

UNIVERSIDAD DE MATANZAS
Facultad de Ciencias Técnicas
Departamento de Informática



Trabajo de Diploma para optar por el título de Ingeniero en Informática.

**Título: PROPUESTA DE SOLUCIÓN PARA LA GESTIÓN DEL PROCESO DE
PLANIFICACIÓN E INSPECCIÓN MEDIO-AMBIENTAL PARA LA EMPRESA CITMA EN
LA PROVINCIA DE MATANZAS.**

Autor: Ariel Del Rio Viamonte

Tutor: MSc. Liana Romero Lovio

Co-tutor: Ing. Yeslaine Cortina Blanco

Matanzas, Cuba

Junio de 2020

Pensamiento:

Una buena ingeniería de software requiere la diferenciación entre la especificación y la implementación.

Andrew Tanenbaum

Agradecimientos:

A mi familia y amigos que siempre están presentes.

A mis tutores Liana y Yeslaine por el apoyo y la confianza brindada.

A aquellos que colaboraron en mi proyecto.

A todos, gracias...

Declaración de autoría

Yo, Ariel del Rio Viamonte, declaro que soy el único autor del trabajo “SISTEMA DE GESTIÓN DEL PROCESO DE PLANIFICACIÓN E INSPECCIÓN MEDIO-AMBIENTAL PARA LA EMPRESA CITMA EN LA PROVINCIA DE MATANZAS.” y autorizo a la Universidad de Matanzas y al Centro de Investigación y Desarrollo de Software, hacer uso del mismo en su beneficio.

Y para que así conste, firmo la presente a los _____ días del mes de junio del año 2020.

Firma del Autor

Firma del Tutor

Resumen:

La empresa CITMA (Ministerio de Ciencia, Tecnología y Medio Ambiente) tiene como misión proponer, dirigir y controlar la política estatal y gubernamental en materia de ciencia, tecnología, medio ambiente y así contribuir con la sostenibilidad y el desarrollo del país. Para dar cumplimiento a esta misión el CITMA ejecuta una serie de procesos como son la planificación de las inspecciones medio-ambientales y el seguimiento de los dictámenes y deficiencias identificados en dichas entidades. Actualmente estos procesos se realizan de forma manual generando atrasos, pérdida de información y desactualizaciones en la planificación y ejecución de las inspecciones. La investigación que se desarrolla tiene como objetivo proponer un sistema informático de gestión para la planificación e inspección medio-ambiental de la empresa CITMA en Matanzas, el cual permita agilizar y facilitar la gestión de la planificación para las inspecciones. La metodología utilizada para el desarrollo de la investigación es Prodesoft, metodología ágil que utiliza un desarrollo iterativo. El sistema será desarrollado en la plataforma Bienestar con la tecnología ProcessMaker. Para la modelación de los diferentes diagramas se utilizó la herramienta Visual Paradigm. Se empleó PHP y JavaScript como lenguajes de programación y PostgreSQL como gestor de bases de datos.

Abstract:

The mission of CITMA (Ministry of Science, Technology and the Environment) is to propose, direct and control state and government policy on science, technology, the environment and thus contribute to the sustainability and development of the country. In order to fulfill this mission, CITMA executes a series of processes such as the planning of environmental inspections and the follow-up of the opinions and deficiencies identified in said entities. Currently these processes are carried out manually, generating delays, loss of information and outdated planning and execution of inspections. The research that is carried out has the objective of design a computerized management system for the planning and environmental inspection of the CITMA company in Matanzas, which makes it possible to streamline and facilitate planning management for inspections. The methodology used for the development of the investigation is Prodesoft, an agile methodology that uses an iterative development. The system will be developed on the Bienestar platform with Process Maker technology. For the modeling of the different diagrams, the Visual Paradigm tool was used. PHP and JavaScript were used as programming languages and PostgreSQL as a database manager.

Índice general:

Introducción	1
Capítulo 1: Marco teórico referencial de la investigación.....	6
1.1 Introducción	6
1.2 Conceptos asociados al dominio del problema	6
1.3 Antecedentes.....	7
1.4 Metodología de desarrollo.....	11
1.5 Herramientas y tecnologías.....	12
1.5.1 Herramienta de modelado.....	12
1.5.2 Lenguaje de modelado	13
1.5.3 Tecnologías y lenguajes de programación.....	14
1.5.4 Herramientas de desarrollo.....	16
1.5.5 Frameworks.....	17
1.6 Conclusiones parciales	18
Capítulo 2: Análisis y diseño de la solución propuesta	19
2.1 Introducción	19
2.2 Modelo de negocio	19
2.2.1 Descripción del negocio	19
2.2.2 Reglas del negocio.....	23
2.2.3 Modelo conceptual	23
2.3 Requisitos del sistema informático.....	24
2.3.1 Requisitos no funcionales	24
2.3.2 Requisitos funcionales.....	25
2.3.3 Diagrama de actividades de los requisitos funcionales.....	28
2.4 Implementación de la arquitectura de software	32
2.5 Patrones de diseño.....	34

2.6 Mecanismo de diseño	36
2.7 Diseño de la base de Datos	37
2.8 Diagrama de clases de diseño	39
2.9 Diagramas de secuencia	42
2.10 Seguridad.....	44
2.11 Análisis de factibilidad.....	45
2.11.1 Costo	45
2.11.2 Beneficios tangibles e intangibles.....	47
2.11.3 Análisis del costo y beneficios.....	47
2.12 Diseño de pruebas de software.....	48
2.12.1 Tipos de pruebas de software	48
2.12.2 Diseños de casos de prueba	49
2.13 Conclusiones parciales.....	53
Conclusiones generales	55
Recomendaciones.....	56
Bibliografía.....	57
Anexos	60

Índice de imágenes:

Figura 1 Diagrama de proceso del negocio: PLANIFICACION_ANUAL	20
Figura 2 Diagrama de proceso del negocio: PLANIFICACION_MENSUAL	21
Figura 3 Diagrama de proceso del negocio: REALIZAR_INSPECCION.....	22
Figura 4 Diagrama del Modelo conceptual.....	24
Figura 5 Diagrama de actividad del requisito funcional adicionar planificación	28
Figura 6 Diagrama de actividad del requisito funcional modificar planificación	29
Figura 7 Diagrama de actividad del requisito funcional eliminar planificación.....	30
Figura 8 Diagrama de actividad del requisito funcional buscar planificación.....	31
Figura 9 Diagrama de actividad del requisito funcional adicionar dictamen de inspección	32
Figura 10 Diagrama de Mecanismo de diseño	37
Figura 11 Modelo lógico	38

Figura 12 Modelo Físico	39
Figura 13 Diagrama de clases de diseño del requisito funcional gestionar planificación de entidad.....	40
Figura 14 Diagrama de clases de diseño del requisito funcional generar dictamen	41
Figura 15 Diagrama de clases de diseño del requisito funcional Gestionar deficiencias ...	42
Figura 16 Diagrama de secuencia adicionar plan de inspección a entidad	43
Figura 17 Diagrama de secuencia modificar disposición	43
Figura 18 Diagrama de secuencia eliminar dictamen.....	44

Índice de tablas:

Tabla 1 Análisis de costo	45
Tabla 2 diseño de caso de prueba para el requisito funcional adicionar planificación.....	49
Tabla 3 Diseño de caso de prueba para el requisito funcional aprobar disposición.....	51

Introducción

Actualmente la sociedad vive en un mundo interconectado donde cada vez está más presente el desarrollo que propicia el uso de las Tecnologías de la información y las comunicaciones. Los campos donde estas tecnologías han mostrado su dominio abarcan desde las tareas más simples y cotidianas hasta procesos realmente complejos y avanzados.

Su uso para la sociedad realmente se pueden presenciar en cada aspecto y campos donde el ser humano intervenga, pero hay un espacio en el cual se ha visto la eficiencia y el avance que pueden llegar a ofrecer y es que las gestiones gubernamentales, los procesos y trámites que las entidades jurídicas de los países realizan y la interacción con sus ciudadanos, siempre ha carecido de cierta transparencia e inmediatez, estas tecnologías aplicadas a los procesos de los gobiernos propician un factor que siempre se ha requerido y que permite el avance en los procesos democráticos de los países, este factor es la interacción y comunicación directa e inmediata entre la población y el gobierno de su país.

Los denominados gobiernos electrónicos son el mayor avance y aporte del uso de las tecnologías de la información y las comunicaciones a los procesos gubernamentales y la relación de los mismos con sus ciudadanos. Cuba reconoce la necesidad de establecer ese gobierno electrónico en su plan de informatización de la sociedad y los avances que provoca en el desarrollo de la sociedad socialista.

Mucho se ha trabajado en el país para fomentar y desarrollar la informatización de la sociedad, y acercar a los ciudadanos a las tecnologías de la información y la comunicación (TIC). Este constituye uno de los sectores estratégicos del país, según lo establecido en la Conceptualización del Modelo Económico y Social Cubano y en las Bases del Plan Nacional de Desarrollo Económico y Social hasta 2030. (Paz, 2019)

Como parte de la implementación del gobierno electrónico en Cuba se han desarrollado diferentes proyectos como son los portales del ciudadano donde cada provincia cuenta con un sitio web que propicia un catálogo en línea con descripciones sobre servicios y trámites, otro proyecto reconocido es el denominado proyecto Bienestar desarrollado e Implementado por el Parque Científico y Tecnológico de Matanzas, proyecto sustentado en Plataformas tecnológicas de integración de Servicios y Gestión de trámites.

En el portal del ciudadano matancero se explica además que el gobierno electrónico es el empleo intensivo de las tecnologías de la información y las comunicaciones para la gestión y administración pública y constituye una nueva forma de ejercicio de gobierno. (Gobierno Matancero, 2019)

El proyecto de gobierno electrónico Bienestar tiene como objetivo implementar en la provincia de Matanzas de manera organizada, controlada y escalonada plataformas de integración de servicios y gestión de trámites a la población. Bienestar tiene el propósito de ofrecer por parte del gobierno y las instituciones un servicio de mayor calidad a la población matancera y perfeccionar los procesos de retroalimentación entre autoridades y sociedad.

En la provincia de Matanzas la implementación del gobierno electrónico se desarrolla en varias entidades, una de ellas es la empresa CITMA. Entre los procesos que se ejecutan en dicha empresa está la planificación de las inspecciones medio-ambientales a entidades de la provincia y el seguimiento de los dictámenes y deficiencias identificados en dichas entidades. Actualmente estos procesos se realizan de forma manual generando atrasos, pérdida de información y desactualizaciones en la planificación y ejecución de las inspecciones, además producto de los escasos mecanismos de comunicación, se dificulta el envío de la información sobre los planes de inspección que requieren los especialistas municipales, así como la confirmación de dichas inspecciones; también se dificulta la generación de reportes y el control en el seguimiento y la gestión de deficiencias y medidas establecidas como resultados de las inspecciones al no existir un mecanismo que mantenga la información centralizada y con una estructura correctamente definida donde se expongan los historiales de los dictámenes establecidos.

En relación a lo propuesto anteriormente se identificó como problema científico: ¿Cómo facilitar la gestión de los trámites necesarios para la planificación e inspección medio-ambiental de la empresa CITMA en Matanzas?

La solución a este problema se inserta en el siguiente objeto de estudio: la gestión de trámites para la planificación e inspección medio-ambiental, lo que propicia penetrar en un campo de acción centrado en: Sistema automatizado para la gestión de trámites para la planificación e inspección medio-ambiental de la empresa CITMA en la provincia de Matanzas.

Como guía a tener en cuenta para llevar a cabo la investigación se formularon las siguientes preguntas científicas.

1. ¿Qué fundamentos teóricos sustentan la gestión de trámites para la planificación e inspección medio-ambiental?
2. ¿Qué metodología utilizar para el diseño de un sistema informático?
3. ¿Qué pruebas se deben diseñar para la validación de la solución propuesta?

Con el fin de dar solución al problema planteado se traza como objetivo general: Proponer un sistema informático de gestión para la planificación e inspección medio-ambiental de la empresa CITMA en Matanzas.

Para dar cumplimiento a dicho objetivo se determinaron los siguientes objetivos específicos:

1. Realizar un estudio que permita conformar el marco teórico referencial de la investigación.
2. Modelar la propuesta de solución haciendo uso de la metodología Prodesoft.
3. Proponer una solución con las herramientas adecuadas para confeccionar un software.
4. Realizar diseños de caso de pruebas para la validación del sistema propuesto.

Los métodos utilizados para la resolución del problema de investigación son:

Métodos teóricos:

- Método de análisis y síntesis: se utilizó para realizar un análisis y una revisión de forma general de toda la información relacionada con el tema. Además de recopilar los aspectos fundamentales para el desarrollo de la investigación, permitiendo analizar soluciones similares que contribuyan como antecedentes de este trabajo.
- Método inductivo – deductivo: Se empleó para inducir las características generales del proceso y a partir de ellas obtener deducciones y conclusiones en la confección del documento y el producto final.
- Lógico-histórico: Se realizó un análisis detallado de los antecedentes, causas y condiciones históricas en que surgió y se desarrolló el objeto y fue utilizado durante la revisión de la literatura.

- Método de Modelación: Se utilizó para la modelación de los procesos de negocio.

Método Empírico:

- Método de Observación: Método utilizado para recolectar información sobre el comportamiento de nuestro objeto de estudio, permitió entender la realidad del proceso de gestión y solución de las opiniones espontáneas de la población.
- Revisión documental: Se empleó para analizar todos los documentos disponibles para la investigación.

Técnica de investigación utilizada:

- Entrevista: Técnica utilizada para obtener la información mediante charlas planificadas con especialistas en el tema; es decir, expertos en los procesos de planificación e inspecciones medio ambientales del CITMA que ayudaron en la investigación, especialmente en el levantamiento de requisitos del negocio.

El resultado esperado de este trabajo es proponer una solución al proceso de inspección y planificación medio ambiental de la empresa CITMA, mediante la realización de la ingeniería de software utilizando la metodología PRODESOF. La gestión de los datos de la solución propuesta posibilitará una mejor organización de la información, logrando de forma segura la integridad, extracción, manipulación y persistencia de los datos. Además, proporcionar una herramienta automatizada que permita a la empresa CITMA realizar una planificación de las inspecciones con mayor facilidad. Propiciar una rápida comunicación mediante el sistema entre los especialistas municipales y los responsables designados por la empresa para agilizar la planificación de las inspecciones. También se proporcionará un mayor control sobre los historiales de inspecciones, así como el seguimiento de deficiencias y medidas tomadas con las entidades evaluadas.

El presente trabajo está compuesto por una introducción, dos capítulos, conclusiones generales, recomendaciones y referencias bibliográficas. A continuación, se presenta una síntesis de cada uno de los capítulos:

- Capítulo 1: “Marco teórico referencial” se presenta una exposición detallada de los referentes teóricos que argumentan la propuesta y permiten un acercamiento al objeto de estudio. Además, se comenta sobre el estado del arte, las tendencias y tecnologías actuales que serán usadas para el desarrollo de la herramienta informática propuesta.

- Capítulo 2: “Análisis y diseño de la solución propuesta” se argumenta la solución que se propone al problema de investigación mediante su descripción. Se presenta el empleo de la metodología de desarrollo de software PRODESOFIT. Se realizan los diseños de las pruebas propuestas a realizar de acuerdo a los requisitos funcionales.

Capítulo 1: Marco teórico referencial de la investigación

1.1 Introducción

En este capítulo se lleva a cabo una descripción teórica de las temáticas que sirven de apoyo a la investigación a desarrollar. Se describen los antecedentes necesarios para su comprensión, las dificultades que se presentan y la propuesta de solución. Se definen las principales herramientas tecnológicas, lenguajes de programación y metodología para su desarrollo.

1.2 Conceptos asociados al dominio del problema

Los conceptos asociados al dominio del problema son los siguientes:

- CITMA: Ministerio de Ciencia, Tecnología y Medio Ambiente.
- IPF: Instituto de Planificación Física.
- ORSA: Oficina regulatoria de seguridad ambiental
- Especialista municipal: Empleado del CITMA con altos niveles de conocimiento en sectores sobre la gestión ambiental el cual labora en un municipio específico y es responsable de mantener el contacto entre las entidades municipales y la ORSA.
- Especialista de la ORSA: Trabajador del CITMA que labora en la ORSA el cual tiene entre sus responsabilidades la gestión de la planificación y el control de las inspecciones medio ambientales de la provincia.
- Disposición: Documento oficial que se emite para la ejecución de la inspección ambiental estatal ordinaria a entidades, el cual contiene los detalles de la inspección que se va a realizar como son la fecha de realización, el responsable designado y los objetivos específicos de la misma.
- Dictamen de inspección: Documento oficial donde se declaran los objetivos generales y específicos de la inspección realizada y las deficiencias detectadas a una entidad inspeccionada, así como el conjunto de leyes violadas y el plan de medidas que se aplica para la corrección de dichas deficiencias y sus fechas de cumplimiento, además de otros pronunciamientos de interés sobre la realización de la inspección.

Estado actual

La empresa CITMA con sede en matanzas ejecuta una serie de procesos como son la planificación de las inspecciones medio-ambientales a entidades de la provincia y el seguimiento de los dictámenes y deficiencias identificados en dichas entidades. Actualmente estos procesos se realizan de forma manual generando atrasos, pérdida de información y desactualizaciones en la planificación y ejecución de las inspecciones, además producto de los escasos mecanismos de comunicación, se dificulta el envío de la información sobre los planes de inspección que requieren los especialistas municipales, así como la confirmación de dichas inspecciones; también se dificulta la generación de reportes y el control en el seguimiento y la gestión de deficiencias y medidas establecidas como resultados de las inspecciones al no existir un mecanismo que mantenga la información centralizada y con una estructura correctamente definida donde se expongan los historiales de los dictámenes establecidos.

Por la gran importancia que este proceso representa para la sociedad, el diseño de sistema sugerido debe contribuir a mejorar la gestión de las planificaciones de inspecciones efectuadas por la empresa CITMA, además de cooperar al desarrollo del programa de gobierno electrónico en el país.

1.3 Antecedentes

Según (OAS, s.f) gobierno electrónico se puede definir como “el uso de las Tecnologías de Información y Comunicación (TIC), por parte de las instituciones de gobierno, para mejorar cualitativamente los servicios e información que se ofrecen a los ciudadanos; aumentar la eficiencia y eficacia de la gestión pública e incrementar sustantivamente la transparencia del sector público y la participación ciudadana”.

Es el resultado de un crecimiento exponencial de internet y las tecnologías de la información durante los últimos 50 años o más. Se puede considerar al gobierno electrónico como un intento por modernizar las relaciones (y servicios) entre los individuos y las autoridades, a través de nuevas reglas y herramientas de gestión modernas.

Muchos gobiernos en el mundo comprenden este cambio y han mostrado gran adaptabilidad en adoptar un verdadero enfoque centrado en el ciudadano, tan rápido y efectivo como las organizaciones o empresas privadas orientadas hacia el consumidor.

Muchos son los beneficios que una estructura de gobierno electrónico provee a la ciudadanía y a las autoridades, sobre todo en sectores muy sensibles para ambos actores:

- Sector fiscal: Con base en nuevas tecnologías de información, el gobierno pone al alcance de sus ciudadanos herramientas en línea que permiten una mejor identificación de sus obligaciones.
- Sector salud: Expedientes médicos electrónicos, telemedicina, seguimiento electrónico a pacientes con enfermedades crónicas, entre otros, son beneficios con impacto directo en la salud de los ciudadanos, al mismo tiempo que reducen costos para los gobiernos.
- Sector social: Identificación y mejora en la gestión de apoyos a los beneficiarios de programas gubernamentales.
- Mejora administrativa: Modernización de los registros civiles, rapidez en servicios de identificación ciudadana (expedición de visas, pasaportes), mejora en servicios provistos por autoridades locales. (El Espectador, 2012)

Cuba tiene como propósito la implementación progresiva del gobierno electrónico, a partir de las condiciones existentes, realizando las transformaciones requeridas en los procesos administrativos, estructuras y funciones actuales, y teniendo en cuenta: Ciudadano como centro del proceso (objeto/sujeto). Implementación escalonada y simultánea de cada una de las etapas. No informatizar procedimientos obsoletos. Mantener la actualización sistemática de los contenidos y servicios que se implementan. (Ministerio de Comunicaciones, 2019)

En Cuba se busca que la interacción de los gobiernos con los ciudadanos sea mayor, para esto se avanza en el desarrollo de gobiernos electrónicos en todas las provincias. La implementación de los gobiernos electrónicos consta de cuatro etapas:

- Presencia (publicación en línea de la información instructiva o de interés para los usuarios, tanto ciudadanos como otras instituciones).

Los Organismos de la Administración Central del Estado (OACE) y los gobiernos territoriales han cumplido, en el primer trimestre del 2019, los 12 elementos previstos para la etapa de Presencia. Todos cuentan con portal institucional de gobierno público en

Internet, bajo el dominio genérico de segundo nivel gov.cu (Ministerio de Comunicaciones, 2019).

- Interacción (se comienza la implementación de los mecanismos de comunicación e intercambio con los usuarios, incrementándose el nivel de interacción de los mismos con el gobierno).

Durante el 2019 se trabajó en el cumplimiento de los 8 elementos para la etapa de interacción. (Ministerio de Comunicaciones, 2019).

- Transacción (se garantizan los trámites y servicios en línea, así como la interacción simultánea de los usuarios con el gobierno).
- Transformación (se genera una transformación en el pensamiento y actuar de los ciudadanos, elevando a un primer plano la interacción con el gobierno a través de las TIC).

El portal del ciudadano aporta una serie de beneficios entre ellos:

- Facilitar información sobre los principales trámites y servicios que puedan realizar los ciudadanos en las entidades estatales.
- Informar sobre los principales temas de actualidad a través de la publicación de noticias.
- Proporcionar información sobre los eventos culturales y deportivos del territorio.
- Proveer un sistema de avisos sobre informaciones de carácter inmediato.
- Propiciar la facilitación social mediante la publicación de anuncios clasificados.
- Informar sobre los principales resultados de las diferentes entidades estatales relacionados con la población.
- Transparencia sobre el trabajo que realiza el gobierno en la provincia y los programas sociales.
- Propiciar la participación del ciudadano y el intercambio con el gobierno sobre diversos temas.
- Mejorar la calidad de vida de los ciudadanos.
- Optimización del tiempo del ciudadano.
- Abre nuevos espacios y formas de participación ciudadana.
- Comunicación con el gobierno desde cualquier lugar, en cualquier momento y con cualquier dispositivo. (Domínguez, 2018)

El 1 de noviembre de 2017 comenzó a funcionar el primer Parque Científico Tecnológico ubicado en la universidad de Matanzas (PCTM), luego de que el Consejo de Administración Provincial de esa provincia aprobara a través de un acuerdo este proyecto. Entre sus objetivos está la integración de todas las fuerzas que intervienen en el programa de informatización del territorio y también las empresas que trabajan la automática. El gobierno como una de las hélices o fuerzas motrices presenta a la Junta Directiva del PCTM todos los proyectos de la provincia que sean necesarios ejecutar, contenidos en el plan de desarrollo y uso de las TIC, y algunos los financiará como es el caso del proyecto Bienestar, que permite informatizar la gestión de los delegados de circunscripciones y de las Asambleas Municipales y Provincial del Poder Popular. Es importante señalar que el PCTM se integra a la concepción del Polo científico- productivo de la provincia de Matanzas, a través de los diferentes frentes científicos tecnológicos. Las áreas temáticas principales que atenderá son: gobierno electrónico, comercio electrónico, gestión del conocimiento, medio ambiente, energía, turismo, agricultura, cultura, gestión empresarial, así como producción y servicios. (Pérez Salomón, 2018)

La labor que se desarrolla en el Parque Tecnológico contribuye a mejorar la calidad de vida y participación de la población matancera. Este avance de digitalización permite agilizar procesos como la solicitud de certificaciones de notas, de turnos médicos, servicios de emergencia médica, de abasto de agua, trámites en la vivienda, la empresa eléctrica, servicios de acueducto y alcantarillado, así mismo se podrá gestionar el otorgamiento de tierras en usufructo, de círculos infantiles y de carreras de cursos por encuentro. La universidad acoge la sede del parque, el cual gestiona el flujo del conocimiento y actúa como puente entre la investigación y el mercado en la transferencia, comercialización de productos, así como la actividad económica sostenible de creación. Como resultado de esa integración resultan la aplicación para celulares Mi Constitución y la Plataforma Bienestar, que conecta a los gobiernos locales a una sala situacional para gestionar la solución de planteamientos de la población. (Granma, 2019)

Bienestar es un proyecto en conjunto entre la Universidad de Matanzas y la Empresa de Tecnología de la Información para la Defensa (XETID) el cual está enfocado en la realización de una plataforma digital y participativa donde se recogerán los planteamientos de la población, es uno de los proyectos que hoy marcan el camino hacia la informatización de la sociedad y el gobierno digital en la provincia Matanzas. Este

proyecto contribuirá al desarrollo del gobierno digital o electrónico en el territorio. (Radio26, 2018)

1.4 Metodología de desarrollo

Prodesoft (Proceso de Desarrollo y Gestión de Proyectos de Software): es una metodología que posee una gran flexibilidad y adaptabilidad, su proceso es iterativo e incremental, es decir, un enfoque en el que el ciclo de vida está compuesto por iteraciones, estas son pequeños procesos compuestos de varias actividades cuyo objetivo es integrar una parte del sistema parcialmente completo, probado, integrado y estable. (Xetid, 2012)

Es basado en componentes, lo que permite alcanzar un mayor nivel de reutilización de software, aún en contextos distintos a aquellos para los que se diseñó. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo. (Xetid, 2012)

Su ciclo de vida consta de 5 fases ([Anexos | Anexo 1](#)):

1. Inicio: En esta fase realiza un análisis de la problemática existente y se establece la estrategia a seguir para realizar la captura de requisitos, así como un estimado del alcance del proyecto.
2. Modelación: Se capturan las partes esenciales del sistema donde se identifican los procesos de negocio fundamentales, se aceptan los requerimientos funcionales y se definen la arquitectura de sistema y de datos.
3. Construcción: Es la fase donde se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. En esta fase todas las características, componentes, y requerimientos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión liberada del producto.
4. Explotación experimental: Durante esta fase se convierte la versión liberada del producto en una solución estable, donde se eliminan los errores que surgen durante las pruebas y se obtiene una certificación funcional y de seguridad del producto.

5. Despliegue: En la fase de despliegue se instala y configura el sistema para un ambiente de producción real, se capacita al personal que usará la aplicación y se continúa dando soporte durante la explotación del sistema.

El empleo de esta metodología de desarrollo de software en la investigación, está justificado por las normativas del proyecto Xetid.

1.5 Herramientas y tecnologías

Luego de un estudio de las tecnologías actuales para el desarrollo de software, se decidió dadas las ventajas y el conocimiento del personal de desarrollo de este trabajo utilizar las herramientas que se caracterizan a continuación.

1.5.1 Herramienta de modelado

Visual Paradigm es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos de Ciclo de Vida de desarrollo de un Software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. La herramienta tiene un buen ambiente de trabajo, lo que facilita la visualización y manipulación del proyecto de modelado. (Visual Paradigm, 2018)

El software de modelado UML permite una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es compatible con los equipos de desarrollo de software de captura de requerimientos, planificación de software (utilizar el análisis de casos), ingeniería de código, modelado de clase, el modelado de datos, etc. (Picazo, 2013)

Además, cuenta con potentes herramientas de modelado visual que lo ayudan a construir y administrar sus diagramas y elementos de modelo:

- Editor de diagramas de arrastrar y soltar
- Soporta UML , BPMN , ArchiMate , DFD , ERD , SoaML , SysML , CMMN
- Herramientas de modelado efectivas como la reutilización de elementos, la transformación de diagramas y elementos, validación de sintaxis, propiedades personalizadas, etc.
- Muchas opciones de formato (Visual Paradigm, sf)

En la presente investigación se utilizará la notación gráfica BPMN, además se empleará para realizar todos los diagramas que propone Prodesoft como parte del modelado del negocio, haciendo de esta herramienta una buena opción para el trabajo en la misma.

1.5.2 Lenguaje de modelado

Modelo y Notación de Procesos de Negocio (BPMN) es una notación gráfica estandarizada que describe la lógica de los pasos de un proceso de Negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (BPD).

BPD es un diagrama diseñado para representar gráficamente la secuencia de todas las actividades que ocurren durante un proceso, incluye además toda la información que se considera necesaria para el análisis. BPD es un diagrama diseñado para ser usado por los analistas, quienes diseñan, controlan y gestionan procesos. Dentro de un Diagrama de Procesos de Negocio BPD se utiliza un conjunto de elementos gráficos, agrupados en categorías, que permite el fácil desarrollo de diagramas simples y de fácil comprensión, pero que a la vez manejan la complejidad inherente a los procesos de negocio. (Nextech, 2016)

Esta notación se usará en Process Maker y Visual Paradigm para la modelación del negocio.

El lenguaje de modelado unificado (UML) ISO / IEC 19501: 2004 describe el lenguaje de modelado unificado (UML) como un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema intensivo en software. UML ofrece una forma estándar de escribir los planos de un sistema, que incluyen elementos conceptuales como procesos de negocios y funciones del sistema, así como elementos concretos como declaraciones de lenguaje de programación, esquemas de bases de datos y componentes de software reutilizables. (ISO, 2005)

Por un lado, el lenguaje de modelado puede servir de modelo para un proyecto y garantizar así una arquitectura de información estructurada; por el otro, ayuda a los desarrolladores a presentar la descripción del sistema de una manera que sea comprensible para quienes están fuera del campo. UML se utiliza principalmente en el

desarrollo de software orientado a objetos. UML es una combinación de varias notaciones orientadas a objetos: diseño orientado a objetos, técnica de modelado de objetos e ingeniería de software orientada a objetos y representa buenas prácticas para la construcción y documentación de diferentes aspectos del modelado de sistemas de software y de negocios. Motivo por el cual se utilizó como lenguaje en la modelación del sistema, dígame diagrama de actividad, modelo conceptual y modelo de datos. (Pressman, et al., 2014)

Se utilizará UML para el modelado de las clases de la solución propuesta.

Lenguaje de modelado de sistemas (SysML) es un lenguaje de modelado de arquitectura de uso general para aplicaciones de ingeniería de sistemas.

- SysML admite la especificación, análisis, diseño, verificación y validación de una amplia gama de sistemas y sistemas de sistemas. Estos sistemas pueden incluir hardware, software, información, procesos, personal e instalaciones.
- SysML es un dialecto de UML 2 y se define como un perfil UML 2. (Un perfil UML es un dialecto UML que personaliza el idioma a través de tres mecanismos: estereotipos, valores etiquetados y restricciones). (Sysml, sf)

Se utilizará para la modelación de requisitos funcionales del proyecto.

1.5.3 Tecnologías y lenguajes de programación

PHP: es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. El objetivo principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas; aunque se puede hacer mucho más con PHP. (Php, sf)

PHP se empleará como lenguaje del lado del servidor para construir la lógica del sistema y de los plugin incorporados.

Lenguaje de Marcas de Hipertexto (HTML): es la pieza más básica para la construcción de la web y se usa para definir el sentido y estructura del contenido en una página web. Otras tecnologías además de HTML son usadas generalmente para describir

la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript). (Mozilla, sf)

Se utilizará para el maquetado y estructura de las interfaces.

Hojas de Estilo en Cascada (CSS) es un estándar diseñado principalmente para separar atributos de diseño (como el color de fondo, el tamaño de la fuente o la sangría) del contenido. Aunque éste puede trabajar con muchos tipos de documentos, es mayormente utilizado como una forma de proveer información de diseño en páginas web.

La separación entre contenido y presentación mejora la flexibilidad, permitiendo a los autores definir un único conjunto de estilos que miles de documentos utilizarán, reduciendo así el tiempo y trabajo que se debe invertir en cada actualización. (León, sf)

CSS se empleará para proveer estilos a las vistas.

JavaScript: es un lenguaje de programación ligero e interpretado, orientado a objetos con funciones de primera clase. Aunque es más conocido como el lenguaje de scripting para páginas web, muchos entornos no relacionados con el navegador también lo usan, como es el caso de node.js. Es un lenguaje script multiparadigma, basado en prototipos, dinámico, soporta estilos orientados a objetos, imperativos y declarativos. (Mozilla, sf)

Se utilizará JavaScript para la creación de interfaces dinámicas, así como para la validación de formularios en las vistas

JQuery es una librería perteneciente al lenguaje de programación JavaScript, uno de los más usados en materia de desarrollo web, de código abierto y con la capacidad de mejorar la interactividad de una página web sin tener nociones de programación o un profundo conocimiento de este lenguaje. Es algo que facilita enormemente la tarea de desarrollo y diseño de páginas web. JQuery sirve para dar una capa extra de usabilidad una web y mejorar el grado de interacción con el usuario. Es una librería indispensable para el desarrollo y la implementación de plugin en cualquier página web, por lo que goza de una doble utilidad que la vuelve indispensable en el entorno digital. (NeoAttack, sf)

Su característica principal es que permite cambiar el contenido de una página web sin necesidad de recargarla mediante la manipulación del árbol DOM y peticiones AJAX.

Otras características de esta biblioteca son que permite la selección de elementos, la manipulación de la hoja de estilos CSS, soporta extensiones y presenta utilidades como obtener información del navegador, operar con objetos y vectores. Además, tiene como una gran ventaja que es compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+. (Villarreal Fuentes, 2013)

Tanto esta librería como el propio JavaScript serán tomadas para realizar las validaciones de los campos en las vistas.

Lenguaje Estructurado de Consultas (SQL): es un lenguaje de programación estándar e interactivo para la obtención de información desde una base de datos y para actualizarla. Aunque SQL es a la vez un ANSI y una norma ISO, muchos productos de bases de datos soportan SQL con extensiones propietarias al lenguaje estándar. Las consultas toman la forma de un lenguaje de comandos que permite seleccionar, insertar, actualizar, averiguar la ubicación de los datos, y más. También hay una interfaz de programación. (Tech Target, sf)

Se utilizará para realizar consultas a la base de datos.

1.5.4 Herramientas de desarrollo

Gestor de Base de Datos PostgreSQL, es un potente sistema de base de datos relacional de objetos de código abierto que usa y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas. Además de ser gratuito y de código abierto, PostgreSQL es altamente extensible y se ejecuta en los principales sistemas operativos (Postgresql, sf)

Se utilizó para la confección del modelo físico de datos y para la gestión de la base de datos desde la interfaz gráfica del gestor.

PhpStorm: es un IDE (entorno de desarrollo integrado) de PHP compatible con PHP 5.3 / 5.4 / 5.5 / 5.6 / 7.0 / 7.1 / 7.2, proporciona prevención de errores sobre la marcha, refactorización de código y autocompletado, depuración de configuración cero y un editor extendido de HTML, CSS y JavaScript. (Jetbrains, sf)

El IDE que se sugiere que se utilice como editor de código para la implementación de plugin.

1.5.5 Frameworks

ProcessMaker es una solución de software de flujos de trabajo, de código abierto simple y rentable. También conocido como Gestor de procesos empresariales, ProcessMaker ayuda a las organizaciones de todos los tamaños para diseñar fácilmente, automatizar e implementar procesos de negocio.

La caja de herramientas ProcessMaker permite a los usuarios de negocio crear formas y mapas de flujos de trabajo completamente funcionales. El software está completamente basado en web, lo que facilita la coordinación del flujo de trabajo entre los usuarios, departamentos y organizaciones. Como una aplicación de SOA (Arquitectura Orientada a Servicios) de gran alcance, ProcessMaker puede interconectarse con sistemas que incluyen la gestión de documentos y aplicaciones de inteligencia empresarial. (Scholarium, sf).

Este framework es el que utiliza la empresa para implementar los procesos de negocios.

Zend Framework es una colección de paquetes PHP profesionales. Se puede usar para desarrollar aplicaciones y servicios web usando PHP 5.6+, y proporciona código orientado a objetos usando un amplio espectro de características del lenguaje. Cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. (Zend Framework, sf)

Se utilizará para la implementación de la lógica del negocio en los plugin.

Doctrine es un ORM (mapeador objeto-relacional) escrito en PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un sistema de gestión de bases de datos (SGBD). (Doctrine, sf)

Doctrine es el ORM que se utilizará para mapear los datos e interactuar entre los modelos y la base de datos.

Ext JS: Librería Javascript que incluye componentes de alto performance y personalización, modelo de componentes extensibles, y un API fácil de usar. Existe un balance entre Cliente – Servidor, la carga de procesamiento se distribuye permitiendo que

el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo. Permite comunicación asíncrona y una mayor eficiencia de la red. (J.E.S, 2008)

Éste framework se empleará para la realización de los componentes visuales de la solución.

1.6 Conclusiones parciales

El desarrollo del capítulo permitió conocer las bases teóricas sobre las cuales se sustenta la propuesta de esta investigación. Se analizó cómo se efectúan los procesos de planificación e inspección medio ambiental por parte de la empresa CITMA y se corroboró la necesidad de diseñar un sistema que agilice y facilite el desarrollo de estos procesos, así como su integración en el proyecto Bienestar como objetivo primordial en el plan de implementación del gobierno electrónico. Además, se definió que la combinación de herramientas, tecnologías y la metodología de desarrollo de software establecida por la empresa XETID, es la apropiada para la realización del sistema a diseñar ya que resuelve la problemática planteada en la investigación.

Capítulo 2: Análisis y diseño de la solución propuesta

2.1 Introducción

El capítulo abarca el análisis y diseño de la solución propuesta. Se detallan los elementos principales del sistema en el modelo conceptual. Se describen las características que deberá cumplir el sistema según la metodología de desarrollo de software Prodesoft en la fase de modelación y se describen los requisitos funcionales y no funcionales. Además, se diseñan las diferentes pruebas que se realizarán para probar la factibilidad y correcto funcionamiento del sistema analizando su concepto, clasificación y exponiendo algunos diseños de casos de pruebas que facilitarán la implementación de dichas pruebas al software.

2.2 Modelo de negocio

2.2.1 Descripción del negocio

El negocio que se analiza y sobre el que se realiza todo el proceso de informatización se dividió en varios subprocesos para una mayor comprensión del mismo.

1- Planificación anual

El proceso inicia cuando el jefe del departamento de regulación y control del CITMA revisa los objetivos de trabajo a nivel nacional dispuestos por "La delegación nacional del CITMA", estos objetivos de trabajo presentan elementos y determinadas ubicaciones geográficas que requieren de inspección medio-ambiental de carácter prioritario. A partir de la consulta realizada a los objetivos de trabajo a nivel nacional, se determinan los municipios y entidades correspondientes que representan dichos objetivos de trabajo, pero a nivel provincial; luego se crean los objetivos de trabajo de la delegación provincial del CITMA que serán establecidos en la realización de la planificación anual. Se confecciona el plan anual estableciendo en el mismo los municipios, entidades, fecha, participantes y responsables de la inspección. Se establecen en el plan las inspecciones de mayor prioridad que se obtienen de los objetivos de trabajo y luego se establecen el resto de municipios y entidades a inspeccionar que determine la delegación provincial del CITMA para el resto del año planificado.

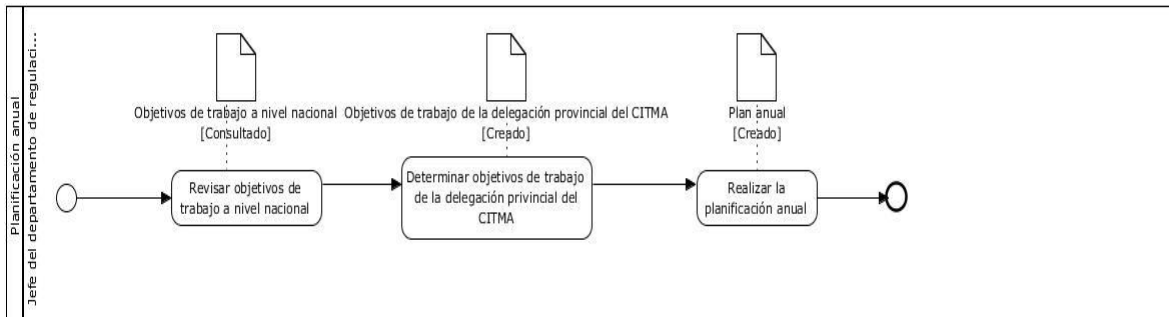


Figura 1 Diagrama de proceso del negocio: PLANIFICACION_ANUAL

2- Planificación mensual

Este proceso se desarrolla una vez que esté confeccionado el plan anual. El proceso inicia cuando el jefe del departamento de regulación y control determina los municipios a los que corresponde la inspección, en el mes a planificar (este proceso se realiza con un mes de antelación aproximadamente).

Para determinar las entidades que se incluirán en el plan mensual se realiza un proceso de obtención de sugerencias por parte de los especialistas municipales correspondientes al municipio a inspeccionar en ese mes. Se realiza la solicitud de dichas sugerencias, luego el especialista municipal determina a partir de criterios definidos sobre las entidades de su municipio, cuáles serán sugeridos para inspeccionar y sus motivos. Luego el especialista le entrega esas sugerencias al jefe del departamento de regulación, el cuál revisa las entidades sugeridas y decide si aprobarlas para incluirlas en el plan de inspección mensual. Una vez que se determinen las entidades que serán inspeccionadas se confecciona el plan mensual estableciendo en el mismo las entidades del municipio, la fecha y hora, los dirigentes y los participantes de las inspecciones medio-ambientales a realizar en el mes.

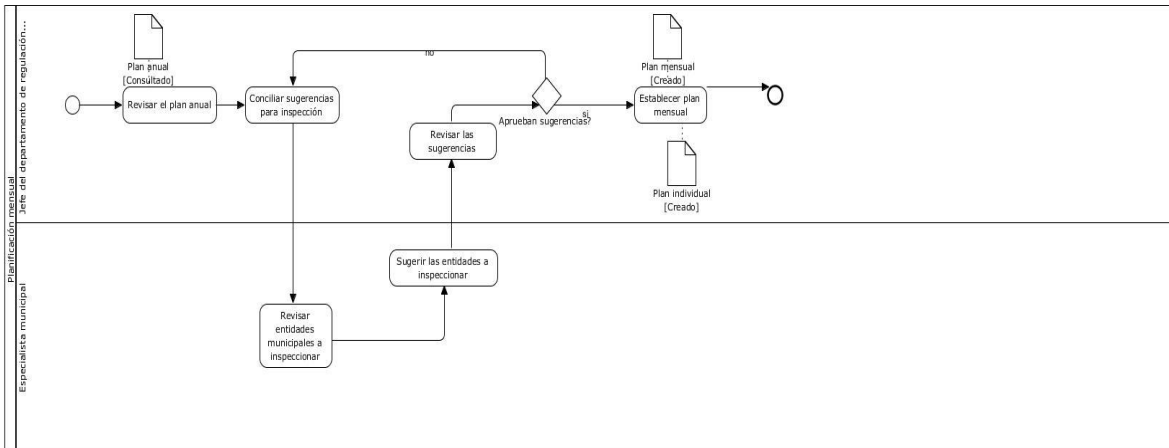


Figura 2 Diagrama de proceso del negocio: PLANIFICACION_MENSUAL

3- Realizar inspección

El proceso inicia cuando el jefe del departamento de regulación y control revisa el plan del mes correspondiente e identifica la entidad que corresponde inspeccionar.

Luego se confecciona la disposición, documento que contiene los detalles de la inspección que se realizará a la entidad correspondiente, además de que este documento formaliza y legaliza la intención de dicha inspección. Dicha disposición se debe entregar con tiempo de antelación a la entidad con un tiempo aproximado de 20-30 días. Esta disposición se le entrega al especialista municipal y a la entidad a inspeccionar correspondiente.

La inspección se realiza el día planificado y a partir de los resultados de la misma se confecciona el dictamen, documento que contiene los detalles de la inspección como son las deficiencias encontradas, los planes de medidas y los responsables de dichos planes, así como las fechas de cumplimiento.

El proceso finaliza con la creación y entrega de dicho dictamen al especialista municipal y a la entidad inspeccionada.

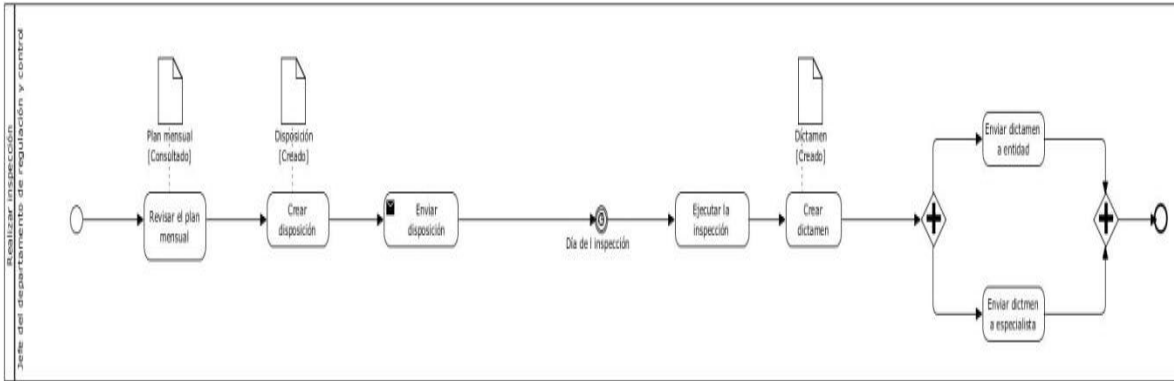


Figura 3 Diagrama de proceso del negocio: REALIZAR_INSPECCION

2.2.2 Reglas del negocio

Las reglas del negocio o conjunto de reglas del negocio describen las políticas, normas, operaciones, definiciones y restricciones presentes en una organización y que son de vital importancia para alcanzar los objetivos misionales. (Navarro Cifuentes, sf).

En el caso de esta investigación se localizaron las siguientes reglas:

R1 Solo puede confeccionar el plan de inspección anual y el plan de inspección mensual el jefe del departamento de regulación y control de la empresa CITMA con sede en Matanzas.

R2 Para la realización de las sugerencias de entidades a inspeccionar para establecer en el plan mensual, el especialista municipal deberá seleccionar entidades de su municipio y exponer el motivo por el cual sugiere la inspección a dicha entidad.

R3 Las sugerencias de las entidades a inspeccionar se deben realizar con al menos un mes de antelación.

R4 No pueden existir dos planes de inspección a la misma entidad en el mismo día

R5 No pueden existir dos planes de inspección a diferentes entidades en el mismo día y hora con los mismos participantes o dirigente.

R6 La disposición solo la puede confeccionar el jefe del departamento de regulación y control del CITMA.

R7 La disposición solo se puede confeccionar a partir de una planificación de inspección mensual ya establecida y debe ser entregada con tiempo de antelación al especialista municipal y a la entidad a inspeccionar a menos que la planificación sea de carácter extraordinaria y entonces el plan no requiera de disposición.

R8 El dictamen solo puede ser confeccionado por el jefe del departamento de regulación y control del CITMA y debe responder solamente a una entidad que haya sido previamente inspeccionada.

2.2.3 Modelo conceptual

Los modelos conceptuales ayudan a organizar, representar y almacenar el conocimiento. Se basan en un esquema de conceptos relacionados jerárquicamente mediante proposiciones o palabras de enlace. Permiten representar y compartir el conocimiento desde una perspectiva constructiva e integradora. Este explica los conceptos más

significativos en el dominio del problema, teniendo como cualidad esencial representar elementos del mundo real. (Los mapas conceptuales interactivos como recursos didácticos en el ámbito universitario. Revista Complutense De Educación, 2015)

El Modelo Conceptual elaborado para la presente investigación se observa en la ilustración que se presenta a continuación.

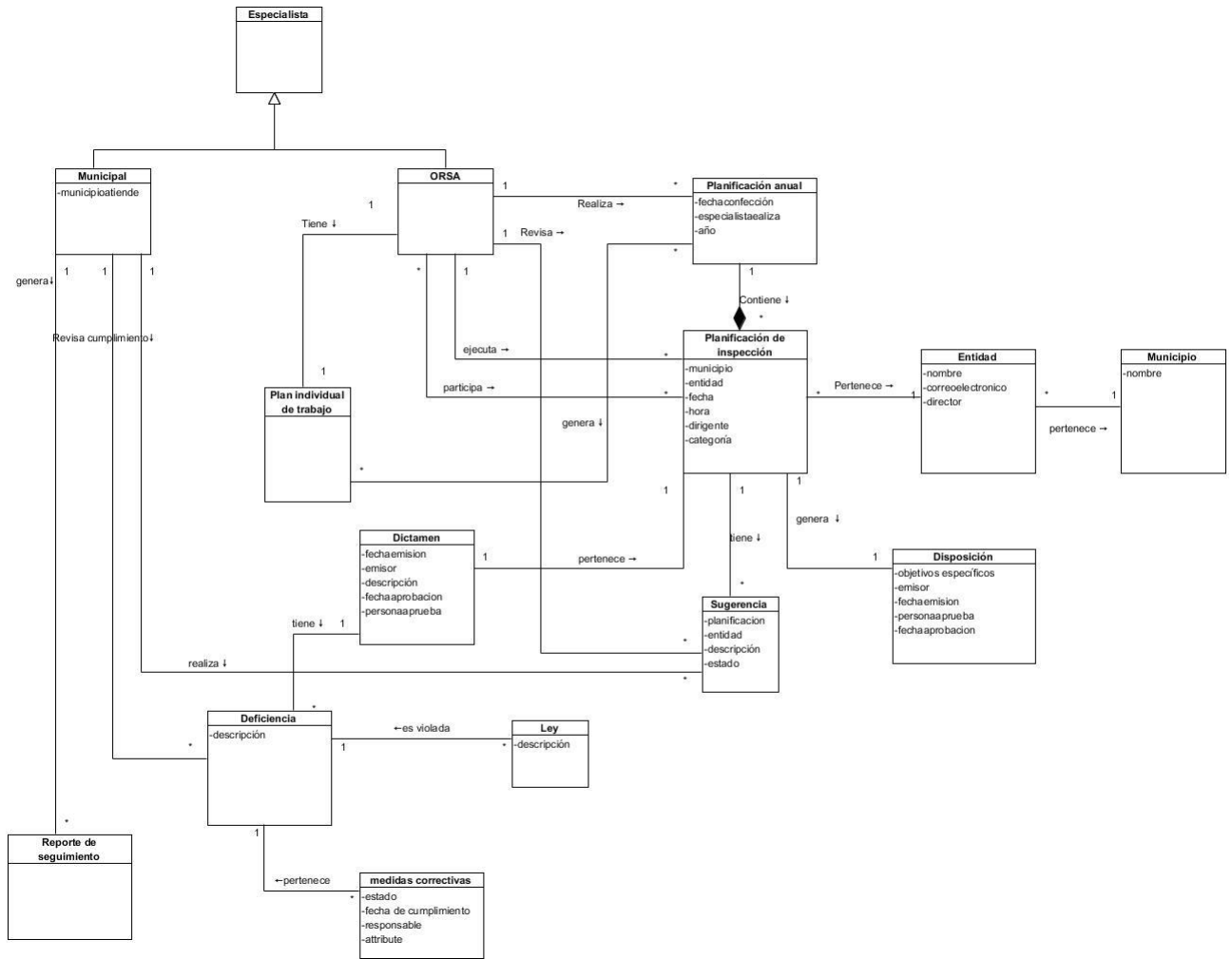


Figura 4 Diagrama del Modelo conceptual

2.3 Requisitos del sistema informático

2.3.1 Requisitos no funcionales

Los Requisitos No Funcionales establecidos para la solución propuesta son los siguientes:

Requisitos de usabilidad

- Contar con manuales de usuario estructurados adecuadamente.
- Proporcionar mensajes de error que sean informativos y orientados a usuario final.
- Ser fácil de utilizar por cualquier persona que tenga un conocimiento básico de computación.
- Implementarse lo más parecido posible a como se realiza el proceso en la actualidad para lograr una mejor comprensión y adaptación al mismo.

Requisitos de confiabilidad:

- Debe existir sistemas de respaldo eléctrico en los locales donde se encuentren los servidores.
- Deben evitarse fallos en la aplicación que colapsen el sistema y en caso de existir se deben establecer mecanismos que aseguren el reinicio del sistema de una forma rápida y eficiente.

Requisitos de seguridad y privacidad:

- El usuario debe autenticarse para acceder al sistema, dependiendo del nivel de acceso se presentarán las interfaces para cada usuario.
- La información que se maneje en el sistema estará protegida de acceso no autorizado y divulgación, a partir de los diferentes roles de los usuarios que empleen el sistema.
- La información existente en el sistema será protegida contra actos ilícitos, de igual manera el origen y fuente de los datos.

Portabilidad:

- El sistema será multiplataforma (Linux y Windows fundamentalmente).

2.3.2 Requisitos funcionales

Los Requisitos Funcionales (RF) identificados son los siguientes:

Gestionar objetivos de trabajo

- Adicionar objetivos de trabajo.
- Modificar objetivos de trabajo.
- Eliminar objetivos de trabajo.
- Buscar objetivos de trabajo.

Gestionar planificación

- Adicionar planificación.
- Modificar planificación.
- Eliminar planificación.
- Buscar planificación.
- Generar planificación.

Gestionar disposición

- Adicionar disposición.
- Modificar disposición.
- Eliminar disposición.
- Buscar disposición.
- Generar disposición.

Gestionar participantes de la inspección

- Adicionar participantes.
- Eliminar participantes.

Gestionar sugerencias para la planificación mensual

- Adicionar sugerencias para la planificación mensual.
- Modificar sugerencias para la planificación mensual
- Eliminar sugerencias para la planificación mensual
- Buscar sugerencias para la planificación mensual.
- Revisar sugerencias

Gestionar dictamen

- Adicionar dictamen de una inspección.
- Modificar dictamen de una inspección.
- Eliminar dictamen de una inspección.
- Buscar dictamen de una inspección
- Generar el documento dictamen de inspección

Gestionar medidas asociadas a una deficiencia.

- Adicionar medidas asociadas a una deficiencia
- Modificar medidas asociadas a una deficiencia.
- Eliminar medidas asociadas a una deficiencia.
- Buscar medidas asociadas a una deficiencia.

Gestionar deficiencias

- Adicionar deficiencias
- Modificar deficiencias
- Eliminar deficiencias
- Buscar deficiencias.

Nomencladores

Gestionar municipios

- Adicionar municipio.
- Modificar municipio.
- Eliminar municipio.

Gestionar entidades

- Adicionar entidades.
- Modificar entidades.
- Eliminar entidades.

Gestionar personas

- Adicionar personas.
- Modificar personas.
- Eliminar personas.

Gestionar Leyes, resoluciones, decretos leyes.

- Adicionar leyes
- Modificar leyes
- Eliminar leyes
- Buscar leyes

Reportes

1. Disposición para la ejecución de la inspección ambiental estatal ordinaria.
2. Dictamen de la inspección ambiental estatal.

2.3.3 Diagrama de actividades de los requisitos funcionales

El diagrama de actividades muestra cómo fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecución de un proceso más complejo. (Universidad Nacional Abierta y a Distancia, sf).

A continuación, se definen los diagramas de actividades de algunos requisitos arquitectónicamente significativos del sistema. Todos los diagramas de actividad desarrollados mediante el diseño de la solución se encuentran en el expediente de proyecto en la dirección: Expediente de proyecto\Gestión de Proyecto\Citma\Ejecución\Modelación\ArchivoVP\citma.vpp\Diagrama UML\Diagrama de actividad.

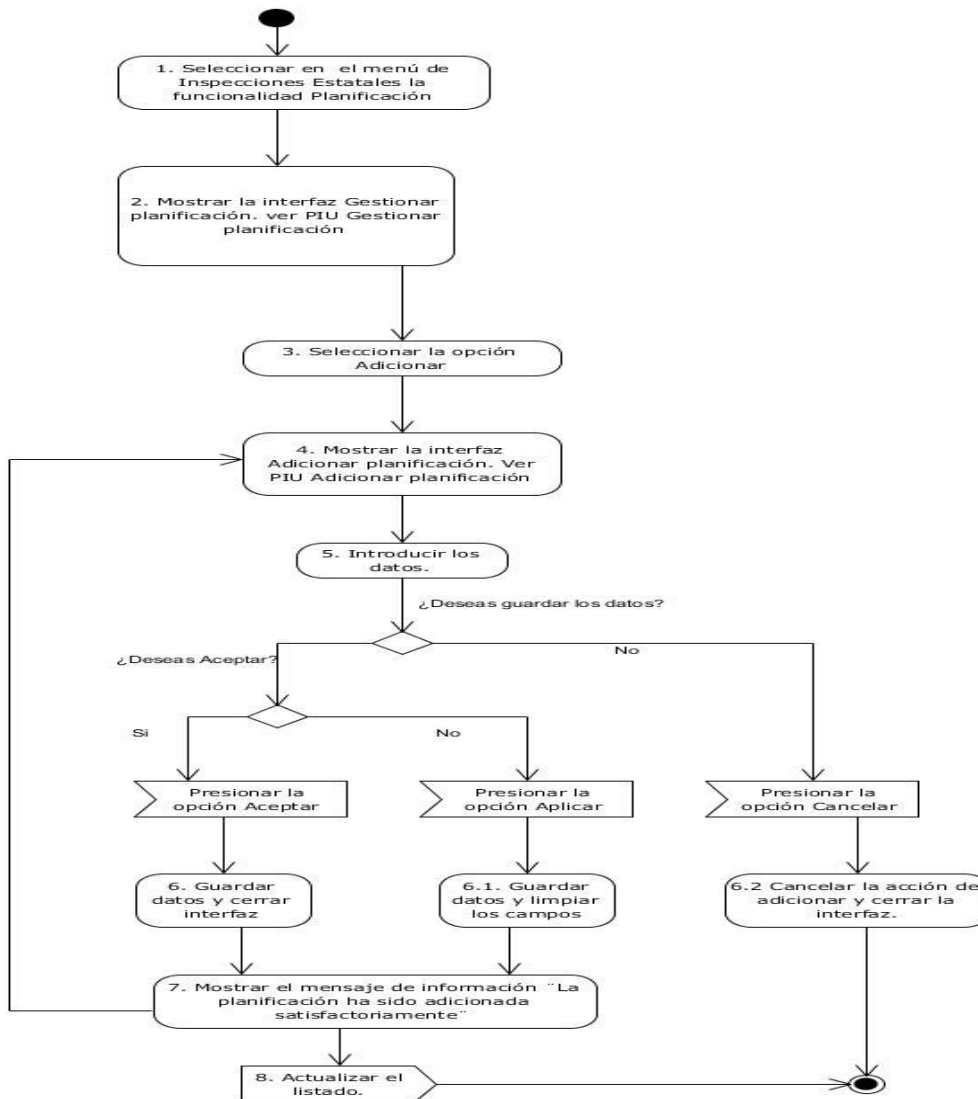


Figura 5 Diagrama de actividad del requisito funcional adicionar planificación

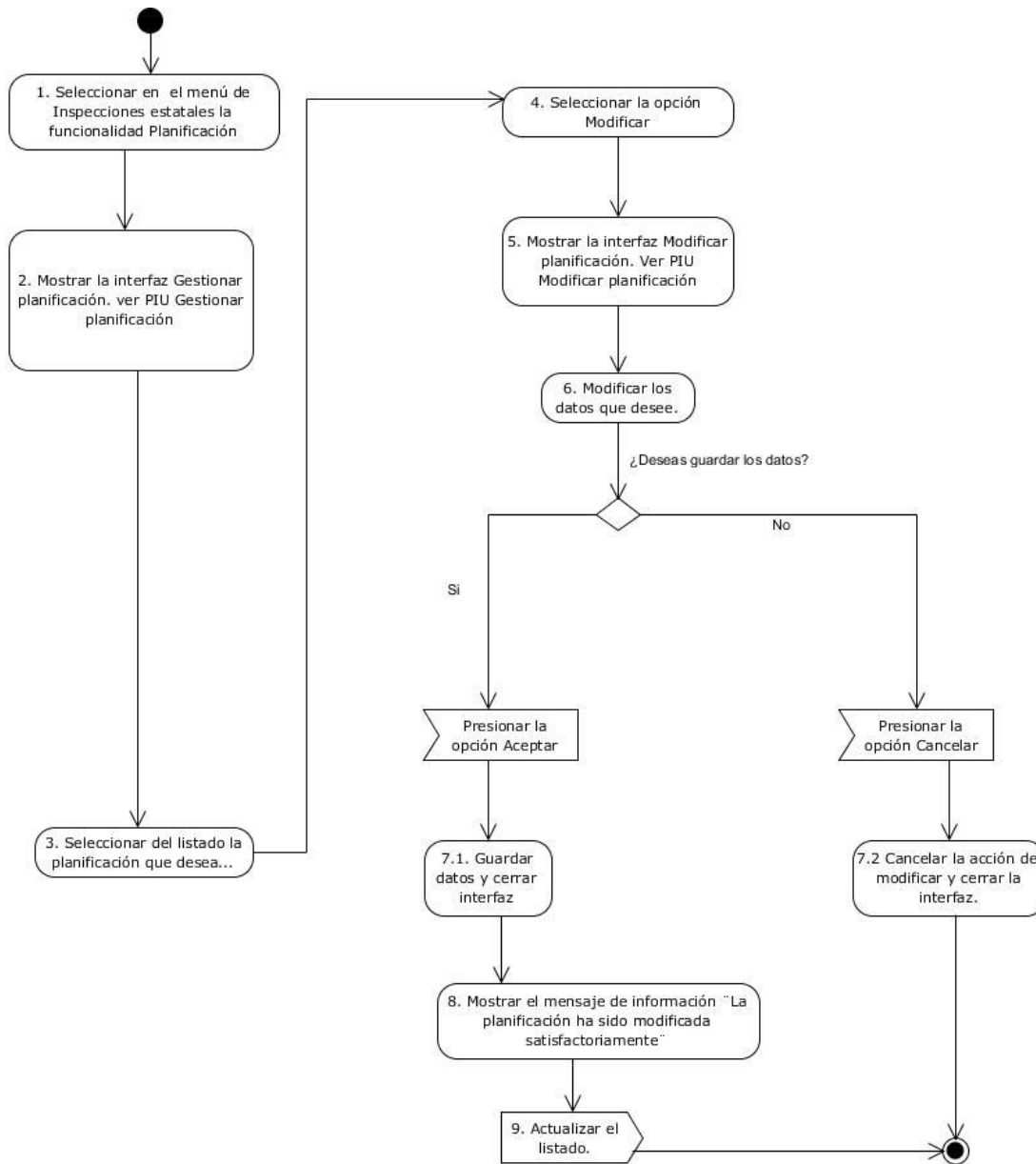


Figura 6 Diagrama de actividad del requisito funcional modificar planificación

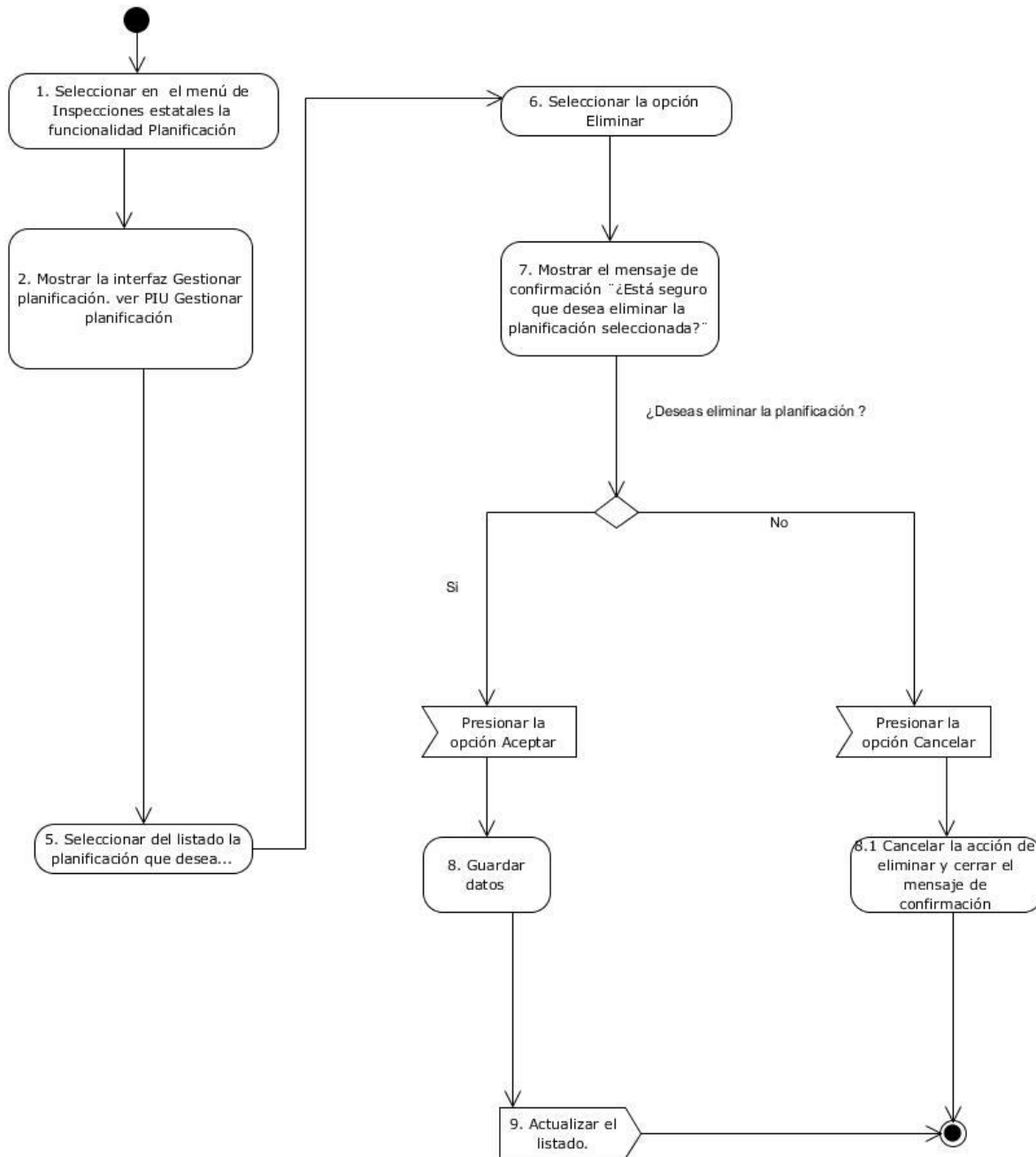


Figura 7 Diagrama de actividad del requisito funcional eliminar planificación

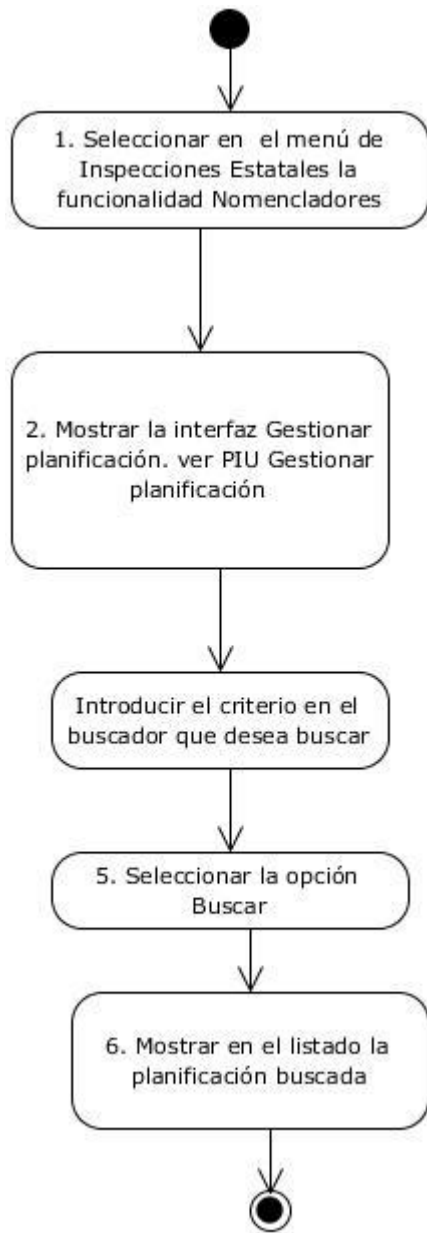


Figura 8 Diagrama de actividad del requisito funcional buscar planificación

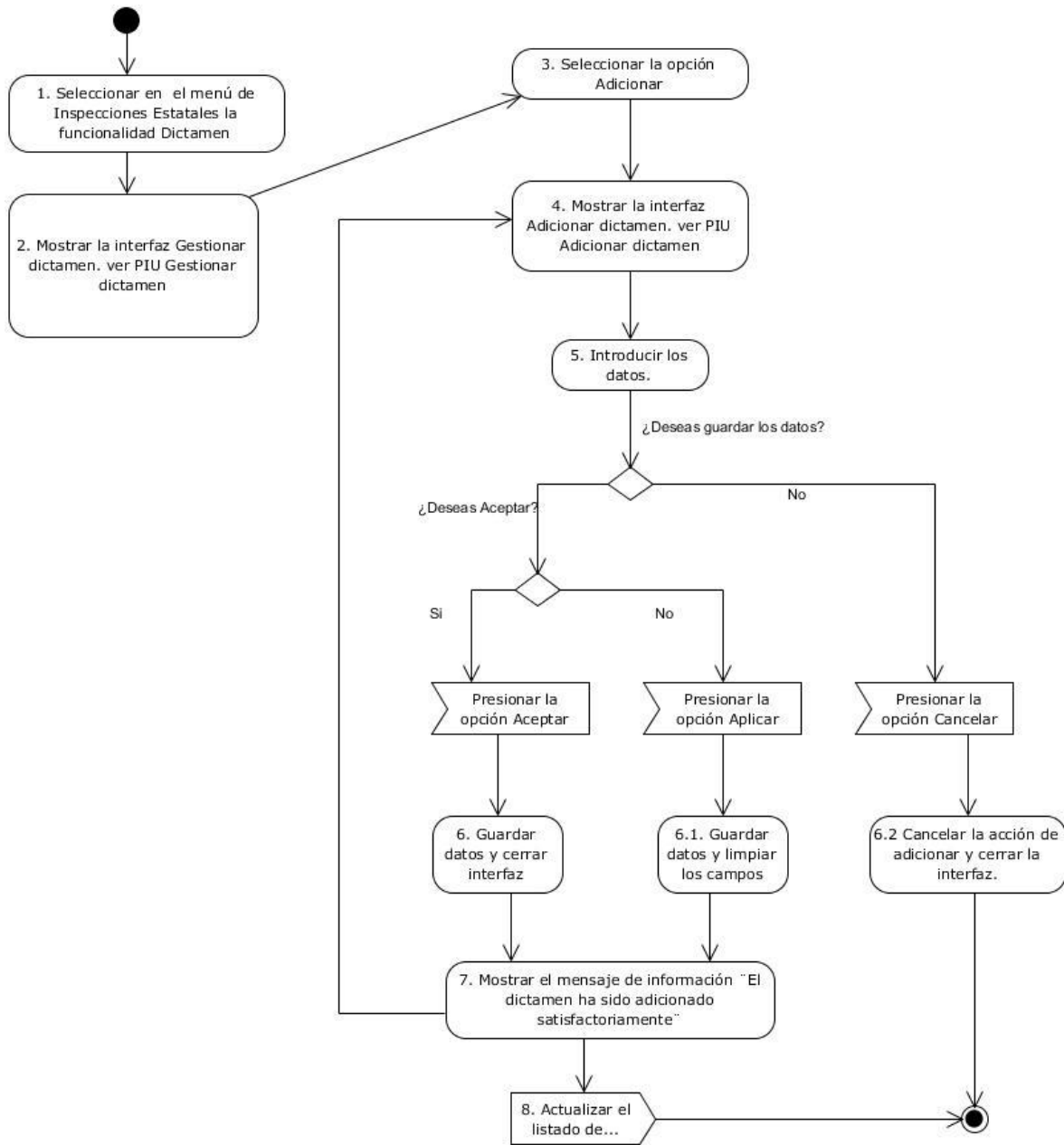


Figura 9 Diagrama de actividad del requisito funcional adicionar dictamen de inspección

2.4 Implementación de la arquitectura de software

La arquitectura del software define la estructura que debe de tener un software, las piezas a construir y el modo en el que se deben juntar y trabajar entre ellas. Se define a alto nivel mediante una serie de patrones y abstracciones que deben seguir para el desarrollo del software y para la interacción entre sus diversas piezas. (García, 2020)

La arquitectura corresponde a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificabilidad), y servir como guía en el desarrollo (Cervantes, sf)

La arquitectura utilizada por la empresa XETID es Cliente/Servidor, la cual es una aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones clientes realizan peticiones a una o varias aplicaciones servidores, que deben encontrarse en ejecución para atender dichas demandas. El modelo Cliente/Servidor permite diversificar el trabajo que realiza cada aplicación, de forma que los clientes no se sobrecarguen. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita el diseño del sistema. Tanto el cliente como el servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. ([Anexos | Anexo 1](#)) (Marini, 2012)

El esquema Cliente/Servidor posee las siguientes ventajas:

- Facilita el suministro de información a los usuarios, ya que por un lado proporciona una mayor consistencia a la organización de la información, al contar con un control centralizado de los elementos compartidos, y por otro, porque facilita la construcción de interfaces gráficas interactivas, las cuales pueden hacer que los datos se conviertan en información.
- Favorece la adaptación a cambios en la tecnología, pues facilita la migración de las aplicaciones a otras plataformas, y al aislar claramente las diferentes funciones de una aplicación, hace más fácil incorporar nuevas tecnologías en esta.

- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando libre la red.

La arquitectura N-Capas

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se modifica el nivel requerido sin tener que revisar entre código mezclado. Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API (Interfaz de programación de aplicaciones) que existe entre niveles. (Administración de Base de Datos, 2012)

Para la realización del proyecto será necesaria la implementación de plugin, los cuales no son más que elementos externos que ayudan a potenciar el desarrollo de un proceso, es decir son funcionalidades que en ProcessMaker no se permiten implementar y que son necesarias para el desarrollo de los procesos. Los plugin no interfieren en el funcionamiento del sistema porque se pueden añadir o quitar y este va a continuar funcionando sin ningún problema. Estos, utilizan un patrón arquitectónico modelo-vista-controlador, el cual se convierte en una arquitectura N-Capas, al añadirseles nuevas capas, donde en la capa de presentación se utilizó EXT JS, Zend Framework en la capa de negocio y Postgre SQL en la capa de datos. Además, se adicionan a estos una capa de configuración y otra de acceso a datos utilizando Doctrine. Se implementó la seguridad RBAC o autenticación basada en roles para el acceso a las funcionalidades. ([Anexos | Anexo 3](#))

2.5 Patrones de diseño

El desarrollo de software es un proceso complejo, en especial si se demanda una solución rápida, adaptable y estable, lo que conlleva enfrentar retos continuamente. Con la utilización de patrones de diseño se puede evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.

Un patrón de diseño nombra, abstrae e identifica los aspectos fundamentales de una estructura común de diseño que la hacen útil para la creación de un diseño orientado a objetos reusable (Gamma, y otros, 1995)

A continuación, se presentan algunos patrones de diseño los cuales se sugieren su implementación para la realización de un sistema optimizado y cumpliendo con buenas prácticas de desarrollo.

Modelo Vista Controlador (MVC): Es un patrón de diseño arquitectónico de software, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las entradas y la salida del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes para crear un sitio es muy fácil generar confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes sin necesidad de afectar a los demás. (García, 2017) ([Anexos | Anexo 4](#))

Singleton: Este patrón se encuentra dentro de los patrones de tipo creación, que son aquellos que abstraen el proceso de creación de instancias de los clientes que lo utilizan. Gracias al patrón Singleton, se puede hacer un sistema independiente de la forma en la que se crean las instancias, además, se hace independiente de la forma en la que se componen y representan las mismas. Esto es debido a que oculta, a los clientes, el proceso de creación y asociación de las instancias. El patrón Singleton es aplicable en cuando, únicamente debe de existir una instancia de una clase y debe ser accesible desde un punto conocido. La instancia única debe ser extensible por los clientes mediante herencia sin necesidad de modificar su código (Time of Software, 2016)

Constructor: Como Patrón de diseño, el Constructor, es clasificado como creacional y es usado para permitir la creación de una variedad de objetos complejos desde un objeto fuente (Producto). El objeto fuente se compone de una variedad de partes que contribuyen individualmente a la creación de cada objeto complejo a través de un conjunto de llamadas a interfaces comunes de la clase constructor-abstracto. Tiene como ventajas que el algoritmo para la creación de objetos complejos sea

independiente de las partes que construyen el objeto y que el proceso de construcción pueda tener diferentes representaciones para el objeto que está construido. (ANDALUCÍA)

Role Based Access Control (RBAC): es un enfoque basado en el concepto de rol para implementar políticas de control de acceso y su intención es lograr un control de acceso más granular, donde las asignaciones usuario-rol y permiso-rol permiten que un usuario pertenezca a varios roles simultáneamente y que un rol posea múltiples usuarios.

2.6 Mecanismo de diseño

El mecanismo de diseño resume cómo construir el diseño de las clases que se pueden implementar en el software, se visualizan las relaciones entre ellas y se muestra gráficamente la interacción de los objetos para comunicarse entre sí. Su propósito es representar los diagramas de paquetes, así como los diagramas de clases y la descripción de las mismas. (Proceso de Desarrollo y Gestión de Proyectos de Software, 2012)

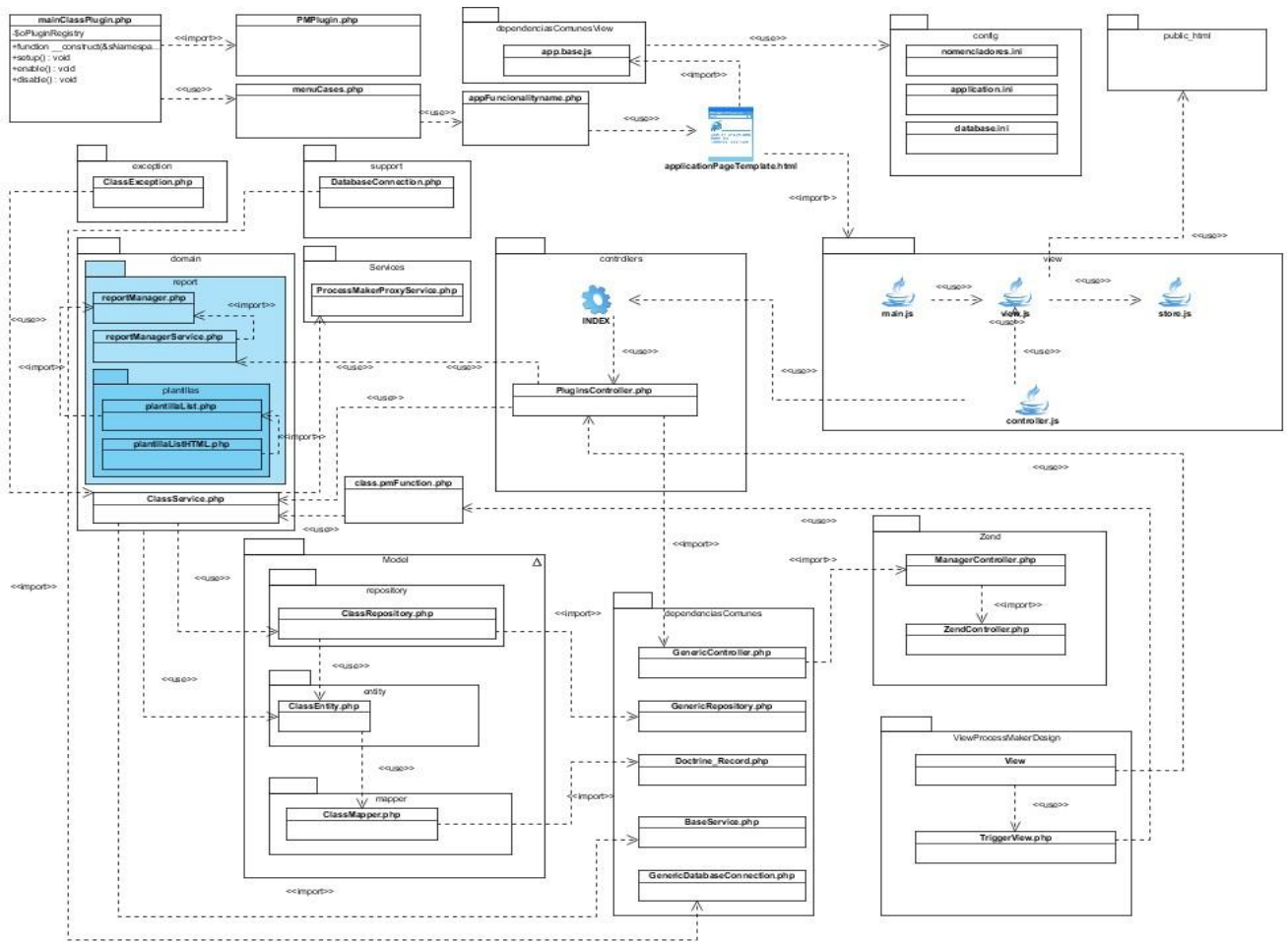


Figura 10 Diagrama de Mecanismo de diseño

2.7 Diseño de la base de Datos

Teniendo como entrada el modelo conceptual, la especificación de la arquitectura de sistemas y la especificación de los requisitos de software se diseñan las tablas, sus atributos y relaciones agrupados por componentes ya sea por paquetes o colores delimitando cada uno de ellos, obteniendo como resultado el modelo de datos. (Proceso de Desarrollo y Gestión de Proyectos de Software, 2012)

A continuación se presenta el modelo lógico obtenido del análisis de los conceptos asociados al problema identificado:

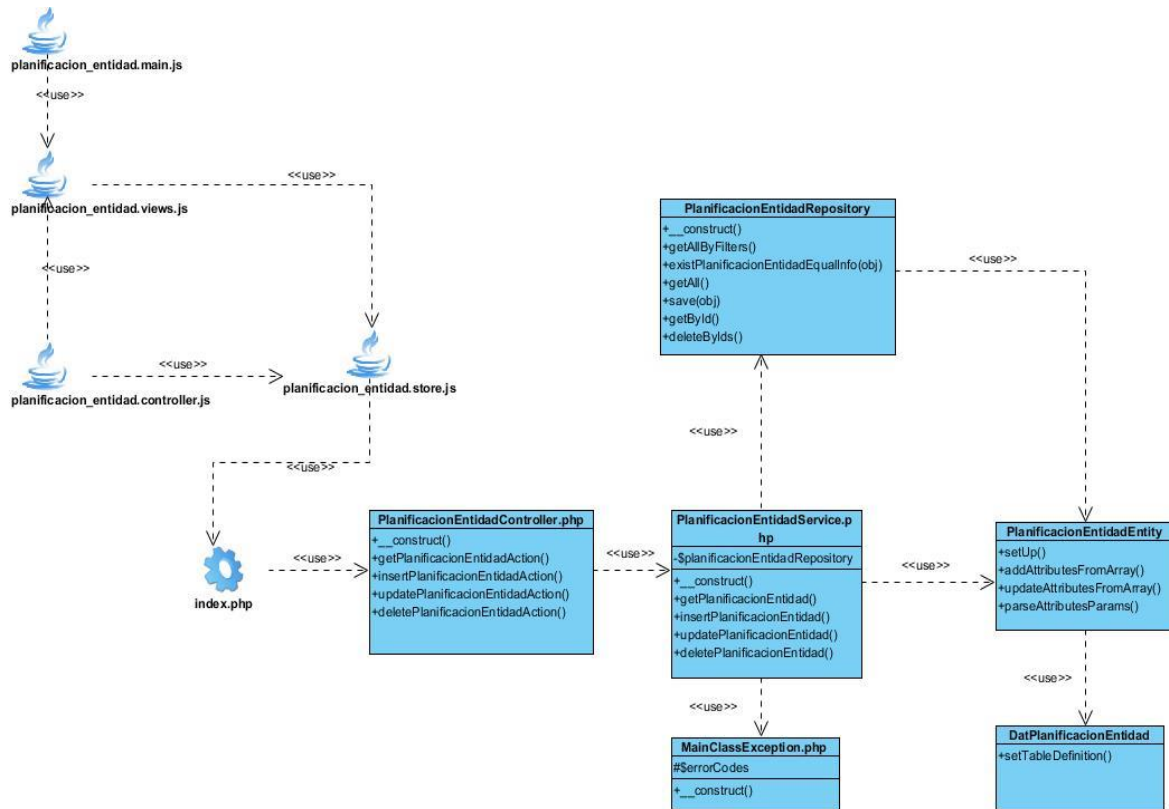


Figura 13 Diagrama de clases de diseño del requisito funcional gestionar planificación de entidad

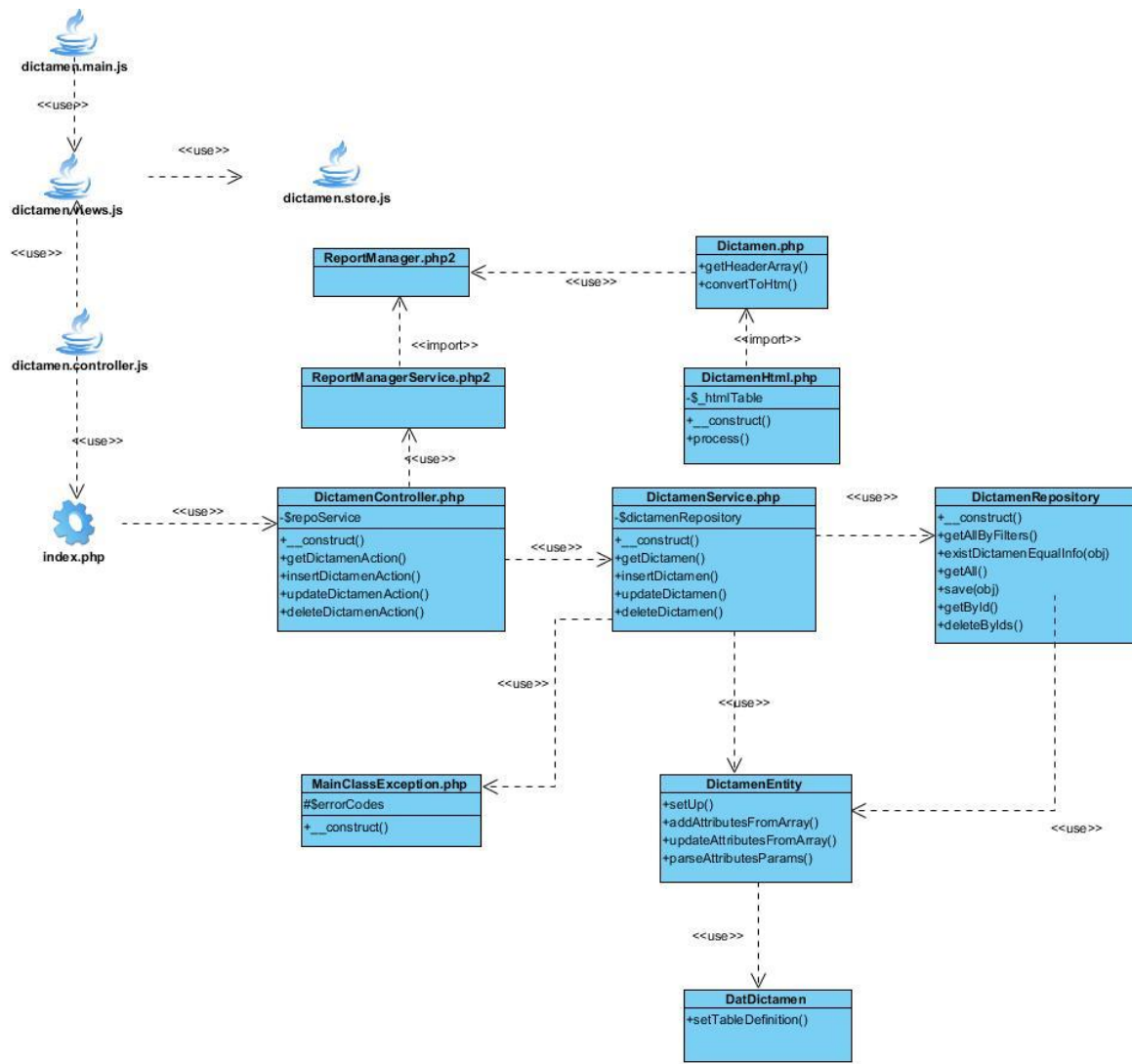


Figura 14 Diagrama de clases de diseño del requisito funcional generar dictamen

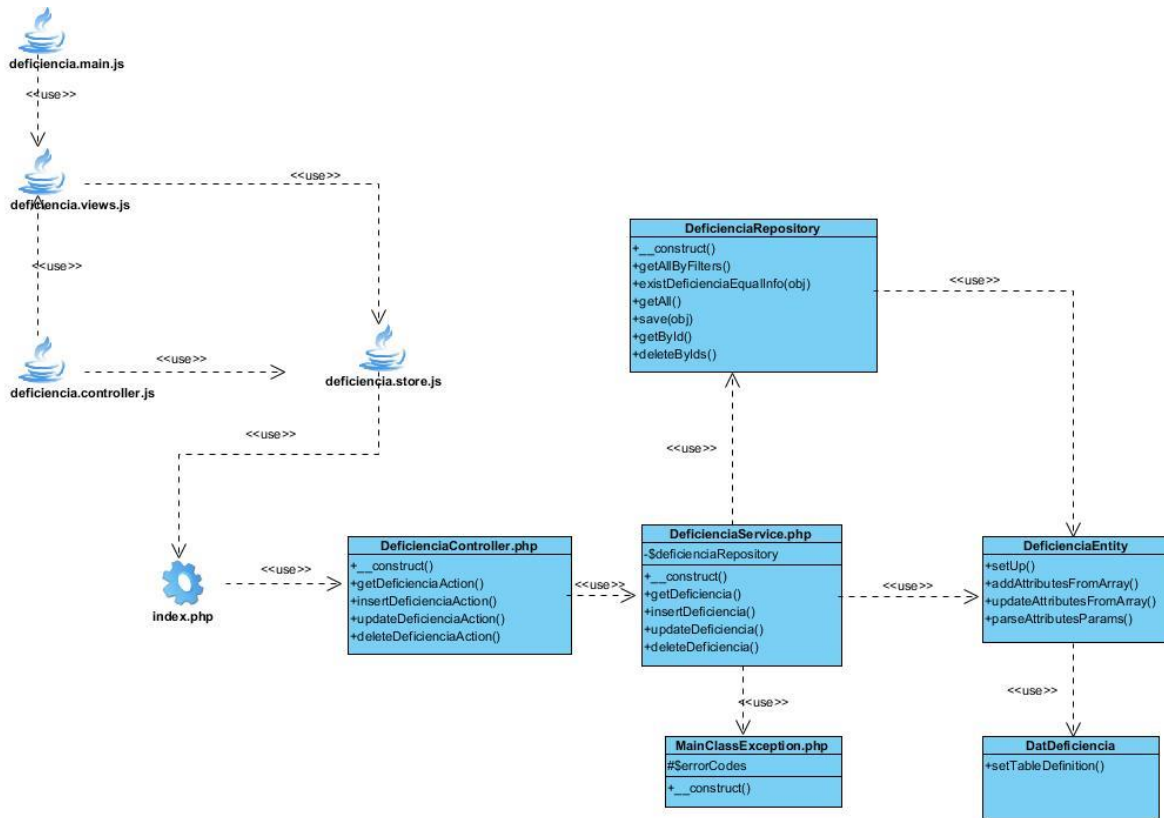


Figura 15 Diagrama de clases de diseño del requisito funcional gestionar deficiencias

2.9 Diagramas de secuencia

Los diagramas de secuencia son aquellos diagramas que permiten mostrar la interacción entre los usuarios, la interfaz y las instancias de los objetos en el sistema. Provee un mapa secuencial del paso de los mensajes entre los objetos a lo largo del tiempo. (Proceso de Desarrollo y Gestión de Proyectos de Software, 2012)

A continuación, se presentan un conjunto de diagramas de secuencia de algunos requisitos funcionales que son considerados arquitectónicamente significativos. Todos los diagramas de secuencia desarrollados mediante el análisis y diseño de la solución se encuentran en el documento anexo de evidencias.

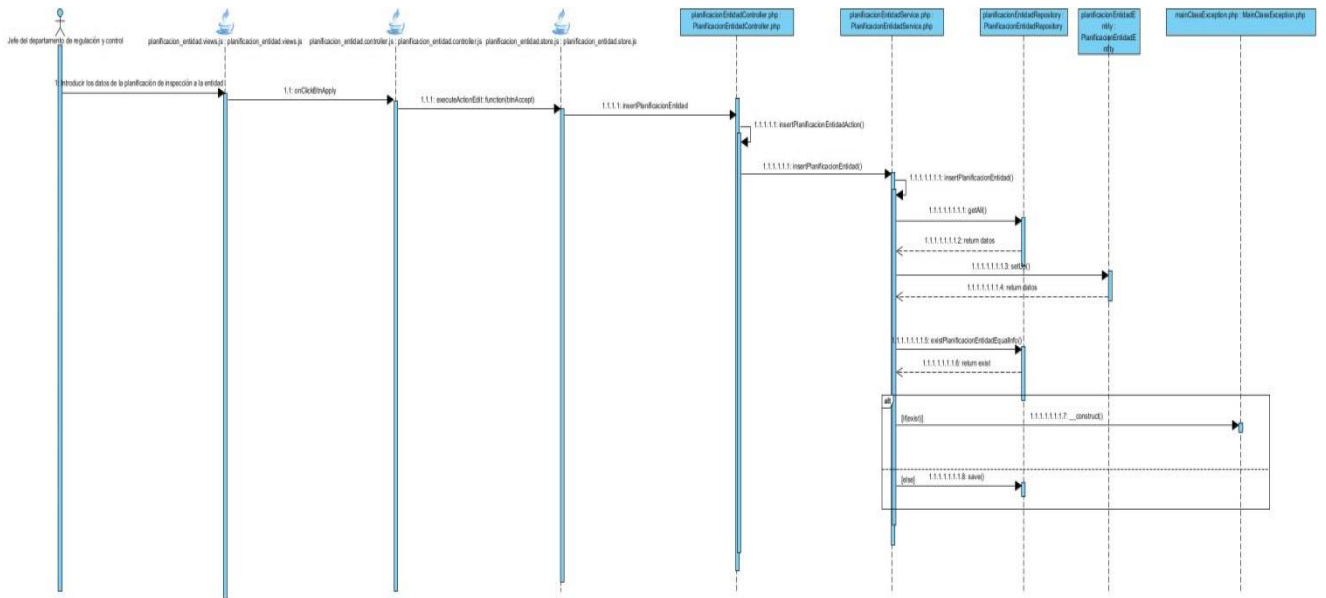


Figura 16 Diagrama de secuencia adicionar plan de inspección a entidad

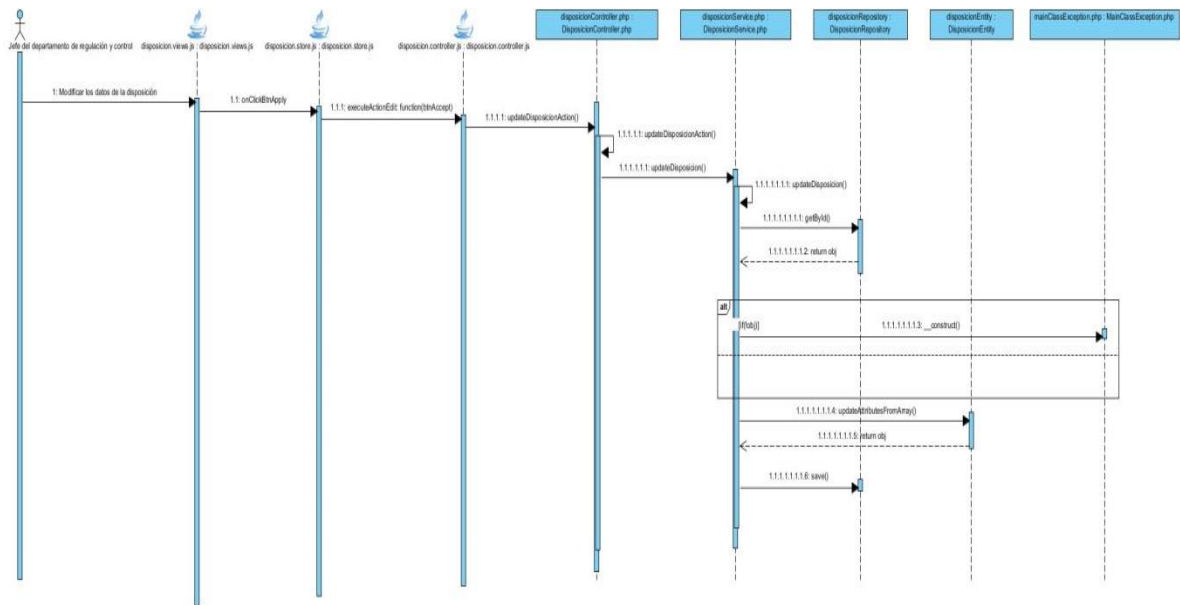


Figura 17 Diagrama de secuencia modificar disposición

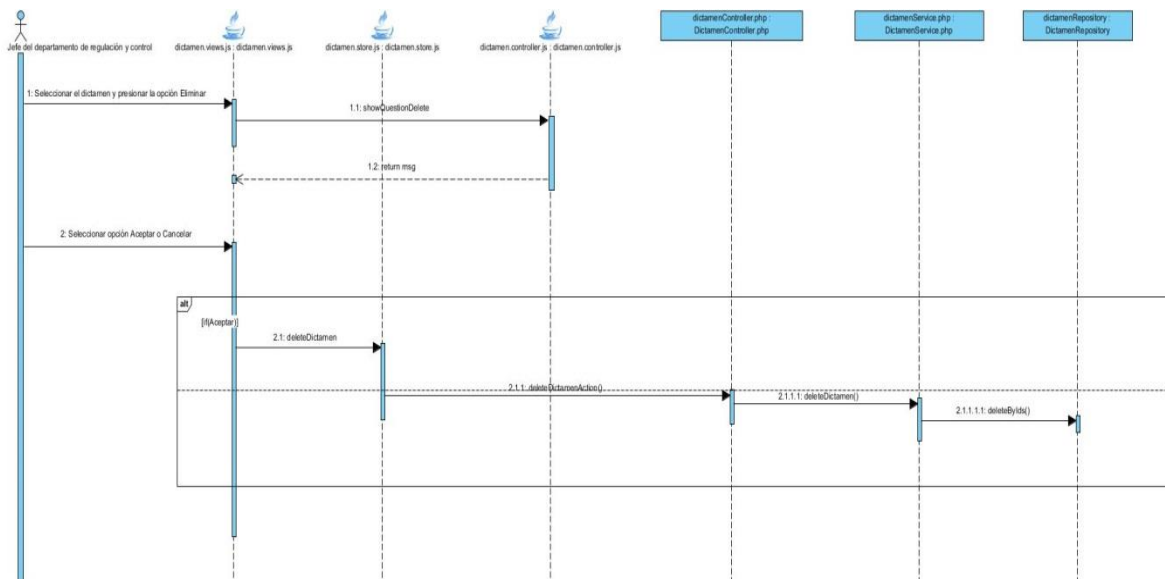


Figura 18 Diagrama de secuencia eliminar dictamen

2.10 Seguridad

La seguridad de software se encarga de proteger las aplicaciones y el software de amenazas exteriores. El sistema deberá implementar diversas políticas de seguridad para garantizar la confidencialidad, integridad, disponibilidad y la privacidad de la información que gestione.

El mecanismo de seguridad utilizado para ello es una seguridad RBAC o control de acceso basado en roles, consiste en asignar derechos de acceso a los usuarios de su organización en función de sus roles y las tareas que realizan. Esto garantiza que los usuarios y equipos solo puedan tener acceso a los niveles que pertenezcan.

A través del estilo arquitectónico N-Capas, se le agregó al patrón de diseño modelo-vista-controlador una capa de seguridad transversal en la cual se encuentra RBAC.

Por otra parte, el framework utilizado para el acceso a los datos, en este caso Doctrine, al igual que la mayoría de los ORM (mapeador de objetos relacional), presenta su propia capa de seguridad para la protección contra los ataques más comunes, como son las inyecciones SQL.

2.11 Análisis de factibilidad

El resultado de los estudios de factibilidad de los trabajos de investigación es la base de las decisiones que se tomen para su introducción, por lo que deben ser lo suficientemente precisas para evitar errores que tienen un alto costo social directo en cuanto a los medios materiales y humanos que involucren; así como por la pérdida de tiempo en la utilización de las variantes de desarrollo más eficientes para la sociedad. Para el análisis del costo y beneficios del sistema en esta investigación se utilizó la metodología de Prodesoft donde desde el inicio se estima de forma empírica la duración de la implementación de cada uno de los requisitos, basado en la experiencia del programador en el trabajo con el lenguaje de programación, el entorno de desarrollo, el conocimiento sobre el tema de investigación y las técnicas de programación necesarias para resolver el problema. Para esto es necesario realizar una estimación del esfuerzo, el tiempo de desarrollo de cada requisito y la cantidad de personas que participarán para poder determinar eficazmente si resulta beneficioso el desarrollo, aunque es importante señalar que solo se habla de una estimación. Esta metodología guía un proceso basado en componentes lo que permite ahorrar gran cantidad de tiempo.

2.11.1 Costo

El análisis de costo es el proceso de identificación de los recursos necesarios para llevar a cabo el trabajo o proyecto eficientemente. Para la estimación del costo del software se empleará la fórmula propuesta por la metodología, para lo cual primeramente se tiene en cuenta los requisitos funcionales a desarrollar, así como su prioridad, complejidad y tiempo estimado de desarrollo, lo que se detalla en la siguiente tabla.

Tabla 1 Análisis de costo

No	Nombre de requisito	Prioridad	Complejidad	Tiempo de desarrollo (semanas)
1	Gestionar objetivos de trabajo	Baja	Baja	2

2	Gestionar planificación	Alta	Alta	3
3	Gestionar disposición	Media	Media	3
4	Gestionar participantes de la inspección	Baja	Baja	2
5	Gestionar sugerencias para la planificación mensual	Baja	Baja	2
6	Gestionar dictamen	Media	Media	3
7	Gestionar medidas asociadas a una deficiencia	Media	Media	3
8	Gestionar deficiencias	Alta	Alta	3
9	Gestionar Leyes, resoluciones, decretos leyes	Media	Media	3

Para la estimación del costo del software se empleó la fórmula propuesta por la metodología, donde se determina:

Costo diario de un trabajador= Tarifa horaria (MT) * Trabajador * Tiempo diario. = \$18.00 * 1 trabajador * 8 horas.
= \$144.00.

Tomando en cuenta que un trabajador cubano, trabaja 20 días laborables
= \$144.00. *20 = \$2 880.00.

Costo mensual de un trabajador = \$2 880.00.

Teniendo en cuenta un tiempo estimado de aproximadamente 6 meses para el desarrollo del proyecto se determina:

Costo total= \$2 880.00.* 6 meses.

= \$ 17 280.00

El costo de desarrollo del sistema estimado es de \$ 17 280.00 aproximadamente, lo que en materia económica constituye una cifra moderada de dinero con respecto a los beneficios que se evidencian a continuación.

2.11.2 Beneficios tangibles e intangibles

El diseño de la solución propuesta para la gestión de los procesos de planificación e inspección medio ambientales de la empresa CITMA permitirá que disminuya el gasto de materiales y medios utilizados en la actualidad en estos procesos que se realizan de forma manual. La gestión de los datos posibilitará una mejor organización de la información, logrando de forma segura la integridad, extracción, manipulación y persistencia de los datos. A través del sistema se realizarán planificaciones más óptimas disminuyendo así el coste de recursos empleados para la comunicación entre los especialistas municipales y los responsables designados por la empresa para realizar dichas planificaciones, así como para optimizar el coste de recursos en el traslado a los municipios al tener un diseño más claro en la planificación de las entidades a inspeccionar.

2.11.3 Análisis del costo y beneficios

Después de la estimación de costos y el análisis de los beneficios tangibles e intangibles, se llega a establecer una correlación entre ambos y arribar a la conclusión de que será factible el desarrollo e implementación de un sistema de gestión de los procesos de planificación e inspección medio ambiental para la empresa CITMA de la provincia de Matanzas.

2.12 Diseño de pruebas de software

Las pruebas de software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estado del desarrollo del sistema, desde su creación hasta su puesta en producción. Lo interesante de las pruebas es que se puedan ejecutar de manera automática, para determinar en cualquier momento si se tiene una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin poder identificar dichos cambios. (Turrado, 2020)

Las pruebas de software tienen como objetivo comprobar el correcto funcionamiento de los sistemas informáticos y detectar la mayor cantidad de errores posibles en la implementación de requisitos funcionales, además de asegurar que estos errores encontrados sean corregidos.

2.12.1 Tipos de pruebas de software

Todos los tipos de pruebas de software que existen, básicamente, se pueden agrupar en dos grupos: las pruebas funcionales y las pruebas no funcionales.

Las pruebas funcionales se definen teniendo como fuente los requisitos del sistema, estas pruebas validan y verifican que el producto cumple con lo especificado, hace lo que debe y cómo lo tiene que hacer obteniéndose también una idea del grado de calidad del software. (testerhouse, 2019)

Tipos de pruebas funcionales:

- Pruebas unitarias.
- Pruebas de aceptación.
- Pruebas de integración.
- Pruebas de regresión.

Las pruebas no funcionales se centran en aspectos muy importantes del comportamiento del producto pero que no están relacionados con las funciones que realiza el sistema. (testerhouse, 2019)

Tipos de pruebas no funcionales:

- Pruebas de carga.
- Pruebas de estrés.
- Pruebas de escalabilidad.
- Pruebas de portabilidad.

Dentro de las pruebas funcionales se empleó la técnica de caja negra que permite exponer mediante los diseños de casos de prueba la funcionalidad de los requisitos funcionales en el sistema y así probar su correcto funcionamiento y satisfaciendo las necesidades del cliente.

Las **Pruebas de Caja Negra**, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

Las pruebas de caja negra, se enfocan solamente en las entradas y salidas del sistema, sin tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles deben ser esas entradas y salidas, se basan en los requerimientos de software y especificaciones funcionales (Terrera, 2017)

2.12.2 Diseños de casos de prueba

A continuación, se presentan una serie de diseños de casos de pruebas que se realizaron en algunos requisitos funcionales que son considerados arquitectónicamente significativos. Todos los diseños de casos de pruebas se encuentran en el documento anexo de evidencias..

Tabla 2 diseño de caso de prueba para el requisito funcional adicionar planificación

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario	Resultados esperados
<i>Adicionar planificación</i>	<i>Permite adicionar una planificación de inspección al listado de planificaciones</i>	<i>EP 1.1: Adicionar planificación correctamente</i>	1. Seleccionar en el menú de Inspecciones Estatales la funcionalidad Planificación 2. Mostrar la interfaz Gestionar planificación. ver PIU Gestionar planificación	1. Cerrar la ventana Adicionar planificación y mostrar el mensaje de información "La planificación ha sido adicionada satisfactoriame

		<p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar planificación. Ver PIU Adicionar planificación</p> <p>5. Introducir los datos.</p> <p>6. Presionar la opción Aceptar</p>	<p>nte".</p> <p>2. Adicionar la planificación y actualizar el listado de planificaciones.</p>
	<p><i>EP 1.2 Adicionar planificación oprimiendo el botón Aplicar.</i></p>	<p>1. Seleccionar en el menú de Inspecciones Estatales la funcionalidad Planificación</p> <p>2. Mostrar la interfaz Gestionar planificación. ver PIU Gestionar planificación</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar planificación. Ver PIU Adicionar planificación</p> <p>5. Introducir los datos.</p> <p>6. Presionar la opción Aplicar</p>	<p>1. Mostrar el mensaje de información: "La planificación se ha guardado satisfactoriamente", se actualiza el listado y limpiar los campos de la interfaz, permitiendo adicionar otra disposición.</p>
	<p><i>EP 1.3: Adicionar planificación incorrectamente.</i></p>	<p>1. Seleccionar en el menú de Inspecciones Estatales la funcionalidad Planificación</p> <p>2. Mostrar la interfaz Gestionar planificación. ver PIU Gestionar planificación</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz</p>	<p>1. Mostrar un mensaje de error en los tooltips de los campos según corresponda.</p>

			<p>Adicionar planificación. Ver PIU Adicionar planificación</p> <p>5. Introducir los datos.</p> <p>6. Presionar la opción Aceptar</p>	
		<p><i>EP 1.4</i> <i>Oprimir botón</i> <i>Cancelar</i></p>	<p>1. Seleccionar en el menú de Inspecciones Estatales la funcionalidad Planificación</p> <p>2. Mostrar la interfaz Gestionar planificación. ver PIU Gestionar planificación</p> <p>3. Seleccionar la opción Adicionar</p> <p>4. Mostrar la interfaz Adicionar planificación. Ver PIU Adicionar planificación</p> <p>5. Introducir los datos.</p> <p>6. Presionar la opción Cancelar</p>	<p>1. Cancelar todas las acciones, cerrar la ventana, y regresar a la interfaz principal.</p>

Tabla 3 Diseño de caso de prueba para el requisito funcional aprobar disposición

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario	Resultados esperados
<i>Aprobar disposición</i>	<i>Permite que un usuario con los permisos requeridos pueda aprobar una disposición correspondiente a una planificación de inspección</i>	<i>EP 1.1: Aprobar disposición correctamente</i>	<p>1. Seleccionar en el menú de Inspecciones estatales la funcionalidad Disposición</p> <p>2. Mostrar la interfaz Gestionar disposición. Ver PIU Gestionar disposición</p> <p>3. Seleccionar del listado la disposición</p>	<p>1. Cerrar la ventana Aprobar disposición y mostrar el mensaje de información "La disposición ha sido aprobada satisfactoriamente".</p> <p>2. Aprobar la disposición y</p>

		<p>que desea aprobar</p> <p>4. Seleccionar la opción Aprobar</p> <p>5. Mostrar el mensaje de confirmación " ¿Está seguro que desea aprobar la disposición seleccionada? "</p> <p>6. Presionar la opción Aceptar</p>	<p>actualizar el listado de disposiciones.</p>
	<p><i>EP 1.2: Aprobar disposición y presionar el botón Cancelar</i></p>	<p>1. Seleccionar en el menú de Inspecciones estatales la funcionalidad Disposición</p> <p>2. Mostrar la interfaz Gestionar disposición. Ver PIU Gestionar disposición</p> <p>3. Seleccionar del listado la disposición que desea aprobar</p> <p>4. Seleccionar la opción Aprobar</p> <p>5. Mostrar el mensaje de confirmación " ¿Está seguro que desea aprobar la disposición seleccionada?"</p> <p>6. Presionar la opción Cancelar</p>	<p>1. Cancelar todas las acciones, cerrar la ventana, y regresar a la interfaz principal.</p>

Se propone que se realicen dentro de las pruebas no funcionales, las pruebas de carga y estrés que permiten conocer el rendimiento real del sistema:

Pruebas de carga

Estas pruebas se hacen con el objetivo de determinar y validar la respuesta de la aplicación cuando está sometida a una carga de un cierto número de usuarios o de peticiones.

Pruebas de estrés

Estas pruebas se realizan para encontrar el número de usuarios, peticiones o tiempos que la aplicación puede soportar. Este tipo de pruebas no funcionales son muy semejantes a las pruebas de carga y rendimiento, pero se diferencian en que deben superar los límites esperados en el ambiente de producción o los límites que fueron determinados en las pruebas. (testerhouse, 2019)

2.13 Conclusiones parciales

Luego de describir la propuesta de solución al problema científico de esta investigación y el análisis de las pruebas de software sugeridas se concluye que:

- La realización del modelado del proceso permitió un mayor entendimiento del negocio.
- El levantamiento de los requisitos funcionales y su descripción fue fundamental para lograr un diseño más apropiado para la realización de la solución propuesta.
- La estimación del costo del software junto con el estudio de la relación beneficio-costo demuestra la factibilidad del desarrollo de la solución propuesta.
- Se analizaron las pruebas de software, sus objetivos y su importancia en el desarrollo de sistemas informáticos y por supuesto su impacto en el sistema a desarrollar.
- Se mostraron diseños de casos de prueba que verifican algunas de las principales funcionalidades del sistema, estas muestran cuales son las

operaciones que se espera que realice el sistema y facilitará la posterior creación de diferentes tipos de pruebas de software más exhaustivas.

Conclusiones generales

Con el desarrollo de esta investigación podemos concluir que se cumplió con los objetivos trazados a partir de:

- El estudio realizado sobre los antecedentes, el estado actual de la temática, la bibliografía y documentos relacionados con el objeto de estudio, permitió contar con los elementos necesarios para dar solución a la problemática planteada.
- Se realizó un estudio acerca de las herramientas y tecnologías a utilizar, determinando así sus eficiencias y deficiencias.
- Se realizó la estimación de costo de implementación del sistema y el estudio de factibilidad, arrojando como resultado la factibilidad de la realización del sistema informático.
- Se realizó el modelado del negocio teniendo en cuenta los pasos que describe la metodología Prodesoft para su realización.
- La aplicación de las pruebas de validación, con resultados satisfactorios, demostraron que el sistema propuesto cumplirá con los requerimientos especificados por el cliente.

Recomendaciones

Para un mayor alcance del proyecto y continuidad de la investigación realizada se recomienda:

- La implementación del sistema informático del proceso de planificación e inspección medio-ambiental para la empresa CITMA en la provincia de Matanzas.
- La validación del software informático.
- La incorporación de nuevos requisitos que sean de utilidad al usuario.

Bibliografía

- Administracion de Base de Datos. 2012.** iutll-abdd.blogspot.com. *iutll-abdd.blogspot.com*. [En línea] Mayo de 2012. [Citado el: 20 de Mayo de 2020.] <http://iutll-abdd.blogspot.com/2012/05/arquitectura-de-n-capas.html>.
- ANDALUCÍA.** Marco de Desarrollo de la Junta Andalucía . *Marco de Desarrollo de la Junta Andalucía* . [En línea] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/185>.
- Cervantes, Humberto. sf.** *sg.com.mx. sg.com.mx*. [En línea] sf. [Citado el: 20 de Mayo de 2020.] <https://sg.com.mx/revista/27/arquitectura-software>.
- Doctrine. sf.** *www.doctrine-project.org. www.doctrine-project.org*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://www.doctrine-project.org/projects/doctrine-orm/en/current/tutorials/getting-started.html>.
- Domínguez, L Eduardo. 2018.** *www.cubadebate.cu. www.cubadebate.cu*. [En línea] 24 de Agosto de 2018. [Citado el: 1 de Mayo de 2020.] <http://www.cubadebate.cu/noticias/2018/08/24/cinco-provincias-cuentan-con-un-portal-del-ciudadano/>.
- El Espectador. 2012.** *www.elespectador.com. www.elespectador.com*. [En línea] 6 de Febrero de 2012. [Citado el: 18 de Febrero de 2020.] <https://www.elespectador.com/tecnologia/el-gobierno-electronico-y-sirve-articulo-325179>.
- Gamma, Erich, y otros. 1995.** *Design Patterns. Elements of reusable object-oriented software*. 1995.
- García, Jose M. Baquero. 2020.** *www.arsys.es. www.arsys.es*. [En línea] 3 de Febrero de 2020. [Citado el: 20 de Mayo de 2020.] <https://www.arsys.es/blog/arquitectura-software/>.
- García, Miriam. 2017.** *coding or not. coding or not*. [En línea] 5 de octubre de 2017. <https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve>.
- Gobierno Matancero. 2019.** *www.matanceros.gob.cu. www.matanceros.gob.cu*. [En línea] 20 de 11 de 2019. [Citado el: 18 de Febrero de 2020.] <http://www.matanceros.gob.cu>.
- Granma. 2019.** *www.granma.cu. www.granma.cu*. [En línea] 16 de octubre de 2019. [Citado el: 1 de Mayo de 2020.] <http://www.granma.cu/cuba/2019-10-16/parque-cientifico-tecnologico-de-matanzas-es-elogiado-por-el-ministro-de-comunicaciones-de-cuba-16-10-2019-09-10-49>.
- ISO. 2005.** *www.iso.org. www.iso.org*. [En línea] Abril de 2005. [Citado el: 1 de Mayo de 2020.] <https://www.iso.org/standard/32620.html>.
- Jetbrains. sf.** *www.jetbrains.com. www.jetbrains.com*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://www.jetbrains.com/phpstorm/features/>.
- León, Diego Ponce de. sf.** *Hojas de estilo en cascada (CSS)*. [En línea] sf. [Citado el: 19 de febrero de 2020.] <https://www.htmlquick.com/es/tutorials/css.html>.

Los mapas conceptuales interactivos como recursos didácticos en el ámbito universitario. Revista Complutense De Educación. Cabero Almenara, Julio, Ballesteros Regaña, Cristóbal y López Meneses, Eloy. 2015. 51-76, 2015, Vol. 26.

Marini, Emiliano . 2012. *El Modelo Cliente/Servido.* 2012.

Ministerio de Comunicaciones. 2019. www.mincom.gob.cu. *www.mincom.gob.cu*. [En línea] 2019. [Citado el: 18 de febrero de 2020.] <https://www.mincom.gob.cu/es/gobierno-electronico>.

Mozilla. sf. developer.mozilla.org. *developer.mozilla.org*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://developer.mozilla.org/docs/Web/HTML>.

Mozilla. sf. developer.mozilla.org. *developer.mozilla.org*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://developer.mozilla.org/docs/Web/JavaScript>.

Navarro Cifuentes, Bernan Osiel. sf. Academia Análisis de Sistemas. *Academia Análisis de Sistemas*. [En línea] sf. [Citado el: 20 de Mayo de 2020.] https://www.academia.edu/40494764/An%C3%A1lisis_de_Sistemas.

NeoAttack. sf. neoattack.com. *neoattack.com*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://neoattack.com/neowiki/jquery/>.

Nextech. 2016. nextech.pe. *nextech.pe*. [En línea] 16 de Septiembre de 2016. [Citado el: 1 de Mayo de 2020.] <https://nextech.pe/que-es-bpmn-y-para-que-sirve/>.

OAS. s.f. OAS. [En línea] s.f. [Citado el: 3 de diciembre de 2019.] https://www.oas.org/es/sap/dgpe/guia_egov.asp.

Paz, Claudia Yilén. 2019. CUBAHORA. [En línea] 8 de julio de 2019. [Citado el: 13 de febrero de 2020.] <https://www.cubahora.cu/ciencia-y-tecnologia/informatizacion-de-la-sociedad-en-cuba-ordenar-el-proceso>.

Pérez Salomón, Omar . 2018. www.cubadebate.cu. *www.cubadebate.cu*. [En línea] 7 de mayo de 2018. [Citado el: 1 de Mayo de 2020.] <http://www.cubadebate.cu/opinion/2018/05/07/parque-cientifico-tecnologico-de-matanzas-una-propuesta-innovadora/>.

Php. sf. www.php.net. *www.php.net*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://www.php.net/manual/preface.php>.

Picazo M. 2013. Herramientas Automatizadas . *VISUAL PARADIGM UML*. [En línea] 17 de noviembre de 2013. http://picazomarianaha.blogspot.com/2013/11/blog-post_16.html.

Postgresql. sf. www.postgresql.org. *www.postgresql.org*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://www.postgresql.org/about/>.

Pressman, Maxim y B. 2014. *Software Engineering: A Practitioner's Approach*. s.l. : 8, 2014.

Proceso de Desarrollo y Gestión de Proyectos de Software. 2012. *Proceso de Desarrollo y Gestión de Proyectos de Software*. 2012.

Radio26. 2018. www.radio26.cu. *www.radio26.cu*. [En línea] 25 de Mayo de 2018. [Citado el: 1 de Mayo de 2020.] <http://www.radio26.cu/2018/05/25/sistema-bienestar-y-gobierno-digital-en-matanzas/>).

Scholarium. sf. scholarium.info. *scholarium.info*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <http://scholarium.info/processmaker-sistema-de-gestion-de-procesos-de-negocio-bpm/>.

Sysml. sf. sysml.org. *sysml.org*. [En línea] sf. [Citado el: 1 de mayo de 2020.] <https://sysml.org/sysml-faq/what-is-sysml.html>.

Tech Target. sf. searchdatacenter.techtarget.com. *searchdatacenter.techtarget.com*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultas-estructuradas>.

Terrera, Gustavo. 2017. testingbaires.com. *testingbaires.com*. [En línea] 26 de Febrero de 2017. [Citado el: 10 de Junio de 2020.] <https://testingbaires.com/2017/02/26/pruebas-caja-negra-enfoque-practico/>.

testerhouse. 2019. testerhouse.com. *testerhouse.com*. [En línea] 26 de Marzo de 2019. [Citado el: 10 de Junio de 2020.] <https://testerhouse.com/teoria-testing/pruebas-funcionales/>.

Time of Software. 2016. Time of Software. *Time of Software*. [En línea] 14 de diciembre de 2016. <http://timeofsoftware.com/patron-singleton/>.

Turrado, Jorge. 2020. www.campusmvp.es. *www.campusmvp.es*. [En línea] 10 de Marzo de 2020. [Citado el: 10 de Junio de 2020.] <https://www.campusmvp.es/recursos/post/que-son-las-pruebas-de-software.aspx>.

Universidad Nacional Abierta y a Distancia. sf. stadium.unad.edu.co. *stadium.unad.edu.co*. [En línea] sf. [Citado el: 20 de Mayo de 2020.] http://stadium.unad.edu.co/ovas/10596_9839/diagramas_de_actividades.html.

—. Universidad Nacional Abierta y a Distancia. *Universidad Nacional Abierta y a Distancia*. [En línea] http://stadium.unad.edu.co/ovas/10596_9836/diagrama_de_clases_de_diseo.html.

Villarreal Fuentes, César Amauri . 2013. docplayer.es. *docplayer.es*. [En línea] 13 de noviembre de 2013. [Citado el: 12 de Mayo de 2020.] <https://docplayer.es/15803354-Noviembre-2013-que-es-jquery-por-cesar-amauri-villarreal-fuentes-que-es-jquery-noviembre-2013.html>.

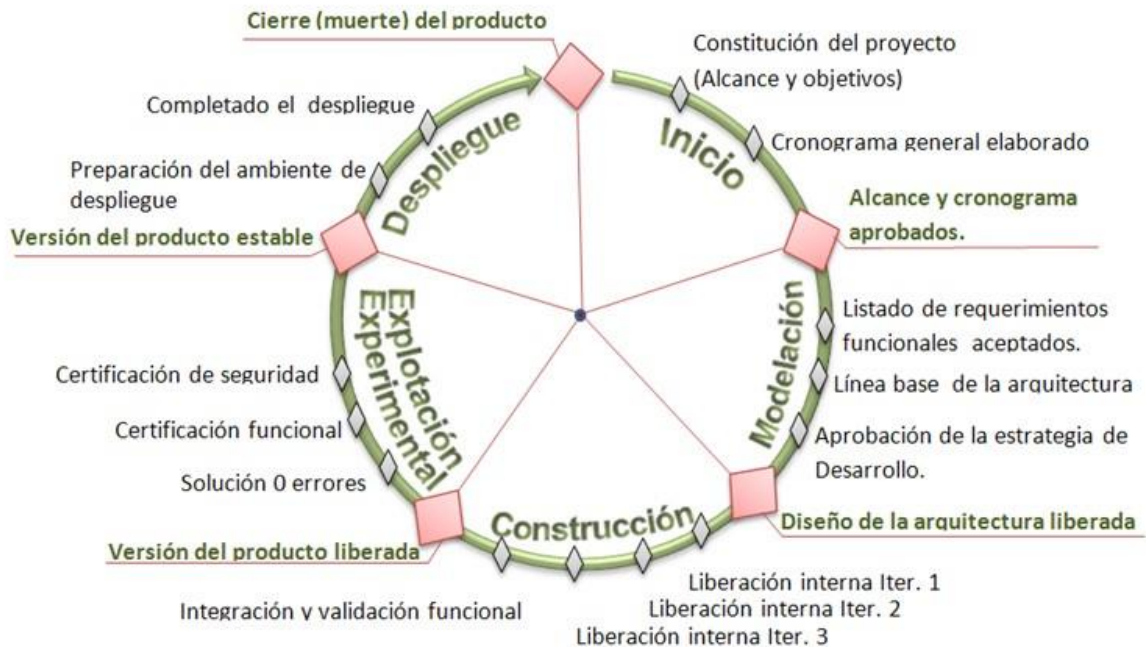
Visual Paradigm. sf. www.visual-paradigm.com. *www.visual-paradigm.com*. [En línea] sf. [Citado el: 1 de Mayo de 2020.] <https://www.visual-paradigm.com/features/>.

—. **2018.** www.visual-paradigm.com. *www.visual-paradigm.com*. [En línea] 2018. [Citado el: 10 de febrero de 2020.] https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html.

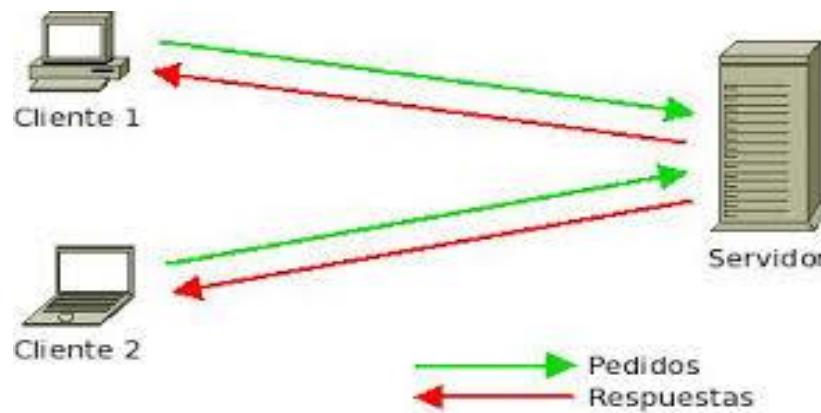
Xetid. 2012. *Proceso de Desarrollo y Gestión de Proyectos de Software*. La Habana : s.n., 2012.

Zend Framework. sf. framework.zend.com. *framework.zend.com*. [En línea] sf. [Citado el: 12 de Mayo de 2020.] <https://framework.zend.com/about>.

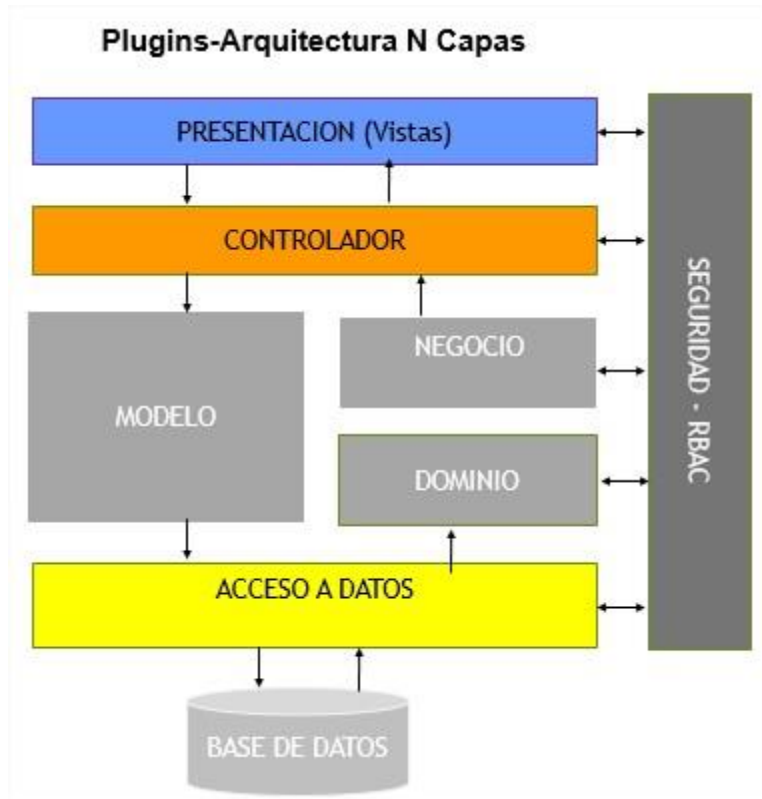
Anexos



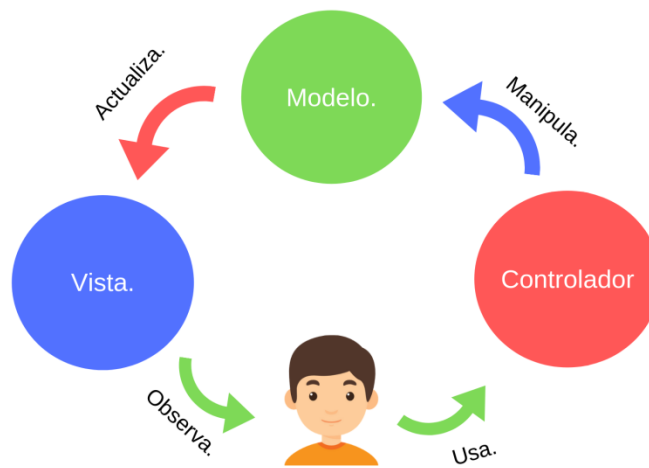
Anexo 1 Hitos por fases



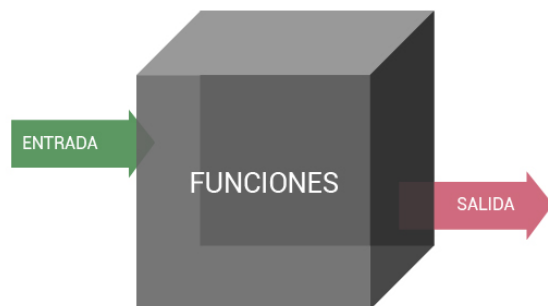
Anexo 2 Arquitectura cliente-servidor



Anexo 3 Arquitectura n-capas



Anexo 4 Patrón de diseño Modelo-Vista-Controlador



CAJA NEGRA

Anexo 5 Prueba de caja negra



Anexo 6 Logotipo de la empresa CITMA