

CONTROL DE INDICADORES DE GESTIÓN MEDIANTE EL DESARROLLO DE
UNA APLICACIÓN WEB

JULIÁN FELIPE BORRAY GUTIÉRREZ

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BOGOTÁ
2015

CONTROL DE INDICADORES DE GESTIÓN MEDIANTE EL DESARROLLO DE
UNA APLICACIÓN WEB

JULIÁN FELIPE BORRAY GUTIÉRREZ

TRABAJO DE GRADO PARA OPTAR POR TÍTULO DE INGENIERO DE
SISTEMAS

AUGUSTO JOSE ANGEL MORENO
DIRECTOR

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BOGOTÁ
2015

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Bogotá mayo de 2015

Dedico este proyecto de grado a mi mamá quien me apoyo incondicionalmente durante el desarrollo de mi carrera siendo indispensable para lograr finalizarla con éxito.

AGRADECIMIENTOS

En primer lugar quiero agradecer a mis padres por ser una fuente de continuo apoyo, por cuidarme y facilitarme lo que pudiera necesitar para desarrollar y culminar mis estudios, gracias por su paciencia, motivación y aliento. Para mí ha sido un privilegio contar con su ayuda.

Gracias a los profesores Erlington Salcedo por sus enseñanzas, su exigencia y por motivarme a ser un mejor ingeniero inspirado en su genialidad y su mente prodigiosa, Omar Bohórquez por enseñar con una pasión única por su profesión, además por sus enseñanzas y por crear un interés hacia la física en mí.

Gracias a mis compañeros de estudio quienes fueron personas clave para mi vida académica y profesional aportándome su conocimiento, ideas y apoyo. Gracias también a mis amigos quienes siempre estuvieron presentes de una u otra manera para apoyarme, ayudarme o distraerme.

CONTENIDO

	Pág.
1. INTRODUCCIÓN	12
2. JUSTIFICACIÓN	13
3. OBJETIVOS	14
3.1 OBJETIVO GENERAL.....	14
3.2 OBJETIVOS ESPECÍFICOS	14
4. MARCO TEÓRICO.....	15
5. INGENIERÍA DEL PROYECTO.....	19
5.1 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL	19
5.2 REQUERIMIENTOS DE LA INFORMACIÓN	19
5.3 MODELAMIENTO DEL SISTEMA	20
5.3.1. DIAGRAMAS DE CASOS DE USO.....	21
5.3.2. DESCRIPCIÓN DE CASOS DE USO	24
5.3.3. DIAGRAMA DE CLASES	31
5.3.4. DIAGRAMA DE SECUENCIA	32
5.3.5. MODELADO DE BASE DE DATOS	34
5.4 DESCRIPCIÓN DEL SISTEMA	36
5.4.1 FUNCIONES ORIENTADAS AL ADMINISTRADOR.....	37
5.4.2 FUNCIONES ORIENTADAS AL USUARIO	37
5.4.3 REQUERIMIENTOS DE HARDWARE Y SOFTWARE	38
6. EVALUACIÓN ECONÓMICA DEL PROYECTO	39
6.1. RIESGO EN LA FASE DE ANÁLISIS	39
6.2. RIESGO EN LA FASE DE DISEÑO	40
6.2.1. FASE DE DISEÑO DE DATOS.....	40
6.2.2. FASE DE DISEÑO DE DATOS.....	41
6.3. RIESGO EN LA FASE DE CODIFICACIÓN	42
6.4. RIESGO EN LA FASE DE PRUEBAS	43
6.5. RIESGO EN LA FASE DE IMPLEMENTACION	44

6.6. RIESGO EN LA FASE DE MANTENIMIENTO	45
7. PRESUPUESTO DETALLADO	45
7.1. COSTO DE LA INFRAESTRUCTURA.....	45
8. BENEFICIOS DE LA IMPLEMENTACIÓN	46
8.1. OPERACIONALES	46
8.2. GESTIÓN.....	46
8.3. ESTRATÉGICOS.....	47
8.4. BENEFICIOS DE TI.....	47
9. ALCANCES Y LIMITACIONES DEL PROYECTO	48
10. CRONOGRAMA.....	50
11. BIBLIOGRAFÍA	52

LISTA DE TABLAS

	Pág
Tabla 1. Ingresar resultados de visita	25
Tabla 2. Cargar resultados de visita	25
Tabla 3. Modificar centros de servicio	26
Tabla 4. Consultar directorio del sistema	27
Tabla 5. Generar informes de gestión	27
Tabla 6. Consultar inventario	28
Tabla 7. Modificar inventario	28
Tabla 8. Reservar vehículo	29
Tabla 9. Crear usuario	29
Tabla 10. Eliminar usuario	30
Tabla 11. Crear centro de servicio	30
Tabla 12. Crear motivos	31
Tabla 13. Crear ítems	31
Tabla 14. Riesgos en el análisis	39
Tabla 15. Riesgos en el diseño	41
Tabla 16. Riesgos en la codificación	42

LISTA DE FIGURAS

	Pág
Figura 1. Diagrama caso de uso resultado de visitas	21
Figura 2. Diagrama caso de uso del control de apertura	22
Figura 3. Diagrama caso de uso información de centros de servicio	23
Figura 4. Diagrama caso de uso informes de gestión	23
Figura 5. Diagrama caso de uso general	24
Figura 6. Diagrama de clases	32
Figura 7. Diagrama de secuencia	33
Figura 8. Modelo entidad relación uno	34
Figura 9. Modelo entidad relación definitivo	35
Figura 10. Cronograma de Actividades	50
Figura 11. Diagrama de Gantt	50

GLOSARIO

- AJAX: acrónimo de Asynchronous Javascript and XML, es decir: Javascript y XML Asíncrono.
- Apache: Servidor web HTTP de código abierto
- Base de datos: "Almacén" que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. "
- CSS: Es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación.
- Hardware: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.
- HTML: Es el lenguaje con el que se definen las páginas web.
- Javascript: Lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web.
- Modelo Entidad relación: Es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades
- MySQL: Sistema de gestión de bases de datos relacional, multihilo y multiusuario.
- NginX: Es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3). Es software libre y de código abierto
- PHP: (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.
- Servidor: Equipo informático que forma parte de una red y provee servicios a otros equipos cliente.
- SLQ: Lenguaje de computación que sirve para trabajar con conjuntos de datos y las relaciones entre ellos
- Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas

RESUMEN

El presente documento tiene como objetivo principal mostrar una visión general de los procesos llevados a cabo para el adelanto del desarrollo de una aplicación web capaz de modelar indicadores de gestión y controlar inventario y ubicación de diversos documentos y activos. En primera instancia se adelantó una breve introducción a los lineamientos cardinales que rigen la ejecución del trabajo. En segundo lugar, se presenta la justificación y objetivos que delimitan el propósito del trabajo ejecutado. La siguiente sección tiene como objetivo especificar las variantes teóricas que se emplearon como directrices para la puesta en marcha del proyecto. Posteriormente se describen los antecedentes, el modelamiento y la metodología de desarrollo del sistema.

En la fase final se adelanta la evaluación económica del proyecto y se procede a fijar el presupuesto, para en última instancia inferir los beneficios y el alcance del mismo.

1. INTRODUCCIÓN

El Diccionario de Oxford define una aplicación web como: “un programa o conjunto de programas que ayudan al usuario de un ordenador a procesar una tarea específica” (Véronneau, 2012). La aplicación web por tanto facilita tareas determinadas para las cuales fue diseñada.

El surgimiento de la aplicación web facilitó la evolución del contenido de las páginas; desde sus inicios en 1987 bajo un lenguaje de programación denominado Perl que permitió el surgimiento primario de las aplicaciones web, hasta su presencia ulterior en el lenguaje PHP en aplicaciones como Google, Facebook y Wikipedia (Véronneau, 2012).

Con el surgimiento de esta tecnología se amplió el acceso a la información a través de internet mejorando las comunicaciones a nivel mundial. Las primeras aplicaciones web agruparon un conjunto de buscadores, redes sociales, páginas web e intranets que hicieron más sencillo el acceder a la información alojada en un servidor.

Codensa es una importante empresa del sector energético creada formalmente el 23 de octubre de 1997 y dedicada a la distribución y comercialización de energía eléctrica, cubriendo adicionalmente la ejecución de obras afines, conexas y complementarias relacionadas con este objeto (Grupo Energía de Bogotá).

Actualmente me encuentro desarrollando mi pasantía en Codensa en la Subgerencia de Mercado Residencial y Pymes, División de Atención Presencial. Dicho segmento posee problemas asociados al control de la información que maneja, el acceso a ella, la obtención de información útil y relevante a partir de los datos registrados y el control de inventarios, documentos y activos del área.

La solución ofrecida es una aplicación web desarrollada con el fin de facilitar los procesos de almacenamiento y acceso a información, la obtención de datos relevantes de lo que se ingresa y el control de documentos, inventarios y activos.

El trabajo relata en términos generales la solución ofrecida a Codensa y se organiza de la siguiente manera:

2. JUSTIFICACIÓN

El deseo de hacer uso de los conocimientos académicos y proyectarlos a escenarios profesionales, con el objetivo de fortalecer y desarrollar competencias a través de la aproximación a un contexto real de trabajo y a las necesidades de un ambiente organizacional confrontando los conocimientos y destrezas adquiridas con los requerimientos profesionales es la principal razón del adelanto de este proyecto.

Durante el desarrollo las labores diarias se observaron varios procesos donde el manejo de la información no era el correcto debido a que se presentaban errores constantemente, problemas para encontrarla y para almacenarla por la gran cantidad de datos manejados, en la mayoría de los casos era almacenada en archivos de Excel lo que podía ocasionar una búsqueda lenta, bloqueos del equipo, pérdida de información y cuando se superaba el límite de registros era necesario pedir ayuda externa para acceder a la información de los archivos.

Además también se detectó que tal cantidad de información no estaba siendo útil de ninguno modo, pues al ser tan difícil de manejar no se ejecutaba ningún proceso para extraer datos que pudieran contribuir a la mejora de los servicios prestados por Codensa, dificultando la constante mejora en la atención lo cual es el objetivo de la división.

La necesidad de la División de Atención Presencial de mejorar el manejo de la información fue el escenario apropiado para llevar a cabo una solución que cubriera a cabalidad los requerimientos de este segmento asintiendo el uso de las tecnologías de desarrollo web; AJAX, CSS, HTML, PHP y Javascript. Para ello se desarrolló una aplicación capaz de solventar estas dificultades asociadas a la información, dicha aplicación tiene la ventaja de permitir la obtención de información útil y relevante a partir de los datos registrados y el control de inventarios, documentos y activos del área.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Poner en práctica los conocimientos adquiridos en la academia mediante la creación de una aplicación web capaz de suplir las necesidades del entorno organizacional de la División de Atención Presencial de Codensa.

3.2 OBJETIVOS ESPECÍFICOS

- Registrar y controlar las No Conformidades que se dan en los Centros de Servicio.
- Controlar la apertura y el funcionamiento de los aplicativos de los Centros de Servicio.
- Mantener la información actualizada y facilitar el acceso a la misma.
- Generar reportes sobre la gestión de los Centros de Servicio.
- Regular el movimiento de los equipos, documentos y activos.

4. MARCO TEÓRICO

Entre las definiciones más completas de la ingeniería de software se encuentra la especificada por Bohem quien define ingeniería de software como la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos (1976) y la del Instituto de Ingenieros Eléctricos y Electrónicos (cuyas siglas son IEEE y provienen del inglés Institute of Electrical and Electronics Engineers) que relaciona la ingeniería del software con un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software (IEEE, 1993).

Pese a que el término ingeniería del software no fue usado sino hasta 1968 en la conferencia de ingeniería de software de la Organización del Tratado Atlántico Norte (OTAN) para evocar la crisis de software que se vivía en ese momento, los problemas a los cuales hizo alusión esta reunión surgieron con anterioridad (Instituto Nacional de Tecnologías de la Comunicación, 2009).

En 1940 se dio la creación de las primeras computadoras digitales, sin embargo, estas trajeron consigo problemas asociados a los tiempos programados para los proyectos y el presupuesto inicialmente planteado. En 1950 aparecieron los primeros lenguajes de programación, entre los cuales se encuentran los asociados a problemas científicos, algorítmicos y de negocios (Fortran, Algol y Cobol respectivamente) (Instituto Nacional de Tecnologías de la Comunicación, 2009).

En la década de los 60s se introdujeron los sistemas operativos que permitían generar sistemas de software para gestionar el hardware, además de introducirse el lenguaje Simula capaz de modelar la programación orientada a objetos. En la década de los 70s y 80s se introdujo el concepto de modularidad y encapsulación para controlar la complejidad de los sistemas de software y se generaron varios lenguajes de programación orientada a objetos inspirados en simula, como C++, Smalltalk y Objective C. (Instituto Nacional de Tecnologías de la Comunicación, 2009).

El software open source empezó a aparecer a principios de los 90s en la forma de Linux y otros software introduciendo el “bazaar” o el estilo descentralizado de construcción de software. Después aparecieron Internet y la World Wide Web a mediados de los 90s cambiando de nuevo la ingeniería del software. Los sistemas

distribuidos ganaron dominio como forma de diseñar sistemas y el lenguaje de programación Java se introdujo como otro paso en la abstracción, teniendo su propia máquina virtual. Varios programadores colaboraron y escribieron el manifiesto ágil que favoreció procesos más ligeros para crear software más barato y en menos tiempo. (Instituto Nacional de Tecnologías de la Comunicación, 2009).

El objetivo primario de la ingeniería del software es construir un producto de alta calidad de una manera oportuna. Trata de conseguir este objetivo primario usando un enfoque de ingeniería. Implica un conjunto de principios fundamentales que deberían seguirse siempre. Incluyen actividades explícitas para el entendimiento del problema y la comunicación con el cliente, métodos definidos para representar un diseño, mejores prácticas para la implementación de la solución y estrategias y tácticas sólidas para las pruebas. Si se siguen los principios básicos, esto resulta en productos de alta calidad.

Para conseguir el objetivo de construir productos de alta calidad dentro de la planificación, la ingeniería del software emplea una serie de prácticas para:

- Entender el problema
- Diseñar una solución
- Implementar la solución correctamente
- Probar la solución

Gestionar las actividades anteriores para conseguir alta calidad

La ingeniería del software representa un proceso formal que incorpora una serie de métodos bien definidos para el análisis, diseño, implementación y pruebas del software y sistemas. Además, abarca una amplia colección de métodos y técnicas de gestión de proyectos para el aseguramiento de la calidad y la gestión de la configuración del software. (Instituto Nacional de Tecnologías de la Comunicación, 2009)

En la ingeniería de software existen distintas formas de notación; entre las cuales se encuentra el lenguaje único de modelado (UML por sus siglas en inglés) y los diagramas de flujo de datos, estas notaciones son expresiones gráficas con las que se puede ver un sistema a modo de plano.

El UML es definido como una notación de modelado visual que utiliza diagramas para mostrar distintos aspectos de un sistema (Fontela, 2011).

La primera versión de UML se lanzó en 1997, esta versión fue recogiendo aportes de grandes gurús del desarrollo orientado a objetos; como David Hasel con sus diagramas de estado y Rebeca Wirf-Brock. En el mismo año, OMG (Object Management Group) aceptó el UML como estándar y con ello nació el primer lenguaje de modelado visual orientado a objetos, desde entonces han desaparecido prácticamente todas las demás metodologías y UML se ha convertido en el estándar en la industria del software. En 1998 OMG lanzó dos versiones más; 1.2 y 1.3. En el año 2000 se presentó UML 1.4, en el 2005 la versión 2 y en el 2012 la versión 2.5 de carácter beta. (Aguilar, y otros, 2014).

En la ingeniería de software para llevar a cabo un desarrollo también se requieren metodologías de trabajo que pueden ser de desarrollo ágil como SCRUM, cascada o espiral entre otras, con el fin de controlar, planificar y estructurar el proceso. Estas metodologías, en su mayoría, responden inicialmente a seis pasos; Captura y validación de requerimientos, análisis, diseño, construcción, pruebas e implementación o despliegue.

La metodología SCRUM se ha destacado por definir roles y conjuntos de buenas prácticas para el desarrollo de un software. Esta metodología se define como un proceso en el que se aplica de manera regular un grupo de prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto, estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos (IBM, 2010).

SCRUM consiste en realizar entregas parciales y regulares del producto final y es una metodología pensada para entornos que cambian constantemente, es decir, donde los requisitos no están bien definidos.

En esta metodología participan una serie de actores que conforman el equipo SCRUM este consiste en un dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master, los Equipos Scrum son auto organizados y multifuncionales.

Los equipos auto organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo y los multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo.

El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. Los Equipos entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación.

Entre los componentes del equipo se encuentran; el Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos, El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo y El Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado. Los ScrumMasters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. El ScrumMaster es un líder que está al servicio del Equipo Scrum. también ayuda a las personas externas al Equipo a entender qué interacciones con él pueden ser de ayuda y cuáles no, ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum. (Sutherland, y otros, 2013).

5. INGENIERÍA DEL PROYECTO

5.1 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

Actualmente en la División de Atención Presencial de Codensa se maneja cuantiosa información acerca de los centros de servicio de una forma desorganizada, descentralizada y poco eficiente. Los encargados de llevar esta información son los gestores cada uno de ellos tiene centros de servicio asignados, en estos deben verificar el correcto funcionamiento del centro de servicio teniendo en cuenta personal, infraestructura física y tecnológica y el funcionamiento de las aplicaciones entre otros.

En la actualidad la información es ingresada en archivos de Excel y no se tiene un formato estandarizado lo que dificulta extraer datos útiles volviendo inservible la información que registran diariamente durante sus visitas a los centros de servicio. Asimismo no existe una forma de generar indicadores para evaluar el desempeño y los resultados generados por los gestores.

5.2 REQUERIMIENTOS DE LA INFORMACIÓN

El levantamiento de los requerimientos se hizo mediante entrevistas, observación y análisis de la documentación para tener una visión y expectativas del usuario claras hacia el sistema. La entrevista se realizó a manera de conversación y de esta se obtuvo gran cantidad de datos respecto a las necesidades y a los usuarios a quienes va dirigido el sistema. Mediante la observación se logró determinar más detalladamente el proceso que siguen los usuarios para obtener la información, registrarla en un archivo y establecer que datos serán ingresados al sistema. Finalmente el análisis de la documentación fue muy significativo para determinar el manejo de los datos y el procesamiento de la información.

Estas técnicas en conjunto permitieron identificar las necesidades y restricciones para llevar a cabo un desarrollo viable y que cumpla plenamente con los objetivos de los usuarios.

Los requerimientos encontrados fueron los siguientes:

Funcionales:

- Permitir el ingreso la información obtenida a partir de las visitas realizadas a los centros de servicio, Cades o Supercades.
- Permitir carga de archivos .csv con información de las visitas.
- Permitir realizar el control de apertura y de funcionamiento de los aplicativos usados en los centros de atención
- Modificar información de los centros de servicio
- Permitir a los usuarios mantener y actualizar un directorio.
- Mostrar gráficamente las no conformidades que más se repiten
- Generar informes de gestión filtrados por fecha
- Consultar y actualizar de manera rápida y sencilla el inventario
- Permitir reservar vehículos por días u horas
- Perfiles de usuario
- Perfil de usuario administrador para crear y eliminar usuarios, centros de servicio, motivos e ítems.

No funcionales:

- Accesible desde cualquier punto de servicio
- Interfaz amigable e intuitiva en cualquier navegador
- Disponibilidad días entre semana
- Copias de seguridad semanales
- Posibilidad de añadir nuevos módulos
- Documentar y comentar el código para facilitar la comprensión para otros desarrolladores

5.3 MODELAMIENTO DEL SISTEMA

El modelamiento del sistema es una técnica con la que es posible crear un plano del software a desarrollar para dar una idea más clara de la complejidad del mismo, obtener aspectos importantes y explorar múltiples soluciones.

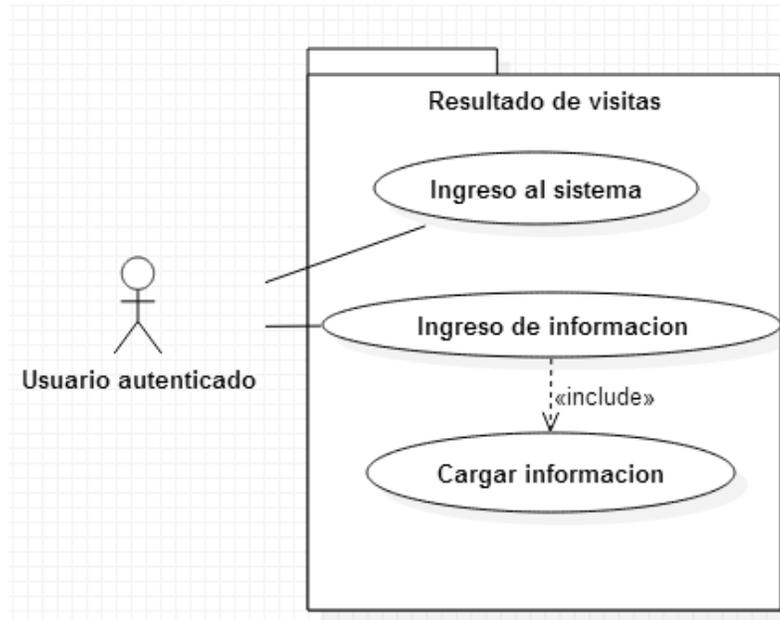
En el proceso de modelar un sistema son utilizados varios tipos de diagramas, los cuales se clasifican en tres clases, estructura, comportamiento e interacción.

Entre los diagramas de comportamiento podemos encontrar los diagramas de casos de uso que se encargan de representar las interacciones de los usuarios con el sistema.

5.3.1. DIAGRAMAS DE CASOS DE USO

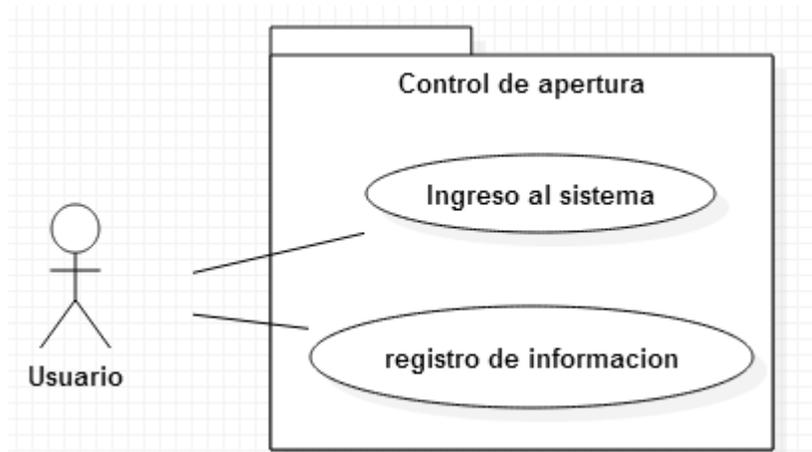
Este diagrama representa la interacción del usuario al ingresar al sistema para cargar información a través del sistema o subiendo un archivo plano.

Figura 1. Diagrama caso de uso resultado de visitas



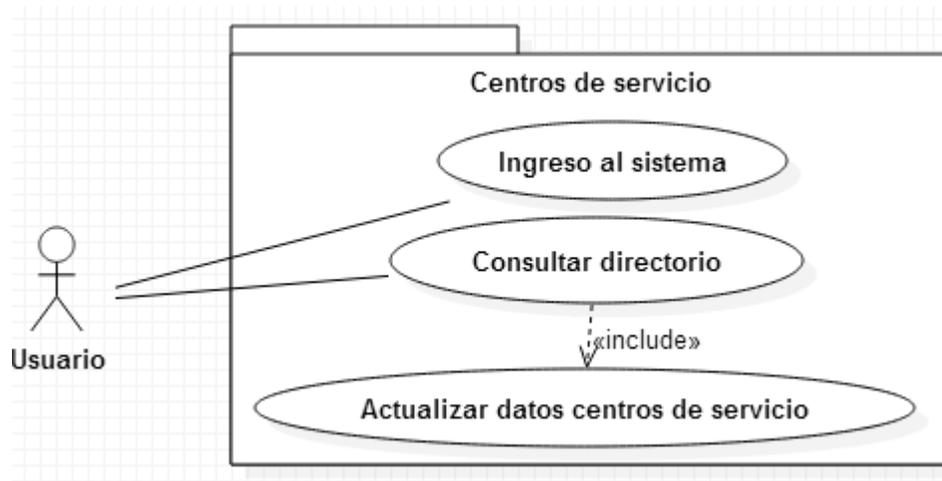
El diagrama número dos representa la interacción del sistema con el usuario cuando se requiere ingresar información del control de apertura de un centro de servicio.

Figura 2. Diagrama caso de uso del control de apertura



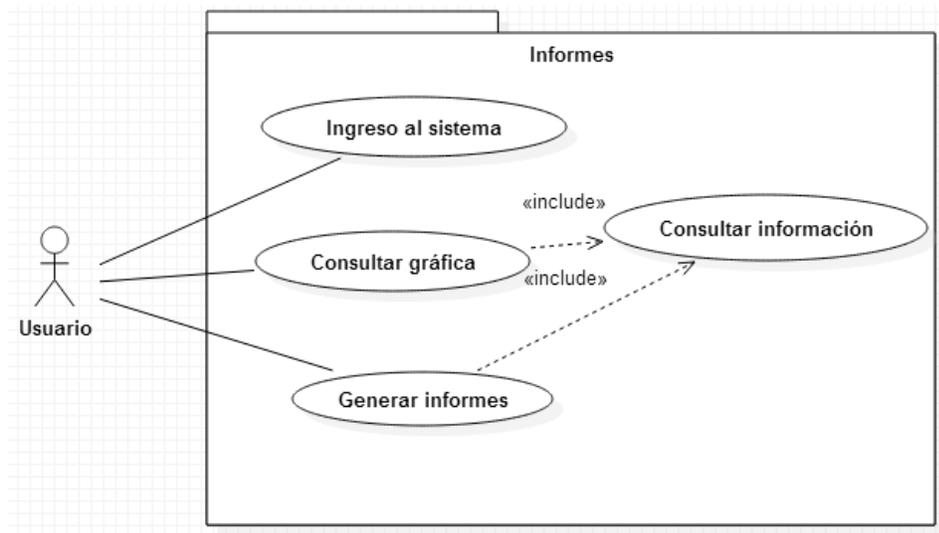
El siguiente diagrama simboliza la interacción del sistema con el usuario cuando se requiere modificar o consultar información de los centros de servicio.

Figura 3. Diagrama caso de uso información de centros de servicio



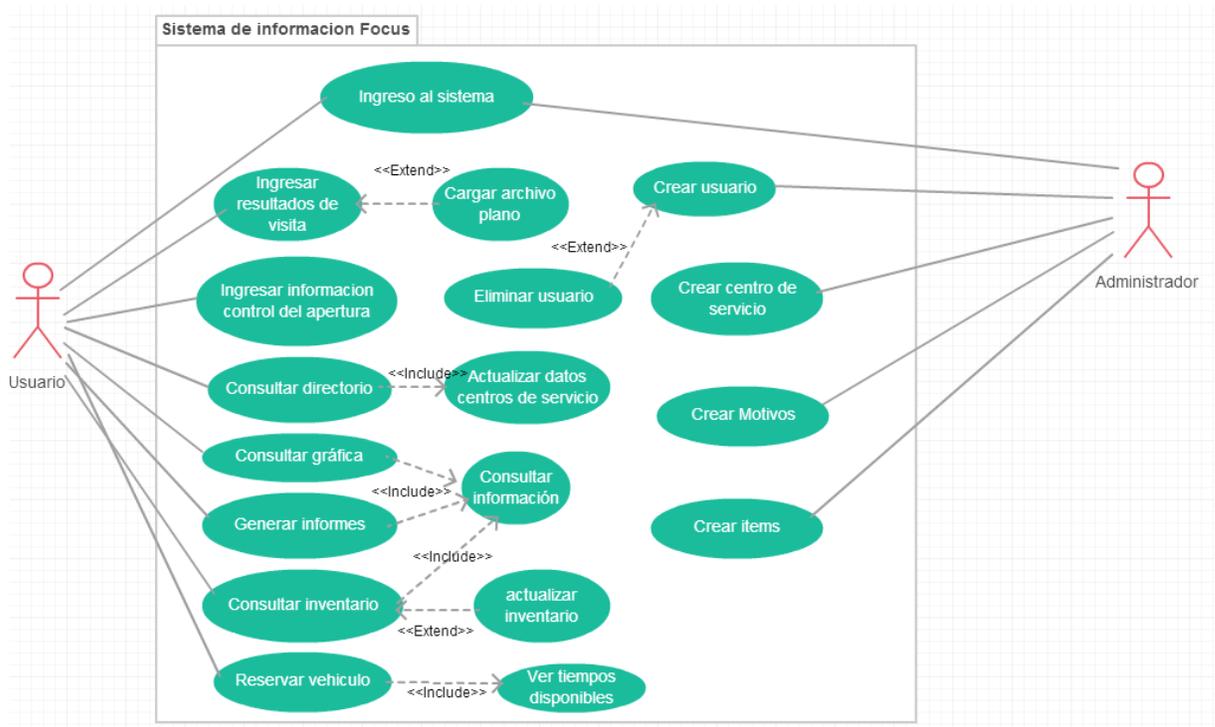
El diagrama de casos de uso cuatro simboliza la interacción del usuario con el sistema al momento de generar informes o ver de manera gráfica la información registrada

Figura 4. Diagrama caso de uso informes de gestión



El siguiente diagrama representa los casos de uso generales del sistema en él se pueden distinguir las funciones que ofrece para el usuario y las posibilidades que tiene el administrador del sistema para adaptarlo a unas necesidades determinadas.

Figura 5. Diagrama caso de uso general



5.3.2. DESCRIPCIÓN DE CASOS DE USO

La descripción de los casos de uso indica los pasos que debe seguir el usuario para llevar a cabo algún proceso indicando el flujo básico de los datos, precondiciones y postcondiciones.

Tabla 1. Ingresar resultados de visita

Caso de uso	Ingresar resultados de visita
Descripción	El usuario ingresa los resultados de la visita al sistema
Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El centro de servicio debe estar asignado al usuario
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema pide al usuario seleccionar un centro de servicio 2. El usuario selecciona ítem y motivo 3. El usuario ingresa la observación 4. El usuario da clic en nuevo para guardar y limpiar los campos
Postcondiciones	El sistema registra la no conformidad guardándola en la base de datos
Excepciones	<p>Paso cuatro, el usuario da clic en guardar sin seleccionar ítem, motivo o escribir una observación.</p> <ul style="list-style-type: none"> • El sistema comprueba los campos y muestra un erro pidiendo completar los campos

Tabla 2. Cargar resultados de visita

Caso de uso	Cargar resultados de visita
Descripción	El usuario ingresa los resultados de la visita a través de un archivo plano .csv
Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El usuario debe llenar el formato estándar para subir al sistema
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema pide al usuario seleccionar el archivo a cargar 2. El usuario selecciona archivo 3. El usuario da clic en subir para cargar los datos
Postcondiciones	El sistema registra las no conformidades encontradas en el archivo .csv
Excepciones	<p>Paso dos, el usuario selecciona un archivo con formato no válido.</p> <ul style="list-style-type: none"> • El sistema verifica el formato del archivo • El sistema emite un mensaje de error si el formato del archivo es distinto al admitido. • El sistema no lee el archivo ni guarda datos del mismo. <p>Paso tres, los datos no tienen el formato correcto</p> <ul style="list-style-type: none"> • El sistema verifica los campos del archivos • El sistema muestra un error pidiendo que se confirme la información de los formatos
Caso de uso	Registrar control de apertura
Descripción	El usuario ingresa los datos de selección de control de apertura

Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema pide al usuario completar los campos 2. El usuario selecciona el centro de servicio 3. El usuario selecciona los campos requeridos 4. El usuario da clic en nuevo para guardar y limpiar el formulario
Postcondiciones	El sistema el estado del centro de servicio y sus aplicaciones al momento de iniciar la actividad.
Excepciones	N/A

Tabla 3. Modificar centros de servicio

Caso de uso	Modificar centros de servicio
Descripción	El usuario desea modificar los datos de un centro de servicio
Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El usuario debe tener centros de servicio asignados
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema muestra los centros de servicio asignados a ese usuario 2. El usuario selecciona el centro de servicio 3. El usuario da clic sobre actualizar 4. El sistema muestra los datos del centro de servicio para que puedan ser actualizados por el usuario 5. El usuario modifica el campo requerido 6. El usuario da clic sobre guardar 7. El sistema da un mensaje de confirmación y direcciona al menú de centros de servicio
Postcondiciones	El sistema actualiza la información del centro de servicio
Excepciones	<p>Paso tres, el usuario da clic sobre actualizar sin seleccionar centro de servicio</p> <ul style="list-style-type: none"> • El sistema lo lleva a la pantalla de actualización, pero no muestra ningún dato para actualizar. <p>Paso cinco, el usuario ingresa datos alfabéticos en un campo numérico.</p> <ul style="list-style-type: none"> • El sistema valida el campo • El sistema muestra un mensaje de error • El sistema limpia el campo

Tabla 4. Consultar directorio del sistema

Caso de uso	Consultar directorio del sistema
Descripción	El usuario requiere datos de un centro de servicio
Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema pide al usuario seleccionar el centro de servicio 2. El usuario selecciona el centro de servicio 3. El usuario hace clic sobre el botón información 4. El sistema muestra todos los datos que tiene sobre el centro de servicio
Postcondiciones	El sistema muestra los datos del centro de servicio sin permitir editarlos
Excepciones	N/A

Tabla 5. Generar informes de gestión

Caso de uso	Generar informes de gestión
Descripción	El usuario genera un archivo de Excel con datos de gestión
Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema pide al usuario seleccionar que tipo de informe requiere 2. El sistema pide seleccionar un rango de fechas para limitar el informe 3. El usuario da clic en el botón correspondiente a ese informe 4. El sistema genera un archivo Excel con los datos establecidos para ese informe
Postcondiciones	El sistema da la opción al usuario de descargar un archivo en Excel con los datos del informe
Excepciones	<p>Paso dos, el usuario no selecciona rango de fechas para el informe.</p> <ul style="list-style-type: none"> • El sistema verifica los campos • El sistema emite un mensaje de advertencia <p>Paso dos, no existen datos registrados en ese rango de fechas</p> <ul style="list-style-type: none"> • El sistema emite un mensaje de advertencia <p>Paso dos, el usuario ingresa la fecha inicial mayor a la final</p> <ul style="list-style-type: none"> • El sistema emite un mensaje de advertencia y limpia los campos

Tabla 6. Consultar inventario

Caso de uso	Consultar inventario
Descripción	El usuario requiere ubicar un activo de la empresa
Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema pide al usuario seleccionar un filtro de búsqueda 2. El usuario llena los campos dependiendo el filtro de búsqueda seleccionado 3. El usuario da clic en buscar 4. El sistema ejecuta una consulta y muestra los datos relacionados a la búsqueda
Postcondiciones	El sistema muestra los datos relacionados con la búsqueda
Excepciones	<p>Paso tres, el usuario da clic en buscar sin llenar campos</p> <ul style="list-style-type: none"> • El sistema verifica los campos • El sistema emite un mensaje de advertencia • El sistema no realiza la búsqueda

Tabla 7. Modificar inventario

Caso de uso	Modificar inventario
Descripción	El usuario desea modificar los datos del inventario
Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El usuario debe tener centros de servicio asignados
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema muestra la lista de activos por centro de servicio 2. El usuario selecciona activo a actualizar 3. El usuario da clic sobre actualizar 4. El sistema muestra los datos del activo para que puedan ser actualizados por el usuario 5. El usuario modifica el campo requerido 6. El usuario da clic sobre guardar 7. El sistema da un mensaje de confirmación y direcciona al menú de inventario
Postcondiciones	El sistema actualiza la información del inventario del centro de servicio
Excepciones	<p>Paso tres, el usuario da clic sobre actualizar sin seleccionar centro de servicio</p> <ul style="list-style-type: none"> • El sistema lo lleva a la pantalla de actualización, pero no muestra ningún dato para actualizar. <p>Paso cinco, el usuario ingresa datos alfabéticos en un campo numérico.</p> <ul style="list-style-type: none"> • El sistema valida el campo

	<ul style="list-style-type: none"> • El sistema muestra un mensaje de error • El sistema limpia el campo
--	--

Tabla 8. Reservar vehículo

Caso de uso	Reservar vehículo
Descripción	El usuario requiere reservar un vehículo para desplazarse
Actores	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El vehículo debe estar disponible
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema muestra al usuario los tiempos ocupados y libres del vehículo por semana 2. El usuario da clic en reservar 3. El usuario ingresa los datos de la reserva 4. El usuario da clic en guardar
Postcondiciones	El sistema reserva por tiempos los vehículos de la división
Excepciones	<p>Paso cuatro, el vehículo ya está reservado en ese espacio de tiempo</p> <ul style="list-style-type: none"> • El sistema informa al usuario sobre una reserva previa • El sistema cancela la reserva • El sistema redirecciona al menú de reservas

Los siguientes son los casos de uso para el usuario Administrador:

Tabla 9. Crear usuario

Caso de uso	Crear usuario
Descripción	El administrador requiere crear un nuevo usuario para el sistema
Actores	Administrador
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El usuario debe tener acceso de administrador
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema muestra al administrador un formulario para crear usuarios 2. El administrador ingresa los datos requeridos 3. El administrador crea el usuario dando clic en nuevo 4. El sistema guarda y muestra un mensaje de confirmación
Postcondiciones	Se crea un nuevo usuario en el sistema
Excepciones	<p>Paso dos, el administrador no completa los campos</p> <ul style="list-style-type: none"> • El sistema verifica los campos • El sistema muestra un mensaje de advertencia • El sistema cancela la creación del usuario

Tabla 10. Eliminar usuario

Caso de uso	Eliminar usuario
Descripción	El administrador requiere eliminar un usuario del sistema
Actores	Administrador
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El usuario debe tener acceso de administrador
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema muestra al administrador un formulario para eliminar usuarios 2. El administrador ingresa el co del usuario a eliminar 3. El administrador elimina el usuario dando clic en Eliminar 4. El sistema elimina y muestra un mensaje de confirmación
Postcondiciones	Se elimina un usuario en el sistema
Excepciones	<p>Paso tres, el usuario no existe</p> <ul style="list-style-type: none"> • El sistema verifica la existencia del usuario • El sistema muestra un mensaje de error

Tabla 11. Crear centro de servicio

Caso de uso	Crear centro de servicio
Descripción	El administrador crear un centro de servicio
Actores	Administrador
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El usuario debe tener acceso de administrador
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema muestra al administrador un formulario con los datos necesarios para crear un centro de servicio 2. El administrador ingresa los datos 3. El administrador da clic en guardar 4. El sistema crear el centro y muestra un mensaje de confirmación
Postcondiciones	Se crea un nuevo centro de servicio en el sistema
Excepciones	<p>Paso dos, los datos no están completos</p> <ul style="list-style-type: none"> • El sistema emite un mensaje de error

Tabla 12. Crear motivos

Caso de uso	Crear motivos
Descripción	El administrador requiere crear un motivo de no conformidad
Actores	Administrador
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El usuario debe tener acceso de administrador
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema muestra al administrador un formulario donde debe ingresar y seleccionar los datos 2. El administrador selecciona el ítem asociado e ingresa los datos 3. El administrador da clic en crear motivo 4. El sistema crea el motivo y emite un mensaje de confirmación
Postcondiciones	Se crea un nuevo motivo de no conformidad
Excepciones	N/A

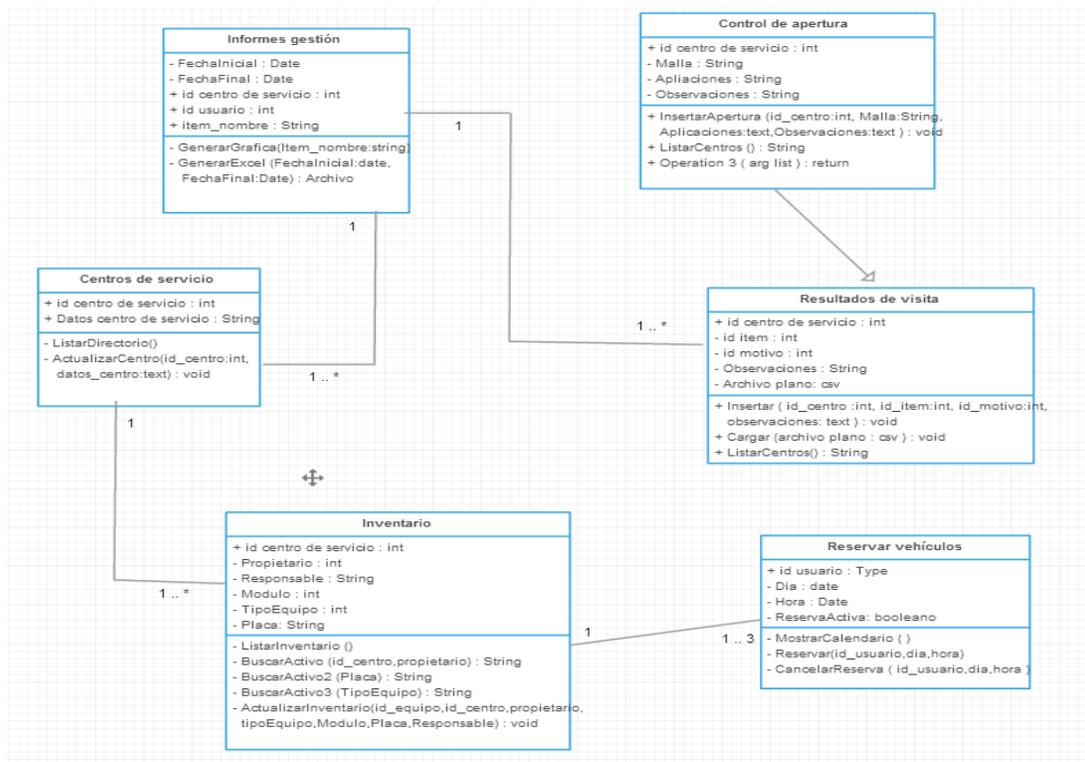
Tabla 13. Crear ítems

Caso de uso	Crear ítems
Descripción	El administrador requiere crear un ítem de no conformidad
Actores	Administrador
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado en el sistema 2. El usuario debe haber ingresado al sistema 3. El usuario debe tener acceso de administrador
Flujo básico de datos	<ol style="list-style-type: none"> 1. El sistema muestra al administrador un formulario donde debe ingresar los datos 2. El administrador ingresa los datos 3. El administrador da clic en crear ítem 4. El sistema crea el ítem y emite un mensaje de confirmación
Postcondiciones	Se crea un nuevo ítem de no conformidad
Excepciones	N/A

5.3.3. DIAGRAMA DE CLASES

El diagrama de clases es el más conocido y utilizado en el modelado de sistemas se usa comúnmente para percibir las relaciones que existen entre las clases del sistema, a través de este diagrama también se pueden visualizar las herencias de las clases, sus métodos y atributos.

Figura 6. Diagrama de clases



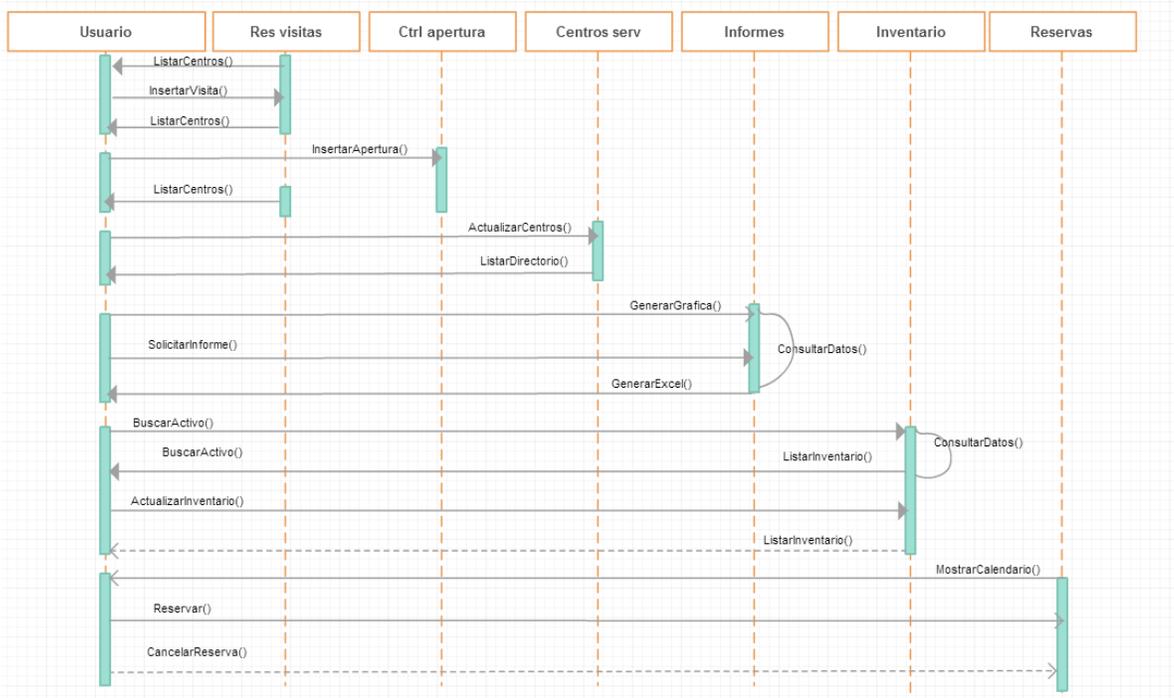
En el diagrama de clases encontramos los objetos fundamentales de este sistema, aquellos con los que el usuario tendrá contacto para llevar a cabo su objetivo. Además podemos percibir la cardinalidad entre las clases donde en su mayoría son de uno a muchos, exceptuando la relación de herencia entre resultados de visita y control de apertura donde esta última extiende los métodos de la anterior.

5.3.4. DIAGRAMA DE SECUENCIA

Los diagramas de secuencia nos muestran la interacción de un conjunto de objetos del sistema a través del tiempo, permiten aclarar detalles de implementación y conocer los mensajes que intercambian los objetos de una manera más clara.

A continuación se puede observar el diagrama de secuencia del sistema.

Figura 7. Diagrama de secuencia



El diagrama nos permite ver como es la comunicación del usuario con el sistema y dentro del mismo entre los objetos, se encuentra basado en los casos de uso y en el diagrama de clases debido a que incluye las clases con sus métodos y el momento en que serán ejecutados, como valor adicional incluye el tiempo de vida de los objetos que para el caso de este sistema no son dependientes debido al tipo de relación que existe entre las clases.

Se puede observar que el usuario interactúa constantemente con el sistema mientras que los objetos en la mayoría de los casos están para dar respuesta al usuario.

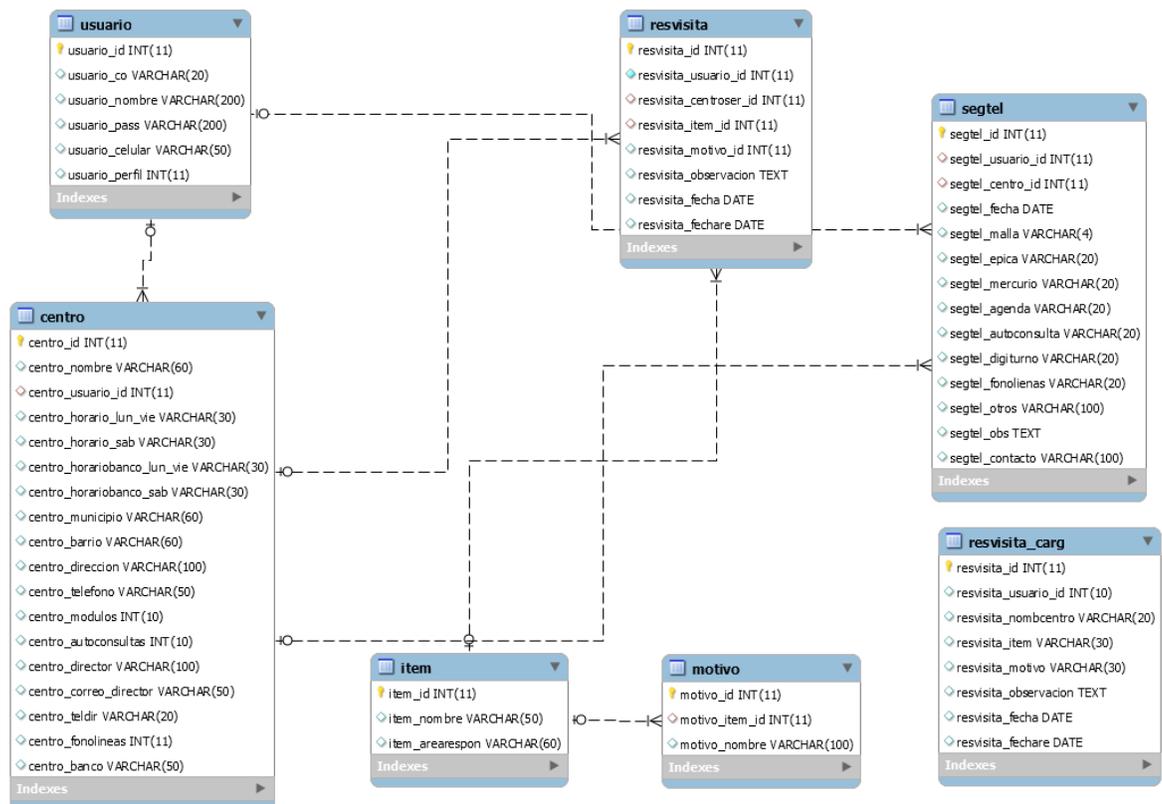
5.3.5. MODELADO DE BASE DE DATOS

La base de datos es el pilar del sistema pues nos permite almacenar y encontrar rápidamente grandes cantidades de información, para este sistema se modelo una base de datos relacional basada en el motor mysql.

Las tablas de la base se crearon a partir de un análisis de los requerimientos, su modelado tuvo varias fases pues hubo algunos cambios en los requerimientos, la base fue modelada creando un diagrama entidad relación lo que permitió representar las entidades, sus propiedades, las relaciones entre ellas y su estructura lógica.

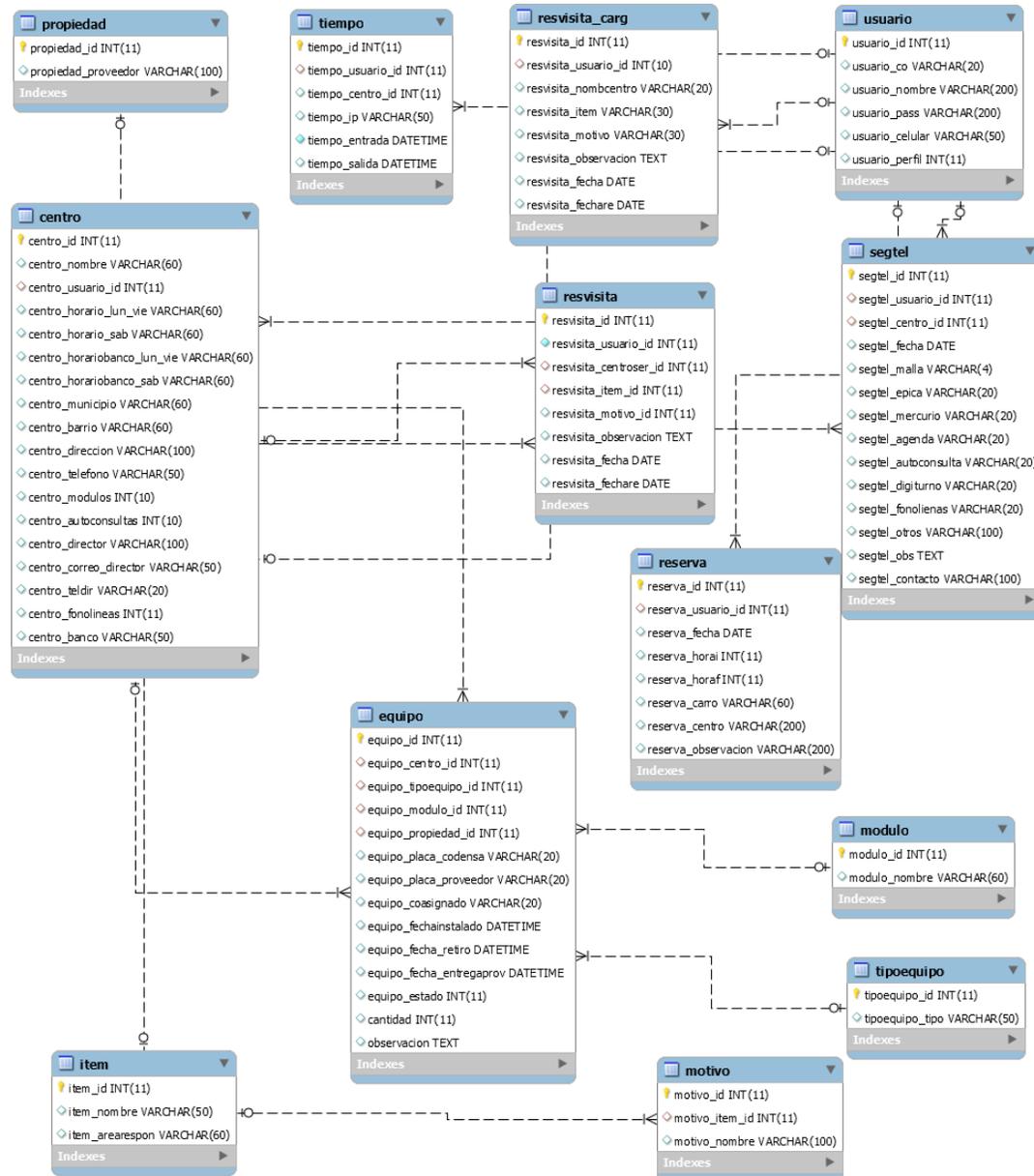
A continuación se puede observar el primer modelo entidad de relación del sistema antes de los nuevos requerimientos y los cambios en algunos de ellos.

Figura 8. Modelo entidad relación uno



Luego de algunos cambios en los requerimientos y la solicitud por parte del usuario de añadir nuevas funciones se volvió a modelar la base de datos usando el diagrama de entidad relación y este fue el resultado final.

Figura 9. Modelo entidad relación definitivo



En el diagrama anterior se puede evidenciar el gran crecimiento del sistema con los nuevos requerimientos, este diagrama no permite redundancia en los datos asegurando mantener la consistencia en la información.

5.4 DESCRIPCIÓN DEL SISTEMA

En la descripción del sistema se dará a conocer el funcionamiento en general, las partes que lo conforman y los requerimientos para su correcto funcionamiento.

El sistema fue desarrollado con las tecnologías web PHP para el manejo de las conexiones y la información del lado del servidor, Mysql como motor de bases de datos y Javascript, AJAX, HTML y CSS del lado del cliente para mejorar su interacción con la interfaz y hacer más dinámico el sistema.

Se encuentra conformado por dos tipos de usuario con distintos privilegios y objetivos, a continuación se darán a conocer los roles y una breve descripción de ellos.

- **Administrador:** Este usuario estará encargado de mantener el sistema actualizado en cuanto a cambios de gestores, centros de servicio o motivos de visita, tendrá los privilegios necesarios para crear centros de servicio, usuarios, ítems y motivos de no conformidades, también podrá eliminar usuarios y hacer cambios de gestores en el sistema.
- **Usuario:** Son los usuarios creados por el administrador, estarán encargados del ingreso de información por lo que tienen la responsabilidad de mantener el sistema actualizado con los datos más recientes, podrán ingresar resultados de visitas, control de apertura de los centros, actualizar datos de los centros de servicio, generar informes, actualizar el inventario y reservar vehículos.

A partir de la planeación y los requerimientos se desarrolló un sistema web de escritorio capaz de organizar la información de la División de Atención Presencial centralizándola, creando un estándar y generando datos útiles desde ella.

5.4.1 FUNCIONES ORIENTADAS AL ADMINISTRADOR

El sistema ofrece grandes opciones de administración para facilitar su manejo y asegurar un poco de escalabilidad. Se diseñó con el fin de poder ser administrado por un usuario común con unos conocimientos mínimos de informática.

Se desarrolló una interfaz muy sencilla e intuitiva que está dividida en cuatro secciones, tres de ellas orientadas al usuario y una administrativa. En las secciones orientadas al usuario se encuentran creación de centros de servicio, ítems y motivos. Dentro de creación de centros de servicio se pueden hacer cambios de gestores y crear centros de servicio. La segunda opción permite crear ítems de no conformidad y la última motivos asociados a ítems.

En cuanto a la sección administrativa permite la creación y eliminación de usuarios, estas funciones del rol de administrador permitirán mantener un estándar en la información y modificarlo de ser necesario.

5.4.2 FUNCIONES ORIENTADAS AL USUARIO

El sistema le brinda al usuario la posibilidad de mejorar el manejo de su información estandarizándola y manteniéndola íntegra y segura.

Durante el diseño del sistema se pensó una interfaz sencilla de fácil navegación donde el usuario final pueda moverse sin problemas para ello se creó un menú en el cual se clasifican las funciones según títulos descriptivos estos son seis y serán explicados a continuación:

- **Resultado de visitas:** Ofrece al usuario la posibilidad de ingresar al sistema lo que observe durante la visita directamente o cargando un archivo plano siempre siguiendo determinados estándares para la información.
- **Control de apertura:** Permite al usuario registrar el estado del centro de servicio al inicio de la actividad, registra datos sobre el personal y las aplicaciones que se usan durante el día.
- **Centros de servicio:** Esta opción permite al usuario modificar la información asociada a los centros de servicio con el fin de mantenerla

actualizada, además ofrece un directorio virtual donde se puede consultar datos de los centros como dirección, teléfono y director entre otros.

- **Informes:** Ofrece al usuario la posibilidad de generar informes de gestión en formato xls.
- **Inventario:** Permite modificar el inventario con el fin de mantenerlo actualizado y hacer búsquedas informativas.
- **Vehículos:** Esta opción brinda al usuario la posibilidad de reservar vehículos para poder realizar más cómodamente las visitas a los centros de servicio.

5.4.3 REQUERIMIENTOS DE HARDWARE Y SOFTWARE

En esta sección se mencionaran los requerimientos mínimos del sistema para su correcto funcionamiento en cuanto a software y hardware.

En los requerimientos mínimos para ejecutar el sistema como cliente se encuentran:

- Sistema operativo Windows 7, Linux, Mac
- Resolución de pantalla de 1000 x 600 px
- Navegador web, preferiblemente Mozilla o Chrome
- Procesador de 32 o 64 bits de 1GHz
- Conexión a internet
- Conexión a la VPN de Codensa
- Memoria RAM 2GB

Para obtener un excelente funcionamiento del lado del servidor son recomendadas las siguientes características:

- Sistema operativo Windows 7
- PHP5 o superior
- Motor de bases de datos Mysql 5 o superior
- Servidor web Apache o Nginx

- Memoria RAM 4GB
- Conexión a internet
- Conexión a la VPN de Codensa
- Procesador CORE i5 vpro 3.20 GHz
- Disco duro con 1GB libre

Al cumplir todos esos requisitos tanto el cliente como el servidor se asegura una ejecución óptima del sistema.

6. EVALUACIÓN ECONÓMICA DEL PROYECTO

6.1. RIESGO EN LA FASE DE ANÁLISIS

La fase de análisis es decisiva para el desarrollo de un proyecto por lo que se realizó con atención especial, buscando conocer las reglas del negocio, los roles de las personas en él y su relación con los centros de servicio. Se buscó detallar sus procesos y conocer la trazabilidad de la información con el objetivo de evitar posibles contratiempos durante la ejecución del proyecto, impedir desfases económicos y garantizar un trabajo fluido.

Durante el análisis se determinaron una lista de requerimientos funcionales y no funcionales de los cuales se obtuvieron los riesgos representados en la siguiente tabla.

Tabla 14. Riesgos en el análisis

Riesgo	Control	Respuesta	Consecuencia
Cambio de requerimientos	Definir de forma clara los requerimientos del usuario desde el inicio del proyecto	Toma y análisis de nueva información	Alta
Errores al tomar los datos	Estandarizar la toma de datos	Revisar de nuevo el proceso	Media

Errores de interpretación de los datos	Revisar con detalle los datos tomados durante el levantamiento	Entrevista con la persona encargada del proceso donde se tomaron esos datos	Baja
--	--	---	------

En el desarrollo del proyecto se presentaron requerimientos como cambios en la interfaz y algunas funcionalidades lo que implicó un aumento significativo en el tiempo presupuestado para el desarrollo de la aplicación y unas modificaciones apenas perceptibles en el alcance del proyecto.

6.2. RIESGO EN LA FASE DE DISEÑO

En la fase de diseño se tuvo el objetivo de representar el software con detalles suficientes para construirlo, se creó a partir de los requerimientos obtenidos durante la etapa de análisis con el fin de satisfacer todas las necesidades del usuario.

El diseño se construyó de tal manera que cumpliera los principios básicos, por lo que representa uniformidad e integración y escalabilidad, esta etapa se dividió en diseño de datos y de interfaz, durante el proceso se presentaron varios contratiempos afectando los tres ítems trabajados.

6.2.1. FASE DE DISEÑO DE DATOS

En esta fase las adversidades fueron causadas por el cambio de requerimientos y el requisito de poder cargar archivos de Excel al sistema, finalmente en lo que se refiere al cambio de requerimientos se aplicó la respuesta establecida para ese riesgo logrando constituirlos de manera rápida y efectiva asegurando perder un tiempo mínimo con esa dificultad.

En cuanto a la carga de archivos de Excel esta fue una dificultad grande pues consumió gran parte de tiempo del diseño debido a factores como la estandarización de los datos; ya que el archivo cargado no cumplía con las características de la información almacenada en la base de datos y arruinaría su grado de normalización.

Para dar solución a este problema se planteó inicialmente hacer un proceso de ETL (Extract, Transform and Load) capaz de ofrecer una solución consistente y efectiva al costo de un tiempo extra mientras se establecía o desarrollaba, debido al tiempo esta solución no fue viable lo que llevo a una solución más sencilla, crear un archivo Excel estandarizado controlando sus datos por medio de listas y filtros predefinidos una vez creado el archivo estándar, fue posible relacionar la tabla donde sería subida la información y mantener la base de datos en un cierto grado de normalización.

6.2.2. FASE DE DISEÑO DE DATOS

Durante el diseño de la interfaz se presentaron problemas menores basados en la compatibilidad de las tecnologías usadas con el navegador oficial de la empresa a lo que se plantearon diversas soluciones, entre estas el uso de Modernizr una librería de Javascript capaz de detectar el soporte de HTML5 y CSS3 para usar solo aquellos elementos soportados por Internet Explorer, otra solución posible fue el uso de Pollyfills (fragmentos de código que añaden funcionalidades HTML5 a los navegadores) estas soluciones fueron descartadas por no tener una compatibilidad completa lo que ocasionaba un cambio en la dimensión de los objetos de la interfaz, finalmente se optó por solo usar efectos básicos de CSS.

Otra dificultad que se presentó durante el diseño de la interfaz fue el cambio de requerimientos por parte del usuario, esto obligo a rediseñar el módulo de resultados de visita fue un cambio prácticamente completo por lo que tomo un tiempo significativo realizar este cambio.

Tabla 15. Riesgos en el diseño

Riesgo	Control	Respuesta	Consecuencia
Falta de compatibilidad	Uso de tecnologías con alta compatibilidad	Cambio de implementación del objeto no compatible	Baja
Información redundante	Pasar por filtros, organizar los datos	Aplicar técnicas de normalización	Alta
Daños o cambios en el formato estandarizado de Excel	Procedimiento almacenado capaz de eliminar datos o información "basura"	Monitoreo de la información en la base de datos	Alta
Uso de un formato no apto para cargar datos	Restricción en el sistema a otros formatos	Dejar el formato siempre disponible para su descarga	Alta

Búsquedas lentas en la base de datos	Tablas con índices correctos según el tipo de dato	Optimizar o realizar tuning a las consultas ejecutadas	Media
Perdida de información	Creacion de backups dos veces por semana	Importar backups al sistema	Alta

En la tabla anterior se resumen los posibles riesgos con sus medidas de control, respuesta y el nivel de impacto o las consecuencias que tienen en el sistema.

6.3. RIESGO EN LA FASE DE CODIFICACIÓN

En la fase de codificación se llevó a cabo todo el desarrollo del software este se dividió en dos etapas, interfaces para el ingreso de información y desarrollo del código para crear los informes descargables.

Durante en desarrollo de las interfaces de ingreso de información se presentaron algunos problemas entre ellos estuvo un cambio en un requerimiento, la interfaz de ingreso de resultados de visita tuvo que ser rediseñada para adaptarse al nuevo requerimiento extendiendo levemente el tiempo de desarrollo, también fue necesario implementar validaciones a los formularios usando JQuery para asegurar que los campos fueran completados correctamente y prevenir del lado del cliente vulnerabilidades que permitieran inyectar código SQL al sistema. Otro de los riesgos que se presentaron durante esta fase que tuvo una influencia significativa en el tiempo del desarrollo fue la falta de experiencia con las tecnologías de AJAX y CSS especialmente con este último debido a la falta de soporte en Internet Explorer.

Tabla 16. Riesgos en la codificación

Riesgo	Control	Respuesta	Consecuencia
Errores o mal funcionamiento en código PHP	Se controlan escribiendo un código limpio y haciendo pruebas constantes del desarrollo	Ubicar la línea, analizarla y reescribirla si es necesario	Alta
Errores en código Javascript	Escribir código limpio usando funciones específicas para cada problema	Usar la consola de los navegadores para detectarlo y corregirlo	Alta
Errores en código SQL	Ejecutar las consultas primero directamente en el motor de bases de datos	Imprimir la consulta y su respectivo error en pantalla para ubicarlo y corregirlo	Alta
Problemas de	Probar el software en	Modificar el código	Baja

visualización del contenido	distintos navegadores con el fin de asegurar buen rendimiento	CSS buscando obtener la máxima compatibilidad	
Seguridad del software	Mantener el software dentro de la VPN, crear copias de seguridad cada determinado tiempo, escapar caracteres no permitidos, cifrar contraseñas, crear niveles de acceso	Restablecer copias de seguridad, modificar contraseñas y/o usuarios	Alta
Perdida de información	Creación de copias de seguridad automatizadas.	Cargar backups de la información	Alta

6.4. RIESGO EN LA FASE DE PRUEBAS

Se aplicaron cinco tipos de pruebas con el fin de detectar distintos tipos de fallos tanto funcionales como estructurales.

Las pruebas en su mayoría fueron realizadas mientras se desarrollaba el software, durante ese proceso se realizaron pruebas unitarias de caja blanca consistieron en escoger distintos valores de entrada para ver la reacción del sistema durante el flujo de la información y comprobar que los datos de salida eran los adecuados, estas pruebas se realizaron en cada módulo creado conforme se avanzaba en él, gracias a estas pruebas se determinó en que módulos del sistema era necesario mejorar o implementar filtros en los datos de entrada, comparando el resultado esperado con el resultado obtenido, luego de esto se aplicaron los nuevos filtros o mejoras lo que afecto el tiempo del proyecto pero aumento la calidad del software.

Al sistema también se le realizaron pruebas de integración de caja negra durante el desarrollo del software, estas pruebas se realizaron con el fin de determinar si los requerimientos de diseño fueron alcanzados, si se producían errores al integrar los módulos probados de manera unitaria. En la mayoría de módulos se encontraron problemas de integración al comparar los resultados esperados con los obtenidos, los problemas se presentaron en la interfaz y el proceso de carga, selección o actualización en la base de datos, por lo que en algunos casos se requirió modificar la interfaz o una tabla específica en la base de datos para obtener los resultados esperados y cumplir con los requerimientos de los usuarios, el tiempo extra requerido para estos cambios fue poco pues a pesar de que se encontraron varios fallos eran muy sencillos de corregir.

La siguiente prueba aplicada fue de regresión la cual se realizó con el objetivo de ver si los nuevos módulos y los avances en el desarrollo tenían algún efecto negativo en los demás componentes del sistema, se probaron tanto las funcionalidades viejas como las nuevas, durante su desarrollo no se encontraron inconvenientes por lo que no retraso el avance del proyecto.

Entre las pruebas realizadas también se encuentra la de integridad de datos, que se realizó con el fin asegurar que todos los métodos de acceso a la base de datos de ejecutaran correctamente sin causar corrupción en los datos, las consultas y procedimientos almacenados fueron probados y optimizados directamente en el motor de la base de datos antes de ser llevados al sistema, luego de esto se analizaron los datos guardados y los datos retornados por las consultas para comprobar su integridad, esta fase retorno unos pocos errores de alto impacto para su corrección se requirió un tiempo significativo.

Por último se aplicó la prueba de control de acceso donde se verifico que cada usuario solo pudiera acceder a las funciones y datos designados para ese perfil, en esta prueba no se tuvo ningún contratiempo ya que desde un inicio el sistema estuvo enfocado en la correcta asignación de roles y permisos a los usuarios.

Durante la aplicación de este conjunto de pruebas también se validó la usabilidad, funcionalidad, la interfaz de usuario y su navegabilidad obteniendo la aprobación del usuario para llevar al sistema a producción.

6.5. RIESGO EN LA FASE DE IMPLEMENTACION

Una vez realizadas las pruebas al sistema se inició la fase de implementación esta se realizó a través de un paquete de herramientas llamado NMP Server el cual incluye un servidor web Nginx, PHP 5.4 y Mysql 5.5, se instaló en un equipo de la División de Atención Presencial dándole permisos al servidor para ser accesible en la VPN de Codensa, esta fase también incluyo capacitaciones y explicaciones del sistema a los usuarios además de acompañamiento con el fin de observar los errores encontrados por los usuarios y corregirlos, tomar sus sugerencias y mejorar las funcionalidades del sistema, durante este proceso se encontraron sugerencias pequeñas para mejorar la funcionalidad que prácticamente no afectaron la duración estimada para el proyecto .

6.6. RIESGO EN LA FASE DE MANTENIMIENTO

Durante esta fase se considera realizar mantenimientos correctivos, preventivos y adaptivos con el fin de mantener el sistema con el mejor rendimiento posible y adecuado a las nuevas necesidades del usuario, en el desarrollo de los mantenimientos se tiene en cuenta el riesgo de afectar la funcionalidad del sistema conforme se vayan implementando nuevos módulos o mejoras de seguridad, también una posible dificultad al leer el código que se puede dar si las personas que lo desarrollaron no son las mismas que le harán mantenimiento, para evitar este último problema se dejó con comentarios el código facilitando así su mantenimiento e integración con nuevas funciones desarrolladas luego de la implementación.

7. PRESUPUESTO DETALLADO

7.1. COSTO DE LA INFRAESTRUCTURA

La inversión en infraestructura física no fue un gran costo en el proyecto pues por las características del sistema no requería grandes capacidades de procesamiento o almacenamiento debido a que la concurrencia máxima esperada no supero los veinte usuarios.

A continuación se relaciona el costo y las características del equipo (servidor) destinado a alojar la aplicación, el servidor web y de bases de datos.

PC HP EliteDesk 800 G1

Sistema operativo: Windows 7 Enterprise 32 bits

Procesador: Intel® Core™ i5-4570 con Intel HD Graphics 4600 (3,2 GHz, 6 MB de caché, 4 núcleos)

Memoria RAM: SDRAM DDR3 1600 MHz de 4 GB

Disco Duro: SATA de 500 GB, 7200 rpm

Interfaz de red: Conexión de red Gigabit Intel® I217LM

El equipo tuvo un costo de un millón quinientos mil pesos, el cual fue el total de la inversión en infraestructura física.

8. BENEFICIOS DE LA IMPLEMENTACIÓN

8.1. OPERACIONALES

En cuanto a los beneficios operacionales los procesos de manejo de información mejoraron considerablemente en los siguientes puntos:

- Centralización de la información: con la implementación del sistema se centralizo toda la información correspondiente a los centros de servicio manejada por sus gestores.
- Estandarización de la información: El sistema unifico la información contribuyendo a mantener la integridad y reducir el desorden que se presenta cuando no se tiene un formato único.
- Facilidad de manejo: Al tener un formato estándar se pueden manejar mejor los datos para obtener información relevante de ellos.
- Accesibilidad a la información: Los gestores pueden acceder con mayor facilidad y orden a la información ingresada por ellos mismos o sus compañeros.
- Compartir información: Los gestores podrán compartir la información con mayor facilidad y claridad entre ellos mismos o con otras personas

8.2. GESTIÓN

Los beneficios de gestión se ven a corto plazo pues el sistema se alimenta de la información ingresada y a partir de ella genera informes que sirven para identificar los problemas que más se presentan en los centros de servicio y el tiempo dedicado a la gestión del centro de servicio entre otros.

La siguiente es una lista de informes que genera el sistema con el fin de tomar decisiones que permitan tener una óptima gestión de los centros de servicio:

- No conformidades: Este reporte muestra las no conformidades registradas en los centros de servicio con su respectiva observación.
- Indicadores: Este reporte indica el número de visitas realizadas por cada gestor a un centro de servicio en un rango de fechas determinado por el usuario
- No conformidades reiterativas: Este reporte indica los problemas que más se repiten en los centros de servicio.

Con base en estos reportes obtenidos con la información cargada los gestores pueden tomar decisiones sobre problemas que se presenten constantemente, cuánto tiempo dedicar a su gestión o a que puntos deben visitar más.

8.3. ESTRATÉGICOS

En cuanto a beneficios estratégicos la División de Atención Presencial es la única con un sistema interno de gestión de información lo que le da la ventaja de mantener la integridad de la información haciendo que los informes y reportes creados con ella sean más veraces y objetivos frente a los de otras divisiones de Codensa, esto le permitirá obtener una posición destacada en la gerencial comercial.

Además el sistema también trae la ventaja de evitar pérdida de tiempo recopilando la información esto implica un aumento de la productividad gracias a que estos tiempos podrán aprovecharse en otras actividades.

Con la información proporcionada por el sistema se mejora la capacidad para tomar decisiones, facilita compartirla y genera conocimiento del negocio con el análisis de los informes por parte de los usuarios.

8.4. BENEFICIOS DE TI

Los beneficios en cuanto a tecnologías de la información que trae el sistema para la División de Atención Presencial son:

- Proporciona ventajas competitivas debido a que es la única división con sistema de información propio
- Integra nuevas tecnologías aprovechando las funciones de internet, mejorando la accesibilidad a la información.
- Permite analizar más fácilmente el valor de las actividades para continuar con ellas, mejorarlas o eliminarlas.
- Facilita enormemente el control de los inventarios.
- Amplia accesibilidad pues se puede agregar información al sistema desde cualquier centro de servicio o desde cualquier lugar conectándose a la VPN
- Las operaciones son más efectivas cuando la información es más acertada

9. ALCANCES Y LIMITACIONES DEL PROYECTO

Este proyecto es el inicio hacia la sistematización de la información de la División de Atención Presencial, estará concentrado la gestión de los datos generados por los gestores, permitiendo generar informes de indicadores de gestión a través de los datos ingresados en él.

Inicialmente se definieron dos versiones entregables del sistema siguiendo una metodología de desarrollo iterativa, en la primera versión se contempló:

- Análisis de requerimientos
- Diseño de base de datos
- Diseño de módulos
- Programación fase uno, ingreso de información
- Programación fase dos, generación de un informe.

En esta entrega se definió como única información que trataría el sistema aquella generada por los gestores al visitar los centros de servicio, no conformidades. Estas se clasificaron en una serie de ítems:

- Mantenimiento locativo
- Vigilancia
- Tecnología
- Autoconsultas
- Fonolíneas
- Malla de turnos
- Estándares de servicio
- Papelería y archivo
- Cajeros
- Buzón de cliente
- Sin novedades

Las funcionalidades definidas para estas no conformidades se centraron en el ingreso de información y la generación de un único reporte donde se vieran reflejadas todas las no conformidades ingresadas filtradas por un rango de fechas.

La segunda entrega se realizó partiendo de las sugerencias de los usuarios y las funciones aún no incluidas en la primera entrega en esta se tuvieron en cuenta los siguientes puntos:

- Análisis de requerimientos
- Adaptación de la base de datos
- Diseño de módulos
- Programación tres, generación de todos los informes
- Programación fase cuatro, Inventario y programación de camionetas
- Fase de pruebas
- Fase de implementación
- Mantenimiento

Para la segunda entrega se definió ampliar la información que manejaría el sistema, este se extendió haciendo crecer la base de datos, se incluyó información de inventarios, reservas de vehículos y se incluyeron motivos aún más específicos en las no conformidades para lograr una mejor clasificación y mayor claridad en la información. En esta entrega se perfeccionaron los módulos y se añadieron dos más quedando el sistema con su funcionalidad completa y abriendo paso a incluir la información tratada por otros roles dentro de la División de Atención Presencial.

10. CRONOGRAMA

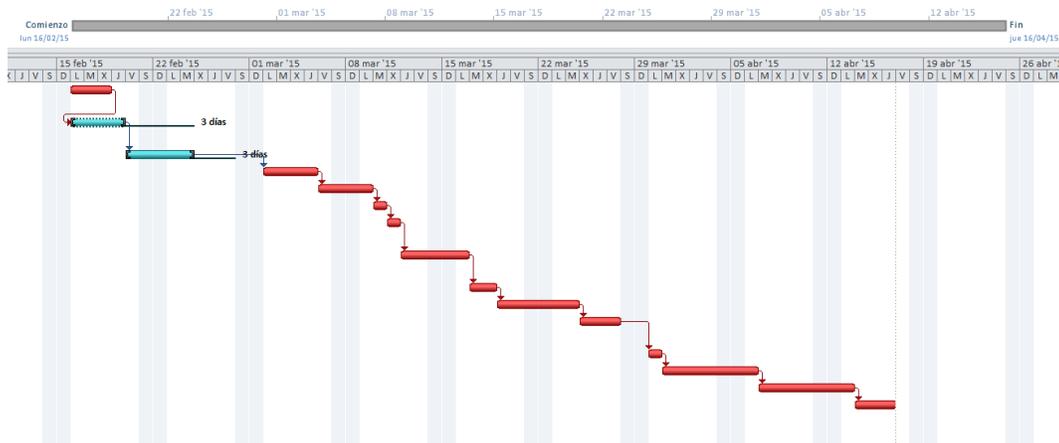
El cronograma se realizo teniendo en cuenta las fases de desarrollo contempladas en la metodología escogida, al ser reiterativa se tienen dos fechas de entrega y se repiten algunas actividades por iteración, en el cronograma se puede detallar la duración y la actividad que debió realizarse previamente.

Figura 10. Cronograma de actividades

	Modo de	Nombre	Retraso por redist.	Duración	Comienzo	Fin	Sucesoras
1		Análisis de requerimientos	0 díast	3 días	lun 16/02/15	mié 18/02/15	2
2		Diseño de base de datos	0 díast	4 días	lun 16/02/15	jue 19/02/15	3
3		Diseño de módulos	0 díast	3 días	vie 20/02/15	mar 24/02/15	4
4		Programación Fase uno	0 díast	4 días	lun 02/03/15	jue 05/03/15	5
5		Programación fase dos	0 díast	2 días	vie 06/03/15	lun 09/03/15	6
6		Primera entrega	0 díast	1 día	mar 10/03/15	mar 10/03/15	7
7		Análisis de requerimientos	0 díast	1 día	mié 11/03/15	mié 11/03/15	8
8		Diseño de base de datos	0 díast	3 días	jue 12/03/15	lun 16/03/15	9
9		Diseño de módulos	0 díast	2 días	mar 17/03/15	mié 18/03/15	10
10		Programación fase tres	0 díast	4 días	jue 19/03/15	mar 24/03/15	11
11		Pogramación fase cuatro	0 díast	3 días	mié 25/03/15	vie 27/03/15	12
12		Entrega	0 díast	1 día	lun 30/03/15	lun 30/03/15	13
13		Pruebas	0 díast	5 días	mar 31/03/15	lun 06/04/15	14
14		Documentacion	0 díast	5 días	mar 07/04/15	lun 13/04/15	15
15		Implementacion	0 díast	3 días	mar 14/04/15	jue 16/04/15	

A continuación en el diagrama de Gantt se puede observar gráficamente el tiempo previsto para las diferentes actividades a lo largo de tiempo total del proyecto.

Figura 11. Diagrama de Gantt



11. CONCLUSIONES

El desarrollo de la pasantía fue indispensable para mi formación como ingeniero de sistemas pues tuve la oportunidad de aplicar mi conocimiento, reforzarlo y aprender más sobre el campo del desarrollo, las bases de datos y el manejo de proyectos. Además aumento mi experiencia laboral, conocí el contexto de trabajo en una empresa grande obteniendo una perspectiva más amplia.

Las actividades propuestas se cumplieron completamente y de forma satisfactoria, todas estas fueron de mi agrado lo que me motivo aún mas a complementar mi carrera enfocándome en las bases de datos

Asimismo el proyecto también fue de gran ayuda para la División de Atención Presencial debido a que les otorgo ventajas sobre otras divisiones y comodidades para el manejo de la información pudiendo organizarla, centralizarla y generar reportes con ella.

12. BIBLIOGRAFÍA

- Aguilar, Luis Joyanes y Zahonero Martinez, Ignacio. 2014.** *Programación en C, C++, Java y UML*. Mexico : Mc Graw Hill, 2014.
- Fontela, Carlos. 2011.** *UML modelado de software para profesionales*. Buenos Aires : Alfaomega, 2011.
- Google. 2010.** AngularJS. [En línea] 2010. [Citado el: 22 de 05 de 2015.] <https://angularjs.org>.
- Grupo Energía de Bogotá.** Grupo Energía de Bogotá. *Grupo Energía de Bogotá*. [En línea] [Citado el: 3 de Abril de 2015.] <http://www.grupoenergiadebogota.com/nuestras-empresas/empresas-participadas/codensa>.
- IBM. 2010.** IBMdeveloperWorks. [En línea] 22 de Noviembre de 2010. [Citado el: 9 de Abril de 2015.] <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Rational+Team+Concert+for+Scrum+Projects/page/SCRUM+como+metodolog%C3%A1Da>.
- Instituto Nacional de Tecnologías de la Comunicación. 2009.** *Ingeniería del Software: Metodologías y Ciclos de Vida*. España : Ministerio de Industria, Turismo y Comercio, 2009.
- Sutherland, Jeff y Schwaber, Ken. 2013.** Scrum Guide. [En línea] [scrumguides.org](http://www.scrumguides.org), 01 de 07 de 2013. [Citado el: 24 de 04 de 2015.] <http://www.scrumguides.org>.
- The jQuery Foundation. 2014.** jQuery. [En línea] 2014. [Citado el: 21 de 04 de 2015.] <http://api.jquery.com>.
- The Object Management Group.** Unified Modeling Language™ (UML®). [En línea] [Citado el: 05 de 05 de 2015.] <http://www.omg.org/spec/UML>.
- The PHP Group. 2015.** <http://php.net>. [En línea] 14 de 05 de 2015. [Citado el: 22 de 05 de 2015.] <http://php.net>.
- Véronneau, Martin. 2012.** Nmédia Solutions Inc. *Nmédia Solutions Inc*. [En línea] 10 de Agosto de 2012. [Citado el: 3 de Abril de 2015.] <http://www.nmediasolutions.com/publications/conseils/histoire-developpement-applications-web>.
- W3. 2013.** W3schools. [En línea] 2013. [Citado el: 15 de 05 de 2015.] <http://www.w3schools.com/css/>.
- . 2015.** World Wide Web . [En línea] W3, 2015. [Citado el: 24 de 04 de 2015.] <http://www.w3.org>.

