior, un grupo de Desarrollo y un grupo de Soporte a la Operación, dentro de este último grupo está la unidad y centro de Taller de Reparaciones de Estaciones Públicas que garantiza la explotación continua de los equipos terminales de telecomunicaciones y otros periféricos, incluyendo las estaciones públicas del territorio.

Este taller tiene entre sus objetivos restablecer la imagen de las Estaciones Públicas que por los años de explotación se ha deteriorado, de acuerdo con las políticas y estrategias a corto y mediano plazo aprobadas para la actividad. Para ello se llevan a cabo varias actividades entre ellas se encuentran la recepción y entrega de piezas a los diferentes municipios siendo esta una actividad crítica, debido a las largas jornadas de trabajo dedicadas a organizar la información, agruparla y procesarla. Además estos datos alcanzan un volumen considerable, que actualmente están guardados en Documentos Excel y Documentos Word sin existir la posibilidad de que esta información esté disponible para que pueda ser consultada por todas aquellas personas pertenecientes a la empresa cuando lo deseen. También se hace difícil obtener reportes capaces de consolidar la información adquirida y pueda ser entregada en un formato determinado, de tal manera que sea útil al usuario al cual va dirigida.

Teniendo en cuenta la situación problémica anterior se define como **problema científico**: ¿Cómo facilitar la gestión de la información relacionada con la reparación de piezas de telefonía pública en la D.T.CF?

Identificándose como **objeto de estudio** el proceso de control de reparación de piezas de telefonía pública en todas las D.T de ETECSA, dentro del cual se define como **campo de acción** el proceso de gestión de la información relacionada con el control de reparación de piezas de telefonía pública en el taller de la D.T.CF.

Considerando el análisis del problema se puede definir como **idea a defender**: la construcción de un sistema informático capaz de gestionar la información referente al control de reparación de piezas de telefonía pública en la D.T.CF, posibilitará una mejora significativa en el proceso.

El **objetivo general** de esta investigación es desarrollar un sistema informático que gestione la información relacionada con la reparación de piezas de telefonía pública en la D.T.CF.

Del objetivo general se definieron los siguientes **objetivos específicos** para lograr un mayor nivel de detalles en la investigación:

- Analizar los elementos que serán automatizados durante el proceso de reparación de piezas de telefonía pública en el taller de reparaciones de la D.T.CF.
- Diseñar un sistema que gestione la información del proceso de reparación de piezas de telefonía pública en el taller de reparaciones de la D.T.CF.
- Implementar el sistema propuesto mediante una aplicación Web.
- Validar el sistema.

Para dar cumplimiento a los objetivos se trazaron las siguientes **tareas**:

- Entrevistas a directivos y trabajadores de la Unidad del Taller de Telefonía Pública para conocer los principales procesos que tienen lugar en la realización de la gestión de los datos.
- Caracterización del problema y la situación actual para identificar las particularidades de los procesos relacionados con el control de piezas de telefonía pública en el taller de reparaciones de la D.T.CF.
- Selección de las metodologías, sistema gestor de base de datos y lenguajes de programación a utilizar.
- Diseño de la Base de Datos que contendrá la información necesaria para informatizar la reparación de las piezas de telefonía pública.
- Diseño de la interfaz de la aplicación.
- Diseño de la encuesta.
- Procesamiento de la encuesta que se aplicó al sistema informático.
- Documentación de la información generada durante el análisis, diseño, implementación y validación del sistema.
- Instalación del sistema informático para su etapa de prueba.

Esta solución tendrá un significativo **aporte práctico** ya que la D.T.CF será provista con un sistema informático que permitirá a los usuarios del mismo generar sus propios reportes de recepción y entrega de una manera sencilla y en un ambiente amigable, sin necesidad de un alto nivel de conocimiento informático. Además la información adquirida en los reportes podrá ser mostrada en diferentes formatos, de tal manera que sea útil al usuario al cual va dirigida, contribuyendo a la toma de decisiones por parte de los mismos. Todo esto conlleva a una considerable reducción de los costos y errores por automatizar tareas de recolección y manipulación humana de información.

Resumen capitular:

Capítulo 1: Fundamentación Teórica. Se describe el objeto de estudio, se especifican los detalles de la construcción de la herramienta y la propuesta del sistema. También incluye una descripción del lenguaje a utilizar para la implementación del sistema; la herramienta utilizada para diseñar la interfaz gráfica; y el sistema de gestión de Bases de Datos. También se describe la metodología que se utilizada para realizar el análisis y diseño del sistema y el lenguaje para modelar, construir y documentar los elementos que conforman el software.

Capítulo 2: Descripción del Sistema. Se describen los procesos del negocio, se enumeran las funcionalidades previstas y los requerimientos no funcionales. Se tienen en cuenta los principios de diseño para crear la interfaz de la aplicación. Se diseña e implementa el modelo lógico y físico de la base de datos. Por último se realiza el estudio de la factibilidad.

Capítulo 3: Construcción del Sistema. En este capítulo se realiza una descripción de la construcción de la solución propuesta haciendo uso de la descripción de la arquitectura y los diagramas de clases por cada capa. Se aborda todo lo relacionado con la seguridad del sistema y el tratamiento de errores. También se realiza la validación del sistema.

Capítulo 1.Fundamentación Teórica.

1.1 Introducción.

En este capítulo se realiza un estudio sobre la existencia de sistemas automatizados vinculados al campo de acción. También se incluye una descripción del lenguaje de modelado y la metodología de desarrollo a utilizar para la implementación del sistema. Además se describen las tecnologías y herramientas de desarrollo a utilizar; y el sistema Gestor de Bases de Datos.

1.2 Sistemas automatizados existentes vinculados al campo de acción.

En la actualidad el gran avance de las tecnologías de la informática y las comunicaciones, ha permitido desarrollar infinidad de aplicaciones de gestión relacionadas con el control y la recuperación de la información en todo el mundo.

El objetivo principal de dichas aplicaciones es el de almacenar grandes cantidades de información y asegurarse de que su recuperación se efectúe de la forma más rápida y óptima.

La presente investigación pretende enfocarse en la rama del control de reparaciones de piezas de telefonía pública.

Actualmente no existe ningún sistema informático en Cuba vinculado al campo de acción.

1.3 Fundamentación de la metodología utilizada y el lenguaje de modelado a considerar para la propuesta.

Es una realidad de nuestros días las tendencias de muchas empresas e instituciones a dedicar tiempo a pensar en el uso o no de algunas tecnologías para construir sistemas informáticos, ya que la correcta elección de estas, garantizarán la flexibilidad, robustez y eficiencia de los productos desarrollados.

1.3.1 Metodología de Desarrollo de Software.

Las metodologías de desarrollo de software definen el cómo trabajar eficientemente evitando las problemáticas que conllevan al fracaso de muchos proyectos de software. El objetivo fundamental de una metodología es aumentar la calidad del software a producir en todas las fases de desarrollo del mismo, haciendo énfasis en la calidad y menor tiempo de construcción del software o lo que es lo mismo “producir lo esperado en el tiempo esperado y con el costo esperado”.

Las metodologías de desarrollo de software se dividen en dos grupos, las llamadas “***pesadas***” y las que se conocen como “***ágiles***”, como sus nombres lo indican ambos grupos tienen marcadas diferencias y la razón del uso o no de alguna de ellas está dada en la medida que el equipo de desarrollo del software determina la grandeza del producto o la simplicidad del mismo.[1]

Las metodologías ***ágiles*** o también llamadas ***livianas*** tienen como características fundamentales el promover iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto minimizando los riesgos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración. Al final de cada iteración, el equipo vuelve a evaluar las prioridades del proyecto.

Existen varias metodologías de desarrollo de software calificadas como livianas o ágiles de las que se pueden mencionar *Crystal Clear, Scrum, eXtreme Programming y Feature Driven Development o* Desarrollo Basado en Funcionalidades (FDD) que fue la seleccionada para el desarrollo de este proyecto.[1]

FDD es un proceso diseñado por Peter Coad, Erich Lefebvre y Jeff de Luca y se podría considerar a medio camino entre RUP y XP, aunque al seguir siendo un proceso ligero es más similar a este último.

FDD está pensado para proyectos con tiempo de desarrollo relativamente corto (menos de un año). Se basa en un proceso iterativo de iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.

Las iteraciones se deciden sobre la base a *features* (de ahí el nombre del proceso) o funcionalidades, que son pequeñas partes del software con significado para el cliente.

Un proyecto que sigue FDD se divide en 5 faces:

- Desarrollo de un modelo general.
- Construcción de una lista de funcionalidades.
- Planeación por funcionalidad.
- Diseño por funcionalidad.
- Construcción por funcionalidad.

Las primeras tres fases ocupan parte del tiempo en las primeras iteraciones, siendo las dos últimas las que absorben la mayor parte del tiempo según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento.[2]

1.3.2 Lenguaje de Modelado Unificado, UML.

Lenguaje de Modelado Unificado: Este lenguaje conocido por sus siglas en inglés como UML, permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Este lenguaje tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue. UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Entre sus objetivos fundamentales se encuentran:

- Ser tan simple como sea posible, pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.
- Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son el encapsulamiento y el uso de componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.[3]

1.4 Aplicaciones Web.

En la ingeniería de software se denomina aplicación Web a aquellas aplicaciones que los usuarios pueden utilizar accediendo mediante el Protocolo de Transferencia de HiperTexto, por sus siglas en inglés (HTTP), a un servidor Web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores Web en la que se confía la ejecución al navegador.

1.5 Elementos de Arquitectura.

El concepto de arquitectura de software está muy extendido, sin embargo no existe una definición única y estándar para este concepto. Un autor, cuya contribución en el mundo de las arquitecturas de software y las componentes ha sido muy aceptada, es David Garlan. Su definición de arquitectura es una de las más utilizadas en la literatura.

“La arquitectura de software está compuesta por la estructura de los componentes de un programa o sistema, sus interrelaciones, y los principios y reglas que gobiernan su diseño y evolución a lo largo del tiempo.”

Por las características del proyecto y por las tendencias de las tecnologías en la actualidad se pretende implementar una arquitectura Modelo - Vista - Controlador y se empleará la programación orientada a objetos pues la aplicación será desarrollada con PHP5, este es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones, está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento. Se hará uso además de frameworks que organicen toda la estructura de la aplicación y minimicen el tiempo de desarrollo.

Modelo Vista Controlador (MVC)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón se observa frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la capa de acceso a datos, y el controlador es el responsable de recibir los eventos de entrada desde la vista y de la lógica del negocio.[4]

1.6 Uso de Lenguajes y Tecnologías Web.

1.6.1 Lenguajes y Tecnologías del lado del cliente.

HTML: acrónimo inglés de Hypertext Markup Language (lenguaje de marcado de hipertexto), es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

El HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos. Este lenguaje nos permite aglutinar textos, sonidos e imágenes y combinarlos a nuestro gusto. Además, y es aquí donde reside su ventaja con respecto a libros o revistas, el HTML nos permite la introducción de referencias a otras páginas por medio de los enlaces hipertexto.[5]

JavaScript: es un lenguaje interpretado, al igual que Visual Basic, Perl, TCL. (Lenguajes de script) sin embargo, posee una característica que lo hace especialmente idóneo para trabajar en Web, ya que son los navegadores que se utilizan para viajar por ella los que interpretan los programas escritos en JavaScript. De esta forma, se puede enviar documentos a través de la Web que llevan incorporados el código fuente de programas, convirtiéndose de esta forma en documentos dinámicos, y dejando de ser simples fuentes de información estáticas.

Las dos principales características de JavaScript son que es un lenguaje basado en el paradigma de programación orientada a objetos, aunque con menos restricciones, y es además un lenguaje orientado a eventos, debido por supuesto al tipo de entornos en los que se utiliza (Windows y sistemas X-Windows). Esto implica que gran parte de la programación en JavaScript se centra en describir objetos y escribir funciones que respondan a movimientos del Mouse, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos.[6]

Ajax: es una combinación de JavaScript, que trabaja del lado del cliente, y de lenguajes que procesan la información en el servidor y la entregan como una cadena de texto o en un archivo XML, en realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML".

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano; brinda más rapidez en las operaciones y está más cerca de crear realmente "Aplicaciones Web" permitiendo que estas sean más atractivas al usuario.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

AJAX tiene a su favor también que es independiente del tipo de tecnología de servidor que se utilice, funciona en cualquier navegador, es perfectamente compatible con cualquier tipo de servidor estándar y lenguaje de programación Web. PHP, ASP. etc. El ser completamente compatible el desarrollo en éstas tecnologías ha ayudado a AJAX a que vaya cada vez más en auge.[7]

Dojo: es un framework que contiene Apis y widgets para facilitar el desarrollo de aplicaciones Web que utilicen tecnología AJAX. Contiene un sistema de empaquetado inteligente, los efectos de UI, drag and drop Apis, widget Apis, abstracción de eventos, almacenamiento de APIs en el cliente, e interacción de Apis con AJAX.

Resuelve asuntos de usabilidad comunes como pueden ser la navegación y detección del navegador, soportar cambios de URL en la barra de URLs para luego regresar a ellas, y la habilidad de degradar cuando AJAX/JavaScript no es completamente soportado en el cliente. Es conocido como "la navaja suiza del ejército de las bibliotecas JavaScript". Proporciona una gama más amplia de opciones en una sola biblioteca JavaScript.[8]

CSS: el concepto de hojas de estilo apareció por primera vez en 1996 cuando W3C publicó una recomendación nueva intitulada "Hojas de estilo en cascada" o CSS, según sus siglas en inglés.

El principio de las hojas de estilo consiste en la utilización de un solo documento para almacenar las características de presentación de las páginas asociadas a grupos de elementos. Esto implica nombrar un conjunto de definiciones y características de presentación de las páginas, y activar esos nombres para aplicarlos a una parte del texto.

Las hojas de estilo pueden utilizarse para lograr una apariencia uniforme de todo el sitio al activar una sola definición de estilo en cada página, cambiar un aspecto en todo el sitio Web con tan sólo editar unas pocas líneas, hacer que los códigos HTML sean más fáciles de leer ya que los estilos se definen por separado, permitir que las páginas se carguen más rápido ya que hay menos cantidad de HTML en cada página y posicionar los elementos de la página de una manera más uniforme.[9]

1.6.2 Lenguajes y Tecnologías utilizadas del lado del servidor.

PHP: lenguaje interpretado de alto nivel embebido en páginas web HTML y ejecutado en el servidor. Es multiplataforma y se utiliza por la comunidad de

programadores para generar páginas web con contenidos dinámicos. El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.[10]

CodeIgniter 2.1.0: Framework para desarrollo de aplicaciones web usando PHP. Se creó con el objetivo de permitirle a los usuarios desarrollar proyectos mucho más rápido que lo que podría hacer si escribiese el código desde cero, proveyéndole un rico conjunto de bibliotecas para tareas comunes, así como una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. CodeIgniter 2.1.0 les permite a sus usuarios enfocarse creativamente en sus proyectos al minimizar la cantidad de código necesaria para realizar una tarea dada.

Los motivos fundamentales para la selección de este framework son los siguientes:

- Se necesitaba un framework con una pequeña impronta.
- Se necesitaba un framework con un desempeño excepcional.
- Existía la necesidad de una alta compatibilidad con cuentas estándar de alojamiento que corren en una variedad de versiones de PHP.
- Se necesitaba un framework que casi no llevase configuración.
- Se necesitaba un framework que no obligase a usar la línea de comandos.
- Se necesitaba un framework que no obligara a escribir reglas de codificación restrictivas.
- Se necesitaba un framework que evitase la complejidad, favoreciendo las soluciones simples.
- Se necesitaba una documentación clara y completa.[11]

1.7 Sistema Gestor de Base de Datos.

PostgreSQL: es un sistema de gestión de base de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de formas desinteresadas, altruistas, libres y/o apoyadas por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Una de sus principales características es:

Alta concurrencia

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.[12]

1.8 Servidor Web, Apache.

Apache: es uno de los servidores más utilizados en la actualidad según las estadísticas, la mayoría de los sitios que se encuentran en estado activo en Internet están soportados por Apache.

Posee un gran número de funcionalidades como son:

- Gran estabilidad, seguridad y facilidad de expansión, además que es un software libre por tanto no es necesario el pago por la obtención de su licencia.
- Es soportado por diversos sistemas operativos como Linux, Solaris, Rhapsody, BeOS, OD/2, Windows, entre otros.
- Permite la personalización de variables de entorno además de soportar la reparación o depuración de errores, lo cual no es muy común en otros servidores.

- Contiene un índice de directorios además de un directorio de alias.
- Brinda un informe de errores HTTP que pueden ser configurables.
- Permite la integración de imágenes del lado del servidor.
- Gestiona recursos para procesos hijos.
- Ejecuta la identificación de los usuarios de programas CGI.[13]

1.9 Herramientas utilizadas.

Visual Paradigm: es una herramienta CASE, soporta el ciclo de vida completo del desarrollo de software y permite a varios usuarios trabajar sobre el mismo proyecto. Dentro de las características del Visual Paradigm se destacan su robustez, usabilidad y portabilidad. Permite realizar diferentes tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta apoya las notaciones de Java y UML, y está diseñada para un amplio grupo de usuarios, incluyendo Ingenieros de software, Analistas de sistema, Analistas de negocio y Arquitectos de sistema, usuarios que están interesados en realizar software a grandes escalas y siguiendo el paradigma de la programación orientada a objeto.[14]

Macromedia Dreamweaver CS4: Es un editor WYSIWYG (What You See Is What You Get) de páginas web, creado por Macromedia. Es el programa de este tipo más utilizado mundialmente en el sector del diseño y la programación web por sus funcionalidades, su integración con otras herramientas como Macromedia Flash y por su soporte de los estándares del World Wide Web Consortium. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas.[15]

1.10 Entorno de Desarrollo Integrado, NetBeans.

NetBeans: es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.
NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. SunMicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.[17]

1.11 Herramienta de Diseño Gráfico, GIMP.

El tratamiento de imágenes se hace necesario cada vez más en un entorno de trabajo donde la elaboración de documentos electrónicos es básica para cualquier tarea diaria. GIMP es una herramienta, de gran potencia y reconocido prestigio en el entorno del software libre. Se trata de un programa que permite retocar fotografías, crear imágenes para la Web y otras muchas utilidades. Se presenta bajo licencia libre, lo que permite al usuario distribuirlo y adaptarlo a sus necesidades. Otra característica interesante es su disponibilidad tanto para plataformas Linux como Windows con lo que se garantiza la posibilidad de una implantación progresiva y adaptada a las necesidades de la administración. La

interfaz de GIMP está disponible en varios idiomas, entre ellos: español, inglés, catalán, gallego, euskera, francés, italiano, ruso, sueco, noruego, coreano y neerlandés.

GIMP es también conocido por ser quizás la primera gran aplicación libre para usuarios no profesionales o expertos. Productos originados anteriormente, como GCC, el núcleo Linux, etc., eran principalmente herramientas de programadores para programadores. GIMP es considerado por algunos como una demostración fehaciente de que el proceso de desarrollo de software libre puede crear aplicaciones que los usuarios comunes, no avanzados, pueden usar de manera productiva. De esta forma, GIMP ha abierto el camino a otros proyectos como KDE, GNOME, Mozilla Firefox, OpenOffice.org y otras aplicaciones posteriores.[18]

1.12 Conclusiones.

Para la construcción de la aplicación se realizó un estudio de las principales herramientas, lenguajes, metodologías y tecnologías que rigen en la actualidad el desarrollo de software. Se llegó a la conclusión de escoger a FDD como metodología de desarrollo y UML como lenguaje de modelado, PostgreSQL como gestor de base de datos, Dojo Toolkit 1.8 como marco de trabajo para la implementación del lado del cliente utilizando JavaScript y CodeIgniter 2.1.0 como su homólogo del lado del servidor para estructurar y dirigir la codificación en PHP. Para hospedar el sistema se escogió el servidor web Apache debido a su alta estandarización en los servicios de hospedaje que existen en la actualidad.

Capítulo 2. Descripción del Sistema.

2.1 Introducción.

La descripción de las distintas fases de la metodología FDD, del flujo actual de los procesos involucrados en el campo de acción junto con el objeto de automatización, la elaboración del modelo de objeto, lógico y físico de la base de datos, la especificación de los requerimientos que debe tener el sistema, el estudio de factibilidad del producto donde se estiman el esfuerzo humano y el tiempo de desarrollo que se requiere para el mismo, así como los costos y los beneficios tangibles e intangibles que reporta la utilización del sistema son los aspectos que se tomarán en cuenta para la confección del siguiente capítulo.

2.2 Desarrollo de un modelo global.

Al inicio de esta fase ya se tiene una idea del contexto y los requerimientos del sistema. Posteriormente el dominio global es dividido en diferentes áreas y se realiza un informe detallado para cada una de dichas áreas por parte de los expertos del dominio. Luego de cada informe, un grupo de desarrolladores realizan un modelo de objetos para el área del dominio. Además, el equipo de desarrollo discute y decide cual es el modelo de objetos más apropiado para cada área del dominio. Simultáneamente, el modelo global del sistema es construido.[2]

2.2.1 Flujo actual de los procesos involucrados al campo de acción.

En el área del taller de reparaciones de la D.T.CF se llevan a cabo varias actividades entre ellas se encuentran la recepción y entrega de piezas de telefonía pública a los diferentes municipios, siendo esta actividad la que se desea automatizar.

La actividad del registro de las piezas funciona de la siguiente forma:

La Unidad del taller de reparaciones de la D.T.CF desea conocer que cantidad de piezas son entregadas por cada municipio y en qué estado se encuentran las piezas que sean particularmente Microteléfono para tener información de esta actividad, además de poder conocer el nivel de vandalismo por cada municipio.
El personal encargado de traer las piezas es recibido por el técnico que está al frente del taller, el cual cuenta la cantidad de piezas con su modelo correspondiente, seguido a esto introduce la información referente a dichas piezas y la guarda en un documento Excel. A la hora de tomar decisiones los directivos del taller se les dificultaban puesto que tienen que imprimir cada documento para tener constancia de la entrada de piezas realizada en determinada ocasión. En dicha Unidad la pieza en particular que sea un Microteléfono se identificará cual es la rotura que tiene considerando estas como un hecho vandálico, que en este caso puede ser de cuatro formas, como son:

- Sin Cable
- Sin Tapa
- Sin Capsula
- Partido

La actividad de entrega de las piezas funciona de la siguiente forma:

La persona encargada llegará cualquier día para recoger todas o las piezas que estén arregladas según centro telefónico y la fecha en la que fueron recepcionadas. Se entregaran las piezas que hayan sido arregladas y las otras cuando estén reparadas. Todo este proceso queda registrado en un documento Excel que se puede imprimir en caso de alguna toma de decisión por parte de los directivos del Taller.

2.2.2 Objeto de automatización.

Se desea automatizar los procesos relacionados con la reparación de las piezas de telefonía pública en el Taller de la D.T.CF. Entre los cuales resalta el proceso de recepción de las piezas siendo los usuarios del taller los encargados de registrarlas y la entrega de dichas piezas según hayan logrado repararse. Asimismo la elaboración de un reporte que tenga consigo toda la información que se recibió por parte de la persona que hizo la entrega de las piezas, donde se podrá conocer cualquier dato que se desee investigar.

2.2.3 Modelo de dominio. Diagrama de Objetos.

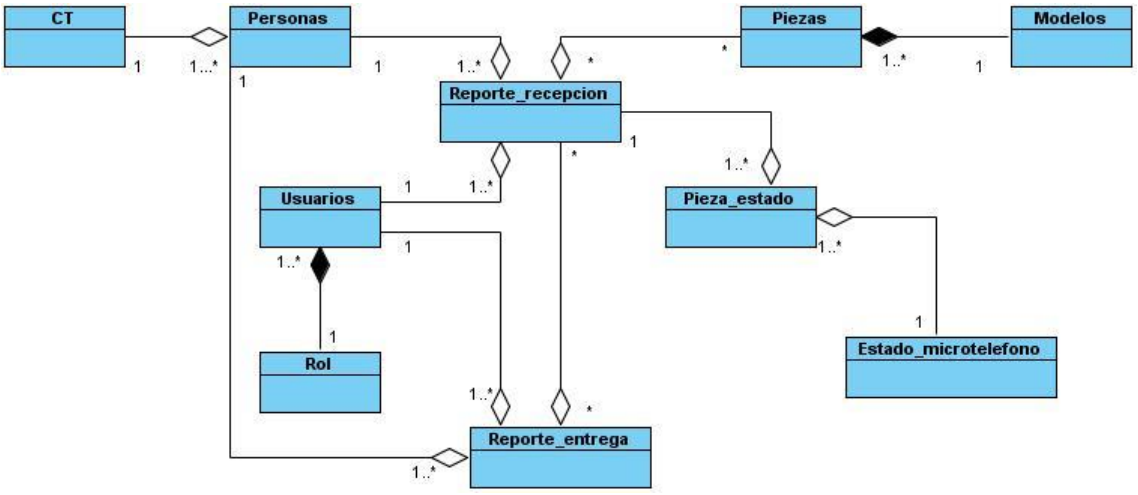

Figura 2.2 Modelo de objetos.

2.2.4 Reglas a considerar.

- El Técnico del Taller es el encargado de entregar las piezas.
- Las actividades de insertar, eliminar o modificar centros telefónicos, piezas, modelos serán realizada por los Técnicos del taller.
- Las actividades de insertar, eliminar o modificar usuarios solo la podrá realizar el Administrador del SW que será uno de los cuatro trabajadores.

- El personal podrá recoger las piezas cualquier día siempre que estas se hayan recepcionado en el taller que se identificara con un número de serie aleatorio.
- El Técnico del Taller es el encargado de recepcionar las piezas.

2.3 Construcción de una lista de funcionalidades.

El modelo de objetos y los requerimientos existentes ofrecen una buena base para construir una lista de funcionalidades que resuma la funcionalidad del sistema a ser desarrollado. En dicha lista, el equipo de desarrolladores presenta cada una de las funcionalidades evaluadas por el cliente.[2]

2.3.1 Requerimientos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. Son capacidades o condiciones que el sistema debe cumplir definiendo el comportamiento interno del software así como los comportamientos del sistema.

Funcionalidades del Software:

Nro.	Funcionalidad	Prioridad
1	Insertar Usuario	A
2	Eliminar Usuario	A
3	Modificar Usuario	A
4	Listar Usuarios	A
5	Autenticarse	A
6	Cambiar Contraseña	A
7	Listar Roles	A
8	Insertar Modelo	A
9	Eliminar Modelo	A
10	Modificar Modelo	A

11	Listar Modelos	A
12	Insertar Persona	A
13	Eliminar Persona	A
14	Modificar Persona	A
15	Listar Personas	A
16	Buscar Persona	A
17	Insertar Pieza	A
18	Eliminar Pieza	A
19	Modificar Pieza	A
20	Listar Piezas	A
21	Insertar CT	A
22	Modificar CT	A
23	Eliminar CT	A
24	Listar CT	A
25	Insertar Estado	A
26	Modificar Micro	A
27	Eliminar Estado	A
28	Listar Estados	A
29	Insertar Reporte de Recepción	A
30	Eliminar Reporte de Recepción	A
31	Listar Reporte de Recepción	A
32	Insertar Estado pieza Microteléfono	A
33	Insertar Reporte de Entrega	A
34	Eliminar Reporte de Entrega	A
35	Listar Reporte de Entrega	A
36	Imprimir Reporte Recepción	A
37	Imprimir Reporte Entrega	A
38	Exportar Reporte Recepción a PDF	A
39	Exportar Reporte Entrega a PDF	A
40	Graficar las piezas que más se rompen por CT y Modelo	A
41	Graficar la cantidad de Recepción y Entrega de piezas	A
42	Graficar el nivel de Vandalismo en los CT por períodos	A

Tabla 2.3 Bloque de funcionalidades.

2.3.2 Requerimientos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto.[1]

Los requerimientos no funcionales del sistema propuesto son los siguientes:

Requerimientos de apariencia o interfaz externa:

- El sistema debe tener una interfaz sencilla, legible y simple de usar.
- La interfaz debe ser diseñada respetando parámetros de diseño.
- La interfaz del sistema se visualizará a través de una página web, personalizada a partir del tipo de usuario que acceda a la misma.

Requerimientos de usabilidad:

- La interfaz de usuario deberá ser tan familiar como sea posible a otras aplicaciones similares que los usuarios hayan utilizado.
- El software debe ser fácil de usar por personas que no tengan mucha experiencia con computadoras.

Requerimientos de rendimiento:

- El sistema propuesto debe ser rápido en el procesamiento de la información así como a la hora de dar respuesta a las solicitudes de los usuarios.
- Los tiempos de respuestas a las peticiones de información tabular deben ser prácticamente instantáneos.

Requerimiento de Portabilidad:

La aplicación propuesta será desarrollada en la plataforma Windows. No obstante los clientes podrán instalar la aplicación en el Sistema Operativo Linux o Windows desde versión 2000 a través de un servidor web y un servidor de

base de datos que soporten el lenguaje PHP y el sistema gestor de base de datos PostgreSQL respectivamente.

Requerimientos de seguridad

Se debe garantizar un control estricto sobre la seguridad de la información teniendo en cuenta el establecimiento de niveles de acceso. Se define una política de usuarios con privilegios de acuerdo a su rol, lo que asegura que la información pueda ser consultada de acuerdo a su nivel de acceso.

Accederán de manera rápida y operativa los usuarios sin que los requerimientos de seguridad se conviertan en un retraso para usar el sistema.

Se utilizará un mecanismo de encriptación para las contraseñas que por cuestiones de seguridad no deben viajar al servidor en texto plano. Se guardará encriptado en la base de datos usando para ello la función hash MD5.

La función hash es un tipo de algoritmo que calcula un mensaje implícito o valor hash para cualquier mensaje que se le indique. El valor generado por el algoritmo no es importante, lo importante es que el resultado sea fijo, es decir, que sea el mismo cada vez que se utiliza una entrada dada, que sea pequeño y que el algoritmo sea rápido. La función MD5 calcula y devuelve el hash MD5 que consiste en una cadena de 32 números en formato hexadecimal.

Requerimiento de Hardware:

- Las máquinas que utilizarán el software tienen que estar conectadas a la red.
- Los requerimientos mínimos para una estación de trabajo deben ser de 128 MB de RAM, velocidad del procesador de 800 MHz, 2GB de disco duro.
- Los requerimientos mínimos para el servidor del software deben ser de 1 GB de RAM, 15 GB de disco duro y 1.4 GHz de velocidad del procesador.

Requerimientos de Software:

- Los requerimientos mínimos para una estación de trabajo deben ser:
    - Navegador Mozilla Firefox 7.0 o superior.
    - Google Chrome 31.0 o superior.
- Se requiere además, una máquina servidor que cuente con:
    - La versión 2.2.6 o superior del servidor Web Apache.
    - Lenguaje de programación PHP 5.2.5
    - Servidor de base de datos PostgresSQL 8.3.4

2.4 Estudio de la factibilidad.

Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de función. Esta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas.

Por otro lado, el SEI (del inglés, Software Engineering Institute) propone desde hace algunos años un método para la estimación del esfuerzo llamado COCOMO II. Éste método está basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code.[19]

En este epígrafe se propone la utilización combinada de dos métodos (Puntos de Función y Cocomo II) los cuales tienden a proporcionar una estimación más precisa de la siguiente forma:

Primero la aplicación del método de Puntos de Función para determinar las sentencias de código del proyecto software.

En segundo lugar la aplicación del método COCOMO II partiendo de la información producida por el anterior para llegar a una estimación precisa de las horas hombre a aplicar y fundamentalmente a la estimación de la duración total del proyecto.

Después clasificar cada una de las funcionalidades previstas se arribó a los siguientes resultados:

Nombre de la entrada externa	*Datos Elementales Referenciados*	*Registros Lógicos Referenciados*	*Complejidad (Simple, Medio y Complejo)*
Insertar Usuario	7	1	simple
Eliminar Usuario	1	1	simple
Modificar Usuario	5	1	simple
Autenticarse	2	1	simple
Cambiar Contraseña	4	1	simple
Insertar Modelo	1	1	simple
Eliminar Modelo	1	1	simple
Modificar Modelo	1	1	simple
Insertar Persona	5	1	simple
Eliminar Persona	1	1	simple
Modificar Persona	5	1	simple
Insertar Pieza	2	1	simple
Eliminar Pieza	1	1	simple
Modificar Pieza	2	1	simple
Insertar CT	1	1	simple
Eliminar CT	1	1	simple
Modificar CT	1	1	simple
Insertar Estado	1	1	simple
Eliminar Estado	1	1	simple
Modificar Estado	1	1	simple
Insertar Reporte Recepción	9	2	medio
Eliminar Reporte Recepción	1	2	simple
Insertar Reporte Entrega	7	2	medio

Eliminar Reporte Entrega	1	2	simple
Insertar Estado pieza	1	1	simple

Tabla 2.4 Entradas externas.

Nombre de la salida externa	*Datos Elementales Referenciados*	*Registros Lógicos Referenciados*	*Complejidad (Simple, Medio y Complejo)*
Graficar las piezas que más se rompen por CT y Modelo.	5	3	simple
Graficar la cantidad de Recepción y Entrega de piezas.	4	3	simple
Graficar el nivel de Vandalismo en los CT por períodos.	5	6	medio
Exportar Reporte de Recepción a PDF	5	6	medio
Exportar Reporte de Entrega a PDF	5	6	medio

Tabla 2.5 Salidas externas.

Nombre de la consulta externa	*Datos Elementales Referenciados*	*Registros Lógicos Referenciados*	*Complejidad (Simple, Medio y Complejo)*
Listar Usuarios	5	1	simple
Listar Roles	1	1	simple
Listar Modelos	1	1	simple
Listar Personas	5	1	simple
Buscar Persona	1	1	simple
Listar Piezas	2	1	simple
Listar CT	1	1	simple
Listar Estados	1	1	simple

Listar Reporte Recepción	3	1	simple
Listar Reporte Entrega	3	1	simple
Imprimir Reporte Recepción	6	4	complejo
Imprimir Reporte Entrega	6	4	complejo

Tabla 2.6 Consultas externas.

Nombre del fichero lógico interno	*Datos Elementales Referenciados*	*Registros Lógicos Referenciados*	*Complejidad (Simple, Medio y Complejo)*
Usuarios	8	2	simple
Rol	2	1	simple
Centro Telefónico	2	1	simple
Personas	6	2	simple
Modelos	2	1	simple
Piezas	3	2	simple
Estado Microteléfono	2	1	simple
Reporte Recepción	4	3	simple
Reporte Entrega	4	3	simple

Tabla 2.7Archivos lógicos internos.

Elementos	*Baja*	*X Peso*	*Media*	*X Peso*	*Alta*	*X Peso*	*Subtotal de puntos de función*
Entradas externas	23	3	2	4	0	6	77
Salidas externas	2	4	3	5	0	7	23
Consultas externas	10	3	0	3	2	6	42
Archivos Lógicos Internos	9	7	0	10	0	15	63
						Total	205

Tabla 2.8 Puntos de función.

$$UFP = 205$$

$$PM_{nominal} = A \times (Size)^{B}$$

A: se toma el valor por defecto del modelo, ajustado en 2.94.

Size: se calcula como el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Se utiliza PHP (factor de conversión medio para lenguajes programación web= 20 SLOC/UFP). Entonces:

$$Size = UFP \times FC$$

$$Size = 205 \times 20 = 4100\ SLOC = 4.1 KLOC$$

B: se calcula ponderando las variables escalares mediante la ecuación siguiente:

$$B = 0.91 + 0.01 \times \sum (W_i)$$

donde las **Wi** se muestran en la siguiente tabla:

Variable	*Ponderación*	*Valor*
PREC	Alto	2.40
FLEX	Nominal	3.04
RESL	Alto	2.83
TEAM	Alto	3.29
PMAT	Nominal	4.68

Tabla 2.9 Variables escalares.

Factores de escala

$$SF = \sum (W_i) = PREC + FLEX + RESL + TEAM + PMAT$$

$$SF = \sum (W_i) = 2.4 + 3.04 + 2.83 + 3.29 + 4.68 = 16.24$$

$$B = 0.91 + 0.01 \times 16.24$$

$$B = 1.0724 \approx 1.07$$

Luego el esfuerzo nominal resulta:

$$\mathbf{PM_{nominal} = A \times (Size)^{B}}$$

$$\boldsymbol{PM_{nominal} = 2.94 \times (4.1\ \text{KLOC})^{1.07} \approx 13.31\ Meses - Hombre}$$

Para completar la estimación hay que ajustar el esfuerzo nominal de acuerdo a las características del proyecto. El ajuste se efectúa aplicando la ecuación siguiente:

$$\boldsymbol{PM_{ajustado} = PM_{nominal} \times \prod(ME_i)}$$

Donde los $\prod(ME_i)$ (multiplicadores de esfuerzo) varían en función del modelo de estimación seleccionado (Diseño Preliminar o Post arquitectura). En este caso se aplica el modelo de Diseño preliminar. Entonces, se cuantifican los multiplicadores de esfuerzo para este modelo de la siguiente forma:

Multiplicador	*Ponderación*	*Valor*
PERS	Alto	0.83
RCPX	Nominal	1
RUSE	Nominal	1
PDIF	Bajo	0.87
PREX	Alto	0.87
SCED	Nominal	1
FCIL	Muy Alto	0.73

Tabla 2.10 Multiplicadores de esfuerzo.

Multiplicador de esfuerzos

$$\boldsymbol{E_M = \prod(ME_i) = PERS \times RCPX \times RUSE \times PDIF \times PREX \times SCED \times FCIL}$$

$$\boldsymbol{E_M = \prod(ME_7) = 0.83 \times 1 \times 1 \times 0.87 \times 0.87 \times 1 \times 0.73}$$

$$\boldsymbol{E_M = 0.45860571 \approx 0.46}$$

Con estos valores, el ajuste del esfuerzo resulta:

$$\boldsymbol{PM_{ajustado} = 13.31 \times 0.46 = 6.1\ Meses - Hombre \approx 6\ Meses - Hombre}$$

Este resultado se interpreta como el tiempo requerido para que una persona desarrolle la aplicación.

Para el costo:

Se asume como salario promedio mensual $675.25.

CH: Cantidad de hombres = 2.

Tiempo: Tiempo total de desarrollo del software.

$$\text{Tiempo} = \text{PM}_{\text{ajustado}} \div \text{CH} = 6 \div 2 = 3 \text{ Meses}$$

$$\boldsymbol{Costo = SalarioPromedio \times CH \times Tiempo}$$

$$\boldsymbol{Costo = 675.25 \times 2 \times 3Meses = \$4051.50}$$

Cálculo	*Valor*
Esfuerzo nominal	$\boldsymbol{13.31\ Meses - Hombre}$
Esfuerzo ajustado	$\boldsymbol{6\ Meses - Hombre}$
Cantidad de hombres	2
Salario promedio	**$675.25**
Tiempo	**3 Meses**
Costo	$\boldsymbol{\$4051.50}$

Tabla 2.11 Resumen de resultados.

Sobre los resultados obtenidos se interpreta que con dos hombres trabajando en la aplicación se desarrolla en 3 meses y su costo total se estima que sea $4051.50 MN y/o 162.06 CUC.

2.4.1 Beneficios tangibles e intangibles.

Los beneficios obtenidos con el desarrollo del software permiten agilizar el proceso de entrega y recepción de las piezas de telefonía pública que se realiza en el taller de reparaciones en la D.T.CF, además brinda ayuda a los directivos del taller en la toma de decisiones y disminuye significativamente la posibilidad de errores de duplicidad o pérdida de la información. Uniendo esto a las ventajas de la digitalización y mejora de la calidad de la información por su integridad y confiabilidad. De esta manera se logra que los esfuerzos empleados en el desarrollo del sistema estén encaminados al cumplimiento de los objetivos planteados.

2.4.2 Análisis de costos y beneficios.

La incorporación del sistema propuesto traerá grandes beneficios como se evidenció en el epígrafe anterior. Permitirá entre otros reducir la pérdida de información por deterioro de documentación, mayor rapidez y confiabilidad del proceso, así como la obtención de reportes y gráficos de fácil entendimiento para los especialistas y para asistir a los directivos en la toma de decisiones. El desarrollo no supone grandes gastos de recursos y las tecnologías empleadas no requieren pagos de licencias.

Para la realización de este sistema no fue necesaria una inversión en los medios técnicos. Estos beneficios implican un ahorro del tiempo que se invierte en esta gestión y control de la información.

2.5 Planeación por funcionalidades.

En esta etapa se incluye la creación de un plan de alto nivel, en el cual la lista de funcionalidades es ordenada en base a la prioridad y a la dependencia entre cada funcionalidad. Además, las clases identificadas en la primera etapa son asignadas a cada programador.[2]

Nro.	Funcionalidad	Prioridad	Fecha Inicial	Fecha Final	Días
1	Insertar Usuario	A	5/12/2013	7/12/2013	2
2	Eliminar Usuario	A	7/12/2013	9/12/2013	2
3	Modificar Usuario	A	12/12/2013	15/12/2013	1
4	Listar Usuarios	A	15/12/2013	19/12/2013	1
5	Autenticarse	A	19/12/2013	22/12/2013	1
6	Cambiar Contraseña	A	9/1/2014	11/1/2014	1
7	Listar Roles	A	11/1/2014	13/1/2014	2
8	Insertar Modelo	A	16/1/2014	18/1/2014	1
9	Eliminar Modelo	A	18/1/2014	20/1/2014	2
10	Modificar Modelo	A	23/1/2014	25/1/2014	2
11	Listar Modelos	A	25/1/2014	27/1/2014	1

12	Insertar Persona	A	30/1/2014	1/2/2014	1
13	Eliminar Persona	A	1/2/2014	3/2/2014	1
14	Modificar Persona	A	6/2/2014	8/2/2014	2
15	Listar Personas	A	8/2/2014	10/2/2014	1
16	Buscar Persona	A	13/2/2014	15/2/2014	1
17	Insertar Pieza	A	15/2/2014	17/2/2014	1
18	Eliminar Pieza	A	20/2/2014	22/2/2014	2
19	Modificar Pieza	A	22/2/2014	24/2/2014	1
20	Listar Piezas	A	27/2/2014	29/2/2014	1
21	Insertar CT	A	29/2/2014	2/3/2014	1
22	Modificar CT	A	5/3/2014	7/3/2014	2
23	Eliminar CT	A	7/3/2014	9/3/2014	1
24	Listar CT	A	12/3/2014	14/3/2014	2
25	Insertar Estado	A	14/3/2014	16/3/2014	1
26	Modificar Micro	A	19/3/2014	21/3/2014	1
27	Eliminar Estado	A	21/3/2014	26/3/2014	1
28	Listar Estados	A	26/3/2014	28/3/2014	1
29	Insertar Reporte de Recepción	A	28/3/2014	30/3/2014	2
30	Eliminar Reporte de Recepción	A	2/4/2014	4/4/2014	2
31	Listar Reporte de Recepción	A	4/4/2014	6/4/2014	1
32	Insertar Estado pieza Microteléfono	A	9/4/2014	12/4/2014	2
33	Insertar Reporte de Entrega	A	12/4/2014	17/4/2014	2
34	Eliminar Reporte de Entrega	A	7/5/2014	8/5/2014	1
35	Listar Reporte de Entrega	A	8/5/2014	9/5/2014	2
36	Imprimir Reporte Recepción	A	9/5/2014	10/5/2014	3
37	Imprimir Reporte Entrega	A	10/5/2014	11/5/2014	3
38	Exportar Reporte Recepción a PDF	A	14/5/2014	16/5/2014	3
39	Exportar Reporte Entrega a PDF	A	16/5/2014	18/5/2014	3
40	Graficar las piezas que más se rompen por CT y Modelo	A	18/5/2014	20/5/2014	2
41	Graficar la cantidad de Recepción y Entrega de piezas	A	20/5/2014	22/5/2014	2
42	Graficar el nivel de Vandalismo en los CT por períodos	A	22/5/2014	24/5/2014	2

Tabla 2.12 planeación por funcionalidades.

2.6 Diseño por funcionalidades.

El diseño y construcción de la funcionalidad es un proceso iterativo durante el cual las funcionalidades seleccionadas son producidas. Una iteración puede llevar desde unos pocos días a un máximo de dos semanas. Este proceso iterativo incluye tareas como inspección del diseño, codificación, prueba unitaria, integración e inspección del código. Luego que la iteración llega a su fin se realiza una construcción de la funcionalidad en el cual esta es integrada.[2]

2.6.1 Principios de diseño.

Los elementos gráficos o textuales que componen la interfaz son claros y de fácil identificación. Se hizo uso de términos y conceptos que se tomaron de la experiencia de las personas que más utilizarán el sistema. Adicionalmente se tuvieron presente los siguientes aspectos:

- Las operaciones comparables se activan de la misma forma.
- El sistema no provoca sorpresa a los usuarios.
- Se incluyeron mecanismos para permitir a los usuarios recuperarse de los errores mediante la confirmación de acciones destructivas.
- La interfaz provee retroalimentación significativa y características de ayuda sensible al contexto.
- La interfaz provee características de interacción apropiada para los diferentes tipos de usuarios.
- La manipulación directa: Interacción directa con los objetos de la pantalla, rápida e intuitiva y fácil de aprender.
- Se requiere teclear poco.
- Pocas opciones en cada menú.
- Introducción de datos sencillos en los campos de un formulario.

2.6.2 Modelo lógico de la base de datos.

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo.

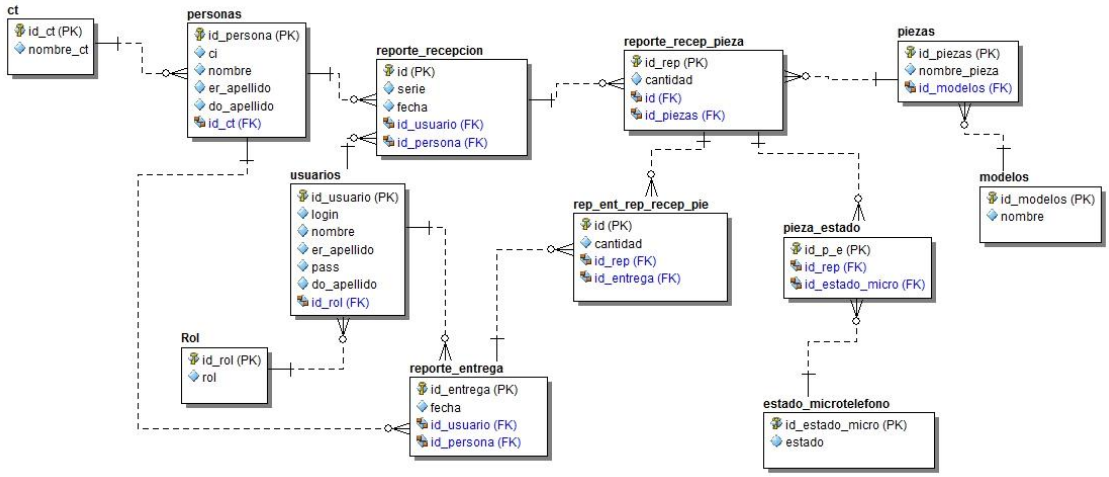

Figura 2.4 Modelo lógico de la base de datos.

2.6.3 Modelo físico de la base de datos.

El modelo físico de datos incluye todos los aspectos de diseño de un modelo de base de datos, que se pueden modificar sin cambiar los componentes de la aplicación.

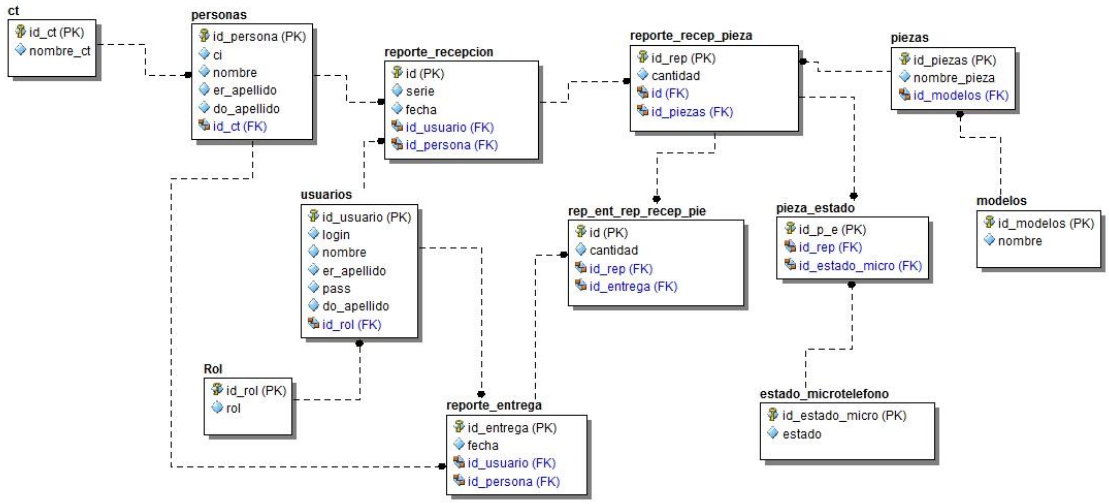

Figura 2.5 Modelo físico de la base de datos.

2.7 Conclusiones.

En este capítulo se crearon los artefactos correspondientes a las fases que propone la metodología FDD utilizando notación UML. Se analizó y describió el flujo de los procesos involucrados en el control de la información que se genera en el taller de reparaciones de piezas de telefonía pública en la D.T.CF. Se especificaron los requerimientos que debe tener el sistema y como parte del diseño por funcionalidades, se elaboró el modelo lógico y físico de la base de datos.

Asimismo se realizó un estudio de factibilidad del sistema el cual arrojó un costo estimado de $4051.50 MN y/o 162.06 CUC con un tiempo de desarrollo por dos persona de aproximadamente tres meses. Analizando el costo y los beneficios de la implantación del sistema y las mejoras que introduce en los procesos del taller de reparaciones de piezas de telefonía pública en la D.T.CF se concluyó que el desarrollo del proyecto es factible.

Capítulo 3. Construcción del Sistema.

3.1 Introducción.

En este capítulo se realiza una descripción del diseño del sistema utilizando los diagramas de clases. Se aborda todo lo relacionado con la seguridad del sistema y el tratamiento de errores. Adicionalmente se realiza la validación de la solución propuesta.

3.2 Descripción de la arquitectura propuesta.

CodeIgniter está basado en el patrón de desarrollo Modelo-Vista-Controlador. MVC es una aproximación al software que separa la lógica de la aplicación de la presentación.

El Modelo representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos. Se implementó una clase del modelo para cada tabla de la base de datos.[11]

La Vista es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página como un encabezado o un píe de página. Las vistas de la aplicación poseen elementos comunes que fueron definidos en la plantilla, además de las vistas diseñadas para cada una de las funcionalidades.[11]

El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web. Se definieron doce clases controladoras, de manera tal que cada una gestiona la información relacionada con una clase entidad, además una para la

administración y la principal que es la que se configura para ser cargada por la vista principal del sistema.[11]

Teniendo en cuenta que la aplicación se implementó utilizando como base CodeIgniter desde el punto de vista técnico y arquitectónico posee las siguientes características:

• **Bajo Acoplamiento:** Acoplamiento es el grado que los componentes de un sistema dependen entre ellos. Mientras menos componentes dependan de otro, más reusable y flexible el sistema se vuelva. [11]

• **Singularidad del Componente:** Singularidad es el grado que más componentes tienen un propósito en el que enfocarse más estrecho. Cada clase y sus funciones son altamente autónomas para permitir máxima utilidad.[11]

De manera general el sistema es poco acoplado con gran singularidad de componente. Posee simplicidad, flexibilidad y buen rendimiento.

3.3 Diagrama de clases.

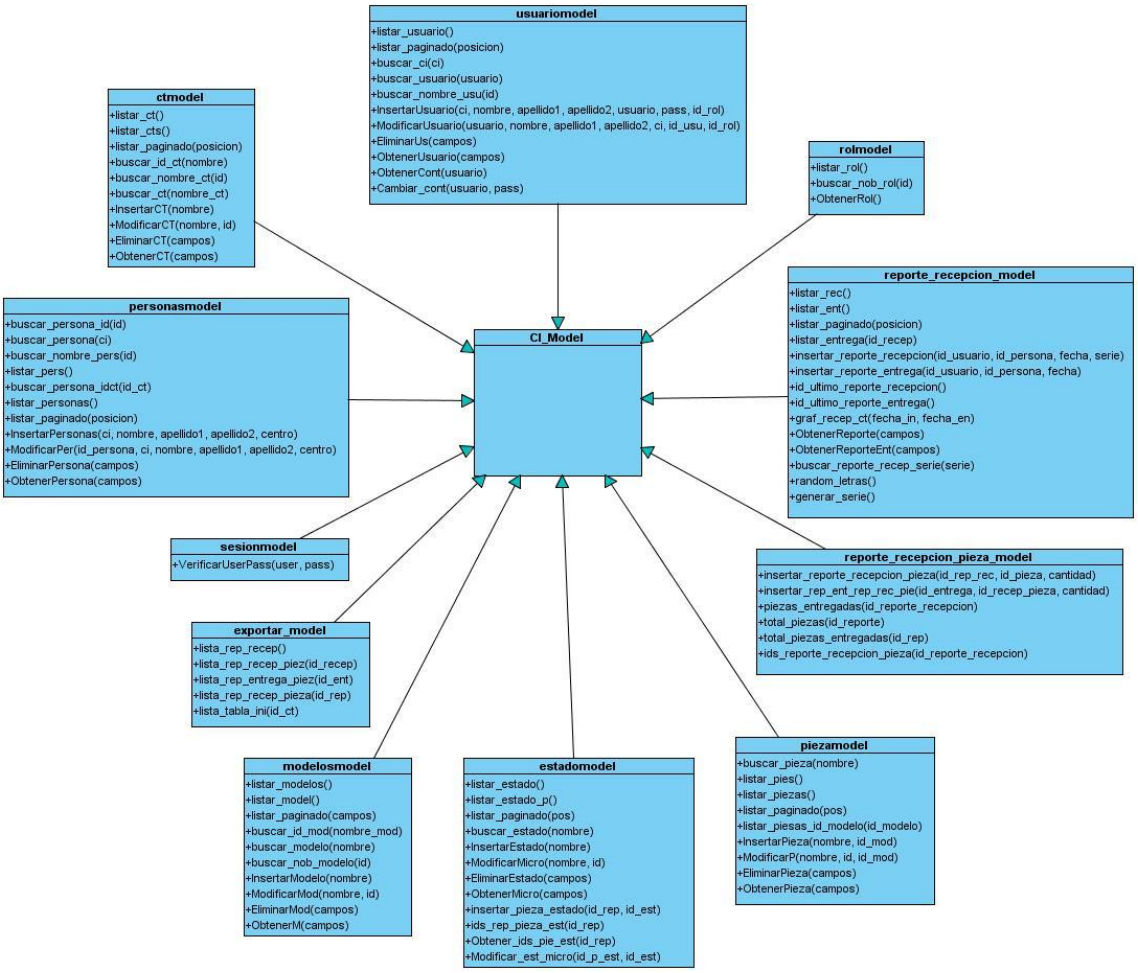

Figura 3.1 Diagrama de clases del modelo.

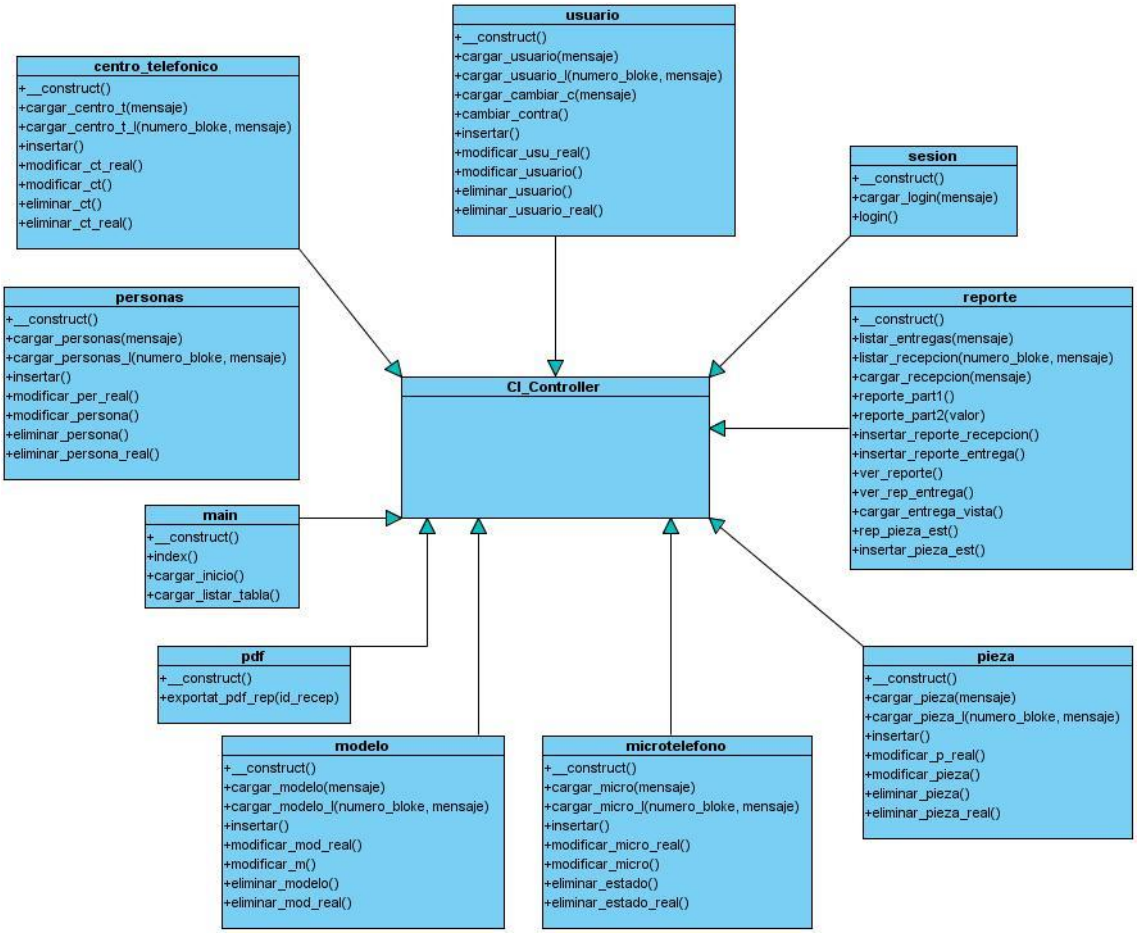

Figura 3.2 Diagrama de clases del controlador.

3.4 Estándar de implementación.

Para el adecuado mantenimiento del código del sistema es necesario establecer un estándar de codificación a emplear. Las variables, nombres de funciones, de clases y objetos fueron declarados en las páginas son cortos, claros, y describen claramente su propósito. Los objetos se nombran según el valor de su contenido. Los inicios ({) y cierre (}) de ámbito se encuentran alineados debajo de la declaración a la que pertenecen. Los signos lógicos y de operación se separan por un espacio antes y después de los mismos. Se emplean comentarios para añadir información sobre determinadas líneas de código en las páginas y así ayudar a entender el objetivo del mismo. Se emplean además nombres similares entre clases de manera tal que se establezca claramente la relación entre la clase del modelo, la clase controladora que maneja sus datos y la tabla de la base de datos de donde obtienen la información necesaria.

3.5 Seguridad del sistema.

- El mecanismo de seguridad y protección de la aplicación se basa en el empleo del nombre de usuario y la contraseña para acceder a las funcionalidades a las que tienen acceso.
- Cada usuario tiene definido a que módulo e información puede acceder, teniendo en cuenta la política de seguridad del taller de reparaciones de la D.T.CF.
- Los usuarios una vez registrados en su módulo no podrán acceder haciendo uso de la url a otra página a la cual no tiene acceso.
- Los usuarios una vez que se desconectan de su módulo no podrán regresar y realizar acciones dentro del mismo sin antes autentificarse.
- Se utilizaron mecanismos de encriptación de los datos usando para ello la función hash MD5.

3.6 Tratamiento de errores.

A menudo las personas que usan sistemas informáticos cometen errores que pueden afectar el adecuado funcionamiento de los mismos. Por lo que es necesario siempre realizar un adecuado tratamiento para estas situaciones excepcionales que pueden suceder. Como por ejemplo la entrada de datos incorrectos o en un formato no soportado por el sistema.

El sistema está diseñado e implementado de forma tal, que las posibilidades de introducir información errónea sean mínimas. Aunque en algunos casos es necesario teclear datos y seleccionar elementos, se mantiene un nivel elevado de validación. Se muestran comentarios sobre el formato de la información a introducir en las áreas de textos de las pantallas. Sobre los botones se colocan etiquetas informativas para brindarle información al usuario sobre cada funcionalidad. Adicionalmente a la validación, los mensajes de error que muestra el sistema se encuentran escritos en un lenguaje de fácil comprensión para los usuarios.

3.7 Diagrama de despliegue.

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.[1]

El siguiente diagrama muestra la configuración hardware del sistema y los nodos físicos que lo componen. El sistema estará estructurado según la arquitectura cliente - servidor. En el lado del servidor estarán en funcionamiento, el servidor de bases de datos PostgreSQL y el servidor Web Apache. Esta se comunicará con el cliente de la Intranet a través del protocolo HTTP. El cliente podrá visualizar la aplicación con el Mozilla Firefox 7.0 o superior. Además se cuenta con una impresora conectada a la máquina cliente que le brindará la posibilidad al usuario de imprimir la información que desee.

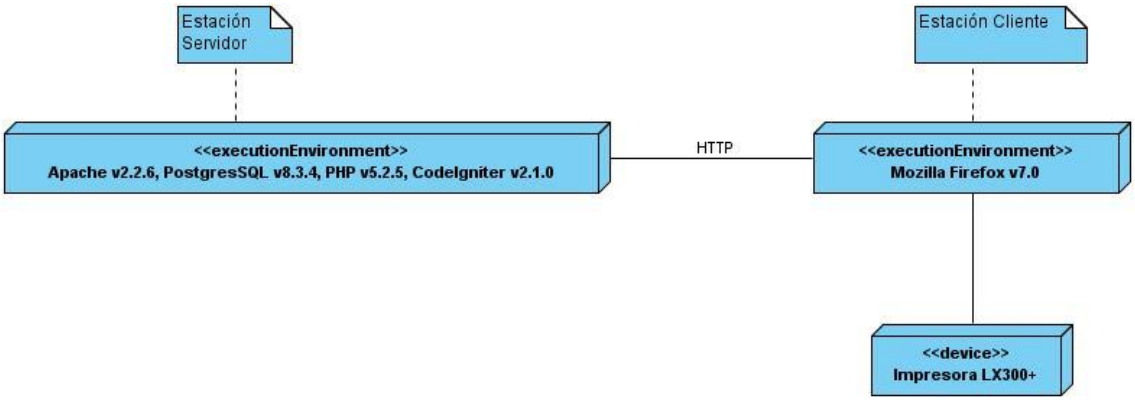

Figura 3.3 Diagrama de despliegue del sistema.

3.8 Validación de la solución propuesta.

Para validar la aplicación Web que se obtiene como resultado del desarrollo de este trabajo se confeccionó una encuesta, cuyas preguntas se diseñaron tomando en consideración determinados aspectos del software (Interfaz

gráfica, Usabilidad, Presentación, Motivación, Funcionalidad y Ayuda), utilizándose la siguiente escala de evaluación:

1 Mal.
2 Regular.
3 Bien.
4 Muy bien.
5 Excelente para cada variable.

Encuesta:

Encuesta sobre "Sistema informático para la gestión de las piezas de telefonía pública en el taller de reparaciones de ETECSA".

Estimado especialista la presente encuesta forma parte de la validación del producto "Sistema informático para la gestión de las piezas de telefonía pública en el taller de reparaciones de ETECSA" con la cual se pretende obtener sus opiniones que contribuirán a la validación del mismo. Muchas gracias por su participación.

Marque con una (X) en la escala que se adjunta a cada variable utilizando la siguiente leyenda:

1. Mal, 2. Regular, 3. Bien, 4. Muy bien, 5. Excelente

Indicadores a valorar	1	2	3	4	5
1. Interfaz gráfica **a.** ¿El software muestra una interfaz innovadora? **b.** ¿El color empleado es adecuado? **c.** ¿El tamaño y tipo de letra son adecuados? **d.** ¿Hay facilidad de navegación entre las distintas pantallas?					
2. Usabilidad **a.** ¿El sistema es fácil de usar para personas sin experiencia?					

b. ¿Es fácil entrar datos a los formularios?					
3. Presentación **a.** ¿Los reportes son precisos y claros?					
4. Motivación **a.** ¿Logra motivar por su manejabilidad? **b.** ¿La interactividad es apropiada para el usuario? **c.** ¿La presentación del software mantiene el interés de usuario hasta el final de la tarea?					
5. Funcionalidad **a.** ¿El software funciona correctamente en su ambiente? **b.** ¿Es adecuado el tiempo de respuesta a las acciones que realiza el usuario?					
6. Seguridad **a.** ¿Es adecuado el mecanismo de autentificación con contraseña encriptada? **b.** ¿El sistema proporciona niveles de acceso a la información por usuarios?					
7. Ayuda **a.** ¿Ofrece una guía detallada para el manejo del software?					

3.8.1 Resultados estadísticos de las encuestas realizadas.

Se utilizó muestreo no probabilístico intencional o a conveniencia, teniendo en cuenta que varias personas pueden acceder al sistema para consultar información.

Se encuestaron un total de 7 personas que fueron las que estaban presentes en el centro en el momento de aplicar la encuesta.

Resultados de la encuesta:

Una vez recogidos los resultados de las encuestas aplicadas se procesan estos utilizando el paquete de programa SPSS (Statistical Package for de Social Sciences) para la realización del análisis estadístico. Los resultados se muestran a continuación:

La primera pregunta sobre interfaz gráfica, oscilando las respuestas entre muy bien y excelente con un 14,3 % y 85,7 % respectivamente.

Interfaz gráfica

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Muy Bien	1	14,3	14,3	14,3
	Excelente	6	85,7	85,7	100,0
	Total	7	100,0	100,0	

La segunda pregunta sobre usabilidad, oscilando las respuestas entre muy bien y excelente con un 14,3 % y 85,7 % respectivamente.

Usabilidad

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Muy Bien	1	14,3	14,3	14,3
	Excelente	6	85,7	85,7	100,0
	Total	7	100,0	100,0	

La tercera pregunta es sobre presentación, para la cual las respuestas están entre muy bien y excelente con un 28,6 % y 71,4 % respectivamente.

Presentación

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Muy Bien	2	28,6	28,6	28,6
	Excelente	5	71,4	71,4	100,0
	Total	7	100,0	100,0	

La cuarta pregunta es sobre motivación, para la cual las respuestas están entre muy bien y excelente con un 14,3 % y 85,7 % respectivamente.

Motivación

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Muy Bien	1	14,3	14,3	14,3
	Excelente	6	85,7	85,7	100,0
	Total	7	100,0	100,0	

La quinta pregunta es sobre funcionalidad, para la cual las respuestas están entre muy bien y excelente con un 14,3 % y 85,7 % respectivamente.

Funcionalidad

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Muy Bien	1	14,3	14,3	14,3
	Excelente	6	85,7	85,7	100,0
	Total	7	100,0	100,0	

La sexta pregunta es sobre seguridad, para la cual todas las respuestas fueron de excelente.

Seguridad

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Excelente	7	100,0	100,0	100,0

La séptima pregunta es sobre la ayuda, para la cual las respuestas están entre muy bien y excelente con un 14,3 % y 85,7 % respectivamente.

Ayuda

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Muy Bien	1	14,3	14,3	14,3
	Excelente	6	85,7	85,7	100,0
	Total	7	100,0	100,0	

Para cumplimentar el análisis anterior se realizó la Prueba no Paramétrica W. de Kendall con el objetivo de demostrar estadísticamente la posible existencia de acuerdo entre los evaluadores. Dicha prueba contrasta la hipótesis nula que plantea que no hay acuerdo contra la hipótesis alternativa en que sí se considera que hay acuerdo entre los evaluadores. Tomando como referencia

un nivel de significación del 5 %, si este es menor que la significación asintótica, entonces rechazamos Ho, de lo contrario aceptamos. Por otra parte los rangos obtenidos en dicha prueba permiten ordenar los criterios analizados según la importancia atribuida por los expertos.

Utilizando un nivel de significación del 5% al comparar con la significación asintótica de los estadísticos calculados se obtuvo 0,899, entonces puede concluirse que se acepta la hipótesis alternativa en los análisis realizados para los encuestados, por lo tanto, existe concordancia de criterios entre los mismos y los planteamientos analizados.

Ranks

	Mean Rank
Interfaz gráfica	4,00
Usabilidad	4,00
Presentación	3,50
Motivación	4,00
Funcionalidad	4,00
Seguridad	4,50
Ayuda	4,00

Test Statistics

N	7
Kendall's W(a)	,053
Chi-Square	2,211
df	6
Asymp. Sig.	,899

a Kendall's Coefficient of Concordance

3.9 Conclusiones

En el presente capítulo se abordaron temas como la descripción da la arquitectura propuesta, estándar de implementación, seguridad del sistema, tratamiento de errores y se elaboró el diagrama de clase y de despliegue.

Además para la validación de la solución propuesta se utilizó muestreo no probabilístico intencional o a conveniencia y se realizó la Prueba no Paramétrica W. de Kendall aceptándose la hipótesis alternativa por lo que existe concordancia de criterios entre los encuestados.

Conclusiones

Teniendo en cuenta los objetivos planteados al inicio del trabajo, se arriban a las siguientes conclusiones.

- Se realizó el análisis y diseño de la aplicación propuesta donde se seleccionaron las metodologías, herramientas y tecnologías más adecuadas para el desarrollo del software.
- Se implementó una aplicación ajustada a las particularidades del Taller de Reparaciones de ETECSA utilizando el lenguaje de programación PHP, el gestor de bases de datos PostgreSQL, el IDE de programación NetBeans 6.8 y se aprovecharon las potencialidades del framework CodeIgniter.
- Según criterios de especialistas, recogido en la encuesta aplicada, el sistema reúne características que permitirán su utilización, facilitándose la gestión de la información relacionada con la reparación de piezas de telefonía pública en la D.T.CF.

Recomendaciones

- Extender el uso de la aplicación a todos los talleres de reparaciones en las Divisiones Territoriales del país.

Referencia Bibliográfica

[1] G. B. I. Jacobson, y J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*, Addison Wesley. 1999.

[2] A. Molpeceres, «Procesos de desarrollo: RUP,XP y FDD.» .

[3] E. H. Orallo., *El Lenguaje Unificado de Modelado (UML).* .

[4] U. y. P. Craig Larman, *Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. 2000.

[5] R. Álvarez, «Introducción al HTML.» [Online]. Available: http://www.desarrolloweb.com/articulos/534.php.

[6] V. Rivas, «Curso JavaScript». [Online]. Available: http://geneura.ugr.es/~victor/cursi llos/javascript/js_intro.html. [Accessed: 13-ene-2014].

[7] V. Rivas, «Curso de Ajax», 2008. [Online]. Available: http://geneura.ugr.es/~victor/cursillos/javascript/ajax_intro.html.

[8] Matthew A. Russell, *Dojo: The Definitive Guide*. 2008.

[9] «CSS: Hojas de estilo». [Online]. Available: http://es.kioskea.net/contents/css/cssintro.php3.

[10] «History of PHP and related projects». [Online]. Available: http://www.php.net/history.

[11] Fernando Velo, *CodeIgniter 2.1.0. Guia de Usuario en español.* 2010.

[12] «Panorámica del sistema de gestión de base de datos PostgreSQL». [Online]. Available: http://dev.postgresql.com/doc/refman/8.3/es/what-is.html. [Accessed: 10-feb-2014].

[13] «Apache». [Online]. Available: www.apache.org. [Accessed: 03-mar-2014].

[14] «Visual Paradigm for UML.» [Online]. Available: http://www.freedownloadmanager.org/es/.

[15] «Manuales de Dreamweaver. Diseño Web». [Online]. Available: http://www.infomanuales.net/Manuales/Dreamweaver.asp.

[16] «Manuales de Adobe. Photoshop CS4». [Online]. Available: http://www.infomanuales.net/Manuales/AdobePhotoshop.asp. [Accessed: 14-abr-2014].

[17] «Netbeans 6.8 liberado.», 2010. [Online]. Available: http://blogultura.com/java/netbeans-6-8-liberado.

[18] «GIMP 2.8.», 2011. [Online]. Available: http://www.gimp.org/.

[19] M. Peralta, *ESTIMACIÓN DEL ESFUERZO BASADA EN CASOS DE USO.*

Bibliografía

[1] «Apache». [Online]. Available: www.apache.org. [Accessed: 03-mar-2014].
[2] Fernando Velo, *CodeIgniter 2.1.0. Guia de Usuario en español.* 2010.
[3] V. Rivas, «Curso de Ajax», 2008. [Online]. Available: http://geneura.ugr.es/~victor/cursillos/javascript/ajax_intro.html.
[4] V. Rivas, «Curso JavaScript». [Online]. Available: http://geneura.ugr.es/~victor/cursi llos/javascript/js_intro.html. [Accessed: 10-feb-2014].
[5] V. Rivas, «Curso JavaScript». [Online]. Available: http://geneura.ugr.es/~victor/cursi llos/javascript/js_intro.html. [Accessed: 13-ene-2014].
[6] Matthew A. Russell, *Dojo: The Definitive Guide*. 2008.
[7] E. H. Orallo., *El Lenguaje Unificado de Modelado (UML).* .
[8] G. B. I. Jacobson, y J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*, Addison Wesley. 1999.
[9] M. Peralta, *ESTIMACIÓN DEL ESFUERZO BASADA EN CASOS DE USO.* .
[10] «GIMP 2.8.», 2011. [Online]. Available: http://www.gimp.org/.
[11] «Manual de Prosedimiento para la restauración de Estaciones Públicas.» .
[12] «Manuales de Dreamweaver. Diseño Web». [Online]. Available: http://www.infomanuales.net/Manuales/Dreamweaver.asp.
[13] «Panorámica del sistema de gestión de base de datos PostgreSQL». [Online]. Available: http://dev.postgresql.com/doc/refman/8.3/es/what-is.html. [Accessed: 10-feb-2014].
[14] A. Molpeceres, «Procesos de desarrollo: RUP,XP y FDD.» .
[15] «Sistema Gestor de Bases de Datos.» [Online]. Available: http://personal.lobocom.es/claudio/sql001.htm. [Accessed: 06-ene-2014].
[16] Harmon, James E., *Using the Dojo JavaScript Library to Build Ajax Application.* Boston: Addison-Weasley, 2009.
[17] «Visual Paradigm for UML.» [Online]. Available: http://www.freedownloadmanager.org/es.