

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

TRABAJO DE DIPLOMA

Título del trabajo: "Sistema para la gestión planificada de actividades en la División Territorial Desoft SS"

Autores del trabajo: Islen del Rio Ramirez

Tutores del trabajo: MsC. Frank Reyes García

Ing: Yunier García Sánchez

Santa Clara, Junio, 2018
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-1419

Pensamiento

“La mejor estructura no garantizara los resultados
ni el rendimiento. Pero la estructura equivocada
es una garantía del fracaso”

Dedicatoria

Este trabajo está dedicado a esa personita que luchó a mi lado, cada segundo, cada minuto para que todo a mi alrededor saliera como deseaba, esa persona que dejó de dormir, de comer para atenderme en todo momento, esa personita eres tú MAMÁ.

Agradecimientos

GRACIAS a Dios.

A mi familia, mis abuelos, mis padres y mi hermano, que siempre estuvieron en todos los malos y buenos momentos. En especial a mi madre por su empeño, dedicación y esfuerzo porque sin ti no sería nada en este mundo. Ojalá pueda llegar a ser la mitad de buena que tú eres.

A mis amigos, por creer en mí y apoyarme siempre. Algunos de ustedes son parte de mi familia.

A mis profesores que fueron de mucha ayuda para que todo esto se hiciera posible.

A mis tutores por soportarme y dedicarme parte de su tiempo.

A todas las personas que nunca dejaron de creer que los sueños podían hacerse realidad.

Resumen

La empresa de Aplicaciones Informáticas Desoft está organizada por una Oficina Central y 16 Unidades Básicas Empresariales a lo largo de todo el territorio nacional, tiene como misión desarrollar y comercializar productos, así como servicios informáticos integrales asociados a Tecnologías de la Información, contribuyendo al desarrollo sostenible de la sociedad. La división Territorial de Sancti Spíritus realiza la gestión y planificación actualmente haciendo uso de las aplicaciones ofimáticas Excel y Word la cual presenta deficiencias en su elaboración, presentando demoras de hasta 2 días en finalizar por completo su realización, lo cual implica falta de agilidad y comodidades para la entidad. Es por ello que el objetivo de este trabajo de diploma es desarrollar un sistema para la gestión de la planificación y control de las actividades en la División Territorial utilizando Angular como marco de trabajo. Como resultado se obtuvo una aplicación web que logra un mejor rendimiento y eficiencia en el trabajo de planificación de actividades en la empresa.

Abstract

Desoft Computer Applications Company is organized by a Central Office and 16 Basic Business Units throughout the national territory, whose mission is to develop and market products, as well as comprehensive IT services associated with Information Technologies, contributing to sustainable development of the society. The Territorial division of Sancti Spíritus performs the management and planning currently making use of Excel and Word office applications which has deficiencies in its preparation, presenting delays of up to 2 days in completing its, which implies lack of agility and comfort for the entity. That is why the objective of this diploma work is to develop a system for managing the planning and control of activities in the Territorial Division using Angular as a framework. As a result, a web application was obtained that achieves a better performance and efficiency in the work of planning activities in the company.

INDICE

| | |
|---|-----------|
| Capítulo 1 Fundamentación teórica | 1 |
| 1.1. Objetivos estratégicos de la organización | 1 |
| 1.2. Objeto de estudio | 1 |
| 1.3. Análisis crítico de la ejecución de los procesos | 2 |
| 1.4. Sistemas automatizados existentes vinculados al campo de acción. | 3 |
| 1.5. Fundamentación de los objetivos | 3 |
| 1.6. Aspectos más significativos de la planificación de actividades | 3 |
| 1.7. Tendencias y Tecnologías Actuales. Fundamentación de la Metodología utilizada | 5 |
| 1.8. Fundamentación del Entorno de Desarrollo, Lenguaje, Gestor de Base de Datos y Tecnología utilizados | 6 |
| 1.8.1. Servidor utilizado: Node.js | 6 |
| 1.8.2. Entorno de Desarrollo: WebStorm | 6 |
| 1.8.3. Lenguajes utilizados | 6 |
| 1.8.3.1. Java Script | 7 |
| 1.8.3.2. HTML | 7 |
| 1.8.4. Gestor de Base de Datos: PostgreSQL | 8 |
| 1.8.5. Framework utilizados: AngularJs | 8 |
| 1.8.5.1. Embarcadero ER/Studio | 9 |
| 1.8.5.2. Visual Paradigm | 9 |
| Conclusiones parciales | 9 |
| Capítulo 2 Modelo del Negocio y Requisitos | 10 |
| 2.1 Modelo del negocio actual | 10 |
| 2.2 Reglas del negocio a considerar | 10 |
| 2.3 Actores del negocio | 11 |
| 2.4 Trabajadores del negocio | 11 |
| 2.5 Diagrama de casos de uso del negocio | 12 |
| 2.6 Descripción de los casos de uso del negocio | 13 |
| 2.7 Actores del sistema a automatizar | 14 |
| 2.8 Definición de los requisitos | 15 |
| 2.9 Requisitos funcionales | 15 |
| 2.10 Definición de los requisitos no funcionales | 19 |
| 2.11 Diagrama de Casos de Uso del Sistema | 20 |
| 2.11.1 Descripción de los casos de uso del Sistema | 21 |
| 2.12 Estimación por Puntos de caso de uso | 28 |

| | |
|---|-----------|
| Conclusiones parciales..... | 31 |
| Capítulo 3 Descripción de la propuesta de solución | 32 |
| 3.1 Arquitectura del sistema..... | 32 |
| 3.2 Diagrama de clases de diseño | 34 |
| 3.3 Diagrama de secuencia | 36 |
| 3.4 Diseño de la Base de Datos | 38 |
| 3.4.1 Modelo Conceptual de datos | 38 |
| 3.4.2 Modelo Lógico de Base de datos | 39 |
| 3.5 Diagrama de despliegue..... | 40 |
| 3.6 Modelo de Componentes..... | 41 |
| Capítulo 4 Pruebas de análisis y factibilidad | 43 |
| 3.1 Pruebas de rendimiento:..... | 43 |
| 3.2 Casos de prueba (caja negra)..... | 45 |
| Conclusiones parciales..... | 48 |
| Conclusiones..... | 49 |
| Recomendaciones..... | 50 |
| Bibliografía..... | 51 |

Lista de Figuras:

| | |
|---|----|
| FIGURA 1 CASO DE USO DEL NEGOCIO..... | 13 |
| FIGURE 2 DIAGRAMA DE CASO DE USO DEL SISTEMA | 21 |
| FIGURE 3 ARQUITECTURA DEL SISTEMA(API REST) | 33 |
| FIGURE 4 ARQUITECTURA DEL SISTEMA(MVC) | 33 |
| FIGURE 5 DIAGRAMA DE CLASES DEL DISEÑO..... | 34 |
| FIGURE 6 DIAGRAMA DE CLASES, CASO DE USO: GESTIONAR ACTIVIDADES PLAN ANUAL..... | 35 |
| FIGURE 7 DIAGRAMA DE CLASES, CASO DE USO: GESTIONAR PLAN ANUAL | 36 |
| FIGURE 8 DIAGRAMA DE SECUENCIA CASO DE USO :INSERTAR PLAN ANUAL..... | 37 |
| FIGURE 9 DIAGRAMA DE SECUENCIA CASO DE USO: ADICIONAR ACTIVIDAD PLAN ANUAL.... | 38 |
| FIGURE 10 MODELO CONCEPTUAL DE BASE DE DATOS | 39 |
| FIGURE 11 MODELO LÓGICO DE BASE DE DATOS..... | 40 |
| FIGURE 12 DIAGRAMA DE DESPLIEGUE..... | 41 |
| FIGURE 13 MODELO DE COMPONENTES CASO DE USO: GESTIONAR PLAN ANUAL..... | 42 |
| FIGURE 14 FORMULARIO INSERTAR PLAN ANUAL..... | 45 |
| FIGURE 15 FORMULARIO INSERTAR PLAN ANUAL CON DATOS CORRECTOS | 47 |
| FIGURE 16 FORMULARIO INSERTAR PLAN ANUAL CON CAMPOS EN BLANCO..... | 47 |

Lista de Tablas:

| | |
|--|----|
| TABLA 1 ACTOR DEL NEGOCIO..... | 11 |
| TABLA 2 TRABAJADORES DEL NEGOCIO | 12 |
| TABLA 3 DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO | 14 |
| TABLA 4 ACTORES DEL SISTEMA A AUTOMATIZAR | 15 |
| TABLA 5 DESCRIPCIÓN DEL CASO DE USO: GESTIONAR NOMENCLADORES-CAPÍTULOS | 24 |
| TABLA 6 DESCRIPCIÓN DEL CASO DE USO: GESTIONAR ACTIVIDADES | 26 |
| TABLA 7 DESCRIPCIÓN DEL CASO DE USO: GESTIONAR PLAN ANUAL | 28 |
| TABLA 8 FACTOR DE COMPLEJIDAD | 29 |
| TABLA 9 FACTORES DE AMBIENTE O ENTORNO..... | 30 |
| TABLA 10 PRUEBA DE CARGA | 44 |
| TABLA 11 PRUEBA DE STRESS..... | 44 |
| TABLA 12 PRUEBA DE RESISTENCIA | 45 |
| TABLA 13 DATOS QUE DEBEN INTRODUCIRSE EN EL FORMULARIO DE INSERTAR PLAN ANUAL..... | 46 |
| TABLA 14 DATOS INSERTADOS CORRECTAMENTE EN EL FORMULARIO INSERTAR PLAN ANUAL..... | 46 |
| TABLA 15 DATOS QUE DEBEN INTRODUCIRSE VACÍOS EN EL FORMULARIO DE INSERTAR PLAN ANUAL..... | 47 |
| TABLA 16 DATOS QUE DEBEN INTRODUCIRSE CON ERROR EN EL FORMULARIO DE INSERTAR PLAN ANUAL. | 47 |

Introducción:

Muchas empresas de hoy en día basan su éxito principalmente en el desarrollo de una aplicación que les permita funcionar de una mucho más eficaz en un mundo empresarial cada vez más complejo. Desoft(Empresa de Aplicaciones Informáticas), es una empresa de soluciones informáticas con más de 15 años de experiencia en el sector TI(Tecnologías de la información), con un capital humano de más de 2 mil empleados. Sus ciclos significan Empresa de Aplicaciones Informáticas. Sus orígenes se deben a una empresa fundada en octubre de 1977 con el nombre Buró SAD (Buró Central de Diseño de Sistemas Automatizados de Dirección) cuya función principal era el desarrollo de software. A partir de 1986, se suceden una serie de transformaciones y fusiones entre empresas dedicadas a las ciencias informáticas, donde Buró SAD es fusionada con Servicec (Empresa de Servicios a Centros de Cálculo), formándose la Empresa Nacional de Aplicaciones, Enap, la que luego a su vez fue fusionada con Servitec, asumiendo esta vez el nombre de ESICH (Empresa de Servicios Informáticos de la entonces Ciudad de la Habana). Años después en 1995, ESICH sufre transformaciones estructurales que la llevan a adoptar el nombre de Empresa de Desarrollo y Producción de Software de Calidad, Softcal. A finales del año 2003, las Empresas de Servicios Informáticos existentes a lo largo de todo el país, la entonces Centersoft y parte de la actual empresa Softel, se unen a Softcal.

Con toda la experiencia y tradición de trabajo acumulada en su personal a través del tránsito por las múltiples transformaciones, a partir del año 2004, SOFTCAL se convierte en la Empresa Nacional de Software, Desoft S.A. teniendo como misión desarrollar y comercializar productos y servicios informáticos integrales asociados a las Tecnologías de la Información, contribuyendo al desarrollo sostenible de la sociedad. Como resultado de estudios y propuestas de perfeccionamiento ministeriales, Desoft S.A. modifica el 5 de enero de 2015 su régimen jurídico constituyéndose como Empresa Estatal Socialista cubana y cambiando su denominación a la actual Empresa de Aplicaciones Informáticas, Desoft.

La empresa de Aplicaciones Informáticas está organizada por una Oficina Central y 16 Unidades Básicas Empresariales a lo largo de todo el territorio nacional, las son las encargadas de cumplir los objetivos de la empresa y una Unidad Integradora de Negocio que gestiona y coordina los proyectos de carácter nacional dígame empresas con dependencias en todo el país. La empresa de Aplicaciones Informáticas Desoft tiene como objeto social:

1. Prestar servicios de desarrollo, despliegue y soporte de aplicaciones informáticas integrales.
2. Brindar servicios públicos de Alojamiento, Hospedaje y Aplicaciones.
3. Gestionar y administrar aplicaciones informáticas en Centros de Datos.

Entre las soluciones informáticas de mayor éxito en el mercado nacional se encuentran las de gestión y control de procesos empresariales, por lo que una adecuada planificación de las actividades permite el mejor desempeño de los colaboradores, de las diferentes dependencias de la organización y del adecuado uso de los recursos. La planificación es un factor clave para lograr un adecuado manejo corporativo del negocio, pues proporciona la seguridad respecto al logro de los objetivos y metas trazadas dentro del marco de la eficiencia, eficacia y economía; éstos se constituyen como los tres pilares que sostienen y resguardan los recursos y bienes de la empresa. La identificación de objetivos constituye un requisito imprescindible para la planificación, ya que asegura el continuo desarrollo del país, con el empleo racional y eficiente de los recursos disponibles a corto, mediano y largo plazos, así como la medición y evaluación sistemática de estos.

Por lo tanto, es indispensable la planificación, con el fin de ofrecer una mejor visión sobre la gestión empresarial y así poder obtener procesos productivos y gerenciales de alto nivel. (Armijo, 2009, Martínez., 2017-2020)

Dicho proceso es realizado de forma manual en la entidad, por lo que carece de agilidad y detección de errores proponiendo como **problema científico** de esta investigación consiste en ¿Cómo mejorar el proceso de la planificación de actividades de la División Territorial Desoft a través de una aplicación web que permita la gestión de tareas y organizaciones?

Problema de investigación:

En la Empresa de Aplicaciones Informáticas Desoft se realiza la planificación de las actividades anualmente, que responda a los objetivos de trabajo de las divisiones hasta llegar al plan individual de cada trabajador. A partir de la planificación enviada desde la empresa nacional, se elabora el plan mensual de la división y en base al anterior cada trabajador elabora su plan individual. Existen actividades que se realizan mensualmente y deben ser ejecutadas por todos los trabajadores. Al final del mes debe chequearse el cumplimiento de las mismas cuantitativamente teniendo cuenta la cumplidas e incumplidas (incluyendo las causas), así como las nuevas tareas incorporadas. Todo el proceso se realiza de forma manual en plazo de

tiempo determinado, probando errores y demoras en el proceso, ¿Cómo automatizar y controlar este proceso entonces?

Objetivo General.

Desarrollar un sistema para la gestión de la planificación y control de las actividades en la División Territorial Desoft SS.

Objetivos específicos:

- ✓ Diseñar una Base de Datos que permita gestionar la información referente a la planificación de actividades de la División Territorial Desoft SS.
- ✓ Diseñar e implementar una aplicación web para la Gestión de la planificación y control de las actividades.
- ✓ Validar la efectividad de la aplicación a partir de pruebas realizadas a la misma.

Estructura del documento:

Este presente trabajo se conforma de 4 capítulos.

Capítulo I: Fundamentación teórica: Se tratan los aspectos de forma general con relación a la planificación de actividades, además se tratan los aspectos relacionados con el uso de las tecnologías y herramientas utilizadas en el desarrollo informático.

Capítulo II: Modelo de Negocio y Requisitos: Se abarca todo modelado del negocio y sus características y así como la captura de requisitos funcionales y no funcionales del sistema.

Capítulo III: Descripción de la solución propuesta: Se describe muy específicamente la propuesta de solución a través de sus diagramas.

Capítulo IV: Pruebas y análisis de factibilidad: Se prueba la aplicación a partir de la estrategia elaborada para la misma.

Capítulo 1 Fundamentación teórica

En este capítulo se presentan varios aspectos fundamentales de la Empresa de Aplicaciones de Software (Desoft) SS. En el mismo se realiza un análisis detallado de los procesos y sus metodologías a utilizar. Se muestran las tecnologías que serán de ayuda para el sistema.

1.1. Objetivos estratégicos de la organización

Los objetivos estratégicos se basarán en la visión o idea que nazca con la empresa, determinando su misión y valores, además de condicionar las acciones que se llevarán a cabo para poder conseguirlos.

La empresa Desoft se apoya en dos pilares fundamentales: un personal calificado y una infraestructura tecnológica que garantice la calidad de los productos y servicios por lo que basado en esto tiene como objeto estratégico:

Ofrecer líneas de servicios que incluya el desarrollo, despliegue y soporte de soluciones informáticas de administración empresarial para la gestión de recursos humanos, contabilidad y finanzas, recursos energéticos, inteligencia de negocios, entre otros.

Desarrollar y comercializar productos y servicios informáticos integrales asociados a las Tecnologías de la Información, contribuyendo al desarrollo sostenible de la sociedad

1.2. Objeto de estudio

Los aspectos más importantes para lograr resultados positivos para un cliente es el establecimiento de objetivos y la identificación y planificación de actividades para tratar sus necesidades por lo cual la empresa de Desoft realiza este proceso de planificación de la siguiente manera:

Se planifica anualmente las actividades para el año a partir de las puntualizaciones que se reciben de los diferentes niveles: subdirecciones, dirección territorial, nivel central y grupo empresarial; además de las principales tareas contenidas en los planes del Partido, Asambleas del Poder Popular, CTC entre otras. Confeccionándose finalmente el Plan Anual de Actividades para el Período (PAP).

A partir del PAP se elaboran mensualmente los planes de trabajo de la entidad. A los cuales se le pueden agregar nuevas actividades o se modifican las que se planificaron inicialmente.

A partir del Plan de Trabajo mensual de la entidad, cada jefe, cuadro o especialista debe elaborar su Plan de Trabajo Individual, el cual debe ser firmado por su jefe inmediato superior.

Al finalizar cada mes, los jefes, cuadros y especialistas deben elaborar el informe de cumplimiento de sus planes de trabajo individuales. A partir de estos se elabora el Informe de cumplimiento del Plan de trabajo de la entidad. Estos informes muestran el total de tareas cumplidas, incumplidas y nuevas; así como un análisis cualitativo del trabajo del mes.

1.3. Análisis crítico de la ejecución de los procesos

El análisis de procesos es el acto de llevar a cabo una revisión exhaustiva y llegar a una comprensión completa de un proceso de negocio (o parte de éste) con el objetivo de mantener o lograr la excelencia del proceso o lograr mejoras incrementales o transformacionales en un proceso empresarial.(Ardila and Pino, 2013) La ejecución del proceso de planificación se conforma de la siguiente forma:

Administrador: tiene como tarea gestionar nomencladores y gestionar los usuarios.

Planificador: gestiona actividades para el plan anual y mensual de la División. Podrá gestionar el Plan anual de actividades y el plan mensual de la División y podrá generar el informe de cumplimiento del plan de trabajo mensual de la División.

Director: gestiona actividades en el plan anual y dar el aprobado del mismo.

Jefe de área: gestiona actividades para los especialistas de su área.

Trabajador: gestiona actividades individuales, generar el plan de trabajo individual y el informe de cumplimiento. Podrá visualizar el plan anual y los planes mensuales de la División.

Supervisor: visualiza los planes de trabajo anuales, mensuales e individuales.

1.4. Sistemas automatizados existentes vinculados al campo de acción.

En la bibliografía consultada se encontraron varios sistemas para la planificación de proyectos de desarrollo de software como: openProj, Open Work bench los que planifican solo las actividades relacionadas con las etapas del desarrollo de un software. (2012) También se identificó el Sistema informático para la planificación y gestión administrativa para el Departamento de laboratorio clínico del hospital nacional Santa Teresa, desarrollado como parte de una tesis de la Universidad de San Salvador. (ANGEL and BARON, 2012)

1.5. Fundamentación de los objetivos

La entidad se decidió crear una aplicación web que facilite el trabajo en la Planificación de las actividades que responda a los objetivos de trabajo de las divisiones hasta llegar al plan individual de cada trabajador para mayor rapidez y eliminar los errores.

1.6. Aspectos más significativos de la planificación de actividades

Planificación:

La planificación trata de identificar qué objetivos desea alcanzar la organización y cómo. El proceso de planificación suele ser escrito porque de esta forma es mucho más fácil comunicar los planes y discutirlos sin ambigüedades. Dentro de un plan, se establecen:

- Objetivos: metas concretas que la empresa desea alcanzar (aumentar la rentabilidad de un 5% a un 15%, aumentar la cuota de mercado del 20% al 30%, etc.).
- Acciones concretas que deben permitir a la empresa alcanzar sus objetivos.
- Asignación de recursos: cómo se van a obtener los recursos económicos necesarios para llevar a cabo las acciones.
- Directrices de implementación: reparto de tareas entre los diferentes actores y calendario de ejecución. (Ruz, 2011)

Plan de Actividades:

Un plan de actividades es un documento que recoge un conjunto de tareas necesarias para la consecución de una acción u objetivo concreto.

Plan anual de actividades:

En el Plan Anual de Actividades se recogen las principales tareas contenidas en los planes del Partido Y de las Asambleas del Poder Popular (Nacional, Provincial y Municipal) que tienen incidencia en el Gobierno; las que aseguran: el cumplimiento de los acuerdos y resoluciones del VI Congreso del Partido Comunista de Cuba y los objetivos de trabajo en cumplimiento de las funciones estatales, ejecutivas y administrativas de cada nivel de dirección, tales como visitas de trabajo, actividades de preparación, productivas y de servicios, cursos, eventos, salidas al exterior, temas de los consejos de dirección o de la administración, reuniones, despachos, entre otros. (Ruz, 2011)

Plan mensual de actividades:

El Plan de Trabajo Mensual se elabora sobre la base de lo aprobado en el Plan Anual de Actividades de cada nivel de dirección, en él se puntualizan las que hayan sufrido cambios y las nuevas, como resultado del proceso de dirección. Se aprueba por el jefe del nivel de dirección que elabora y se archiva por el término de un año. (Ruz, 2011)

Plan individual de actividades:

El Plan de desarrollo individual es aquel que operacionaliza la aceptación racional, el compromiso social y la toma de conciencia del trabajador en la empresa. contiene lo que va a hacer la persona, por qué lo va a hacer, cuándo se hará y cómo se van a medir los resultados.

En el Plan de Trabajo Individual, se recogen todas las tareas que cumplen los jefes, cuadros o especialistas para el periodo de un mes (Anexo No. 2) y la base para su elaboración constituye el Plan Anual de Actividades, el Plan de Trabajo Mensual de la organización y las indicaciones dadas por los jefes facultados. El Plan de Trabajo Individual es revisado y aprobado por el jefe inmediato superior y el análisis de su cumplimiento debe tenerse en cuenta para la evaluación del cuadro o funcionario. Para su elaboración se podrá utilizar las aplicaciones informáticas existentes, sobre la base de la disponibilidad de cada organización. Se archiva por el término de un año. (Ruz, 2011)

1.7. Tendencias y Tecnologías Actuales. Fundamentación de la Metodología utilizada

El Proceso Unificado de Desarrollo (RUP)

Es una metodología de desarrollo de software que está basado en componentes e interfaces bien definidas, y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Es un proceso que puede especializarse para una gran variedad de sistemas de software, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. (Jacobson et al., 1990)

RUP: ciclo de vida

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo). (Gordon et al., 2007)

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML. (Jacobson et al., 1990, Gordon et al., 2007)

Iterativo e Incremental: Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. (Booch et al., 1999)

1.8. Fundamentación del Entorno de Desarrollo, Lenguaje, Gestor de Base de Datos y Tecnología utilizados

En este epígrafe se mostrarán las diferentes tecnologías que serán utilizadas en el desarrollo de la aplicación.

1.8.1. Servidor utilizado: Node.js

Es una plataforma construida o basada en la máquina virtual de java script del navegador Chrome, lo que nos permite construir aplicaciones de red escalables y muy rápidas, usa el modelo de eventos que usa Chrome, un modelo no bloqueante y un modelo de entradas y salidas que lo hace ligero y eficiente, perfeccionando y haciendo énfasis especial con todo lo que tiene que ver con aplicaciones en tiempo real y que tenga que estar consumiéndose en dispositivos distribuidos. Leguaje de java script para la realización de chats. (Monzón, 1999)

1.8.2. Entorno de Desarrollo: WebStorm

WebStorm es un IDE para JavaScript que facilita el desarrollo en dicho lenguaje. Para cada objeto muestra sus posibles métodos y propiedades (objetos predefinidos del lenguaje). También tiene un editor HTML que muestra el CSS asociado a cada etiqueta de la página que se está editando. Es compatible con una amplia gama de tecnologías modernas relacionadas con el lenguaje de programación JavaScript, HTML y CCS, además ofrece la experiencia completa para el desarrollo Web productivo. Proporciona soporte de primera clase para JavaScript, Node.js, HTML y CSS, así como sus sucesores modernos.(Talekar, 2008)

1.8.3. Lenguajes utilizados

Un lenguaje de programación debe resultar correcto, es decir, determinar qué debe hacer antes de desarrollarlo y compararlo, una vez acabado, con lo que realmente hace. Debe ser lo más claro, conciso y legible posible, con una estructura sencilla y entendible. Eficaz, que sea capaz de gestionar de la mejor manera posible los recursos que utiliza. Portable, con capacidad para ser ejecutado en una plataforma diferente a aquella en la que se elaboró inicialmente ya sea hardware o software. Con todo ello conseguiremos llegar a un mayor número de usuarios más

fácilmente. El lenguaje de programación es la base del éxito de un buen programa. (Joyanes Aguilar, 2003)

1.8.3.1. Java Script

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems.(Pérez, 2007)

1.8.3.2. HTML

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la

página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.(Robbins, 2013)

1.8.4. Gestor de Base de Datos: PostgreSQL

PostgreSQL es gratuito y libre además de que hoy nos ofrece una gran cantidad de opciones avanzadas. Es considerado el motor de base de datos más avanzado en la actualidad. Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez a el sistema. En Postgres no se requiere usar bloqueos de lectura al realizar una transacción lo que nos brinda una mayor escalabilidad. También PostgreSQL tiene Hot-Standby. Este permite que los clientes hagan búsquedas (sólo de lectura) en los servidores mientras están en modo de recuperación o espera. Así podemos hacer tareas de mantenimiento o recuperación sin bloquear completamente el sistema. Como ventaja es que está disponible para muchas plataformas y ofrece el código fuente desde el sitio oficial.(Momjian, 2001)

1.8.5. Framework utilizados: AngularJs

AngularJS es un proyecto de código abierto, realizado en Javascript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso sobre el lenguaje Javascript con programación del lado del cliente. Lógicamente, AngularJS viene cargado con todas las herramientas que los creadores ofrecen para que los desarrolladores sean capaces de crear ese HTML enriquecido. La palabra clave que permite ese HTML declarativo en AngularJS es "directiva", que no es otra cosa que código Javascript que mejora el HTML. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles. se puede combinar con el entorno en tiempo de ejecución Node.js, el framework para servidor Express.js y la base de datos MongoDB para formar el conjunto MEAN(Dorta, 2014 - 2016)

1.8.5.1. Embarcadero ER/Studio

Embarcadero ER/Studio es una aplicación para modelar visual usada para el análisis de arquitectura de datos lógico plataforma-independiente y diseño, además del plan de la base de datos físico plataforma-específico y construcción. Su ambiente del plan poderoso, multi-nivelado se dirige las necesidades cotidianas de administradores de la base de datos, diseñadores, y arquitectos de los datos que construyen y mantienen la base de datos grande, las aplicaciones de la base de datos complejas y se esfuerza por consolidar, informar, y reuse el metadata por la empresa.(Belletti and Motz, 2002)

1.8.5.2. Visual Paradigm

Visual Paradigm para UML(Unified Modeling Language) es una herramienta para desarrollo de aplicaciones utilizando modelado UML. Soporta las últimas versiones del mismo y la Notación para Modelado de Procesos de Negocios (BPMN). Es un producto galardonado que facilita la diagramación visual y el diseño de proyectos, brinda la posibilidad integrar y desplegar aplicaciones y sus bases de datos subyacentes. Este software incorpora UML como lenguaje de modelado que permite modelar, construir y documentar los elementos que integran un sistema software orientado a objetos. Posee una notación gráfica expresiva que permite representar todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, entre otros. Utiliza gráficos y textos de manera que los modelos pueden ser interpretados por personas que no participaron en su diseño.(Rondón et al., 2011)

Conclusiones parciales

En este capítulo se realizó un análisis de los objetivos del departamento de planificación. Se realizó un estudio de la metodología y tecnologías a emplear para dar solución al problema planteado, utilizando: Bootstrap 4 como framework o librería del lado del cliente para el uso tecnologías HTML5, CSS3 y JavaScript; RUP como metodología de desarrollo de software; UML como lenguaje de modelado; NodeJs para lenguaje de programación del lado del servidor y Angular como framework para el desarrollo, en este caso sobre el lenguaje Javascript con programación del lado del cliente y PostgreSQL como gestor de base de datos para la herramienta.

Capítulo 2 Modelo del Negocio y Requisitos

En este capítulo se abordan aspectos con relación al modelado del negocio, en el cual se describe de una forma detallada el proceso de Gestión de la planificación de actividades. Se incluye la descripción de actores del negocio, así como del sistema y sus respectivos diagramas de casos de uso. Además, se exponen los requisitos funcionales y no funcionales, con que debe contar la aplicación.

2.1 Modelo del negocio actual

En la Empresa de Aplicaciones Informáticas DESOFT Sancti Spíritus el proceso de planificación se realiza actualmente haciendo uso de las aplicaciones ofimáticas Excel y Word. Debiendo mensualmente elaborarse los planes de la entidad y de cada trabajador; al finalizar el periodo se debe confeccionar el informe de cumplimiento de estos planes. Este proceso demora al menos dos días y siempre ocurren retrasos por errores humanos. El desarrollo de una aplicación web lograría mayor agilidad en el proceso y se eliminarían algunos errores que se introducen por la forma en la que hoy se desarrolla.

2.2 Reglas del negocio a considerar

Las reglas estimulan (pero no obligan) a considerar las cuestiones más importantes de análisis, diseño e implementación que llevan a tales sistemas a convertirse en bien formados con el paso del tiempo. Las reglas de negocio como una descripción de las operaciones, definiciones y restricciones aplicables a una organización para lograr sus metas. (Leonard et al., 1998)

RN1: El usuario siempre debe autenticarse

RN2: Cada trabajador realiza su plan de trabajo individual

RN3: El director y el planificador son los que realizan el plan anual y el mensual,

RN4: Solo el director aprueba el plan anual

RN5: Todos los trabajadores pueden gestionar las actividades asociadas a los planes que puedan gestionar.

RN6: Se deben confeccionar los informes de cumplimiento de los planes individuales y del mensual.

2.3 Actores del negocio

Un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor. Un candidato a actor del negocio es cualquier individuo, grupo, organización o máquina que interactúa con el negocio. (González, 2005)

En la siguiente tabla se observa el actor del negocio actual:

| Actor del Negocio | Descripción |
|--|---|
| Especialista de control interno | Es quien supervisará los planes anual, mensual e individual |

TABLA 1 ACTOR DEL NEGOCIO

2.4 Trabajadores del negocio

Un trabajador del negocio representa un rol que juega una persona (o grupo de personas), una máquina o un sistema automatizado; actuando en el negocio. Son los que realizan las actividades, interactuando con otros trabajadores del negocio y manipulando entidades. (González, 2005)

En la siguiente tabla se observa los trabajadores del negocio actual:

| Trabajadores del Negocio | Descripción |
|--------------------------|--|
| Directora | Realiza la planificación de actividades anual y aprobar el mismo |

| | |
|--|--|
| Planificadora | Realiza las actividades del plan anual y mensual y genera informe |
| Jefes de área | Jefe del área con permiso de realizar acciones en los planes individuales del trabajador de sus área |
| Trabajadores | Trabajador que realiza su propio plan individual de actividades |
| Especialista de control interno | Trabajador que supervisa los planes |

TABLA 2 TRABAJADORES DEL NEGOCIO

2.5 Diagrama de casos de uso del negocio

Un caso de uso es una descripción de un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un actor particular. Un caso de uso se utiliza para estructurar los aspectos de comportamiento en un modelo. (Bonch and Lukewood, 1998)

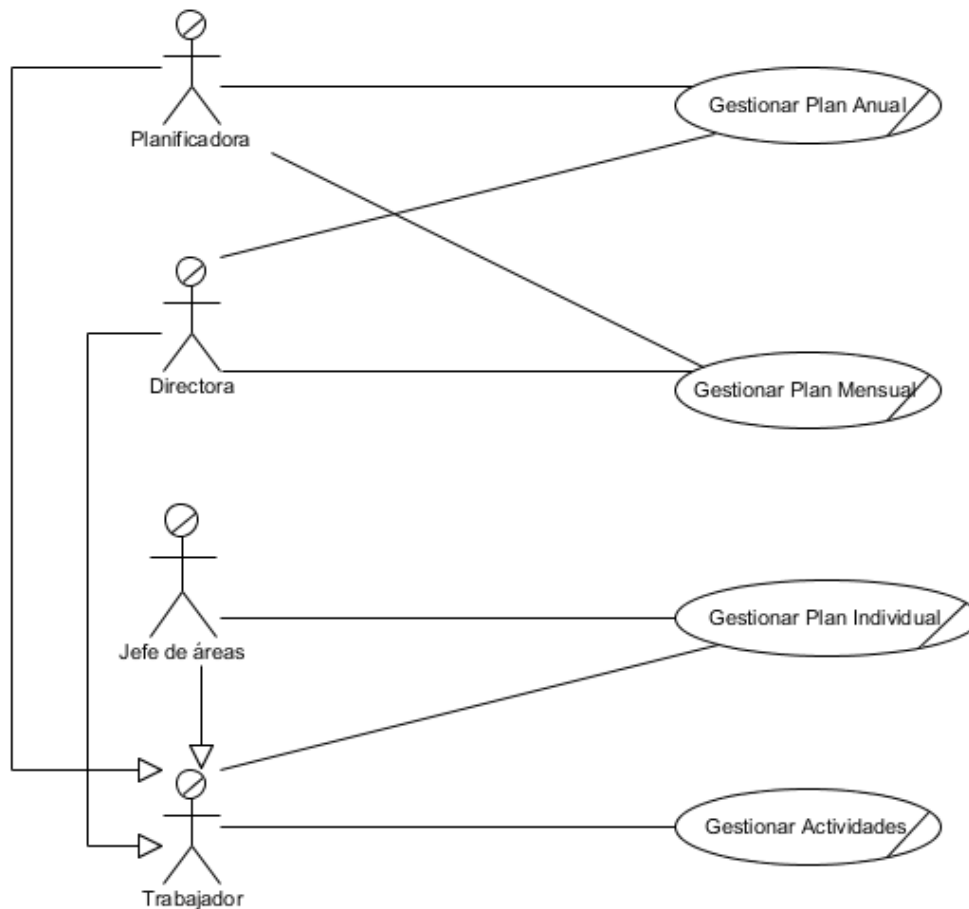


FIGURA 1 CASO DE USO DEL NEGOCIO

2.6 Descripción de los casos de uso del negocio

| Caso de uso del negocio | Descripción |
|-------------------------|---|
| Directora | La directora inicia el caso de uso una vez que son enviadas a la empresa las actividades de la Oficina Central de Desoft y conforma junto con la planificadora el plan anual de actividades y el mensual. |

| | |
|----------------------|---|
| Planificadora | La planificadora inicia el caso de uso una vez que son enviadas a la empresa las actividades de la Oficina Central de Desoft y conforma junto con la planificadora el plan anual de actividades y el mensual. |
| Jefes de área | El jefe de área inicia el caso de uso una vez que fue recibido las actividades del mes para su área específica y conforma las actividades que se le asigna a cada trabajador.. |
| Trabajadores | El trabajador inicia el caso de uso una vez que comienza el mes y conformar su plan individual de actividades. |

TABLA 3 DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO

2.7 Actores del sistema a automatizar

El primer tipo de actor es una persona física, o un usuario. Un actor es cualquiera o algo que actúa recíprocamente con el sistema a construirse. Son algo que está fuera del alcance del sistema(puede ser interior a la organización). (Guide, Aurelia and Ana, 2008)

| Actores del sistema a automatizar | Descripción |
|--|--|
| Directora | Procesa la información planificación de las actividades anual y aprobar el mismo |
| Planificadora | Procesa la información de todas las actividades y genera informe |
| Jefes de área | Procesa la información de los planes individuales del trabajador de su área |

| | |
|---------------------|-------------------------------|
| Trabajadores | Procesa su propia información |
|---------------------|-------------------------------|

TABLA 4 ACTORES DEL SISTEMA A AUTOMATIZAR

2.8 Definición de los requisitos

El análisis de requisitos es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle el ámbito del software y se crean modelos de los requisitos de datos, flujo de información y control, y del comportamiento operativo. Se analizan soluciones alternativas y se asignan a diferentes elementos del software. El análisis de requisitos permite al desarrollador o desarrolladores especificar la función y el rendimiento del software, indicaron otros elementos del sistema la interfaz del software y establece las restricciones que debe cumplir el software. (Eden and Kazman, 2003)

2.9 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que este reaccionará a entradas particulares y de cómo se comportará en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer. (Sommerville, 2002)

RF1: Gestionar usuario

La aplicación debe permitir que se le asigne a cada trabajador uno o más roles para operar en el sistema. Estos roles tendrán asociados determinados privilegios de acceso a las funcionalidades del sistema.

RF1.1: Insertar usuario

El sistema debe permitir incorporar nuevos usuarios al sistema

RF1.2: Actualizar usuario

El sistema debe permitir modificar los datos de los usuarios

RF2: Gestionar nomencladores

RF2.1: Gestionar capítulos

El sistema debe obtener un listado de los capítulos en los que se divide el plan anual de actividades, que están definidos en la Instrucción 1 del Presidente de los Consejos de Estado y de Ministros

RF2.1.1: Insertar capítulos

El sistema debe permitir incorporar nuevos capítulos al sistema

RF2.1.2: Actualizar capítulos

El sistema debe permitir modificar los datos de los capítulos al ser cambiados por el Consejo de Estado

RF2.2: Gestionar subcapítulos

El sistema debe obtener un listado de los subcapítulos dentro de cada capítulo del plan anual.

RF2.2.1: Insertar subcapítulos

El sistema debe permitir incorporar nuevos subcapítulos al sistema

RF2.2.2: Actualizar subcapítulos

El sistema debe permitir modificar los datos de los subcapítulos

RF2.3: Gestionar niveles

El sistema debe obtener un listado de lo cada nivel de dirección que dirige una actividad

RF2.3.1: Insertar niveles

El sistema debe permitir incorporar nuevos niveles al sistema

RF2.3.2: Actualizar niveles

El sistema debe permitir modificar los datos de los niveles

RF2.4: Gestionar estado de cumplimiento

El sistema debe obtener un listado de los estados de cumplimiento que tienen las actividades

RF2.4.1: Insertar estado de cumplimiento

El sistema debe permitir incorporar nuevos estados de cumplimiento al sistema

RF2.4.2: Actualizar estado de cumplimiento

El sistema debe permitir modificar los datos de los estados de cumplimiento

RF3: Gestionar actividades

El sistema debe obtener un listado de todas las actividades que se realizarán

RF3.4.1: Insertar actividades

El sistema debe permitir que se editen los datos de una actividad, una vez que esta se ha seleccionado. Las actividades que se puedan editar estarán en correspondencia con el rol del usuario

RF3.2: Actualizar actividades

El sistema debe permitir modificar los datos de las actividades

RF3.4: Evaluar cumplimiento de las tareas

El sistema debe permitir que cada especialista pueda evaluar el cumplimiento de sus tareas, definiendo el estado de cumplimiento para la actividad

RF4: Gestionar plan anual de la División

El sistema debe permitir crear el modelo del Plan Anual de Actividades, a partir de habilitar cada capítulo y subtítulos que contendrá el plan.

RF4.1: Incorporar actividades al plan

El sistema debe permitir que, cuando se cree el modelo del Plan Anual, se muestren todas las actividades con estado planificada y que sean del tipo anual

RF4.2: Generar PAP

El sistema debe permitir que una vez que se ha elaborado el plan se puedan crear versiones de los planes anuales y se guarden en el sistema las versiones anteriores.

RF5: Gestionar plan mensual de la División

El sistema debe permitir que se seleccione el año y el mes que se desea, y a partir de las tareas que se definieron en el Plan anual de actividades para ese mes, se conforme el modelo del plan mensual de la División

RF5.1: Incorporar actividades al plan

El sistema debe permitir que una vez aprobado el plan mensual de la División, se agreguen actividades con estado nuevas que hayan sido creadas por el planificador, director o jefes de áreas.

RF5.2: Generar plan mensual

El sistema debe permitir que una vez que se ha elaborado el plan mensual de la División se puedan crear versiones de los planes y se guarden en el sistema las versiones anteriores

RF6: Gestionar plan individual de la División

El sistema debe permitir que se seleccione el año y el mes que se desea, y a partir de las tareas que se definieron en el Plan mensual de la División y el carnet de identidad del trabajador, se conforme el modelo del plan individual del trabajador

RF6.1: Incorporar actividades al plan

El sistema debe permitir que una vez aprobado el plan individual por su jefe inmediato superior, se pueden agregar actividades con estado nuevas que hayan sido creadas por el jefe o por el propio trabajador. .

RF6.2: Generar plan individual

El sistema debe permitir que una vez que se ha elaborado el plan individual se puedan crear versiones del plan y se guarden en el sistema las versiones anteriores

2.10 Definición de los requisitos no funcionales

Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo, estándares, etcétera (Sommerville, 2002).

RNF1: Requisitos de interfaz

RNF1.1: El sistema debe poseer una interfaz amigable, así como menús despleables que permitan interactuar y navegar con el sistema de forma fácil y rápida, sin necesidad de conocimientos avanzados o medios de computación.

RNF2: Requisitos de software: Los requisitos software son la descripción de las características y las funcionalidades del sistema

RNF2.1: El sistema debe contar con un lenguaje de programación en java script

RNF2.2: El sistema debe utilizar como gestor de base de datos PostgreSQL

RNF2.3: Se utilizará como servidor de aplicaciones, node.js

RNF3: Requisitos de seguridad: Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.

RNF3.1: El sistema debe garantizar que la información sea editada únicamente por las personas que tienen permisos para realizar esta actividad.

RNF4: Requisitos de usabilidad: La usabilidad es el grado en el cual un producto puede ser utilizado por sus usuarios para lograr metas con efectividad, eficiencia y satisfacción en un determinado contexto.

RNF4.1: El sistema será de fácil acceso y uso para cualquier usuario que acceda a él con conocimiento básicos de computación.

RNF5: Requisitos de disponibilidad: Garantizar un esquema adecuado que permita la posible falla en cualquiera de sus componentes.

RNF5.1: Debe estar disponible las 24 horas del día.

RNF5.2: La interacción con la aplicación y la base de datos debe ser lo más rápido posible para mayor respuesta del usuario.

2.11 Diagrama de Casos de Uso del Sistema

Los diagramas de Caso de Uso muestran las interacciones entre los casos del uso y actores de un sistema a automatizar, así como las funcionalidades del mismo. (Wendy Boggs, 2002)

Los casos de Uso del sistema son:

CU1: Gestionar Usuario.

CU2: Gestionar Nomencladores.

CU3: Gestionar Plan Anual

CU4: Gestionar Plan Mensual.

CU5: Gestionar Plan Individual.

CU6: Gestionar Actividades.

CU7: Exportar a PDF.

En la siguiente figura se muestra el diagrama de Casos de Uso del sistema:

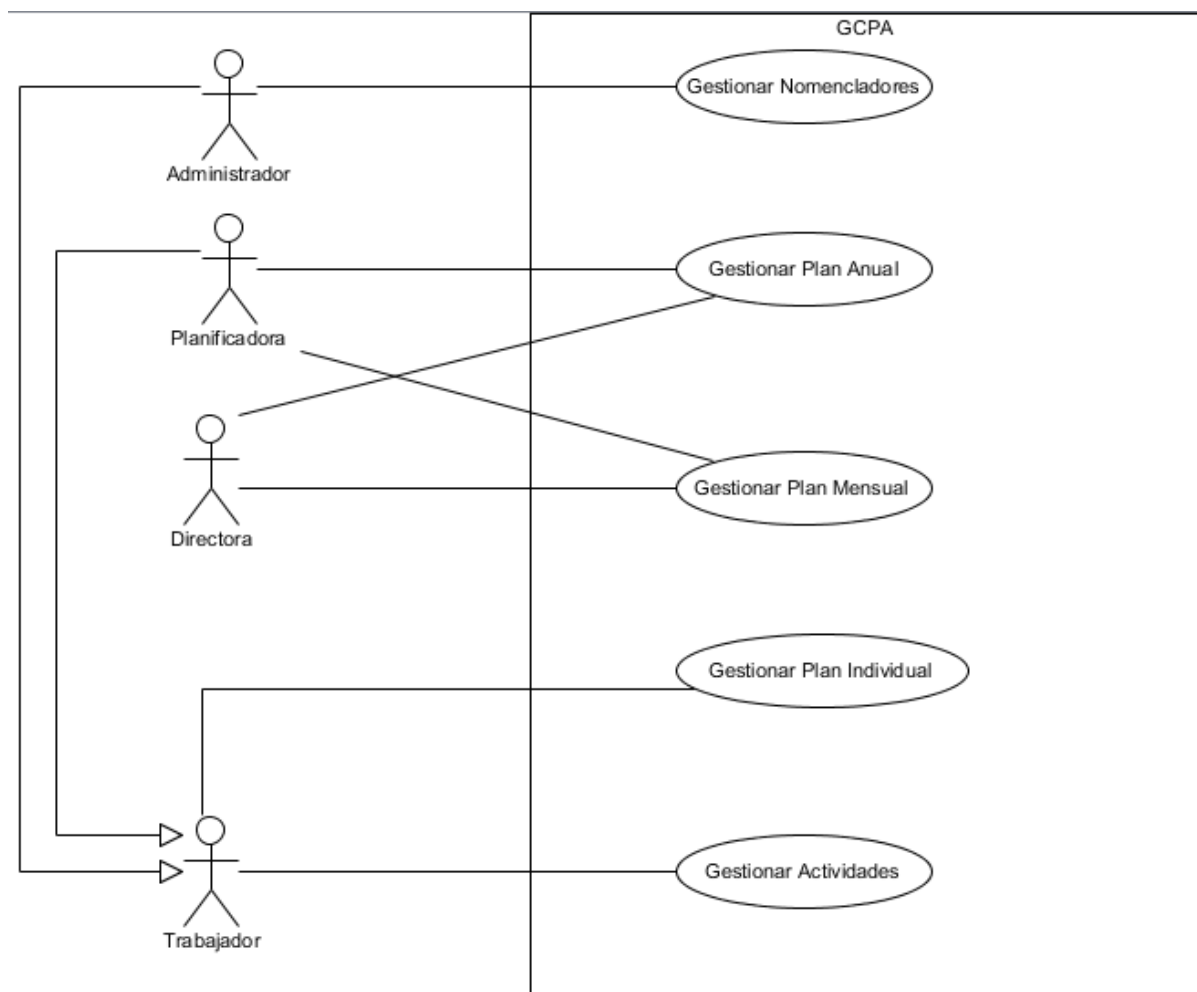


FIGURA 2 DIAGRAMA DE CASO DE USO DEL SISTEMA

2.11.1 Descripción de los casos de uso del Sistema

Caso de uso: Gestionar Nomencladores-Capítulos: El administrador gestiona cada nomenclador que será usado por el sistema en la entidad. Incluye nuevas inserciones de nomencladores, así como actualizar los que ya existen.

| Caso de uso del sistema | Gestionar Nomenclador-Capítulos |
|-------------------------|--|
| Actor | Administrador |
| Propósito | Gestionar el nomenclador capítulo que se utilizará en las actividades del plan anual |

| | |
|------------------------|--|
| Resumen | Inicia cuando el administrador selecciona la opción Gestionar nomencladores y dentro la opción gestionar capítulos, luego pude insertar, actualizar un capítulo y finaliza cuando el administrador termina con las operaciones |
| Responsabilidades | Gestionar el capítulo que se utilizará en el plan anual de actividades |
| Casos de uso asociados | No tiene |
| Precondiciones | El actor que inicia el caso de uso debe estar autenticado en el sistema. |

The screenshot shows a web application interface for managing chapters. On the left is a sidebar with the 'Financiero' module selected and options like 'Gestionar Capítulos'. The main area is titled 'Insertar' and contains a 'Listado de capítulos' table and a 'Formulario de Insertar Capítulo'. The table lists existing chapters with their names, states (all 'activo'), and edit actions. The form allows adding new chapters with a 'Nombre' field, an 'Activo' checkbox, and 'Guardar'/'Cancelar' buttons.

Flujo normal de eventos. Sección A: Nuevo Capítulo

| Acción del actor | Respuesta del Sistema |
|---|---|
| 1.El administrador selecciona la opción de insertar un nuevo capítulo. 3.El administrador llena los campos y acepta. | 2.El sistema muestra un formulario con los datos necesarios para insertar un capítulo. 4.El sistema introduce los datos en la base de datos. |
| Flujos alternativos 1 | |

| | |
|--|--|
| 3.El administrador llena todos los campos y acepta. | 4.El sistema muestra un mensaje de error donde indica que existen campos vacíos. |
| 5.El administrador verifica todos los campos y acepta. | |
| Flujos alternativos 2 | |
| 3.El administrador llena los campos y acepta. | 4.El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos. |
| 5.El administrador verifica todos los campos y acepta. | |
| Flujo normal de los eventos. Sección B: Modificar capítulo | |
| Acción del actor | Respuesta del Sistema |
| 1. El administrador hace clic en el ícono Editar capítulo | |
| 3.El administrador edita los campos del capítulo deseado y acepta. | 2.El sistema carga el capítulo y muestra sus datos en un formulario. 5.El sistema actualiza los datos del capítulo seleccionado en la base de datos |
| Flujos alternativos 1 | |
| 1.El administrador hace clic en el botón Editar Capítulo | 2.El sistema muestra un mensaje de error donde indica que no se ha seleccionado ningún capítulo para actualizar. |
| Flujos alternativos 2 | |
| 3.El administrador edita los campos del capítulo que desee y acepta. | 4.El sistema muestra un mensaje de error donde indica que algún campo ha quedado vacío |

| | |
|--|---|
| 5.El administrador verifica que los campos estén llenos y acepta | |
| Flujos alternativos 3 | |
| 3.El administrador edita los campos del capítulo que desee y acepta. 5.El administrador verifica que los campos estén correctos y acepta. | 4. El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos. |

TABLA 5 DESCRIPCIÓN DEL CASO DE USO: GESTIONAR NOMENCLADORES-CAPÍTULOS

Caso de uso: Gestionar actividades: La planificadora y directora gestiona las actividades para el plan anual y mensual de la División. Incluye nuevas inserciones, modificaciones, así como eliminaciones de las mismas.

| Caso de uso del sistema | Gestionar Actividades |
|--------------------------------|---|
| Actor | Planificadora y Directora |
| Propósito | Gestionar las actividades que serán asignadas en la empresa y a los trabajadores |
| Resumen | Inicia cuando el planificador selecciona la opción Gestionar actividades en el plan que se encuentra, luego puede insertar, actualizar o eliminar una actividad y finaliza cuando el planificador termina con las operaciones |
| Responsabilidades | Gestionar las actividades |
| Casos de uso asociados | No tiene |
| Precondiciones | Conexión a la base de datos |

Flujo normal de eventos. Sección A: Nueva Actividad

| Acción del actor | Respuesta del Sistema |
|--|--|
| <p>1.La planificadora o directora selecciona la opción de insertar una nueva actividad.</p> <p>3 La planificadora o directora llena los campos y acepta.</p> | <p>2.El sistema muestra un formulario con los datos necesarios para insertar una actividad.</p> <p>4.El sistema introduce los datos en la base de datos.</p> |
| Flujos alternativos 1 | |
| <p>3. La planificadora o directora llena todos los campos y acepta.</p> <p>5. La planificadora o directora verifica todos los campos y acepta.</p> | <p>4.El sistema muestra un mensaje de error donde indica que existen campos vacíos.</p> |
| Flujos alternativos 2 | |
| <p>3. La planificadora o directora llena los campos y acepta.</p> <p>5. La planificadora o directora verifica todos los campos y acepta.</p> | <p>4.El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos.</p> |

| Flujo normal de los eventos. Sección B: Modificar actividad | |
|---|--|
| Acción del actor | Respuesta del Sistema |
| 1. La planificadora o directora hace clic en el ícono editar actividad | 2.El sistema carga la actividad y muestra sus datos en un formulario. 4.El sistema actualiza los datos de la actividad seleccionada en la base de datos |
| 3. La planificadora o directora edita los campos de la actividad deseada y acepta. | |
| Flujos alternativos 1 | |
| 3.La planificadora o directora edita los campos de la actividad que desea y acepta. | 4.El sistema muestra un mensaje de error donde indica que algún campo ha quedado vacío |
| 5.La planificadora o directora verifica que los campos estén llenos y acepta | |
| Flujos alternativos 2 | |
| 3.La planificadora o directora edita los campos de la actividad que desea y acepta. | 4. El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos. |
| 5. La planificadora o directora verifica que los campos estén correctos y acepta. | |

TABLA 6 DESCRIPCIÓN DEL CASO DE USO: GESTIONAR ACTIVIDADES

Gestionar Plan Anual: La planificadora y directora realizan los planes anual t mensual, para ello incluye la inserción de los mismos.

| Caso de uso del sistema | Gestionar Plan Anual |
|--------------------------------|--|
| Actor | Planificadora y directora |
| Propósito | Gestionar el plan anual de la división |

| | |
|---|---|
| Resumen | Inicia cuando la planificadora o directora seleccionan la opción Plan anual, luego puede insertar, actualizar las actividades del plan y finaliza cuando la planificadora o directora termina con las operaciones |
| Responsabilidades | Gestionar el plan anual |
| Casos de uso asociados | No tiene |
| Precondiciones | Conexión a la base de datos |
| Flujo normal de eventos. Sección A: Nuevo plan anual | |
| Acción del actor | Respuesta del Sistema |
| 1.La planificadora o directora seleccionan la opción de insertar un nuevo plan 3. La planificadora o directora llenan los campos y acepta. | 2.El sistema muestra un formulario con los datos necesarios para insertar un plan. 4.El sistema introduce los datos en la base de datos. |
| Flujos alternativos 1 | |
| 3. La planificadora o directora llenan todos los campos y acepta. 5 La planificadora o directora verifican todos los campos y acepta. | 4.El sistema muestra un mensaje de error donde indica que existen campos vacíos. |
| Flujos alternativos 2 | |
| 3. La planificadora o directora llenan los campos y acepta. 5. La planificadora o directora verifican todos los campos y acepta. | 4.El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos. |
| Flujo normal de los eventos. Sección B:Actualizar estado del plan anual | |
| Acción del actor | Respuesta del Sistema |
| 1. La planificadora o directora hacen clic en el icono actualizar estado | 2.El sistema realiza la acción y cambia el estado. |

| Flujo normal de los eventos. Sección B: Crear versión del plan anual | |
|---|--|
| Flujos alternativos 1 | |
| 1. La planificadora o directora hacen clic en el icono crear versión. | 3. El sistema muestra el nuevo plan con las actividades del plan anterior. |

TABLA 7 DESCRIPCIÓN DEL CASO DE USO: GESTIONAR PLAN ANUAL

2.12 Estimación por Puntos de caso de uso

La estimación por puntos de caso de usos consiste en un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de “pesos” a un cierto número de factores que lo afectan, para que luego a partir de esos factores se pueda contabilizar el tiempo total estimado para el proyecto en cuestión. (Peralta, 2004)

1. Clasificación de los actores

Se tienen 1 actores de tipo complejo con un peso de 3

$$UAW=1*3=3$$

2. Clasificación de los casos de uso

Se tienen 25 casos de uso de tipo simple con un peso de 5

$$UUCW=25*5=125$$

3. Calcular puntos de casos de uso sin ajustar

$$UUCP=UAW+UUCW$$

$$UUCP=3+125=128$$

Calcular el Factor de Complejidad Técnica (FCT)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente se muestran los factores tenidos en cuenta y el peso de cada uno de estos:

| Factor | Descripción | Peso | Influencia | Resultado |
|--------|---------------------|------|------------|-----------|
| F1 | Sistema Distribuido | 2 | 0 | 0 |

| | | | | |
|------------|--|-----|---|---|
| F2 | Tiempo de Respuesta y Desempeño | 1 | 1 | 1 |
| F3 | Eficiencia respecto al usuario final | 1 | 2 | 3 |
| F4 | Procesamiento interno complejo | 1 | 1 | 2 |
| F5 | Código reutilizable en otras aplicaciones | 3 | 1 | 2 |
| F6 | Facilidad en la instalación | 0,5 | 2 | 1 |
| F7 | Usabilidad (Fácil de usar) | 0,5 | 1 | 1 |
| F8 | Portabilidad | 2 | 0 | 0 |
| F9 | Facilidad en mantener | 1 | 1 | 1 |
| F10 | Accesos simultáneos (concurrentes) | 1 | 1 | 1 |
| F11 | Incluye objetivos especiales de seguridad | 0.5 | 1 | 1 |
| F12 | Provee acceso directo a terceros | 1 | 0 | 0 |
| F13 | Se requiere facilidades especiales de entrenamiento a usuarios | 1 | 1 | 1 |

TABLA 8 FACTOR DE COMPLEJIDAD

- ✓ **TCP:** Factor de complejidad técnica.
- ✓ **E:** esfuerzo estimado en horas-hombre.

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i + \text{Valor Asignado}_i)$$

$$TCF = 0.6 + 0.01 * 13$$

$$TCF = 0.73$$

1. Determinar los factores de ambiente o entorno

| Factor | Descripción | Peso | Influencia | Resultado |
|---------------|--|-------------|-------------------|------------------|
| E1 | Familiarizado con el proceso de desarrollo (RUP) | 1 | 3 | 3 |
| E2 | Experiencia en la aplicación | 1 | 4 | 4 |

| | | | | |
|-----------|--------------------------------------|----|---|---|
| E3 | Experiencia en orientación a objetos | 1 | 3 | 3 |
| E4 | Capacidades de análisis | 1 | 4 | 4 |
| E5 | Motivación | 1 | 5 | 5 |
| E6 | Requisitos estables | 1 | 5 | 5 |
| E7 | Trabajadores a tiempo parcial | -1 | 0 | 2 |
| E8 | Lenguaje complejo | 1 | 4 | 4 |

TABLA 9 FACTORES DE AMBIENTE O ENTORNO

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i + \text{Valor Asignado}_i)$$

$$EF = 1.4 - 0.03 * 30$$

$$EF = 0.5$$

2. Calcular los puntos de casos de uso ajustados

$$PCU = UUCP * TCF * EF$$

$$PCU = 128 * 0.73 * 0.5$$

$$PCU = 46.72$$

3. Calcular el esfuerzo

$$E = PCU * CF$$

$$E = 46.72 * 13$$

$$E = 607.4 \text{ h/h}$$

$$\frac{607.4}{X} = \frac{70}{100}$$

$$X = 867$$

El esfuerzo total es de 867 h/hombres

Se trabaja de lunes a viernes 8 horas al día.

$$8 * 5 = 40 \text{ horas}$$

$$867 / 40 = 21.6$$

El tiempo de duración es de aproximadamente 22 semanas con una sola persona en el proyecto.

Costo:

Se ha tomado como referencia el salario por hora del ingeniero informático de la empresa 3 pesos por hora.

$$607.4 * 3.65 = 2217.01$$

El costo del proyecto es de aproximadamente 2217 pesos.

Conclusiones parciales

En este capítulo se abordó acerca del modelado del negocio, presentando los diagramas de caso de uso, la descripción de los casos de uso del sistema, se trataron los diferentes requisitos que presentaría la aplicación, así como se estimó el tiempo por casos de uso del sistema.

Capítulo 3 Descripción de la propuesta de solución

En este capítulo se describen los principales artefactos utilizados para el diseño e implementación del software para realizar la Gestión de la planificación de actividades. Se describe la arquitectura del sistema y se definen los diagramas de clases, secuencia de los casos de uso significativos, así como el diagrama de despliegue y el modelo conceptual de la Base de Datos.

3.1 Arquitectura del sistema

La arquitectura del software define el sistema en términos de sus componentes computacionales y de las relaciones entre ellos. Estructura o estructuras del sistema que comprende componentes de software, propiedades visibles de esos componentes y las relaciones entre ellos.(Sommerville, 2005)

Arquitectura API REST:

Es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP, es una arquitectura cliente-servidor.

API: Es una forma de describir la forma en que los programas o los sitios webs intercambian datos. El formato de intercambio de datos normalmente es **JSON** o XML.

REST: es un estilo de arquitectura para diseñar aplicaciones en red. Una API podría considerarse REST si su arquitectura se ajusta a ciertas reglas o restricciones.(Zulian, 2011)

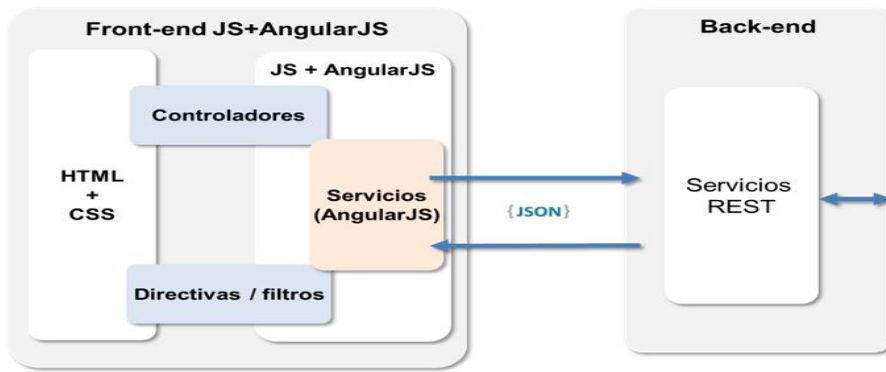


FIGURA 3 ARQUITECTURA DEL SISTEMA (API REST)

Arquitectura MVC

En el MVC, (Modelo-Vista-Controlador) es una patrón de arquitectura de las aplicaciones de software. Separa la lógica del negocio de la interfaz de usuario. Era originalmente aplicado en el usuario gráfico modelo de la interacción de entrada, proceso y rendimiento. (Web, 2015)

Modelo: Un modelo representa los datos de una aplicación y contiene la lógica por acceder y manipular esos datos. Cualquier dato que es parte del estado persistente de la aplicación debe residir en los objetos del modelo

Controlador: El controlador es responsable para interceptar y traducir la entrada del usuario en las acciones a ser realizadas por el modelo. El controlador es responsable para seleccionar la próxima vista basada en el usuario, la entrada y el resultado de funcionamientos ejemplares.

Vista: La vista es responsable para dar el estado del modelo. La semántica de la presentación es encapsulada dentro de la vista, por consiguiente, pueden adaptarse los datos ejemplares para varios tipos diferentes de los clientes. (Gulzar, 2002)

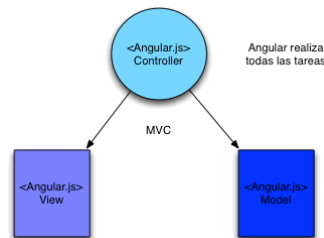


FIGURA 4 ARQUITECTURA DEL SISTEMA (MVC)

3.2 Diagrama de clases de diseño

Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos. Los diagramas de clases que cubren la vista de diseño estática de un sistema. Los diagramas de clases que incluyen clases activas cubren la vista de procesos estática de un sistema. (Bonch and Lukewood, 1998)

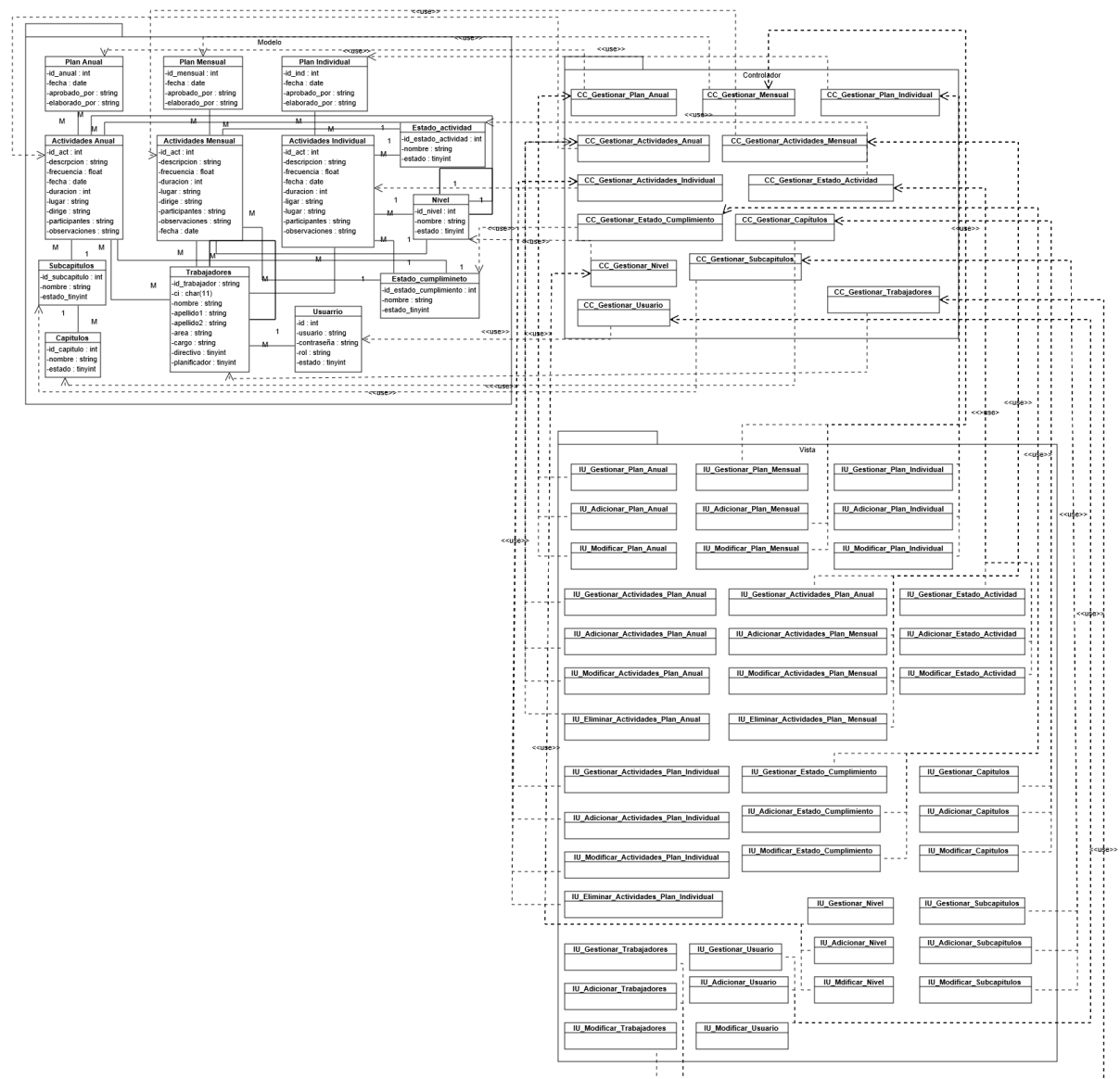


FIGURA 5 DIAGRAMA DE CLASES DEL DISEÑO

En las siguientes figuras se muestran las vistas de los diagramas de clases de diseño para los casos de usos .

Gestionar Actividades plan anual:

Se observa que la clase controladora construye la vista del listado de actividades del plan anual desde esta vista se pueden hacer link hacia las vistas de adicionar y modificar actividades, la vista está compuesta por el formulario de datos el cuál envía su información hacia la controladora. Es la controladora la encargada de interactuar con la entidad de la base de datos.

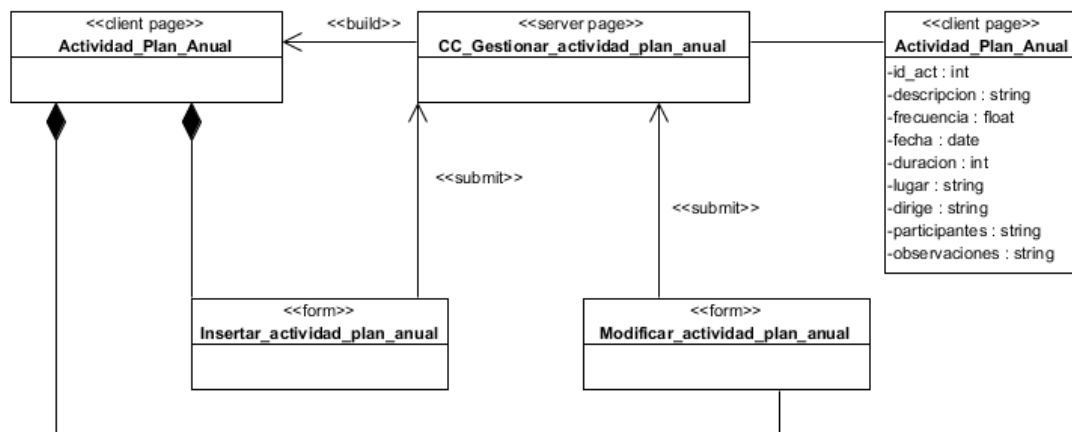


FIGURA 6 DIAGRAMA DE CLASES, CASO DE USO: GESTIONAR ACTIVIDADES PLAN ANUAL

Gestionar Plan Anual:

Se observa que la clase controladora construye la vista del listado de plan anual, desde esta vista se pueden hacer link hacia las vistas de adicionar, la vista está compuesta por el formulario de datos el cuál envía su información hacia la controladora. Es la controladora la encargada de interactuar con la entidad de la base de datos.

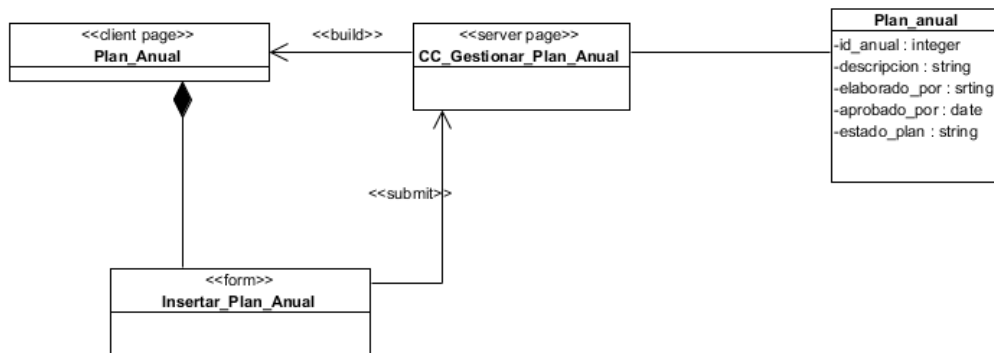


FIGURA 7 DIAGRAMA DE CLASES, CASO DE USO: GESTIONAR PLAN ANUAL

3.3 Diagrama de secuencia

Un diagrama de secuencia es un diagrama de interacción que resalta la ordenación temporal de los mensajes. Un diagrama de secuencia presenta un conjunto de objetos y los mensajes enviados y recibidos por ellos. Los objetos suelen ser instancias con nombre o anónimas de clases, pero también pueden representar instancias de otros elementos, tales como colaboraciones, componentes y nodos. Los diagramas de secuencia se utilizan para describir la vista dinámica de un sistema.(Zapata and Olaya, 2000)

En las siguientes figuras se muestran las vistas de los diagramas de secuencia para los casos de usos

Adicionar plan anual:

En este CU el actor Planificadora es quien lo inicia cuando selecciona la opción Plan Anual y hace link hacia la controladora. Luego la controladora construye la vista con el listado de los planes y es ahí donde el actor puede seleccionar la opción adicionar y con ello un link hacia la vista de adicionar que está compuesta por el formulario de datos, es ahí donde el actor introduce los datos que serán enviados a la controladora. Es en la controladora donde están programados los métodos para adicionar los datos a la base de datos

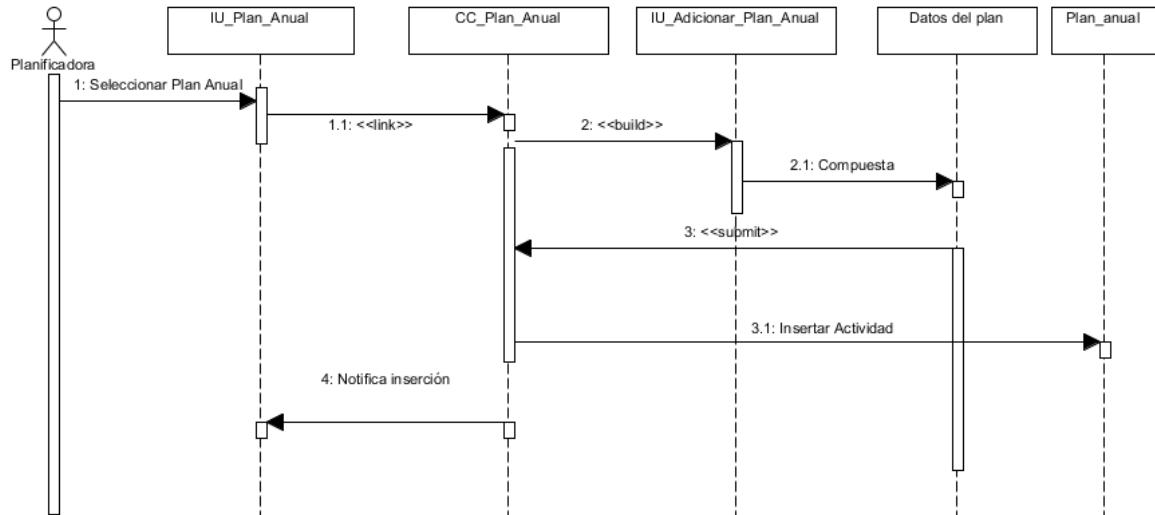


FIGURA 8 DIAGRAMA DE SECUENCIA CASO DE USO :INSERTAR PLAN ANUAL

Adicionar Actividad Plan Anual:

En este CU el actor Planificadora es quien lo inicia cuando selecciona la opción Gestionar Tares en la pantalla de plan anual y hace link hacia la controladora. Luego la controladora construye la vista con el listado de las actividades y es ahí donde el actor puede seleccionar la opción adicionar y con ello un link hacia la vista de adicionar que está compuesta por el formulario de datos, es ahí donde el actor introduce los datos que serán enviados a la controladora. Es en la controladora donde están programados los métodos para adicionar los datos a la base de datos

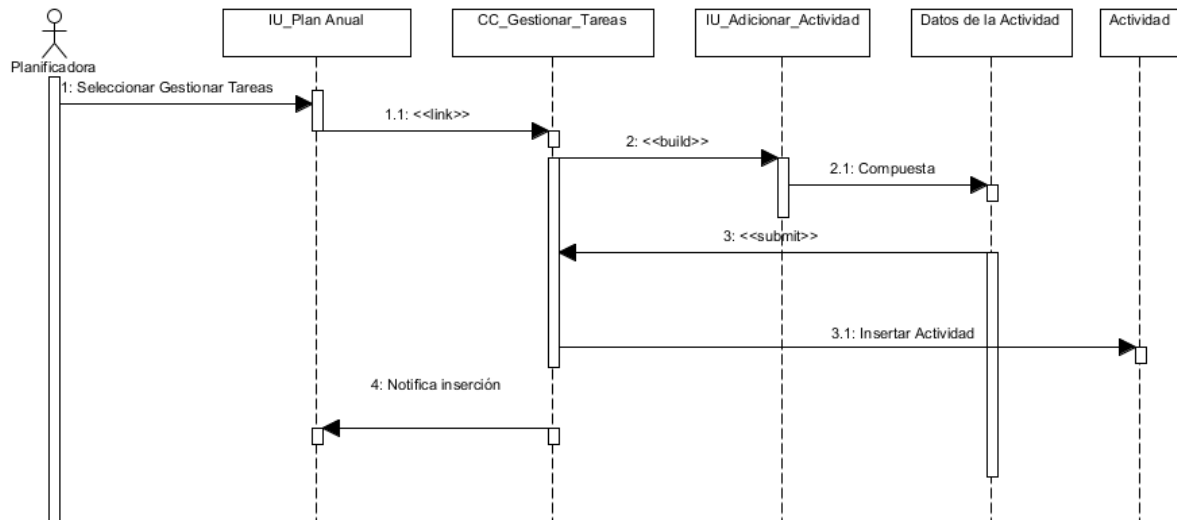


FIGURA 9 DIAGRAMA DE SECUENCIA CASO DE USO: ADICIONAR ACTIVIDAD PLAN ANUAL

3.4 Diseño de la Base de Datos

Un modelo de datos es básicamente una “descripción” de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.(Co-Requisitos and Aprobó, Martínez, 2002)

3.4.1 Modelo Conceptual de datos

Un modelo conceptual no es intrínsecamente correcto ni erróneo, sino que ofrece una utilidad variable. En consecuencia, contiene decisiones que pueden ponerse en tela de juicio o modificarse por buenas razones, sobre todo al momento de elegir las asociaciones. Pero nos da una buena descripción de los conceptos y de sus relaciones importantes, lo bastante útil para sentar las bases de investigaciones posteriores y del trabajo en la fase de solución. Los estudios subsecuentes seguramente detectarán deficiencias y omisiones en el modelo actual, cosa por lo demás previsible. El modelo se actualizará conforme se vayan obteniendo mejoras. (Larman, 2003, Larman)

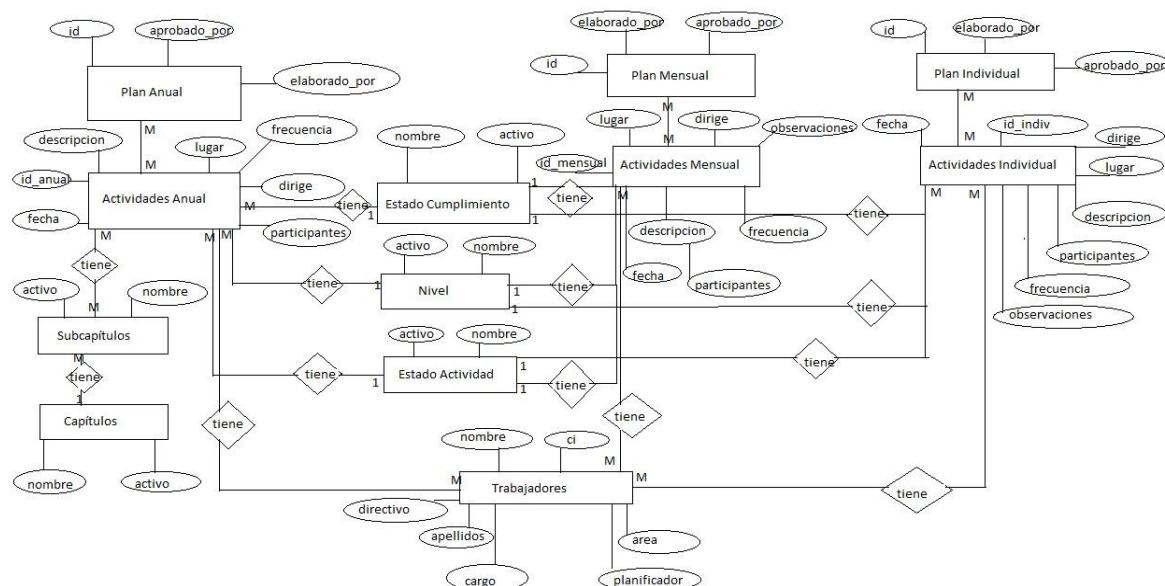


FIGURA 10 MODELO CONCEPTUAL DE BASE DE DATOS

3.4.2 Modelo Lógico de Base de datos

Es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD (SISTEMA DE GESTION DE BASE DE DATOS). Un modelo lógico es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red, etc.). El diseño lógico depende del tipo de SGBD que se vaya a utilizar, no depende del producto concreto. (González Rei and García Mateo, 2000)

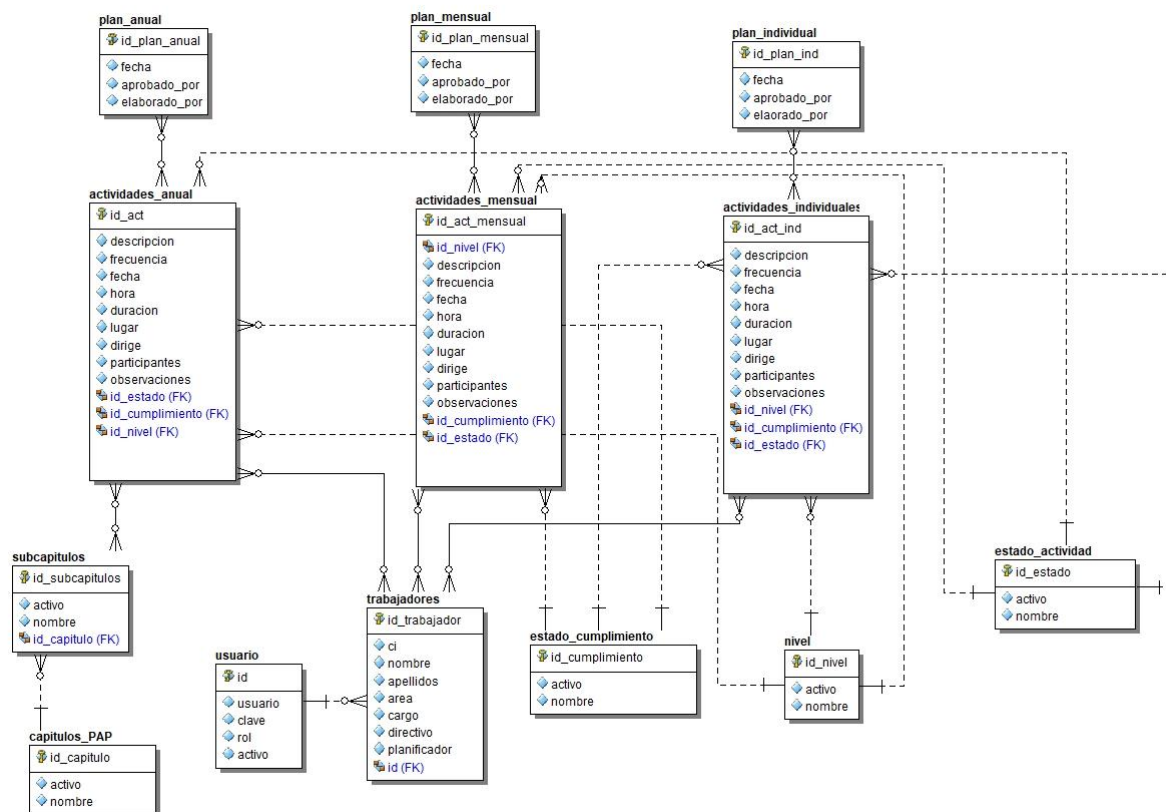


FIGURA 11 MODELO LÓGICO DE BASE DE DATOS

3.5 Diagrama de despliegue

Un diagrama de despliegue muestra un conjunto de nodos y sus relaciones. Los diagramas de despliegue se utilizan para describir la vista de despliegue estática de una arquitectura. Estos diagramas se relacionan con los diagramas de componentes en que de un modo u otro normalmente incluye uno o más componentes. (Bonch and Lukewood, 1998, Grady Booch)

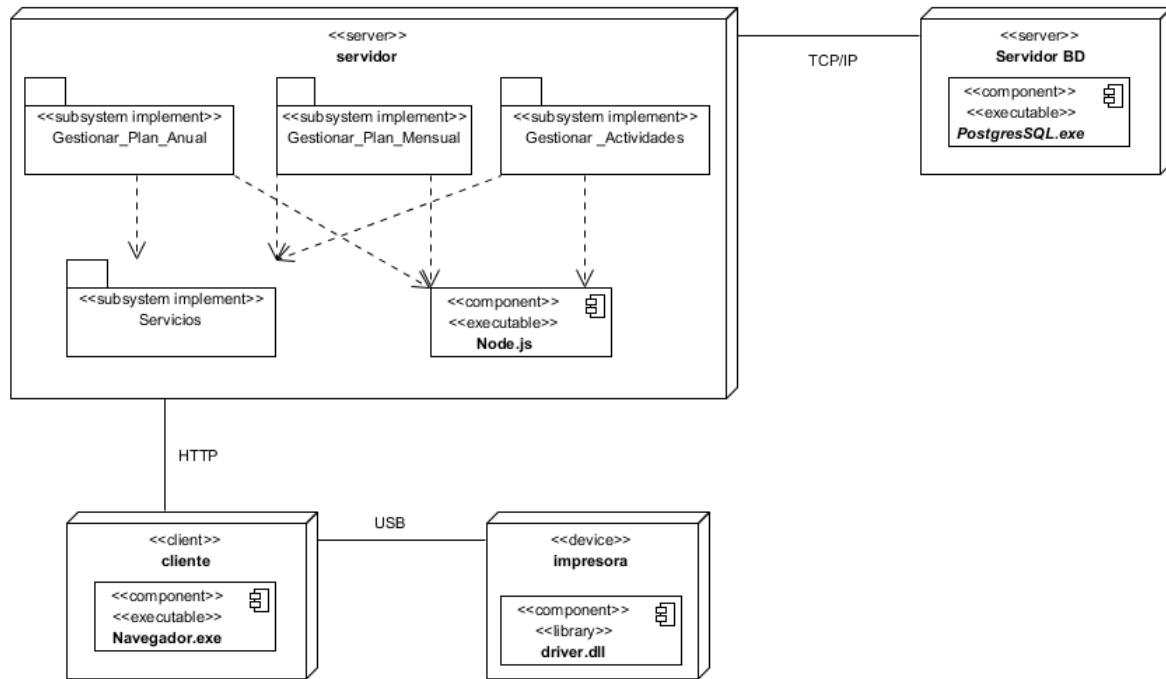


FIGURA 12 DIAGRAMA DE DESPLIEGUE

3.6 Modelo de Componentes

Muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes cubren la vista de implementación estática de un sistema. Se relacionan con los diagramas de clases en que un componente se corresponde, por lo común, con una o más clases, interfaces o colaboraciones. (Bonch and Lukewood, 1998)

En la siguiente ilustración se muestra el modelo de componentes del sistema para el CU Gestionar Plan Anual donde se encuentran los archivos js y .css.

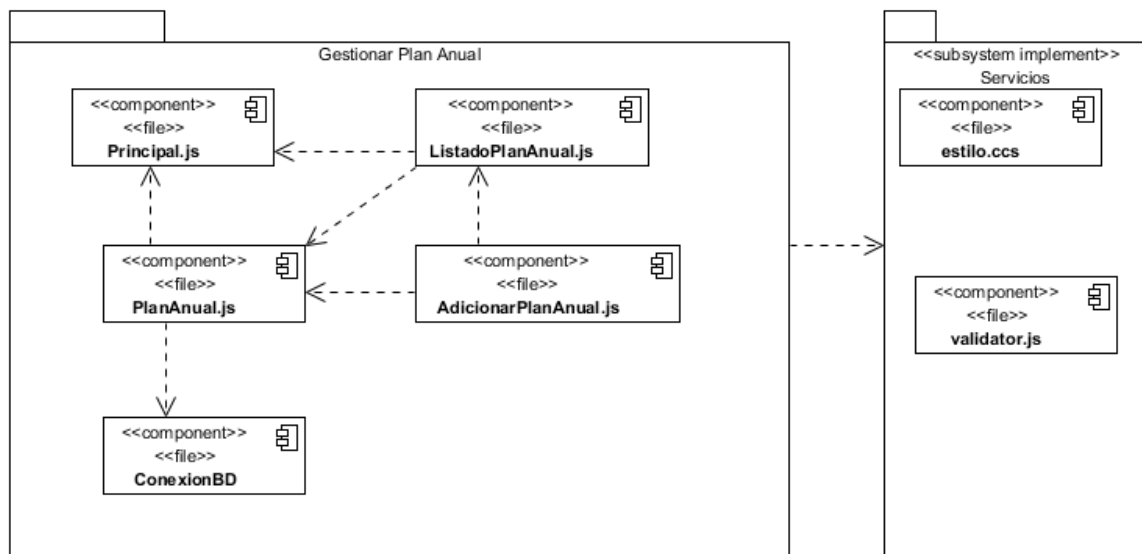


FIGURA 13 MODELO DE COMPONENTES CASO DE USO: GESTIONAR PLAN ANUAL

CONCLUSIONES PARCIALES

En este capítulo se obtuvo un diseño mediante el modelado de las características principales del sistema utilizando la notación UML partiendo de la arquitectura usada en la aplicación, se define una descripción de las clases del diseño. También se abordan los diferentes modelos como el modelo de componentes y el modelo del diseño de la base de datos.

Capítulo 4 Pruebas de análisis y factibilidad

En este capítulo se discuten las generalidades del proceso de pruebas que se incluye en el ciclo de vida de un software, así como las pruebas específicas bajo las cuales se prueba la herramienta del software. Para ello se utilizará un software de distribución gratuita llamado JMeter. También explicaremos cómo se puede realizar un Plan de Prueba que será lo que testeara la aplicación. En la segunda parte veremos las pruebas realizadas y sacaremos las conclusiones. También se realizarán pruebas de caja negra.

3.1 Pruebas de rendimiento:

Las pruebas de rendimiento realizados sobre computadoras, redes, software u otros dispositivos, son utilizados para determinar la velocidad y eficiencia de los mismos. Este procedimiento puede involucrar tanto pruebas cuantitativas como, medir tiempos de respuesta o cantidad en millones de líneas de código, como pruebas cualitativas, en los cuales se evalúa fiabilidad, escalabilidad e interoperabilidad. Estas pruebas de rendimiento pueden ser realizadas a través de herramientas que proveen pruebas de estrés, que permiten determinar la estabilidad del sistema.(Diaz et al., 1900)

Prueba de carga:

La prueba de carga de una estructura es un proceso que, mediante la reproducción de un estado de carga sobre la misma, pretende obtener datos suficientes de su respuesta frente a dicho estado, de forma que pueda deducirse Su comportamiento frente a las cargas actuantes.(Sommerville, 2005)

En la siguiente ilustración se muestra el resultado de una prueba de carga para el sistema con el Apache JMeter donde se ha estimado 5 usuario con una petición por segundo, no obteniendo ni ningún error en el sistema y se comprueba que el rendimiento es de baja carga .

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | KB/sec | Avg. Bytes |
|---|-----------|---------|-----|-----|-----------|---------|------------|--------|------------|
| /success.txt | 50 | 0 | 0 | 1 | 0.50 | 100.00% | 37.3/sec | 60.44 | 1660.0 |
| /planificacion/plan_anual/getplananual/2... | 5 | 6 | 5 | 7 | 0.89 | 0.00% | 6.2/sec | 3.29 | 544.0 |
| /planificacion/plan_anual/getplananual/2... | 5 | 106 | 4 | 507 | 200.50 | 0.00% | 3.8/sec | 3.80 | 1017.0 |
| /planificacion/plan_anual | 5 | 6 | 5 | 8 | 1.36 | 0.00% | 3.8/sec | 2.07 | 554.0 |
| /planificacion/plan_anual/inicio | 5 | 7 | 6 | 8 | 0.89 | 0.00% | 3.8/sec | 30.73 | 8220.0 |
| TOTAL | 70 | 9 | 0 | 507 | 59.97 | 71.43% | 52.2/sec | 98.07 | 1923.9 |

TABLA 10 PRUEBA DE CARGA

Prueba de stress:

Las pruebas de stress van realizando pruebas acercándose a la máxima carga del diseño del sistema hasta que el sistema falle. Son particularmente relevantes para los sistemas distribuidos basados en red de procesadores. (Sommerville, 2005, Grimán et al., 2003)

En la siguiente ilustración se muestra el resultado de una prueba de stress para el sistema con el Apache JMeter donde se ha estimado un usuario con 10 peticiones por segundo, no obteniendo ni un error en el sistema y se comprueba que el rendimiento es de baja carga.

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | KB/sec | Avg. Bytes |
|--|-----------|---------|-----|-----|-----------|---------|------------|--------|------------|
| /success.txt | 90 | 0 | 0 | 1 | 0.43 | 100.00% | 72.6/sec | 117.75 | 1660.0 |
| /planificacion/plan_anual/getplananual/... | 10 | 104 | 4 | 505 | 199.95 | 0.00% | 8.2/sec | 12.30 | 1530.5 |
| /planificacion/plan_anual | 10 | 5 | 4 | 8 | 1.33 | 0.00% | 8.2/sec | 4.38 | 545.0 |
| /planificacion/plan_anual/inicio | 10 | 6 | 5 | 9 | 1.02 | 0.00% | 8.3/sec | 118.74 | 14712.0 |
| TOTAL | 120 | 9 | 0 | 505 | 64.44 | 75.00% | 96.9/sec | 250.07 | 2644.0 |

TABLA 11 PRUEBA DE STRESS

Pruebas de resistencia(SOAK):

Representa una técnica de simulación cuyo objetivo es determinar la capacidad de estabilidad de una entidad o del sistema bancario. Para ello, se someten tanto las carteras de activos como de pasivos de las entidades financieras a diferentes situaciones para conocer sus posibles reacciones. (Rodríguez López, 2015)

En la siguiente ilustración se muestra el resultado de una prueba de carga para el sistema con el Apache JMeter donde se han estimado 10 usuarios haciendo 10 peticiones por segundo cada usuario, no obteniendo ni un error en el sistema y se comprueba que el rendimiento es de baja carga.

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | KB/sec | Avg. Bytes |
|--|-----------|---------|-----|-----|-----------|---------|------------|--------|------------|
| /success.txt | 700 | 0 | 0 | 3 | 0.50 | 100.00% | 203.3/sec | 329.59 | 1660.0 |
| /planificacion/actividad_anual/2 | 200 | 39 | 8 | 80 | 14.12 | 0.00% | 58.4/sec | 499.18 | 8746.0 |
| /planificacion/actividad_anual/getactividadanualbyid/1 | 200 | 36 | 6 | 74 | 14.74 | 0.00% | 58.7/sec | 43.50 | 759.0 |
| /planificacion/combo/estadocumplimiento | 100 | 34 | 4 | 72 | 15.36 | 0.00% | 29.8/sec | 13.78 | 474.0 |
| /planificacion/estadocumplimiento/getestadocumplim... | 100 | 33 | 5 | 75 | 14.59 | 0.00% | 29.7/sec | 10.13 | 349.0 |
| /planificacion/actividad_anual/ | 100 | 31 | 4 | 61 | 13.05 | 0.00% | 29.7/sec | 11.70 | 403.0 |
| TOTAL | 1400 | 18 | 0 | 80 | 20.60 | 50.00% | 406.6/sec | 903.55 | 2275.4 |

TABLA 12 PRUEBA DE RESISTENCIA

3.2 Casos de prueba (caja negra)

Las pruebas de caja negra analizan principalmente la compatibilidad entre sí, en cuanto a las interfaces. Son pruebas funcionales sin acceso al código fuente de las aplicaciones, se trabaja con entradas y salidas. Se centran en los requisitos funcionales del software, es decir, permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. El objetivo es descubrir errores en el sistema para su corrección, previo a la entrega del proyecto. (Tuya et al., 2007)

✓ Insertar plan anual

En la siguiente figura se muestra el formulario de insertar plan anual para la realización de las pruebas de caja negra.

Insertar plan anual

Fecha desde - hasta

Elaborado por

Aprobado por

Desde-hasta

Guardar

Cancelar

FIGURA 14 FORMULARIO INSERTAR PLAN ANUAL

Insertar plan anual:

Fecha desde-hasta: Campo date.

Elaborado por: Campo de texto.

Aprobado por: Campo de texto.

En la siguiente tabla se muestran los datos que deben introducirse en el formulario de insertar plan anual.

| Condición de entrada | Tipo | Clase de equivalencia válida | Clase de equivalencia no válida |
|----------------------|------------------|------------------------------|---|
| Fecha desde-hasta | Valor específico | 1: Cualquier fecha | 2: En blanco |
| Elaborado por | Valor específico | 3: Cualquier cadena de texto | 4: En blanco 5: Cadena de números y un carácter especial |
| Aprobado por | Valor específico | 6: Cualquier cadena de texto | 7: En blanco 8: Cadena de números y un carácter especial |

TABLA 13 DATOS QUE DEBEN INTRODUCIRSE EN EL FORMULARIO DE INSERTAR PLAN ANUAL.

En la siguiente tabla se muestran los valores 04/02/2017 - 05/02/2011 para el atributo fecha desde-hasta, el valor Marta Denia para el atributo elaborado por y el valor María Alonso para el atributo aprobado por que van a introducirse en el formulario de insertar plan anual.

| Condición de entrada | Casos de Prueba | Clases | Salida |
|----------------------|-------------------------|--------|--------|
| Fecha desde –hasta | 04/02/2017 - 05/02/2011 | 1 | OK |
| Elaborado por | Marta Denia | 3 | OK |
| Aprobado por | María Alonso | 6 | OK |

TABLA 14 DATOS INSERTADOS CORRECTAMENTE EN EL FORMULARIO INSERTAR PLAN ANUAL

En la siguiente figura se muestran los valores 04/02/2017 - 05/02/2011 para el atributo fecha desde-hasta, el valor Marta Denia para el atributo elaborado por y el valor María Alonso para el atributo aprobado que van a introducirse en el formulario de adicionar pieza.

Insertar plan anual

Fecha desde - hasta: 04/02/2017 - 05/02/2011

Elaborado por: Marta Denia

Aprobado por: María Alonso

[Guardar](#) [Cancelar](#)

FIGURA 15 FORMULARIO INSERTAR PLAN ANUAL CON DATOS CORRECTOS

| Condición de entrada | Casos de Prueba | Clases | Salida |
|----------------------|-----------------|--------|--------|
| Fecha desde –hasta | | 2 | Error |
| Elaborado por | | 4 | Error |
| Aprobado por | | 7 | Error |

TABLA 15 DATOS QUE DEBEN INTRODUCIRSE VACÍOS EN EL FORMULARIO DE INSERTAR PLAN ANUAL.

Insertar plan anual

Fecha desde - hasta: Desde-hasta

Elaborado por:

Aprobado por:

* El campo es requerido.

[Guardar](#) [Cancelar](#)

FIGURA 16 FORMULARIO INSERTAR PLAN ANUAL CON CAMPOS EN BLANCO

| Condición de entrada | Casos de Prueba | Clases | Salida |
|----------------------|-----------------|--------|--------|
| Fecha desde –hasta | | | Error |
| Elaborado por | | 5 | Error |
| Aprobado por | | 8 | Error |

TABLA 16 DATOS QUE DEBEN INTRODUCIRSE CON ERROR EN EL FORMULARIO DE INSERTAR PLAN ANUAL.

Conclusiones parciales

La etapa de pruebas funcionales de una aplicación es sumamente importante, ya que a partir de las mismas se podrá contar con la aprobación del usuario final, el tipo de pruebas mencionadas en el presente capítulo son también críticas utilizando para algunas de ellas la herramienta JMeter y obteniendo satisfactorios resultados en cada una de ellas, mostrando que el sistema está en perfectas condiciones para ser usado por el cliente.

Conclusiones

- ✓ Se diseñó la base de datos que permite registrar todo el proceso de gestión de las actividades en la División Territorial Desoft SS.
- ✓ Se diseñó una aplicación web adaptable a diferentes dispositivos que permite conocer en tiempo real el estado de las actividades para la división y sus trabajadores.
- ✓ Se realizaron pruebas satisfactorias a la aplicación que midieron el rendimiento, la disponibilidad y confiabilidad de los datos.

Recomendaciones

- ✓ Seguir perfeccionando el sistema pues será de ayuda para otras empresas con el mismo interés que la División Territorial Desoft SS.
- ✓ Realizar en un futuro a través del diseño realizado luego de definir la nueva estructura el plan individual de desarrollo.

Bibliografía

2012. *20 Software gratuitos para la Gestión de Proyectos* [Online].
- ANGEL, J. S. M. & BARON, F. E. O. 2012. SISTEMA INFORMÁTICO PARA LA PLANIFICACION Y GESTION ADMINISTRATIVA PARA EL DEPARTAMENTO DE LABORATORIO CLINICO DEL HOSPITAL NACIONAL SANTA TERESA.
- ARDILA, C. A. & PINO, F. J. 2013. Panorama de gestión cuantitativa de procesos de desarrollo de software en pequeñas organizaciones. *Sistemas & Telemática*, 11, 29-46.
- ARMIJO, D. M. 2009. Manual de Planificación Estratégica.
- AURELIA, P. & ANA, T. 2008. A document management system modeling. *ANALELE UNIVERSITĂȚII DIN ORADEA*, 1479.
- BELLETTI, C. & MOTZ, R. Análisis de Herramientas CASE para uso didáctico en Diseño de Bases de Datos. VIII Congreso Argentino de Ciencias de la Computación, 2002.
- BONCH, G. & LUKEWOOD 1998. Et LENGUAJE UNIFICADO DE MODELADO.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I., MARTÍNEZ, J. S. & MOLINA, J. J. G. 1999. *El lenguaje unificado del modelado*, Addison Wesley Madrid.
- CO-REQUISITOS, C. C. P.-R. & APROBÓ, E. R. Base de Datos. *Sistema*, 2, 6.
- DIAZ, F. J., BANCHOFF, C. M. T., RODRÍGUEZ, A. S. & SORIA, V. 1900. Usando Jmeter para pruebas de rendimiento.
- DORTA, M. J. R. 2014 - 2016 AngularJs Paso a Paso.
- EDEN, A. H. & KAZMAN, R. Architecture, design, implementation. proceedings of the 25th International Conference on Software Engineering, 2003. IEEE Computer Society, 149-159.
- GONZÁLEZ, A. H. 2005. Identificación de procesos de negocio. *Ingeniería Industrial*, 26, 4.
- GONZÁLEZ REI, B. & GARCÍA MATEO, C. 2000. Diseño de una base de datos tipo SpeechDat para el idioma gallego. *Procesamiento del lenguaje natural*, nº 26 (septiembre 2000); pp. 197-203.
- GORDON, P. H., MOORE, D. H., MILLER, R. G., FLORENCE, J. M., VERHEIJDE, J. L., DOORISH, C., HILTON, J. F., SPITALNY, G. M., MACARTHUR, R. B. & MITSUMOTO, H. 2007. Efficacy of minocycline in patients with amyotrophic lateral sclerosis: a phase III randomised trial. *The Lancet Neurology*, 6, 1045-1053.
- GRADY BOOCH, J. R., IVAR JACOBSON 1996. Lenguaje del modelado.
- GRIMÁN, A., PÉREZ, M. & MENDOZA, L. Estrategia de pruebas para software OO que garantiza requerimientos no funcionales. III Workshop de Ingeniería de Software, Chillán, Chile, 2003.
- GUIDE VP-UML.
- GULZAR, N. 2002. StrutsFastTrack.
- JACOBSON, I., BOOCH, G. & RUMBAUGH, J. 1990. <El proceso Unificado de Software.pdf>.
- JOYANES AGUILAR, L. 2003. *Fundamentos de programación: algoritmos y estructura de datos y objetos*.
- LARMAN, C. UML y patrones.
- LARMAN, C. 2003. *UML y Patrones*, Pearson Educación ^ eMadrid Madrid.
- LEONARD, M. C., DO PRADO LEITI, J. C. S. & ROSSL, G. 1998. Estrategias para la Identificación de Reglas de Negocio.
- MARTÍNEZ, G. 2002. Base de Datos. *Documento en línea* Disponible: <http://www.monografias.com/trabajos27/bases-datos/bases-datos.shtml> [Consulta: 2009, Marzo 18].
- MARTÍNEZ, I. A. F. H. I. J. O. G. 2017-2020 Estrategia Integral

- MOMJIAN, B. 2001. *PostgreSQL: introduction and concepts*, Addison-Wesley New York.
- MONZÓN, A. M. D. L. T. 1999. Introduccion a node.js.
- PERALTA, M. 2004. Estimación del esfuerzo basada en casos de uso. *Reportes Técnicos en Ingeniería de Software. Buenos Aires-Argentina*, 6, 1-16.
- PÉREZ, J. E. 2007. Introducción a JavaScript.
- ROBBINS, J. N. 2013. HTML5.
- RODRÍGUEZ LÓPEZ, W. A. 2015. Diseño y ejecución de pruebas no funcionales en la aplicación sistema de gestión documental basados en el esquema MDI 829.
- RONDÓN, Y., DOMÍNGUEZ, L. & BERENGUER, A. 2011. Diseño de la base de datos para sistemas de digitalización y gestión de medias. *Revista de Informática Educativa y Medios Audiovisuales*, 8, 17-25.
- RUZ, R. C. 2011. INSTRUCCION No. 1 DEL PRESIDENTE DE LOS CONSEJOS DE ESTADO Y DE MINISTROS.
- SOMMERVILLE, I. 2002. Ingeniería de Software.
- SOMMERVILLE, I. 2005. *Ingeniería del software*, Pearson Educación.
- TALEKAR, S. 2008. *WebStorm: Web based support tool for organization of requirements modeling*, University of Nevada, Reno.
- TUYA, J., ROMÁN, I. R. & COSÍN, J. J. D. 2007. *Técnicas cuantitativas para la gestión en la ingeniería del software*, NetBiblo.
- WEB, L. 2015. La arquitectura MVC. *Recuperado el*, 12.
- WENDY BOGGS, M. B. 2002. UML whit Rational Rose.
- ZAPATA, C. M. & OLAYA, Y. 2000. Ingeniería de Software.
- ZULIAN, E. R. 2011. *Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST*. Universidad de Belgrano. Facultad de Tecnología Informática.