



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 1**

***“Sistema de Gestión de Ilegalidades para la Dirección Provincial  
de Planificación Física de la provincia Granma.”***

**Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.**

**Autores: Pedro Luis Chacón.**

**Dannel David Ramos Nieves.**

**Tutores: Ing. Yailén Pozo Laserra.**

**Ing. Karina Barbara Martinez Casas.**

**Ing. Ariel Ramirez Alvarez.**

**Habana, Cuba, junio 2018.**

**“Año 60 de la Revolución”**

**Declaración de Autoría:**

Declaramos ser los únicos autores del presente trabajo de diploma y reconocemos a la XETID, Empresa de Tecnología de la Información para la Defensa, los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firman la presente a los \_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

**Firma del Autor**

**Pedro Luis Chacón**

---

**Firma del Autor**

**Dannel David Ramos Nieves**

---

**Firma del Tutor**

**Ing. Yailén Pozo Laserra**

---

**Firma del Tutor**

**Ing. Karina B. Martinez Casas**

---

**Firma del Tutor**

**Ing. Ariel Ramirez Alvarez.**



Frase:

*“El tema relativo al conocimiento y la tecnología es de especial relieve en nuestra agenda, porque en él abordamos los problemas que deciden, en buena medida, el futuro de nuestros países.”*

*Fidel Castro Ruz.*



**Agradecimientos.**

Agradecemos a todas las amistades, profesores, tribunales de corte, tutores y familiares que sacrificaron parte de su tiempo libre para apoyarnos y guiarnos en el transcurso de la elaboración de este trabajo de diploma, que gracias a su esfuerzo desinteresado cumple con la calidad y la profesionalidad requerida. Un agradecimiento especial para nuestros padres que siguieron de cerca y con la mayor preocupación nuestro recorrido por la carrera que hoy culminamos.



## Resumen.

La presente investigación describe la realización del sistema informático, de tipo web, para la gestión de las ilegalidades de la Dirección Provincial de Planificación Física (DPPF) de Granma; desarrollado en la División Sistema de Apoyo a la Toma de Decisiones (SATD) perteneciente a la Empresa de Tecnologías de la Información para la Defensa (XETID). Este sistema permite a los inspectores y especialistas de la DPPF el registro de: ilegalidades referentes a la vivienda, inspectores, multas y bases legales. Además, contribuye al proceso de gestión de las ilegalidades y la toma de decisiones disminuyendo la probabilidad de: duplicidad de información, generación de reportes erróneos y reportes poco confiables. El sistema cuenta con funcionalidades tales como el seguimiento de una ilegalidad a través de un flujo de trabajo y el diseño *responsive* o adaptativo para los dispositivos, desde ordenadores de escritorio a *tablets* y móviles. El proceso de software estuvo guiado por la metodología ágil “Proceso de Desarrollo y Gestión de Proyectos de Software” (PRODESOF) en su versión 1.5. Para la implementación del sistema se utilizó Zeolides como marco de trabajo, PostgreSQL como sistema gestor de bases de datos y NetBeans como entorno integrado de desarrollo.

Palabras-clave: **gestión, ilegalidades, flujo de trabajo, web.**

**Abstract.**

The present investigation describes the realization of the computer system for the management of the illegalities in the Provincial Direction of Physical Planning (PDPP) of Granma; using web technology and developed in the *División de Soporte a la Toma de Decisiones (SATD)* belonging to *Empresa de Información para la Defensa (XETID)*. This system allows inspectors and specialists of the PDPP to register: illegalities related to the houses, inspectors, fines and legal bases. In addition, it contributes to the process of managing illegalities and making decisions, reducing the probability of: duplication of the information, generation of defectives and unreliable reports. The system has features such as tracking an illegality through a workflow and responsive or adaptive design for devices, from desktops to tablets and mobiles. The software process was guided by the agile methodology "Proceso de Desarrollo y Gestión de Proyectos de Software" (PRODESOF) in its version 1.5. For the implementation of the system was used Zeolides framework, PostgreSQL as a database management system and NetBeans as an integrated development environment.

**Keywords: management, illegalities, workflow, web.**

**Índice**

Introducción..... 1

Capítulo 1: Fundamentación teórica ..... 8

    1.1 Introducción..... 8

    1.2 Conceptos Relacionados..... 8

    1.3 Soluciones Existentes ..... 9

    1.3.1 Resultado del análisis de las soluciones existentes.....10

    1.4 Tendencias, tecnologías y herramientas a utilizar .....11

        1.4.1 Metodología para el Proceso de Desarrollo de Software.....11

        1.4.2 Marco de trabajo.....13

        1.4.3 Lenguajes de programación.....15

        1.4.4 Lenguaje Unificado de Modelado UML 2.5.1.....18

        1.4.5 Herramientas de desarrollo.....18

    1.5 Conclusiones del capítulo.....21

Capítulo 2: Análisis y diseño de la propuesta de solución ..... 22

    2.1 Introducción.....22

    2.2 Modelado de los procesos de negocio.....22

        2.2.1 Identificación de los procesos de negocio.....22

        2.2.2 Especificación de los procesos de negocio.....24

        2.2.3 Especificación de los conceptos del dominio del problema .....26

    2.3 Descripción de la propuesta de solución .....27

    2.4 Requisitos .....28

        2.4.1 Fuentes y técnicas para la obtención de requisitos .....28

        2.4.2 Identificación y clasificación de requisitos .....29

        2.4.3 Especificación de requisitos funcionales .....34

    2.5 Análisis y diseño.....35

        2.5.1 Diseño arquitectónico.....35

        2.5.2 Modelado de datos.....40

        2.5.3 Modelado del Diseño .....42

    2.6 Diagrama de despliegue.....44

    2.7 Conclusiones del capítulo.....46

Capítulo 3: Implementación y validación de la propuesta de solución ..... 47



3.1 Introducción.....	47
3.2 Implementación .....	47
3.2.1 Estándares de implementación .....	47
3.3 Interfaces de usuario .....	49
3.4 Pruebas de software.....	50
3.4.1 Pruebas de unidad.....	51
3.4.2 Pruebas de sistema (funcionales).....	55
3.4.3 Pruebas de sistema (carga y estrés).....	59
3.4.4 Pruebas de integración .....	63
3.4.5 Pruebas de aceptación .....	64
3.4.6 Resultado de las pruebas de software .....	65
3.5 Validación de los resultados obtenidos.....	66
3.5.1 Validación de la hipótesis.....	69
3.6 Conclusiones del capítulo.....	71
Conclusiones generales.....	72
Recomendaciones .....	73
Referencias bibliográficas .....	74
Bibliografía consultada.....	77
Anexos.....	82
Anexo 1: Diagramas de requisitos funcionales (DRF) y descripción de requisitos funcionales (DERF).82	
Anexo 2 Prototipos de interfaz de usuario (PIU).....	90
Anexo 3: Diagramas Entidad Relación. ....	96
Anexo 4: Diagramas de clase del diseño con estereotipos web (DCD).....	97
Anexo 5: Interfaces de usuario (IU). ....	100
Anexo 6: Diseños de casos de pruebas (DCP).....	104
Anexo 7: Cuestionario #1 .....	111
Anexo 8: Cuestionario #2 .....	113
Anexo 9: Modelos de reportes generados (MRG).....	115
Anexo 10: Prueba de sistema (carga y estrés). ....	117
Anexo 11: Casos de prueba de integración (CPI).....	118
Anexo 12: Usuarios encuestados .....	120
Anexo 13: Acta de aceptación del cliente .....	121





**Índice de Figuras**

Figura 1: Ciclo de vida del proceso de desarrollo de un producto de software ..... 12

Figura 2: Mapa de procesos de negocio ..... 23

Figura 3: Diagrama de actividad ..... 25

Figura 4: Mapa conceptual..... 26

Figura 5: Arquitectura Cliente – Servidor ..... 36

Figura 6: Representación del Patrón N-Capas con Modelo Vista Controlador ..... 37

Figura 7: Diagrama de componentes del sistema ..... 39

Figura 8: Diagrama entidad-relación mod\_sginf.dat\_incidente ..... 41

Figura 9: Diagrama de clase del diseño. Gestionar Incidencia ..... 43

Figura 10: Diagrama de despliegue ..... 45

Figura 11: Interfaz de usuario Exportar reporte ..... 50

Figura 12: Gráfico piramidal..... 52

Figura 13: Valores de las métricas evaluadas en el gráfico piramidal ..... 54

Figura 14: Gráfico de Abstracción/Inestabilidad ..... 55

Figura 15: Resultado de la 1ra iteración de la prueba de carga y estrés con la herramienta Jmeter ..... 61

Figura 16: Distribución del IP de la encuesta #1 ..... 68

Figura 17: DRF Gestionar concepto..... 82

Figura 18: DERF Adicionar concepto ..... 83

Figura 19: DERF Modificar concepto ..... 84

Figura 20: DERF Eliminar concepto ..... 84

Figura 21: DRP Mostrar concepto ..... 85

Figura 22: DRF Asociar concepto ..... 86

Figura 23: DERF Asociar concepto ..... 86

Figura 24: DRF Buscar concepto ..... 87

Figura 25: DERF Buscar concepto..... 88

Figura 26: DRF Exportar concepto..... 89

Figura 27: DERF Exportar concepto ..... 89

Figura 28: PIU Adicionar incidente ..... 90

Figura 29: PIU modificar contraseña ..... 91



Figura 30: PIU Buscar multa .....	91
Figura 31: PIU Gestionar multa .....	92
Figura 32: PIU Gestionar inspector .....	93
Figura 33: PIU Gestionar infractor .....	94
Figura 34: PIU Exportar listado de multas .....	94
Figura 35: PIU Exportar reporte .....	95
Figura 36: PIU Exportar listado de inspectores .....	95
Figura 37 - Diagrama entidad-relación mod_multa.....	96
Figura 38: DCD Exportar listado de inspectores .....	97
Figura 39: DCD Exportar reporte resumen.....	98
Figura 40: Gestionar inspector.....	99
Figura 41: IU Listado de ilegalidades .....	100
Figura 42: IU Adicionar base legal .....	101
Figura 43: IU Adicionar infractor.....	101
Figura 44: IU Adicionar nomenclador .....	102
Figura 45: IU Buscar concepto.....	103
Figura 46: IU Flujo de trabajo.....	103
Figura 47: Representación en por ciento de los usuarios encuestados.....	120
Figura 48: Acta de aceptación del cliente.....	121



**Índice de Tablas**

Tabla 1: Operacionalización de variable dependiente .....4

Tabla 2: Operacionalización de la variable independiente.....5

Tabla 3: Requisitos de hardware.....31

Tabla 4: Requisitos de software .....32

Tabla 5: Estrategia de pruebas de software .....51

Tabla 6: DCP Adicionar base legal .....57

Tabla 7: Umbrales para sistemas web .....62

Tabla 8: Integración con el módulo Seguridad .....64

Tabla 9: Indicadores de la escala Likert .....66

Tabla 10: Valoración de expertos referente al sistema.....67

Tabla 11: Resultados de la encuesta #2 Sección I.....69

Tabla 12: Resultados de la encuesta #2 Sección II.....69

Tabla 13: Resultados de la encuesta #3 .....70

Tabla 14: DCP Adicionar ilegalidad..... 105

Tabla 15: DCP Eliminar ilegalidad..... 105

Tabla 16: DCP Modificar ilegalidad ..... 106

Tabla 17: DCP Adicionar inspector ..... 107

Tabla 18: DCP Modificar base legal..... 108

Tabla 19: DCP Eliminar inspector ..... 109

Tabla 20: DCP Exportar multa ..... 109

Tabla 21: DCP Exportar reporte..... 110

Tabla 22: MRG Instrucción #2..... 115

Tabla 23: MRG Resumen ..... 115

Tabla 24: MRG Resumen por categoría..... 116

Tabla 25: MRG Resumen por organismo ..... 116

Tabla 26: MRG Resumen de ilegalidades por resolver en el trimestre ..... 116

Tabla 27: Resultado de la prueba de carga y estrés de la herramienta Jmeter ..... 117

Tabla 28: CPI Módulo Seguridad ..... 118

Tabla 29: CPI Módulo Flujo de Trabajo..... 118



Tabla 30: CPI Módulo Persona .....	119
Tabla 31: CPI Módulo Nomenclador .....	119

## Introducción.

La explotación de las Tecnologías de la Información y las Comunicaciones (TIC) beneficia a las organizaciones dotándolas de múltiples mecanismos que mejoran el almacenamiento y análisis de la información que se maneja en cada una de ellas. La mayoría de las empresas en el mundo han centrado su atención en la implantación de nuevos sistemas informáticos que automaticen sus procesos de negocio mediante la unificación de las actividades y datos de la empresa; lo que contribuye a la consulta constante y en tiempo real de los mismos, facilita la toma de decisiones, agiliza los procesos concebidos dentro de la organización y trae consigo un incremento de la productividad.

Cuba, a pesar de ser un país subdesarrollado, estar sometida a un férreo bloqueo económico y no contar con acceso a los principales avances de la ciencia y la técnica a nivel mundial, no está exenta del desarrollo tecnológico. Se ha incrementado el uso de las TIC potenciando el despliegue de nuevas propuestas de aplicaciones informáticas con el objetivo de mejorar el funcionamiento de las organizaciones dentro del país.

Los Sistemas de Soporte a la Toma de Decisiones (*DSS* por sus siglas en inglés) no han quedado ajenos a los avances de las TIC, convirtiéndose estos en herramientas esenciales para la gestión empresarial. Estos sistemas permiten almacenar, organizar y efectuar consultas de datos, en breves períodos de tiempo y de forma eficaz, proporcionando nuevos métodos para analizar los datos, encontrar correlaciones y dependencias entre ellos. Los *DSS* se utilizan frecuentemente en la gestión empresarial.

La Empresa de Tecnologías de la Información para la Defensa (XETID) creada en Cuba durante el año 2013, pertenece a las Fuerzas Armadas Revolucionarias (FAR) y fue concebida como pilar estratégico para apoyar el proceso de informatización de la sociedad. Esta empresa posee una división llamada Sistema de Apoyo a la Toma de Decisiones (SATD) que se especializa en los *DSS* y posee convenios con diferentes entidades, logrando avances significativos en la informatización de las mismas. Dentro de estas entidades se encuentra el Instituto Nacional de Planificación Física (IPF) y sus departamentos subordinados: las Direcciones Provinciales de Planificación Física (DPPF) y las Direcciones Municipales de Planificación Física (DMPF).

Entre los procesos que manejan las DPPF están los relacionados con la inspección urbanística a nivel provincial. Se controlan la cantidad y tipo de ilegalidades detectadas en cada uno de los municipios, ya sean de infractores particulares u organismos infractores de acuerdo al Decreto Ley 272 por el cual impone las multas y la obligación de hacer<sup>1</sup> de acuerdo a la ilegalidad. La Dirección Provincial de Planificación Física entrega sistemáticamente reportes (basados en modelos definidos a nivel nacional) al IPF en los cuales se relacionan las multas, infractores e inspectores asociados con las ilegalidades. Los reportes pueden estar definidos en rangos de fecha, por tipo de ilegalidad, propietario del suelo ya sea estatal o privado, por Organismo Responsable, por obligaciones de hacer y si están en el Plan Integral de Enfrentamiento de Ilegalidades (PIEI) o son Nuevas Detectadas<sup>2</sup>. Vale destacar además que las ilegalidades siguen un flujo de trabajo transitando desde un estado inicial (Detectada) hasta un estado final (Cerrada).

Actualmente la gestión de las ilegalidades generadas por el proceso de inspección urbanística, que ejecuta la Dirección de Inspección Estatal (DIE) de la DPPF de Granma, resulta muy compleja. Los inspectores son los encargados de detectar, archivar en formato duro las ilegalidades y entregarlas a sus respectivas DMPF. En el nivel municipal las ilegalidades son transformadas a formato digital, almacenadas en hojas de cálculo de *Microsoft Office Excel* y posteriormente se elaboran los reportes estadísticos extrayendo la información y clasificándola de manera manual. Dichos reportes son entregados a la DPPF de Granma, entidad encargada de la elaboración de los reportes a nivel de provincia y de su posterior envío al IPF. Realizar un análisis o efectuar búsquedas de información referente a las ilegalidades es un proceso que resulta complicado debido a que las herramientas con las que trabajan han demostrado ser ineficaces para apoyar este trabajo. Esta situación trae consigo que la tramitación de los datos a los reportes no sea un proceso rápido ni efectivo. Los reportes que se generan son poco fiables, por lo difícil que resulta mantener la integridad de la información al transcurrir largos períodos de tiempo, y al ser tramitada por tantas personas y organismos diferentes. Sin el uso de una base de datos centralizada aumenta el riesgo de la duplicidad en los datos, la pérdida o modificación de la información a su paso por todos los canales pertinentes, además de que el control de los cambios sobre la misma se torna complicado.

Todos estos conflictos traen consigo deficiencias en la toma de decisiones, así como en la entrega de los reportes en el tiempo establecido y con la calidad requerida a las instancias superiores del país. Teniendo

---

<sup>1</sup> Obligación de hacer: medida adoptada para dar solución a la ilegalidad (Legalizar, Demoler o Tramitar)

<sup>2</sup> Nuevas detectadas: Ilegalidades detectadas de manera fortuita por los inspectores y no están recogidas en el PIEI.

en cuenta las dificultades descritas anteriormente se define como **problema a resolver**: ¿Cómo contribuir al proceso de gestión de las ilegalidades y al apoyo en la toma de decisiones en la DPPF de Granma?

En función del problema identificado y con el objetivo de alcanzar una solución satisfactoria se determina como **objeto de estudio**: los procesos de gestión de la información para la inspección urbanística, enmarcado en el **campo de acción** constituido por: los procesos de gestión de las ilegalidades en la DPPF de Granma.

Para dar solución a lo anteriormente expuesto se define como **objetivo general** del presente trabajo: desarrollar un sistema informático con el uso de tecnologías web para contribuir al proceso de gestión de las ilegalidades y al apoyo en la toma de decisiones en la DPPF de Granma. Para cumplir con el objetivo propuesto se definen los siguientes **objetivos específicos** de la investigación:

1. Caracterizar los sistemas y conceptos relacionados con la inspección urbanística y definir las tecnologías, herramientas y metodologías a utilizar para el cumplimiento del objetivo de la investigación.
2. Capturar los requerimientos y diseñar el Sistema de Gestión de Ilegalidades en la DPPF de Granma.
3. Implementar el Sistema de Gestión de Ilegalidades en la DPPF de Granma.
4. Validar el Sistema de Gestión de Ilegalidades en la DPPF de Granma a través de las pruebas de software.

Se propone como **hipótesis de investigación**: Si se desarrolla un sistema para la gestión de las ilegalidades en la DPPF de Granma se contribuirá al proceso de gestión de las ilegalidades y al apoyo en la toma de decisiones en dicha entidad.

Se define como **variable independiente**: Sistema de Gestión de Ilegalidades en la DPPF de Granma y **como variable dependiente**: proceso de gestión de ilegalidades y apoyo en la toma de decisiones.

Operacionalización de variables:

**Tabla 1: Operacionalización de variable dependiente**

Fuente: Creación de los autores

Variable Dependiente	Dimensiones	Indicador	Unidad
Proceso de gestión de las ilegalidades y apoyo en la toma de decisiones	Disponibilidad de la información para la gestión de las ilegalidades	Tiempo de respuesta del sistema	Segundos Minutos Horas
	Calidad de la información para la toma de decisiones	Calidad de la información	Alto [100 % ; 70%] Medio [69% ; 50%] Bajo [49% ; 0%]
	Usabilidad de la Información disponible	Tiempo para generar reportes de las ilegalidades	Segundos Minutos Horas
		Nivel de Subjetividad	Alto [100 % ; 70%] Medio [69% ; 50%] Bajo [49% ; 0%]
	Satisfacción del decisor sobre la calidad de la información brindada por el sistema	Nivel de Satisfacción	Alto [100 % ; 70%] Medio [69% ; 50%] Bajo [49% ; 0%]
	Retroalimentación brindada por el sistema entre los diferentes niveles del Instituto de Planificación Física	Nivel de retroalimentación	Alto [100 % ; 70%] Medio [69% ; 50%] Bajo [49% , 1%]



**Tabla 2: Operacionalización de la variable independiente**

Fuente: Creación de los autores

Variable independiente	Dimensiones	Indicadores	Unidad métrica
Sistema de Gestión de Ilegalidades de la DPPF de Granma	Grado de satisfacción del usuario con el sistema	Nivel de satisfacción con el sistema	Alto [100 % ; 70%] Medio [69% ; 50%] Bajo [49% ; 0%]

Para dar cumplimiento a los objetivos específicos se proponen como **tareas de investigación**:

1. Caracterización del estado del arte de sistemas y tendencias existentes relacionadas con la inspección urbanística.
2. Estudio del proceso de negocio del sistema actual.
3. Revisión del diseño de los componentes visuales.
4. Caracterización de la arquitectura establecida por la XETID.
5. Caracterización de las herramientas establecidas por la XETID.
6. Caracterización de las tecnologías de desarrollo establecidas por la XETID.
7. Identificación de los requerimientos.
8. Modelación de la base de datos del sistema.

Con el propósito de desarrollar las tareas planteadas, se utilizaron los **métodos de investigación** siguientes:

**Métodos teóricos:**

**Histórico-Lógico:** este método fue utilizado para estudiar cómo ha evolucionado y se han desarrollado los DSS y los sistemas para la inspección urbanística desde sus surgimientos hasta la actualidad, sus herramientas, formas de trabajos, modelos, entre otros. Realizando así el estudio de los procesos de la gestión de la información en la Dirección Provincial de Planificación Física de Granma.

**Análisis y síntesis:** Es un método que consiste en la separación de las partes de un todo para estudiarlas en forma individual (Análisis), y la reunión racional de elementos dispersos para estudiarlos en su totalidad (Síntesis) (Sampieri et al, 2014).

Este método fue utilizado para el análisis de la bibliografía referente al trabajo que se realizará. La confección del diseño teórico y metodológico de la investigación, análisis de las tendencias y tecnologías actuales de los sistemas de soporte a la toma de decisiones y los sistemas para la inspección urbanística.

**Modelación:** este método es empleado para modelar los diferentes diagramas de ingeniería de software con el objetivo de lograr una mayor comprensión del trabajo que se realiza y los objetivos que se deben cumplir.

#### **Métodos Empíricos:**

**Análisis documental:** se utilizó para consultar trabajos y buscar información relacionada con los *DSS* y los sistemas de inspección urbanística reflejada en la elaboración del marco teórico de la investigación.

Luego de terminada la investigación y comenzar a utilizar los resultados de la misma, se contribuirá al proceso de gestión de ilegalidades y al apoyo en la toma de decisiones en la DPPF de Granma la cual contará con una base de datos única para el almacenamiento de la información referente a las ilegalidades, inspectores, multas asociadas y base legal y dispondrá de una aplicación web que le facilitará la retroalimentación entre todas las áreas de la organización.

El presente trabajo está estructurado en 3 capítulos de la siguiente forma:

**Capítulo 1. Fundamentación teórica:** abordará los conceptos relacionados a la investigación, se referenciarán sistemas existentes los cuales pueden a través de sus funcionalidades llegar a realizar funciones semejantes al sistema deseado. Igualmente se realiza la selección de las tecnologías, la metodología para el proceso de desarrollo y herramientas actuales que se van a emplear para el desarrollo del sistema.

**Capítulo 2. Análisis y diseño de la propuesta de solución:** contendrá toda la información referente al modelado de la aplicación, contando con el modelo de dominio, los diagramas de clases, requisitos

funcionales y no funcionales que debe cumplir el sistema para lograr una mayor aceptación y uso de los usuarios, así como propuesta y diseño de la arquitectura.

**Capítulo 3. Validación de la propuesta de solución:** se describe la elaboración y realización de las pruebas al software a través de las estrategias de pruebas seleccionadas y se representan los resultados de las mismas. Además, se evalúa el cumplimiento de los requisitos no funcionales por parte del sistema y se expone los valores tomados por las variables de la hipótesis de la investigación y si dichos valores satisfacen los objetivos de la presente investigación.

## Capítulo 1: Fundamentación teórica

### 1.1 Introducción

En este capítulo se muestran los conceptos fundamentales asociados a la investigación, además se puede apreciar el resultado del análisis del estudio del arte sobre las soluciones informáticas existentes, con las funcionalidades que de alguna manera responden parcialmente al objetivo general de la investigación. También se evidencia la metodología para el proceso de desarrollo de software y las herramientas a utilizar durante el proceso de diseño e implementación del Sistema de Gestión de Ilegalidades en la DPPF de Granma.

### 1.2 Conceptos Relacionados

En la actualidad existen varios autores que definen los **DSS** como sistemas de apoyo enfocados en el análisis de los datos de la empresa y que sirven para ayudar a los directivos a tomar decisiones que a menudo son cambiantes y se deben tomar rápidamente; tal es el caso de Joaquin Guiral y Rafael Lapiedra (2012). Otros autores como Nada y Mona Nasr (2015) los definen como sistemas que usan bases de datos, software para combinar un gran número de modelos para realizar la toma de decisiones de manera científica. Se puede concluir que los **DSS** constituyen herramientas para el manejo óptimo de las organizaciones, debido a que al almacenar y analizar los datos que son manejados dentro de las mismas, estos sistemas brindan facilidades a los directivos para elegir un curso de acciones de acuerdo a los resultados que reflejan.

Se hizo necesario para la aplicación del **DSS** en la DPPF de Granma estudiar conceptos relacionados con la investigación tales como:

**IPF:** el Instituto de Planificación Física es la entidad nacional subordinada al Consejo de Ministros que dirige la aplicación de la política del Estado y el Gobierno en materia de ordenamiento territorial y **urbanismo** (IPF, 2015). El término **urbanismo** es definido según el sitio web oficial de IPF como el conjunto de disciplinas que se encargan del estudio de los asentamientos humanos para su diagnóstico, comprensión e intervención (IPF, 2018). Dentro de estas disciplinas se encuentra la detección de las **ilegalidades de la vivienda y urbanismo**; definido por el IPF como todo proceso o acción que viole los estatutos y leyes de

urbanización de la localidad. El proceso para el levantamiento de dichas ilegalidades es denominado **Inspección Urbanística** y se realiza como parte del plan de trabajo de las DPPF del país. Se define como la actividad que los órganos administrativos competentes, locales y autonómicos, ejercen en materia de edificación y uso del suelo, con la finalidad de comprobar si los actos en que se materializan se ajustan a la legalidad urbanística y a las especificaciones del planeamiento urbanístico (IPF, 2018).

### 1.3 Soluciones Existentes

Para alcanzar el objetivo de la investigación fue necesario el análisis de aplicaciones desarrolladas para la inspección urbanística tanto en el contexto nacional como internacional. El análisis se realizó con el fin de obtener una aproximación de las funcionalidades que debe tener la solución. En la investigación se hace énfasis en sistemas que permiten el almacenamiento, análisis y control de la información referente a las ilegalidades. Posterior a la detección de estos sistemas se obtuvieron características comunes que fueron utilizadas como criterios de comparación entre ellos.

- Uso de tecnología web
- Tipo de licencia
- Generación de Reportes
- Seguimiento de un flujo de trabajo

A continuación, se muestran algunas soluciones existentes que responden parcialmente a los objetivos de la presente investigación:

- **Sistema Informatizado de Planificación Física (SIPLAF):** sistema que utiliza tecnología web y está desplegado en la DPPF de Las Tunas, así como en sus Direcciones Municipales. SIPLAF es capaz de registrar ilegalidades y realizar reportes estadísticos, pero dichos reportes no cumplen con los parámetros establecidos por el cliente y las ilegalidades registradas no son capaces de seguir con el flujo de trabajo lógico establecido por el IPF. Dicha solución, no muestra la base legal para su consulta por parte de los usuarios de la aplicación y no gestiona las multas e inspectores asociados a cada ilegalidad.

- **Sistema de Información Territorial Software Libre para Entidades Locales (LocalGIS 3):** sistema que emplea tecnología web que surge como iniciativa del Ministerio de Industria, Energía y Turismo de España (MINETUR). El mismo permite la gestión de las ilegalidades a nivel de municipio, ofrece funcionalidades para publicar la información de los ciudadanos vía internet, y se integra con otros sistemas realizados para el monitoreo de las localidades, además posee una licencia *Open Source*. A pesar de esto, no es capaz de conciliar las multas asociadas a las ilegalidades y no genera reportes estadísticos asociados al proceso.
- **Portal para la Información y Gestión de la Inspección Técnica de Edificios (EUSKOREGITE):** aplicación web usada por el Gobierno Vasco. Permite a los inspectores la realización de los informes técnicos de la inspección de edificios, obras en construcción y el registro de las ilegalidades detectadas a través de plantillas prediseñadas. Cuenta con gráficas estadísticas para realizar el análisis de la situación territorial y sigue un flujo lógico de trabajo para las ilegalidades. A pesar de sus ventajas es utilizado específicamente para la información asociada a los edificios y no cubre el ordenamiento territorial en su totalidad. EUSKOREGITE, no es capaz de conciliar las multas correspondientes a cada ilegalidad.

### 1.3.1 Resultado del análisis de las soluciones existentes

Después de analizar los sistemas anteriores se ha concluido que aportan conocimientos acerca de:

- Formas de gestión de las ilegalidades respecto a su almacenamiento centralizado, tratamiento, análisis y manejo de la información para la toma de decisiones.
- La integración con sistemas existentes que apoyan al proceso.
- Formas de generar reportes estadísticos utilizando plantillas prediseñadas y cómo organizar y mostrar la información de los reportes a los usuarios de manera que posibiliten una sencilla comprensión de los datos.
- Flujos de trabajo a seguir por las ilegalidades durante su transcurso por el sistema.
- Tipos de gráficos estadísticos utilizados para el trabajo con ilegalidades asociadas al proceso de inspección urbanística.

Las soluciones analizadas evidencian particularidades que no son requeridas, pues no son independientes, lo que impide su utilización como componente de la propuesta de solución. Los reportes generados por estas aplicaciones no se ajustan a los modelos establecidos por el IPF, para consultar los modelos remitirse al **¡Error! No se encuentra el origen de la referencia.** Las ilegalidades registradas no son capaces de seguir con el flujo de trabajo lógico establecido por el usuario. Además, no muestran la base legal para su consulta por parte de los usuarios de la aplicación y no gestionan las multas e inspectores asociadas a cada ilegalidad.

Basándose en estas limitaciones y con el objetivo de contribuir al proceso de gestión de ilegalidades y al apoyo a la toma de decisiones se decide crear una solución nueva que tenga en cuenta las necesidades y preferencias de los usuarios de la DPPF de Granma. Dicha solución debe integrarse con otros sistemas existentes en la empresa XETID que perfeccionarían el funcionamiento de la solución.

### **1.4 Tendencias, tecnologías y herramientas a utilizar**

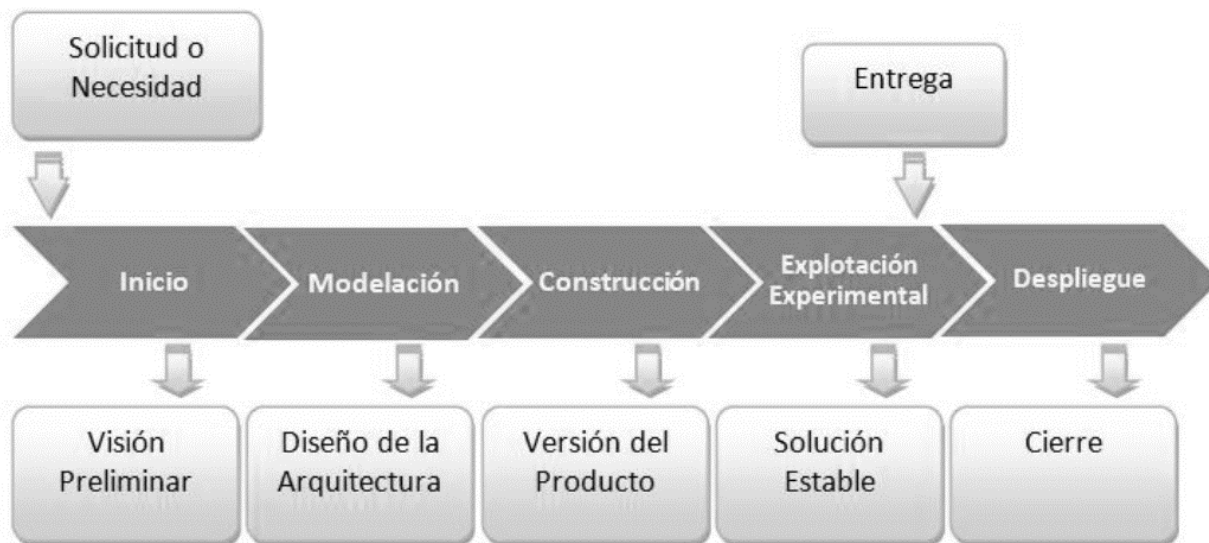
Las tecnologías caracterizadas a continuación, para el desarrollo de la solución, se definieron por el grupo de arquitectura de la XETID.

#### ***1.4.1 Metodología para el Proceso de Desarrollo de Software***

Diversas tendencias y metodologías de desarrollo de software han aparecido en años recientes, buscando resolver los problemas que proyectos más tradicionales, no han conseguido enfrentar. (...), las metodologías ágiles y los modelos de medición de madurez. (...) han permitido a los equipos de desarrollo producir software más robusto, predecible, reutilizable y de fácil mantenimiento (Curbelo Oliva et al, 2017).

Cada proyecto tiene un comienzo y un final definido, alcanzando este último cuando se han logrado los objetivos, cuando los objetivos del proyecto no podrán ser alcanzados, o cuando la necesidad del proyecto ya no exista y el proyecto sea cancelado. Durante la ejecución de un proyecto de software es necesario desarrollar un conjunto de actividades dirigidas a gestionar los recursos materiales, financieros y humanos asignados al proyecto que facilite el éxito del mismo (Curbelo Oliva et al, 2017).

Para ello la XETID cuenta con la metodología PRODESOFIT en su versión 1.5. Esta metodología dicta las pautas para el ciclo de vida del proceso de desarrollo de un producto de software; apreciable en la Figura 1.



**Figura 1: Ciclo de vida del proceso de desarrollo de un producto de software**

**Fuente: ( Curbelo Oliva et al, 2017)**

Ciclo de vida compuesto por 5 fases: **inicio**, **modelación**, **construcción**, **explotación experimental** y **despliegue**, que comprenden todas las actividades, desde el momento en que surge la idea de crear un nuevo producto de software, hasta aquel en que el producto deja definitivamente de ser utilizado por el último de sus usuarios (Curbelo Oliva et al, 2017).

Durante la **fase de Inicio** se logra una visión preliminar de la problemática a resolver y se definen los recursos relevantes para la ejecución del proyecto. Es decir, se describen los objetivos y el alcance del proyecto, se identifican los involucrados y ejecutores (entidades involucradas), se estima de manera general las actividades a realizar durante todo el ciclo de desarrollo del proyecto (Cronograma General), se establece la estrategia a seguir para realizar la modelación del negocio y la captura de requisitos y de ser necesario, se estiman los recursos materiales que deberán ser adquiridos (Curbelo Oliva et al, 2017).



En la **fase de Modelación** se capturan las partes esenciales del sistema, donde se identifican los procesos de negocio fundamentales y se aceptan los requisitos funcionales, obteniéndose la línea base de la arquitectura y una estrategia de construcción de la aplicación aprobada por los implicados en el proyecto. El hito fundamental de esta fase es la liberación de la arquitectura de sistema, datos y despliegue (Curbelo Oliva et al, 2017).

En la **fase de Construcción** se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. Las fases anteriores sólo dieron una arquitectura básica que es aquí refinada de manera incremental, conforme se construye el producto. En esta fase todas las características, componentes, y requisitos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión liberada del producto (Curbelo Oliva et al, 2017).

Durante la **fase de Explotación Experimental** 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. 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, culminando de ser preciso con transferencias tecnológicas (Curbelo Oliva et al, 2017).

El uso de la metodología PRODESOFTE v1.5 permitirá mantener un proyecto organizado, planificado y bien estructurado. Gracias al desarrollo iterativo e incremental se crean las entregas a nivel de versiones del producto de manera ágil y concurrente. Estas versiones garantizan un producto de calidad y la satisfacción del cliente en etapas tempranas del proyecto.

### **1.4.2 Marco de trabajo**

Un Marco de trabajo es un conjunto de herramientas, librerías, convenciones y buenas prácticas que pretenden encapsular las tareas repetitivas en módulos genéricos fácilmente reutilizables (Curbelo Oliva et al. 2017). El marco de trabajo constituye un elemento importante a la hora de desarrollar una solución, pues es capaz de proveer una arquitectura sólida y más extensible para la implementación de los diferentes componentes en un sistema informático.

### 1.4.2.1 Zeolides v2.2.0

Conjunto de componentes de software, librerías, herramientas y tecnologías libres integradas que permiten el desarrollo ágil y basado en componentes de aplicaciones web empresariales (múltiples propósitos, gran complejidad y grandes volúmenes de datos), centrando el desarrollo en el negocio, los requerimientos y las interfaces de usuario. Este Marco de trabajo presenta una estructura que facilita la comprensión por parte de los desarrolladores que trabajen sobre el mismo.(XETID, 2015).

El empleo de Zeolides es requerido debido a que los módulos a integrar con la propuesta de solución fueron desarrollados sobre su tecnología y de esta manera se garantiza la compatibilidad. Dicho marco de trabajo posee un grupo de ventajas de cara a los desarrolladores que lo condicionan como óptimo para utilizarse en el desarrollo del Sistema de Gestión de Ilegalidades de la DPPF de Granma las cuales son:

1. Gestor de metadatos: Automatiza la gestión de los “datos de los datos”. Esta gestión contiene la información de las tablas de datos, los campos y las relaciones que se establecen entre tablas.
2. Gestor de paquetes de instalación: Automatiza la gestión de instaladores de aplicaciones construidas sobre el marco de trabajo. Provee de una estructura para instalar nuevos componentes al menú que muestra el portal de aplicaciones. Esta funcionalidad facilita la integración entre componentes que es requerida por la propuesta de solución.
3. Subsistema de seguridad: Está dividido en seis módulos: Configuración general, Desbloqueo de usuarios y direcciones IP, Administrar sesiones, Configurar sistemas, Configurar usuarios y Compartimentación. Estos módulos y sus funcionalidades son aplicadas y utilizadas en el módulo de Seguridad a integrar con la propuesta de solución.
  - a. El módulo Configuración general permite el manejo de los dominios, servidores, idiomas, temas, escritorios, expresiones y claves, para el manejo posterior de los sistemas y usuarios.
  - b. El módulo Desbloqueo de usuarios y direcciones IP proporciona la gestión de las principales funcionalidades del desbloqueo permitiendo el manejo de los usuarios y direcciones IP del componente de seguridad.
  - c. El módulo Administrar sesiones tiene como propósito gestionar la administración de sesiones permitiendo eliminar las sesiones activas.

- d. El módulo Configurar sistemas permite la gestión de las principales funcionalidades de los sistemas permitiendo el manejo de los mismos, así como sus funcionalidades, acciones y recursos del componente de seguridad.
  - e. El módulo Configurar usuarios permite la gestión de los usuarios, roles, grupos, perfiles de usuario y los campos del perfil de usuario.
  - f. El módulo de Compartimentación permite compartimentar los roles, usuarios y sistemas basados en los dominios de seguridad.
4. Subsistema de gestión y monitoreo de trazas: Está dividido en dos componentes Configuración y Monitoreo de trazas. Esta funcionalidad de marco de trabajo le permitirá a la propuesta llevar un monitoreo de las trazas de acuerdo a los usuarios autenticados a través de módulo de Seguridad.
- a. El módulo Configurar traza permite activar y desactivar las trazas desde una interfaz de configuración.
  - b. El módulo Monitoreo tiene como propósito monitorear las trazas por los diferentes reportes de cantidad por categoría, por fecha, por usuario y por tipo. Además, muestra el monitoreo mediante gráficas y detalles por cada uno de los tipos de trazas (XETID, 2015).

#### **1.4.2.2 AngularJs v1.5**

AngularJs es un marco de trabajo de JavaScript con código abierto mantenido por Google utilizado para el desarrollo de aplicaciones web basado en el patrón Modelo Vista Controlador (Dorta, 2016).

El uso de este marco de trabajo aporta las siguientes ventajas:

- Vinculación de datos en vías.
- Inyección de Dependencias.
- Permite la creación de aplicaciones de una sola página.
- Posee un motor de plantillas.

#### **1.4.3 Lenguajes de programación**

Los lenguajes de computadoras son una sintaxis codificada usada por los programadores para comunicarse con ella. Es el único lenguaje que entienden las computadoras, tanto los programas de software como el

hardware. El lenguaje le permite al usuario dictar los comandos que la computadora debe entender para procesar los datos (Wilson y Leslie, 1993).

En el centro SATD de la XETID se emplean para el desarrollo de aplicaciones web los lenguajes descritos a continuación:

#### **1.4.3.1 Lenguaje de Marcas de Hipertexto HTML v5.0**

Lenguaje de Marcas de Hipertexto (*HTML*, por sus siglas en inglés), es el lenguaje de marcado predominante para la construcción de páginas web. En sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto e imágenes. Es un estándar a cargo de la *World Wide Web Consortium (W3C*<sup>3</sup>, por sus siglas en inglés), organización dedicada a la estandarización de varias tecnologías ligadas a la web en lo referente a su escritura e interpretación. Es un lenguaje de hipertexto, que permite escribir texto de forma estructurada y agradable. Su nivel de complejidad es bajo por lo que no necesita de grandes conocimientos cuando se cuenta con un editor de páginas web. Sus archivos son pequeños, permite un despliegue rápido, es fácil de aprender y lo admiten la mayoría de los navegadores (Lapuente, 2013).

#### **1.4.3.2 AngularJs Material v1.1.4**

AngularJs Material es tanto un marco de componentes de interfaz de usuario como una implementación de referencia de la especificación de diseño de materiales de Google. Este proyecto proporciona un conjunto de componentes de interfaz de usuario reutilizables, bien probados y accesibles basados en el diseño de materiales (GOOGLE, 2017).

#### **1.4.3.3 Procesador de Hipertexto PHP v5.4**

"*Hypertext Preprocessor*" es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. El cliente solamente recibe el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar que código ha producido el resultado recibido (Bakken et al, 2001).

El uso de este lenguaje provee las siguientes ventajas:

---

<sup>3</sup> Consorcio internacional que produce recomendaciones para la World Wide Web (WWW).

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL4 y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).
- Posee una amplia documentación en su página oficial ([www.php.org](http://www.php.org)), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

### 1.4.3.4 JavaScript v1.8

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, es utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y lenguaje C. Se utiliza principalmente en su forma del lado del cliente implementado como parte de un navegador web, permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Es orientado a objetos, al disponer de herencia, la cual se realiza siguiendo el paradigma de programación basada en prototipos y las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. JavaScript es una excelente solución para poner en práctica la validación de datos de un formulario en el lado del cliente. (Pérez, 2008)

---

<sup>4</sup> Sistema de gestión de bases de datos relacionales.

#### 1.4.3.5 Lenguaje de Consulta Estructurado SQL

Lenguaje de Consulta Estructurado (*SQL*, por sus siglas en inglés), es un lenguaje declarativo de alto nivel vinculado con la gestión de bases de datos de carácter relacional, que permite la especificación de distintas clases de operaciones entre estas. *SQL*, al manejar conjuntos de registros y no registros individuales, ofrece una elevada productividad en la codificación y en la orientación a objetos. Se habla por tanto de un lenguaje normalizado que permite trabajar con varios lenguajes de programación (ejemplo *PHP*) en combinación con cualquier tipo de base de datos (Martínez, 2015).

#### 1.4.4 Lenguaje Unificado de Modelado UML 2.5.1

El Lenguaje Unificado de Modelado es un lenguaje estándar para escribir planos de software. *UML* puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. *UML* es apropiado para modelar desde sistemas de información en empresas hasta aplicaciones distribuidas basadas en la web. Es un lenguaje de modelado que proporciona un vocabulario y reglas para su uso en la representación conceptual y física de un sistema (Rebaza, 2009).

Sus funciones son resumibles en cuatro fundamentales:

**Visualizar:** permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

**Especificar:** permite especificar cuáles son las características de un sistema antes de su construcción.

**Construir:** a partir de los modelos especificados se pueden construir los sistemas diseñados.

**Documentar:** los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

#### 1.4.5 Herramientas de desarrollo

Las herramientas informáticas son programas, aplicaciones o simplemente instrucciones usadas para efectuar otras tareas de modo más sencillo. Cada herramienta se crea y diseña para una o varias funciones determinadas, por lo que existen diversos tipos de herramientas informáticas según el campo al que se dediquen (Mastermagazine, 2012).

En el centro SATD de la XETID se emplean para el desarrollo de aplicaciones web las herramientas descritas a continuación:

### 1.4.4.1 Herramienta de Modelado Visual Paradigm v8.0

Herramienta multiplataforma de modelado visual *UML* y una herramienta de Ingeniería de Software Asistida por Computadoras (*CASE*<sup>5</sup>, por sus siglas en inglés) muy potente y fácil de utilizar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción y despliegue. Se puede realizar el modelado, la captura de requisitos, diseño de base de datos, modelado de procesos de negocio. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta también proporciona una mejor interfaz gráfica de usuario y una mayor base de datos de esquema de apoyo (Visual Paradigm, 2010).

### 1.4.4.2 Administrador de base de datos PgAdmin III v1.20.0

Herramienta de código abierto para la administración de bases de datos PostgreSQL. Diseñada para responder a las necesidades de los usuarios, desde escribir consultas *SQL* simples hasta desarrollar bases de datos complejas. La aplicación incluye un editor *SQL* con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar comandos programados y soporte para la replicación. La conexión al servidor puede hacerse mediante la familia de protocolos de internet (*TCP/IP* por sus siglas en inglés) y encriptarse mediante el protocolo de Capa de Conexión Segura (*SSL* por sus siglas en inglés) para mayor seguridad (PgAdmin, 2018).

### 1.4.4.3 Sistema Gestor de Base de Datos (SGBD) PostgreSQL v9.4

PostgreSQL es un sistema gestor de bases de datos objeto-relacional, distribuido bajo la licencia *BSD*<sup>6</sup> y de código fuente libre; por lo que puede ser utilizado, modificado y distribuido por todo el mundo de forma gratuita. Incluye características de la programación orientada a objetos, como: herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Tiene soporte completo para claves

---

<sup>5</sup> Conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

<sup>6</sup> Licencia de software otorgada principalmente para los sistemas desarrollado bajo *Berkeley Software Distribution* (BSD).

foráneas, uniones, vistas, disparadores y procedimientos almacenados. PostgreSQL utiliza un modelo Cliente-Servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Martínez, 2015).

#### **1.4.4.4 Entorno Integrado de Desarrollo NetBeans v8.2**

Plataforma de desarrollo modular para una amplia gama de tecnologías de desarrollo de aplicaciones. Incluye un editor avanzado en varios idiomas, depurador y perfiles, así como herramientas para el control de versiones y la colaboración de desarrolladores. Admite diferentes lenguajes de programación mediante los que se pueden crear aplicaciones gráficas. Posibilita que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Las aplicaciones construidas a partir de módulos se pueden extender agregándole nuevos módulos, debido a que permiten ser desarrollados independientemente. Las aplicaciones implementadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software (Oracle Corporation, 2016).

#### **1.4.4.5 Servidor para aplicaciones web Apache v2.2**

Un servidor web notable por su utilidad es *Apache HTTP Server* comúnmente conocido como Apache. Es publicado bajo la licencia Apache, esta permite la distribución de derivados de código abierto a partir de su código fuente original. Es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo los auspicios de la *Apache Software Foundation*.

Entre algunas de las ventajas que posee se pueden encontrar las siguientes: es modular, flexible rápido y eficiente. Es utilizado para mostrar páginas web con contenidos estáticos y dinámicos. Cuenta con una amplia aceptación en la red por lo que se puede conseguir ayuda fácil. Por todas las características y ventajas que ofrece, es el servidor web que se escoge para el desarrollo del sistema, pues entre todas ventajas se puede mencionar además que es un servidor web HTTP para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido (Apache Foundation, 2017).



### 1.5 Conclusiones del capítulo

Posterior al análisis realizado en el capítulo se concluye:

1. El estudio realizado a los temas relacionados con los sistemas de apoyo a decisiones sirvió de punto de partida para fundamentar la base teórica de la investigación.
2. El análisis de las principales características y aplicaciones de los sistemas homólogos, así como la experiencia adquirida, posibilitó identificar elementos a tener en cuenta en el desarrollo del Sistema de Gestión de Ilegalidades de la DPPF de Granma.
3. La caracterización de las tecnologías y metodologías establecidas por XETID permitieron identificar las ventajas de su uso en el desarrollo del Sistema de Gestión de Ilegalidades de la DPPF de Granma.

## Capítulo 2: Análisis y diseño de la propuesta de solución

### 2.1 Introducción

Este capítulo contiene lo referente al análisis y diseño de la solución informática que permitirá gestionar las ilegalidades de la DPPF de Granma. Se muestra el modelado de negocio con el modelo conceptual y una descripción detallada de los procesos del negocio sobre el que se va a trabajar. Contiene todo lo referente al levantamiento de los requisitos funcionales y no funcionales, su especificación, una breve descripción y su administración. El capítulo también contiene todo el diseño dentro del cual se recogen los diagramas de negocio y de base de datos necesarios para implementar la solución informática.

### 2.2 Modelado de los procesos de negocio

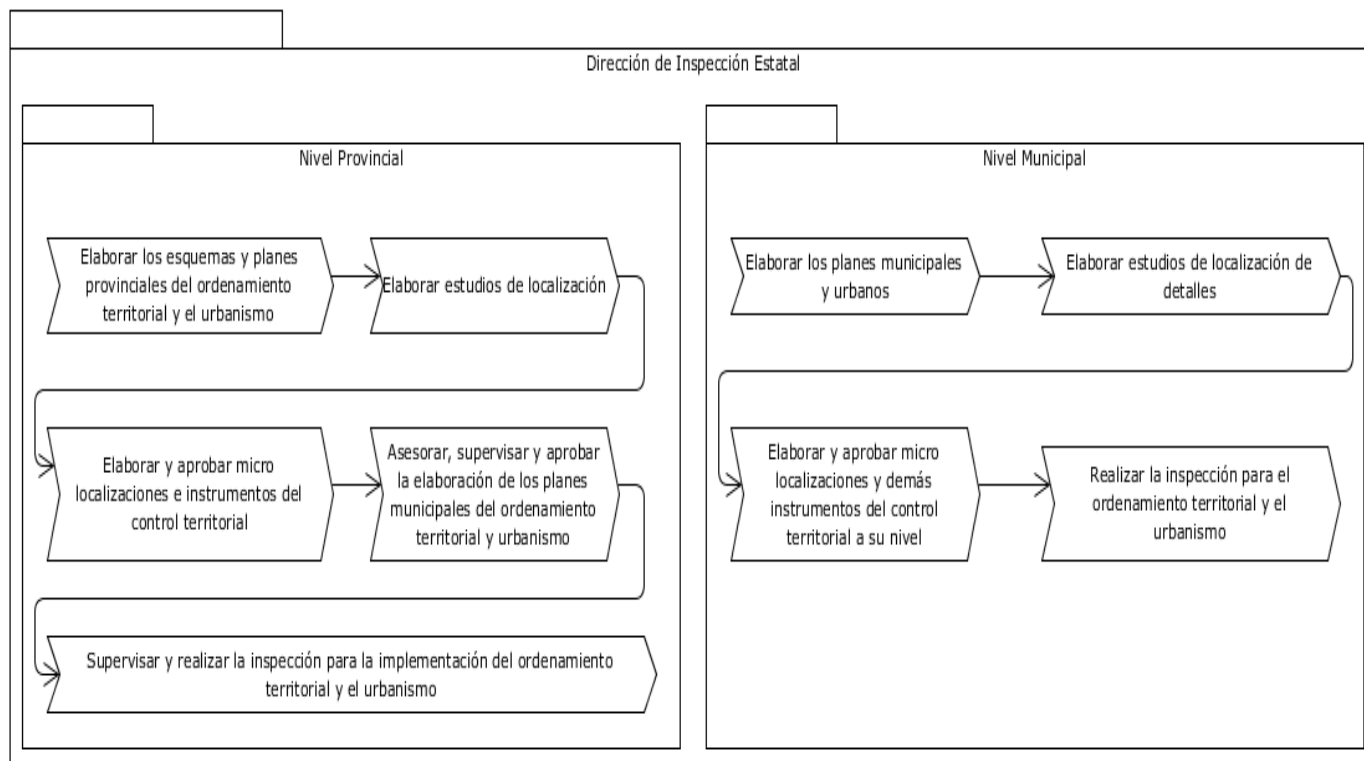
En este epígrafe se muestra cómo está conformado el negocio y la relación entre los diferentes conceptos dentro del mismo. También se describe el funcionamiento de los procesos que se realizan en la DIE de la DPPF de Granma y en sus DMPF para levantar las incidencias o ilegalidades dentro de sus respectivos territorios. Además, se expone el funcionamiento de los procesos de gestión de dichas ilegalidades para darle la resolución final de acuerdo a las leyes establecidas por el estado y el IPF.

#### 2.2.1 Identificación de los procesos de negocio

Una parte fundamental de la modelación de los procesos de negocio su identificación para obtener de este modo una visión global de la entidad y las relaciones establecidas entre los procesos. Una herramienta muy empleada para acometer este objetivo son los mapas de procesos de negocio.

Los mapas de procesos de negocios tienen como propósito representar gráficamente la estructura de los procesos de la entidad y deben ser desarrollados por niveles dado que un proceso puede contener varios subprocesos (Curbelo Oliva et al, 2017).

En este mapa de procesos se describe el negocio y el flujo de trabajo que sigue la DIE de la DPPF de Granma y en sus DMPF.



**Figura 2: Mapa de procesos de negocio**

**Fuente: Creación de los autores**

Como se puede observar en la Figura 2: Mapa de procesos de negocio, la provincia tiene el papel de supervisar a los municipios. Los mismos son los encargados de la realización de la inspección para el ordenamiento territorial y el urbanismo; proceso que se desea informatizar con la propuesta de solución. Vale destacar que siguiendo esta estructura las DPPF son a su vez controladas por el IPF al cual deben entregar los reportes.

Para lograr una mayor comprensión de los términos específicos del negocio se define el siguiente Glosario de términos.

**Inspección para el ordenamiento territorial y el urbanismo:** proceso que comprende el flujo de trabajo de las ilegalidades desde que se detectan por el inspector hasta que se cierran por el especialista de la DPPF.

**Plan municipal:** comprende los reportes referentes a las ilegalidades pertenecientes al PIEI y a las nuevas detectadas a nivel municipal.

**Estudios de localización:** estudios estadísticos realizados a una localidad específica con el objetivo de detectar el índice de ilegalidades, cantidad de ilegalidades por categoría, cantidad de ilegalidades dentro del PIEI y cantidad de ilegalidades de tipo nueva detectadas.

**Estudios de localización de detalles:** estudios estadísticos realizados a una localidad específica por un indicador de interés por el especialista de la DPPF.

**Micro localizaciones:** áreas geográficas seleccionadas dentro de un consejo popular para su análisis.

### ***2.2.2 Especificación de los procesos de negocio***

En esta actividad se realiza una descripción detallada de los procesos de negocio mediante una representación gráfica de las actividades que ocurren en cada proceso. Estos diagramas facilitan la interpretación de las actividades en su conjunto, permitiendo visualizar el flujo y la secuencia de las mismas, incluyendo los responsables de cada una de ellas, las entradas y salidas necesarias para la actividad, así como los límites del mismo (PMOINFORMATICA, 2016).

A continuación, se describe el proceso de **inspección para el ordenamiento territorial y el urbanismo** a través del Diagrama de procesos de negocio.

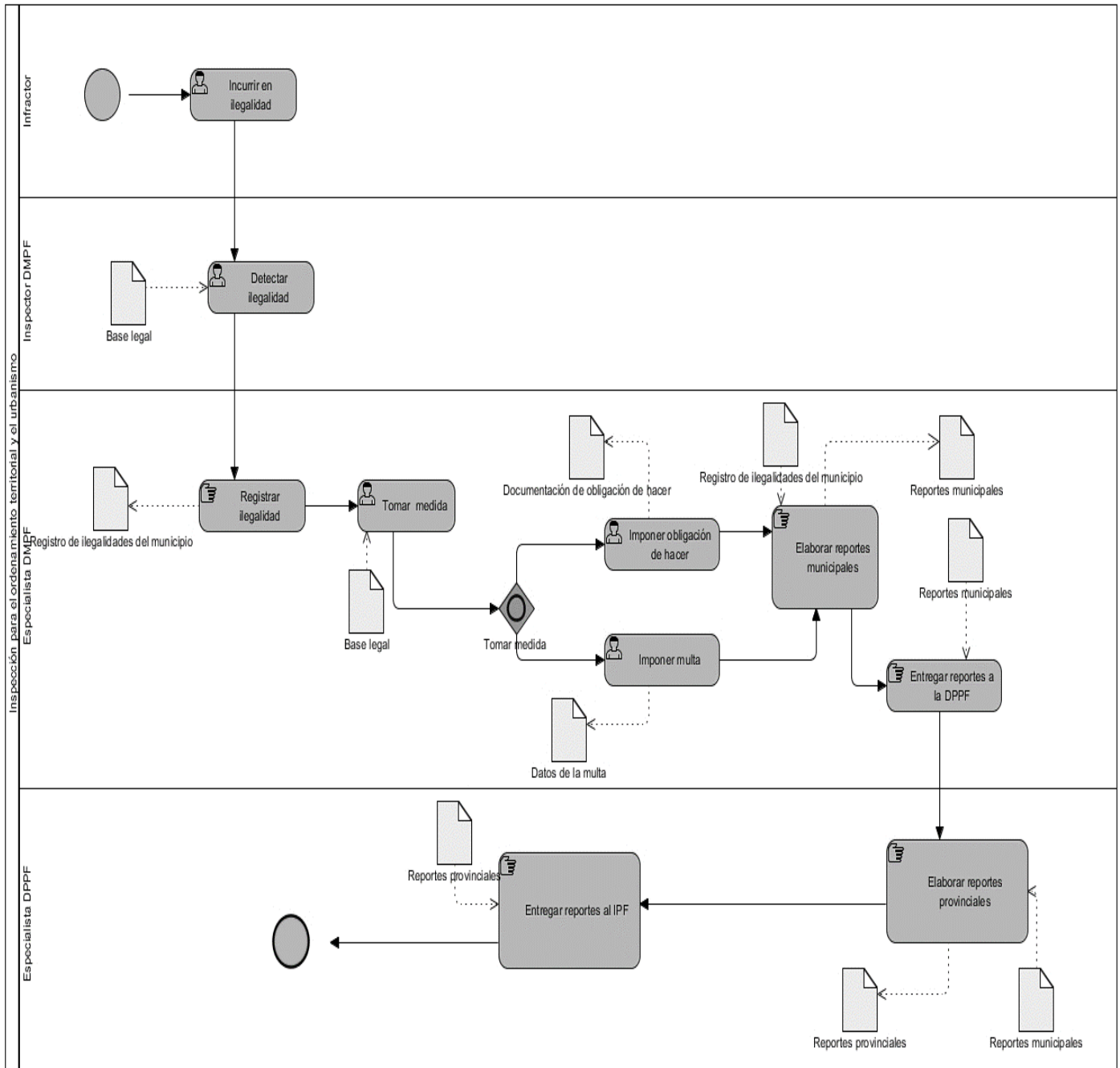


Figura 3: Diagrama de actividad

Fuente: Creación de los autores

### 2.2.3 Especificación de los conceptos del dominio del problema

A continuación, se muestra el modelo conceptual donde se aprecian los diferentes conceptos asociados a la gestión de ilegalidades por parte de la DIE de la DPPF de Granma y la relación estructural entre los mismos.

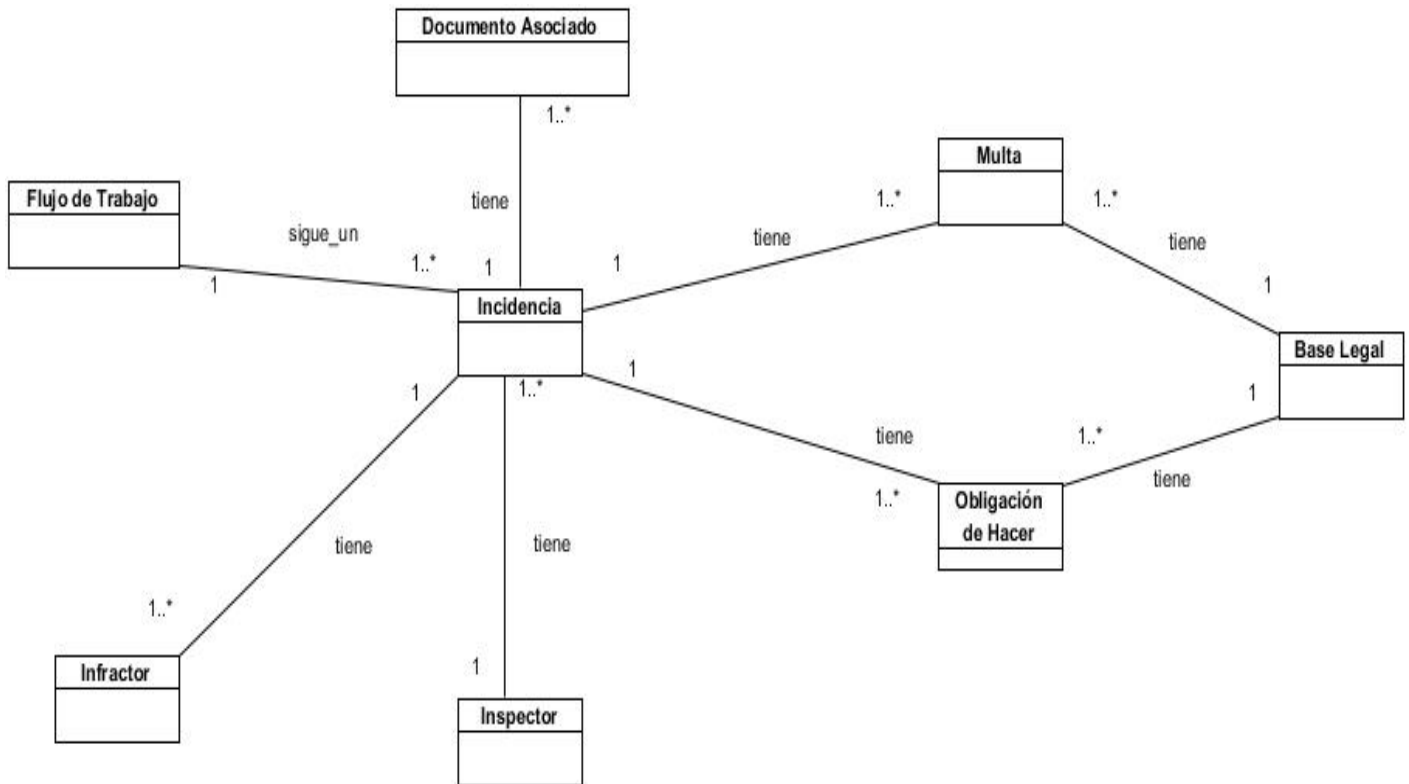


Figura 4: Mapa conceptual

Fuente: Creación de los autores

Para un mayor entendimiento del modelo de dominio y con el objetivo de lograr un lenguaje que sea común para la comprensión del contexto del sistema se muestran a continuación los conceptos más importantes utilizados en el modelo:

- **Incidente:** ilegalidad detectada.

- **Documento asociado:** documentos asociados a una ilegalidad.
- **Infractor:** personas naturales o jurídicas que incurren en una ilegalidad.
- **Multa:** penalización impuesta al infractor.
- **Base legal:** resoluciones y leyes que regulan las medidas impuestas a los infractores.
- **Inspector:** trabajador de la DPPF responsable de detectar las ilegalidades.
- **Flujo de trabajo:** ciclo de vida de una ilegalidad desde que se registra hasta que se cierra y de los conceptos asociados a ella.
- **Obligación de hacer:** acción que debe llegar a cabo el infractor para resolver la ilegalidad.

### 2.3 Descripción de la propuesta de solución

La descripción de cualquier producto o solución informática es una idea, modelo o visión del software que se quiera implementar y esto permite que el cliente y el equipo de trabajo puedan imaginar cómo quedará el producto y la utilidad del mismo.

El Sistema de Gestión de Ilegalidades de la DPPF de Granma aportará las siguientes ventajas al negocio:

1. El sistema centralizará el almacenamiento de las ilegalidades en una base de datos única, lo que disminuirá el riesgo de la duplicidad y pérdida de la información.
2. Al utilizar tecnologías web el sistema brindará constante retroalimentación entre los diferentes niveles dentro de la DPPF de Granma.
3. Los reportes estadísticos que solicita el IPF serán generados automáticamente en formato Excel. Dichos reportes contendrán el formato establecido y los exportará el sistema con una consulta directa a la base de datos, esto eliminará el error humano de este proceso y dará un mayor nivel de confiabilidad a los reportes estadísticos.
4. Al utilizar tecnologías web y un módulo de Seguridad la información solo podrá ser gestionada por los usuarios autorizados mejorando el aseguramiento de la integridad de la información.
5. La información se actualizará en tiempo real para todos los niveles de la DPPF de Granma eliminando los niveles de tramitación y aumentando el rendimiento en el proceso de gestión de las ilegalidades.

6. La introducción del sistema contribuirá a los procesos relacionados con la gestión de las ilegalidades, debido a que automatizará la generación de reportes y la tramitación de las ilegalidades entre los niveles pertinentes dentro de la DPPF.
7. El sistema facilitará los procesos de registro de una ilegalidad, asignación de multa y la consulta de la base legal para la toma de decisiones.
8. El sistema permitirá llevar un registro de los inspectores responsables.
9. Se utilizará un diseño *responsive* o adaptativo para su uso en dispositivos móviles o *tablets*.

Para una mejor comprensión de la propuesta de solución a continuación se ofrecen los artefactos ingenieriles que ayudan a entender de una forma técnica el desarrollo de la solución.

### 2.4 Requisitos

En este epígrafe se evidencia el trabajo con los requisitos del sistema, la fuente para su obtención y los métodos utilizados para el levantamiento de los mismos. Contiene la especificación de los requisitos donde se clasifican en funcionales y no funcionales, así como la descripción de los principales requisitos funcionales del sistema.

#### 2.4.1 Fuentes y técnicas para la obtención de requisitos

La fuente principal para la obtención de los requisitos de sistema fue la documentación que brindó la DPPF de la provincia Granma. La técnica utilizada fue el análisis de dicha documentación, lo que permitió tener una visión bastante amplia del funcionamiento del negocio. Posterior a este análisis se procede a obtener los diferentes requisitos funcionales (RF) y los requisitos no funcionales (RNF).

La **técnica** de obtención de requisitos **análisis de documentación**: Consiste en obtener la información sobre los requerimientos funcionales y requerimientos no funcionales de software a partir de documentos que ya están elaborados. Dicha técnica es útil cuando los expertos en la materia no están disponibles para ser entrevistados o ya no forman parte de la organización (PMOINFORMATICA, 2017).

#### Documentación consultada:



1. Reportes generados por el proceso de gestión de ilegalidades.
2. Correos electrónicos de los directivos de la DPPF y su especialista funcional.
3. Documentación elaborada por el especialista funcional donde plasma lo que se esperaba del sistema.
4. Documentos explicativos sobre el proceso de negocio preparado por el especialista funcional de la DPPF de Granma.

### **2.4.2 Identificación y clasificación de requisitos**

En esta actividad se identifican los requisitos del sistema definiendo cómo el producto se ajusta a las necesidades del negocio y cómo va a ser utilizado por los usuarios. Además, se definen los requisitos no funcionales, encargados de definir el comportamiento del sistema y sus cualidades.

#### **2.4.2.1 Requisitos funcionales**

En este epígrafe se pueden observar los requerimientos funcionales del sistema agrupados de acuerdo a las características de los mismos para su mejor entendimiento y posterior especificación. Entiéndase por incidente la ilegalidad registrada y por concepto todos los valores a los que se le realiza la acción ya sea incidente, multa, inspector o archivos.

#### **Gestionar concepto:**

##### **RF 1 Gestionar incidente.**

- RF 1.1 Adicionar incidente.
- RF 1.2 Modificar incidente.
- RF 1.3 Eliminar incidente.
- RF 1.4 Ver detalles de un incidente.

##### **RF 2 Gestionar archivos.**

- RF 2.1 Adicionar archivos.

- RF 2.2 Eliminar archivos.
- RF 2.3 Ver detalles de un archivo.

**RF 3      Gestionar infractor.**

- RF 3.1 Adicionar infractor.
- RF 3.2 Modificar infractor.
- RF 3.3 Eliminar infractor.
- RF 3.4 Ver detalles de un infractor.

**RF 4      Gestionar inspector.**

- RF 4.1 Adicionar inspector.
- RF 4.2 Modificar inspector.
- RF 4.3 Eliminar inspector.
- RF 4.4 Ver detalles de un inspector.

**RF 5      Gestionar multa.**

- RF 5.1 Adicionar multa.
- RF 5.2 Modificar multa.
- RF 5.3 Eliminar multa.
- RF 5.4 Ver detalles de una multa.

**RF 6      Buscar concepto:**

- RF 6.1 Buscar incidente.
- RF 6.2 Buscar inspector.
- RF 6.3 Buscar multa.
- RF 6.4 Buscar infractor (a través del componente externo: Ciudadano).

**RF 7      Asociar concepto:**

- RF 7.1 Asociar infractor a un incidente.
- RF 7.2 Desasociar infractor de un incidente.
- RF 7.3 Asociar base legal a un incidente (a través del módulo pre implementado Base Legal).
- RF 7.4 Desasociar base legal de un incidente.

**RF 8 Exportar concepto:**

- RF 8.1 Exportar listado de inspectores.
- RF 8.2 Exportar listado de multas.
- RF 8.3 Exportar reporte (Resumen).
- RF 8.4 Exportar reporte (Resumen por resolver en el trimestre).
- RF 8.5 Exportar reporte (Resumen declaradas por organismo).
- RF 8.6 Exportar reporte (Resumen por categorías).
- RF 8.7 Exportar reporte (Instrucción 2).

**2.4.2.2 Requisitos no funcionales**

**RNF 1 - Requisitos de hardware.**

**Tabla 3: Requisitos de hardware**

**Fuente: Creación de los autores**

<b>Característica</b>	<b>Servidor de aplicaciones</b>	<b>Servidor de bases de datos</b>
Memoria RAM	8 GB	8 GB
Capacidad de almacenamiento	10 GB	500 GB
Capacidad de procesamiento	3.0 GHz	3.0 GHz
Capacidad de transmisión de datos	1MB	1MB

**RNF 2 - Requisitos de software.**

**Tabla 4: Requisitos de software**

**Fuente: Creación de los autores**

PC Cliente	Servidor de aplicaciones	Servidor de bases de datos
Navegador web compatible con HTML v5.0.  Ofimática.  Soporte para formato PDF.	Sistema operativo Debian server 7.0.  Servidor Web Apache 2.4 o superior, con módulo PHP 5.	PostgreSQL 9.4 como Sistema Gestor de Base de Datos.

Se sugieren dichos requisitos de software y hardware para garantizar el funcionamiento óptimo de la solución informática. Esta conclusión está basada en las experiencias del centro SATD en el despliegue de soluciones semejantes, desarrolladas utilizando las mismas tecnologías y herramientas a emplear para el desarrollo del Sistema de Gestión de Ilegalidades de la DPPF de Granma.

**RNF 3 - Rendimiento/Eficiencia:**

RNF 3.1 - Los tiempos de respuesta y velocidad de procesamiento de la información deben ser rápidos, no mayores de 5 segundos para las actualizaciones y 10 para las recuperaciones excepto algunas consultas complejas.

RNF 3.2 - El tiempo máximo de espera está en dependencia de las operaciones realizadas, generalmente no deben exceder los 30 segundos.

**RNF 4 – Aplicación:** la cantidad máxima de usuarios concurrentes en el sistema está regulada principalmente por las prestaciones de los servidores tanto el servidor Web como el de Bases de datos, a mayores prestaciones aumenta la cantidad de usuarios soportados por el sistema. Para los requerimientos mínimos descritos en este documento de hardware pueden interactuar con el sistema un total de 200 usuarios sin afrontar dificultades de rendimiento y hasta 250 usuarios obteniendo tiempos de respuesta fuera de los rangos establecidos anteriormente.

**RNF 5 - Confiabilidad:** la información del sistema debe estar disponible todo el tiempo que el usuario solicite. El sistema debe prever casos de excepciones y errores.

**RNF 6 - Seguridad:**

**RNF 6.1** - La información debe estar protegida contra accesos no autorizados utilizando mecanismos de validación tales como: cuenta, contraseña y nivel de acceso. De esta manera cada usuario debe tener disponible solamente las opciones relacionadas con su actividad y datos de acceso propios, garantizando así la confidencialidad.

**RNF 6.2** - El sistema deberá contar con 3 roles (administrador, especialista e inspector).

**RNF 6.3** - Se debe utilizar el protocolo de comunicación HTTPS<sup>7</sup>.

**RNF 6.3 - Integridad:** la información manejada por el software es objeto de cuidadosa protección contra la corrupción y estados inconsistentes, mediante el mantenimiento de las bases de datos, así como la salva de la información se realiza a través de cada componente, de igual manera el origen y autoridad de los datos.

**RNF 6.4 - Disponibilidad:** tanto el sistema como los datos deben estar disponibles para los usuarios autorizados en el momento que así lo requieran.

**RNF 6.5** - Solo deben tener acceso al componente las personas calificadas para interactuar con el marco de trabajo el cual contiene el componente externo de seguridad que garantiza la gestión de roles y usuarios para el sistema.

**RNF 6.6 - Confidencialidad:** la información manejada por el sistema debe estar protegida de accesos no autorizados y divulgación.

**RNF 6.7** – El sistema debe utilizar el ORM Doctrine para evitar la Inyección SQL.

---

<sup>7</sup> Protocolo seguro de transmisión de hipertexto.

**RFN 7 - Usabilidad:**

**RFN 7.1** - El sistema debe emplear barras de progreso para indicar el estado de los procesos que por su complejidad requieran de un tiempo de procesamiento apreciable por los usuarios.

**RFN 7.2** - Se debe utilizar el idioma español para todo el sistema.

**RFN 7.3** - El software debe tener siempre visible la opción de Ayuda, lo que posibilita un mejor aprovechamiento de sus funcionalidades.

**RFN 8 - Restricciones de diseño:** el producto de software está diseñado sobre una arquitectura cliente-servidor y emplea los estándares establecidos en la entidad.

**RFN 9 - Interfaz:** para el desarrollo del sistema se deben utilizar estándares empleados en la empresa. La distribución mínima de la pantalla debe ser de 800 x 600 y debe soportar los periféricos mouse y teclado.

**RFN 10 - Portabilidad:** el sistema debe ser multiplataforma (Linux y Windows fundamentalmente). Al tratarse de una aplicación web puede accederse desde cualquier dispositivo que cumpla con las especificaciones de hardware y software establecidas.

**RFN 12 - Licencias y patentes:** el sistema utilizará la licencia *General Public License (GPL)*.

**RFN 13 - Aplicación de estándares:** el sistema deberá ser regido por diferentes estándares establecidos:

- 1 Estándar para la modelación de procesos de negocio y gestión de requisitos.
- 2 Estándar de codificación.
- 3 Estándar de documentación.

**2.4.3 Especificación de requisitos funcionales**

El propósito de la definición de requisitos es especificar las condiciones o capacidades que el sistema debe cumplir y las restricciones bajo las cuales debe operar, logrando un entendimiento entre el equipo de desarrollo y el especialista funcional, y especificando las necesidades reales de forma que satisfaga sus

expectativas (PMOINFORMATICA, 2017). Para el logro de esta actividad se utilizaron los diagramas de requisitos y sus correspondientes descripciones de requisitos como propone la metodología PRODESOF. Para consultarlos, remitirse al Anexo 1: Diagramas de requisitos funcionales (DRF) y descripción de requisitos funcionales (DERF). Además consultar los prototipos de interfaz de usuario en el Anexo 2 Prototipos de interfaz de usuario (PIU).

## **2.5 Análisis y diseño**

En este epígrafe se hará alusión a los diferentes estilos y patrones del diseño arquitectónico empleados como buenas prácticas para el desarrollo del Sistema de Gestión de Ilegalidades para la DPPF de Granma.

### **2.5.1 Diseño arquitectónico**

La arquitectura de un software es la estructura u organización de un sistema que incluye los componentes de este, las propiedades visibles externas de esos componentes y las relaciones que existen entre ellos. Su diseño considera dos niveles: el diseño de datos y el diseño arquitectónico. El primero permite representar los componentes de la arquitectura y las definiciones de clases, el segundo se concentra en representar la estructura de software, sus propiedades e interacciones (Curbelo Oliva et al, 2017).

La Arquitectura enfoca el diseño del software desde varias perspectivas, de la calidad de estos procesos dependerá el éxito del diseño detallado, implementación e integración de la aplicación (CeO2 Software, 2010). Este epígrafe contiene todo lo referente a el diseño arquitectónico dígase vistas, estilos utilizados y patrones de la arquitectura.

#### **2.5.1.1 Estilo arquitectónico**

En el desarrollo de la investigación el estilo arquitectónico propuesto es Cliente-Servidor, que es un modelo de aplicación distribuido en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes (CeO2 Software, 2010). Un cliente realiza peticiones a otro programa conocido como servidor que le da respuesta. Normalmente, el servidor es una máquina, de altas prestaciones, que puede actuar como servidor de aplicaciones, de depósito de datos, o sencillamente para brindar determinados servicios. Por otro lado, los clientes suelen ser estaciones de

trabajo que realizan varias solicitudes o peticiones al servidor. Ambas partes deben estar conectadas entre sí mediante una red. La Figura 5 es una representación gráfica del estilo arquitectónico.

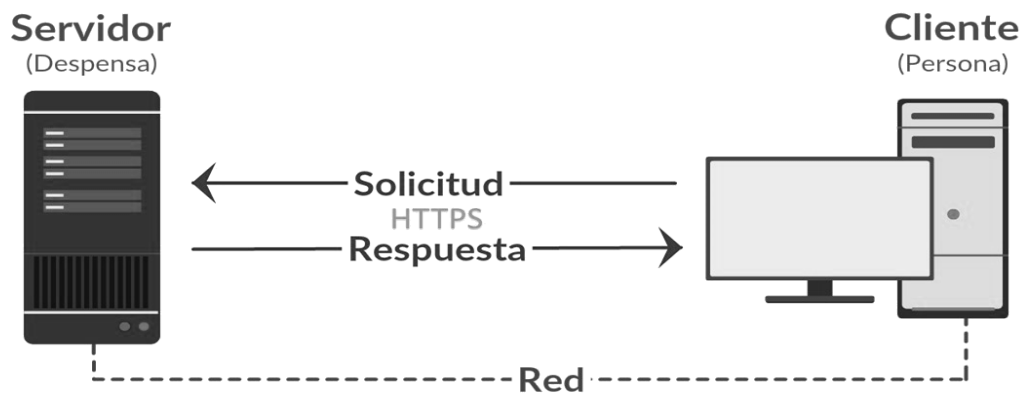


Figura 5: Arquitectura Cliente – Servidor

Fuente: Creación de los autores

Cliente-Servidor es una arquitectura muy usada, debido a las ventajas y potencialidades que proporciona. La separación entre cliente y servidor es de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un solo programa. Los tipos específicos de servidores incluyen los servidores web, de archivo, correo y bases de datos (Aplicaciones Web, 2008).

Se decide la utilización de este estilo arquitectónico debido a la forma de despliegue del Sistema de Gestión de Ilegalidades de la DPPF de Granma. El sistema debe ser web para poder manejarlo a los diferentes niveles dentro de la DPPF a nivel provincial. La información debe ser tratada al mismo tiempo por varios usuarios y actualizada en tiempo real.

### 2.5.1.2 Patrón de arquitectura

Se define el uso de los patrones: **Arquitectura en Capas** y **Modelo-Vista-Controlador**.



**Arquitectura en Capas:** organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior (MICROSOFT, 2018).

Se utilizan 2 capas principales: capa de presentación (web) y capa de lógica de negocio (app) como se muestra en la Figura 6.

El patrón conocido como **Modelo-Vista-Controlador (MVC)** separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes.

- **Modelo:** administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- **Vista:** maneja la visualización de la información.
- **Controlador:** interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado (MICROSOFT, 2018).

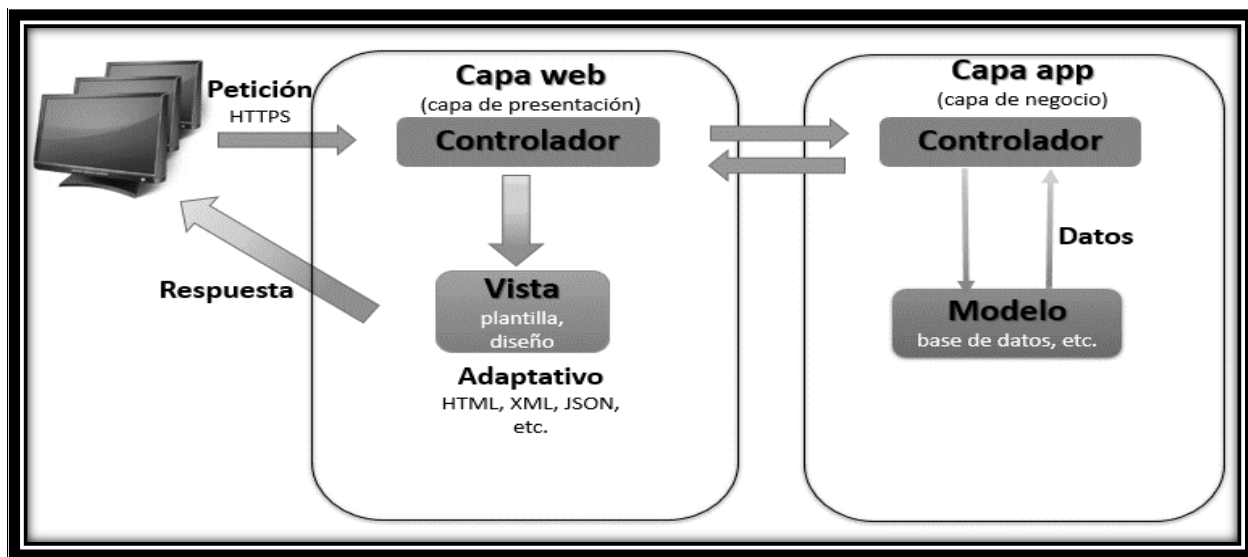


Figura 6: Representación del Patrón N-Capas con Modelo Vista Controlador

Fuente: Creación de los autores



Este patrón se emplea en cada una de las capas de manera distinta. En la **capa de presentación** se encuentra un controlador encargado de recibir las peticiones de las PC clientes y posteriormente enviarlas a la capa de negocio. Una vez recibida la respuesta de la petición el controlador de esa capa construye la vista y muestra los resultados al usuario. Mientras que en la **capa de negocio** se encuentra un controlador encargado de obtener los datos necesarios para responder las peticiones de la capa de presentación del modelo.

### 2.5.1.3 Diagrama de componentes

Este diagrama contiene los componentes del sistema ya sea desarrollados exclusivamente para la nueva solución o reutilizados de sistemas desarrollados anteriormente dentro del centro SATD.

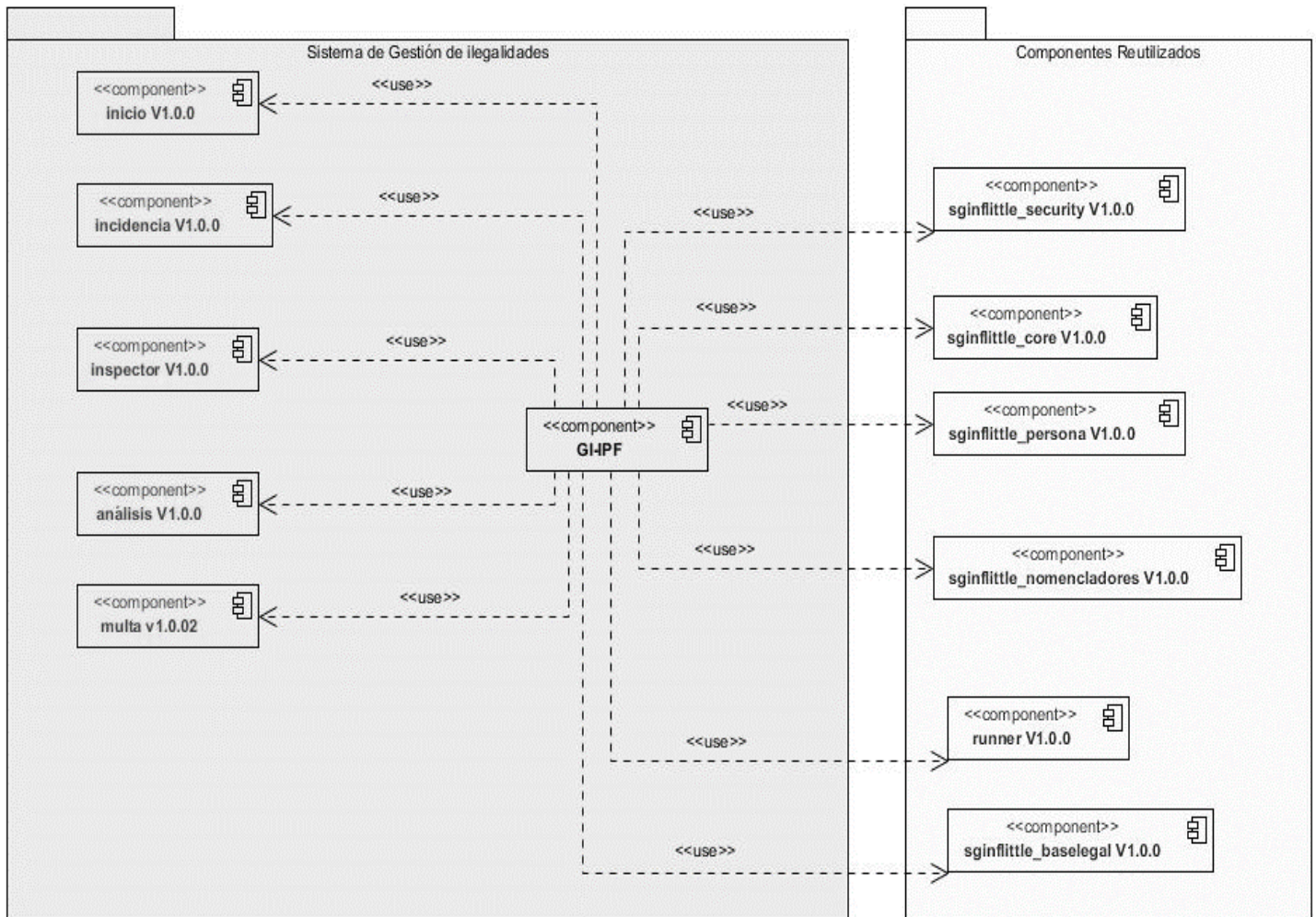


Figura 7: Diagrama de componentes del sistema

Fuente: Creación de los autores

A continuación, se muestra una breve descripción de las funciones que cumplen los componentes reutilizados dentro de la propuesta de solución.

1. **sginflittle\_security**: gestión de los usuarios, los roles y la autenticación. Este módulo se emplea para garantizar el requisito no funcional de seguridad.
2. **sginflittle\_core**: estructura, funcionamiento, validación.
3. **sginflittle\_persona**: búsqueda de personas naturales.

4. **sginflittle\_nomencladores:** nomencladores del sistema.
5. **runner:** notificaciones y alertas.
6. **sginflittle\_baselegal:** gestionar Base Legal.

### **2.5.2 Modelado de datos**

Para el modelado de la base de datos el centro SATD utiliza un híbrido entre bases de datos relacionales y no relacionales, se evidencia la relación entre tablas con el uso de las llaves foráneas como método de unificación entre las mismas y con el objetivo de ganar en rendimiento se hace uso de json<sup>8</sup> donde se almacenan los valores de los nomencladores para minimizar el uso de sentencias *INNER JOIN*<sup>9</sup> debido a que almacena los datos que se necesitarían utilizando estas sentencias.

**A continuación, se muestra uno de los diagramas entidad relación definidos para el Sistema de Gestión de Ilegalidades de la DPPF de Granma (ver**

Anexo 3: Diagramas Entidad Relación. para consultar los demás diagramas).

---

<sup>8</sup> Acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos.

<sup>9</sup> En base de datos es una sentencia que combina cada fila de una tabla con cada una de las filas de la otra tabla.

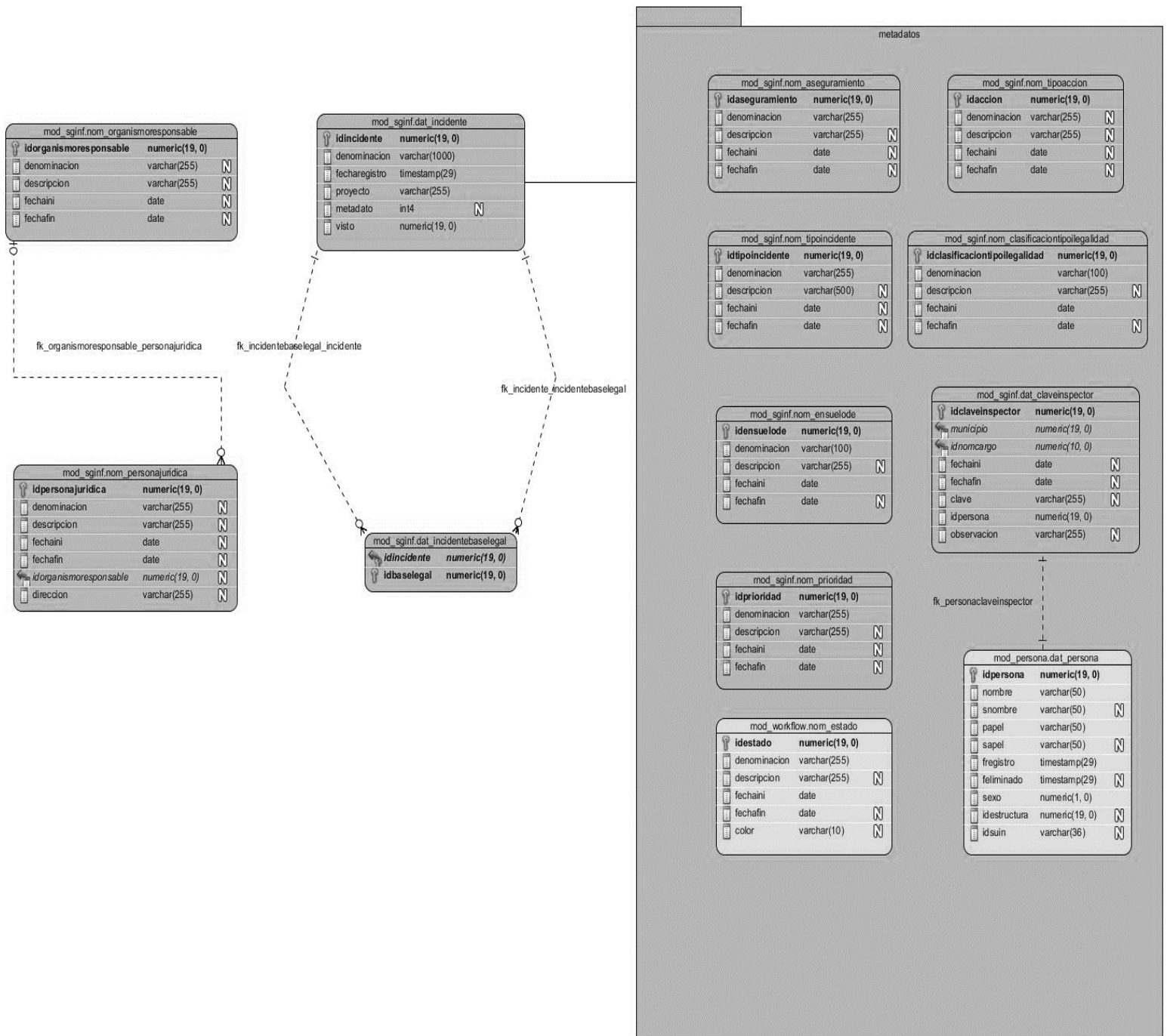


Figura 8: Diagrama entidad-relación mod\_sginf.dat\_incidente

Fuente: Creación de los autores

### 2.5.2.1 Patrones de diseño de bases de datos.

Los patrones de diseño de bases de datos son una plantilla que ya ha sido evaluada como la responsable de resolver un problema, es la guía para el trabajo, permiten crear una de base de datos más fortalecida por el usuario. Para el diseño y la construcción de una base de datos se requiere del mayor análisis posible pues a partir de este diseño se crea la base de datos, en la actualidad suelen ser muy grandes y a veces el trabajo con los patrones de diseño hacen que el trabajo sea más fácil, además asegura un resultado correcto (Blaha, 2010). El patrón empleado en el modelado de la base de datos del sistema de gestión de ilegalidades para la DPPF de Granma fue:

**Llaves subrogadas:** este patrón es muy utilizado por facilitar la interacción con la base de dato en un futuro. El mismo plantea que se genere una llave primara única para cada entidad, en vez de usar un atributo identificador en el contexto dado. El uso del patrón permite que las tablas sean más fáciles de consultar a partir del identificador, pues los tipos de datos son iguales en cada una de las tablas. El patrón se manifiesta en la definición de la llave primaria de todas las tablas del modelo físico porque se generan llaves primarias independientes a los atributos de las entidades.

### 2.5.3 Modelado del Diseño

El modelado del diseño es la actividad técnica que sigue a la selección de arquitectura. Su objetivo es dejar el proyecto preparado para la implementación, parte de los resultados de la fase de arquitectura, y termina con un diseño completo y listo para la implementación. Es decir, se debe ser capaz de implementar un diseño detallado, concentrándose en cuestiones puramente de codificación (Buschmann et al, 2013).

#### 2.5.3.1 Diagrama de clases del diseño

A continuación, se muestra el diagrama de clases del diseño con estereotipos web para el requisito funcional Gestionar Incidencia. Para consultar los demás diagramas, ver Anexo 4: Diagramas de clase del diseño con estereotipos web (DCD).

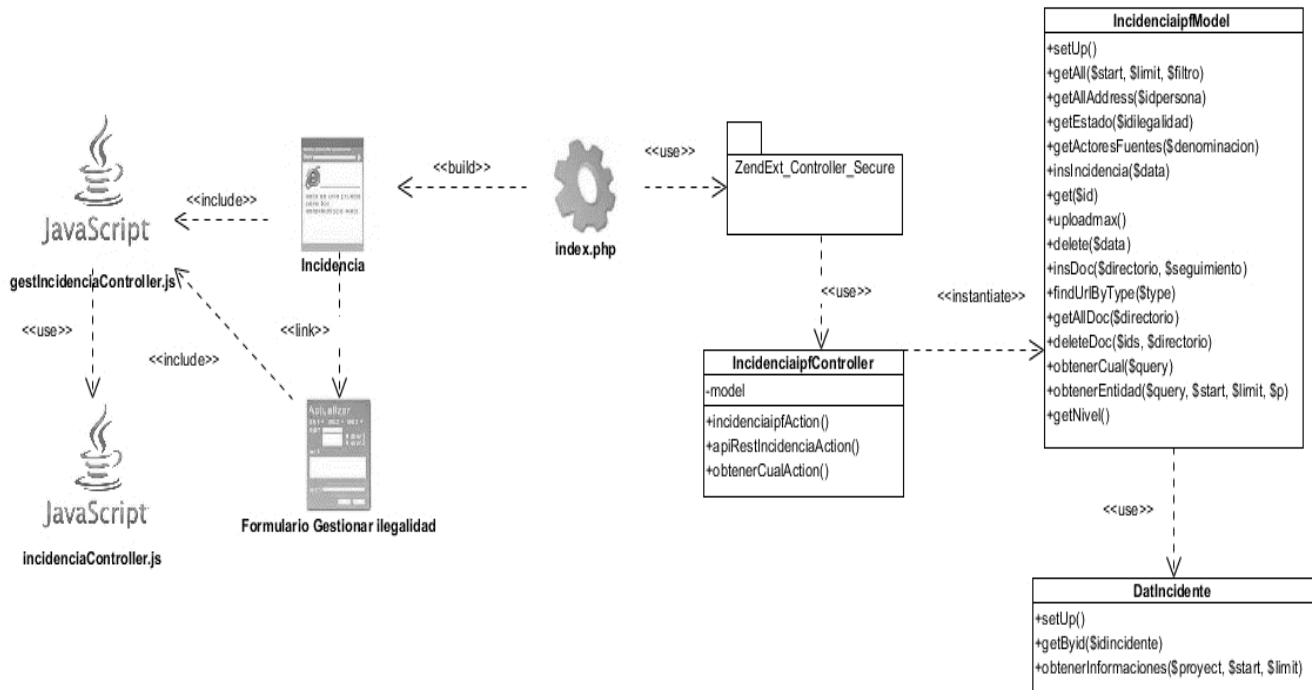


Figura 9: Diagrama de clase del diseño. Gestionar Incidencia

Fuente: Creación de los autores

### 2.5.3.2 Patrones de diseño

Los **patrones de diseño**: expresan un esquema de organización estructural para los sistemas de software. Proporcionan un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluyen reglas y lineamientos para organizar la relación entre ellos (Buschmann et al, 2013). En el diseño del sistema se utilizaron los siguientes patrones.

**Patrones GRASP** (*General Responsibility Assignment Software Patterns*): patrones de software para la asignación de responsabilidades generales (Pressman, 2010).

1. **Creador**: patrón que guía la asignación de responsabilidades relacionadas con la creación de objetos; usado en las clases controladoras, las cuales son las únicas responsables de crear objetos de las clases modelos.



2. **Controlador:** patrón que responde a qué clase debería de encargarse de atender un evento del sistema; usado cuando se hace una petición al servidor, dicha petición llega al controlador frontal, el cual es el encargado de hacer la llamada al método especificado en la clase controladora correspondiente.
3. **Bajo Acoplamiento:** patrón que responde a la asignación de las responsabilidades de forma tal que cada clase se comunique con el menor número de clases, minimizando el nivel de dependencia. Se usa en la integración de componentes, los cuales se comunican mediante servicios IOC (*Inversion of Control*), lo que permite que el componente cambie en su interior, pero se mantiene brindando los mismos servicios.

**Patrones GOF (*The Gang of Four*):** patrones de software que describen soluciones simples y elegantes a problemas específicos en el diseño de software (Guerrero et al, 2013).

1. **Singleton:** patrón que asegura que una clase tiene una sola instancia y proporciona un punto de acceso global a ella. Este patrón brinda una manera centralizada de gestionar un recurso.
2. **Mediador:** patrón que comprende la asignación de responsabilidades entre objetos y algoritmos. Estos no solo conciernen a los objetos y las clases sino también la comunicación entre estas, además caracteriza flujos de control complejos que son difíciles de seguir en tiempo de ejecución. Se evidencia en las librerías, las cuales son mediadoras entre las clases controladoras y los modelos o acceso a datos

## 2.6 Diagrama de despliegue

El diagrama de despliegue es un diagrama estructurado que muestra la arquitectura desde el punto de vista del despliegue. Se utiliza para mostrar la estructura física del sistema, incluyendo las relaciones entre el hardware y el software que se despliega, estas relaciones son representadas por los protocolos de comunicación que se utilizan para acceder a cada uno (Jacobson et al, 2000).

A continuación, se muestra cómo se propone desplegar la propuesta de solución a través del diagrama del despliegue. Se muestran además los protocolos de conexión entre cada componente y los puertos a usar para establecer la comunicación entre ellos.



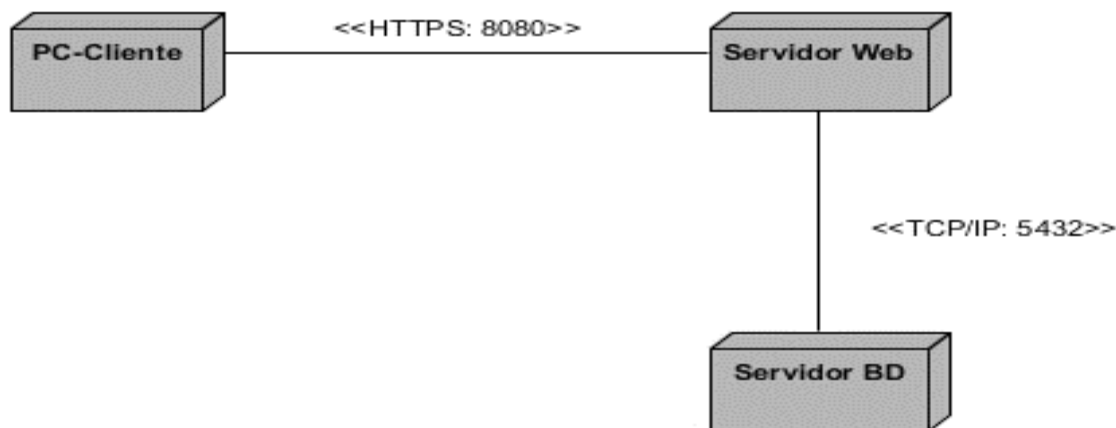


Figura 10: Diagrama de despliegue

Fuente: Creación de los autores

**PC-cliente:** su función es acceder e interactuar con el sistema según sus necesidades. Al estar la aplicación desarrollada sobre la web, la PC cliente necesita de muy pocas prestaciones puesto que solo precisa un navegador web para poder acceder al sistema y realizar las operaciones necesarias.

**Servidor Web:** aquí reside la capa de presentación del Sistema de Gestión de Ilegalidades de la DPPF de Granma, la cual es accedida por las PC-cliente a través de un navegador web.

**Servidor BD:** encargado de almacenar la información registrada y generada por el Sistema de Gestión de Ilegalidades de la DPPF de Granma.

**HTTPS:** protocolo de transferencia de hipertexto seguro, es un estándar de red que sigue el esquema petición respuesta entre un cliente y un servidor.

**TCP/IP:** familia de protocolos de Internet, es un conjunto de estándares de red en la que se basa la red global y que permite la transmisión de datos entre redes de computadoras sin importar el tamaño.

## 2.7 Conclusiones del capítulo

Posterior al análisis del negocio y diseño de la propuesta de solución se arriba a las siguientes conclusiones:

1. El modelado de los procesos de negocio permitió representar cómo se realiza el proceso de gestión de ilegalidades en la DPPF de Granma, lo que permite desarrollar con mayor facilidad la propuesta de solución.
2. El uso de técnicas para la obtención de requisitos permitió comprender, identificar y describir los requerimientos funcionales y no funcionales que deberá cumplir la solución.
3. La utilización de patrones de diseño y arquitectura permitieron diseñar una solución robusta, flexible y escalable.
4. A partir del análisis, el diseño realizado y los artefactos generados quedan sentadas las bases para la implementación y validación de la solución propuesta.

## Capítulo 3: Implementación y validación de la propuesta de solución

### 3.1 Introducción

Los artefactos generados a través del flujo de trabajo de análisis y diseño constituyen la base para las tareas de implementación y pruebas. Las actividades de implementación se contemplan en el desarrollo del sistema que se necesita y las pruebas que son un elemento vital para garantizar la calidad del software y representan una revisión final de las especificaciones, del diseño y la codificación. El objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado y constituyen una actividad en la cual un sistema y componentes son ejecutados bajo condiciones o requerimientos específicos y los resultados son observados y registrados para su posterior evaluación o corrección (Oliva et al, 2017). En el presente capítulo se describe la construcción de la solución propuesta teniendo en cuenta los estándares de codificación. Además, se definen y aplican las pruebas al producto, especificando las evaluaciones realizadas a la propuesta y los resultados obtenidos.

### 3.2 Implementación

La Implementación comienza con el resultado obtenido del diseño detallado. El objetivo principal de este es desarrollar el diseño de la arquitectura propuesta y el sistema como un todo. De forma más específica, los propósitos de la Implementación son:

- 1 Planificar las integraciones de sistema necesarias en cada iteración. Siguiendo para ello un enfoque incremental.
- 2 Implementar clases, componentes y subsistemas encontrados durante el diseño.
- 3 Integrar componentes (Pressman y Troya, 2013).

#### 3.2.1 Estándares de implementación

Un código fuente, debe reflejar un estilo armonioso, que dé la impresión de que fue escrito por un solo programador. La legibilidad del código fuente, repercute directamente en el entendimiento que pueda tener otro programador del mismo, aspecto crucial ya que todo software tiene que someterse constantemente a mantenimiento y mejora de sus funcionalidades, por lo que en la implementación del Sistema de Gestión de

Ilegalidades de la DPPF de Granma se utilizaron los siguientes estándares de implementación definidos por los especialistas de la XETID:

- El nombre de una clase comienza con la primera letra en mayúscula.
- El nombre de los atributos de una clase comienza con la primera letra en minúscula.
- Los nombres de las clases pertenecientes al dominio deben empezar con el prefijo “Dat” y el de las clases del modelo deben tener el prefijo “Model” al final.

En las clases controladoras:

- Los nombres de los métodos serán definidos con la palabra “Action” al final.
- Cada método será público.
- El nombre de cada clase controladora debe terminar con la palabra “Controller”.

Otros estándares que se utilizaron son los siguientes, y se agrupan en las siguientes categorías:

### Estructura:

- Utilizar un solo espacio después de cada delimitador (**coma**).
- Añadir un solo espacio alrededor de los operadores (**==, &&, ||**).
- Añadir una línea en blanco antes de las declaraciones **return**, a menos que el valor devuelto solo sea dentro de un grupo de declaraciones (**tal como una declaración if**).
- Usar llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga.

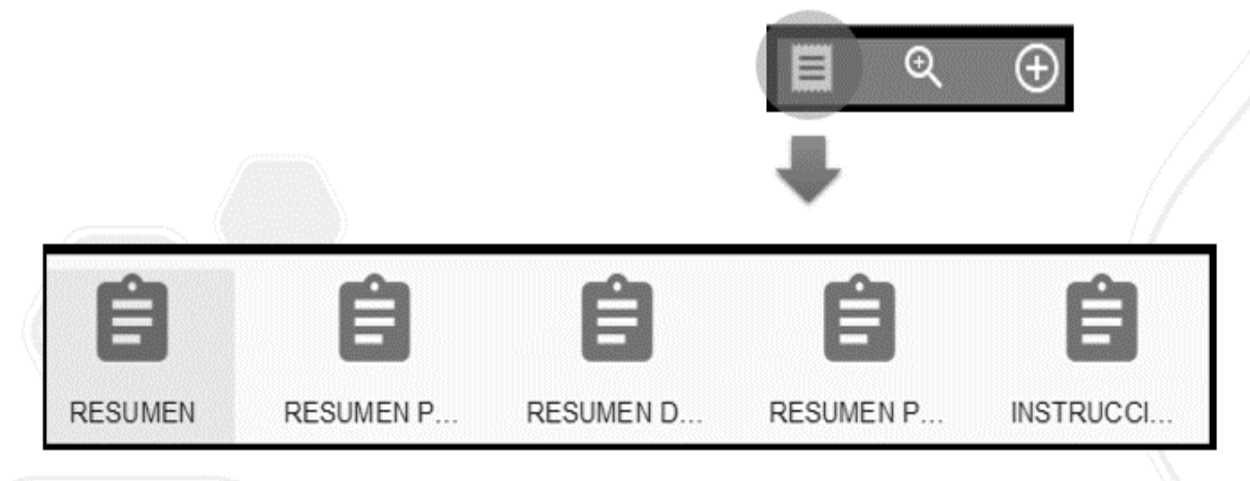
### Convenciones de nomenclaturas:

- Utilizar mayúsculas intercaladas sin guiones bajos, en nombres de variable, función, método o argumentos.
- Usar guiones bajos para nombres de opción y nombres de parámetros.
- Utilizar espacios de nombres para todas las clases.
- Prefijar las clases abstractas con *Abstract*.

- Sufijar las interfaces con *Interface*.
- Sufijar las características con *Trait*.
- Sufijar las excepciones con *Exception*.
- Utilizar caracteres alfanuméricos y guiones bajos para los nombres de archivos.
- Hacer uso de los comentarios utilizando las etiquetas `/** */` para comentarios en varias líneas y `//` para comentarios de una única línea.
- Las estructuras de control deben cumplir un conjunto de normas para su correcto funcionamiento; las estructuras (***if, while, for, entre otras.***) y el primer paréntesis deben tener un espacio intermedio, para no confundirlas con la nomenclatura de las funciones; la llave de apertura (`{`) estará en la primera línea separada por un espacio y es recomendado usar las dos llaves (`{ }`) aunque el código permita no usarlas; y las estructuras ***else*** y ***else if*** serán escritas en la línea siguiente de la llave de cierre anterior (`}`).
- Los valores dentro de un arreglo deben ser separados por un espacio después de la coma. El operador `=>` debe separarse por un espacio a ambos lados.

### 3.3 Interfaces de usuario

Las interfaces de usuario se implementan utilizando formularios, controles u otros elementos, que permitan procesar y dar formato a los datos de los usuarios, así como adquirir y validar los datos entrantes procedentes de éstos (Pressman y Troya, 2013). A continuación, se muestra el ejemplo de la interfaz de usuario del requisito funcional 8.3 exportar reporte; se accede a esta interfaz accionando el botón reporte de la barra de herramientas del sistema.



**Figura 11: Interfaz de usuario Exportar reporte**

**Fuente: Creación de los autores**

Para ver las interfaces de usuario del Sistema de Gestión de Ilegalidades de la DPPF de Granma remitirse al Anexo 5: Interfaces de usuario (IU).

### 3.4 Pruebas de software

Las pruebas del software son la actividad más común de control de la calidad realizada en los proyectos para asegurar el correcto funcionamiento del software. Tienen como objetivos la verificación de la correcta implementación de los requisitos explícitamente establecidos, la adecuada integración de los componentes que conforman el sistema y la ejecución de casos de prueba que permitan detectar el mayor número de no conformidades y corregirlas antes de la entrega del software al cliente. Es importante destacar que las pruebas reducen la probabilidad de que aparezcan defectos ocultos en el software, pero incluso si no se encuentra ningún defecto, nunca será una garantía para la perfección del funcionamiento del software (Curbelo Oliva et al, 2017).

En la siguiente tabla se muestra la estrategia de pruebas de software diseñadas para su aplicación en el Sistema de Gestión de Ilegalidades de la DPPF de Granma.

**Tabla 5: Estrategia de pruebas de software**

**Fuente: Creación de los autores**

Nombre de la prueba	Método	Técnica
Unidad	Caja Blanca	Diagrama Piramidal y Diagrama de Abstracción/Inestabilidad generado por la herramienta Jenkins
Sistema (funcionales)	Caja Negra	Partición de equivalencia
Sistema (carga y estrés)	Caja Negra	Automática utilizando la herramienta Jmeter
Integración	Caja Negra	Casos de prueba de integración
Aceptación	Caja Negra	Alfa

No se comprende la realización de pruebas de seguridad debido a que el Sistema de Gestión de Ilegalidades de la DPPF de Granma utiliza el módulo externo **sginflittle\_security**. Dicho componente fue validado por el equipo de desarrollo del centro SATD y es utilizado en todos los sistemas desarrollados en dicho centro.

### **3.4.1 Pruebas de unidad**

Las pruebas de unidad se centran en la verificación de los elementos más pequeños del software que se puedan probar examinando las estructuras de datos locales. Su objetivo es asegurar que las funcionalidades mantienen su integridad durante los pasos de ejecución de los algoritmos (Pressman y Troya, 2013). Para la prueba se utiliza el método de caja blanca o estructural que se basa en un minucioso examen de los detalles procedimentales del código a evaluar.

Para la realización de estas pruebas se utilizó la herramienta Jenkins, la cual integra varios *plugins*<sup>10</sup>, entre los cuales podemos encontrar el *PHP Depend*. Dicho *plugin* devuelve como resultado, dos tipos de gráficas:

- La primera es la gráfica piramidal, la cual se utiliza para visualizar un sistema completo de software de una manera muy compacta. Para ello se recoge un conjunto de métricas de las categorías de herencia, de enganche, tamaño y complejidad relacionándolas.
- La segunda es un gráfico de Abstracción/Inestabilidad, muestra las dependencias entre paquetes y la abstracción de paquete.

### 3.4.1.1 Resultados de las pruebas de unidad

A continuación, se muestra el gráfico piramidal resultante de aplicar la herramienta Jenkins al Sistema de Gestión de Ilegalidades de la DPPF de Granma.

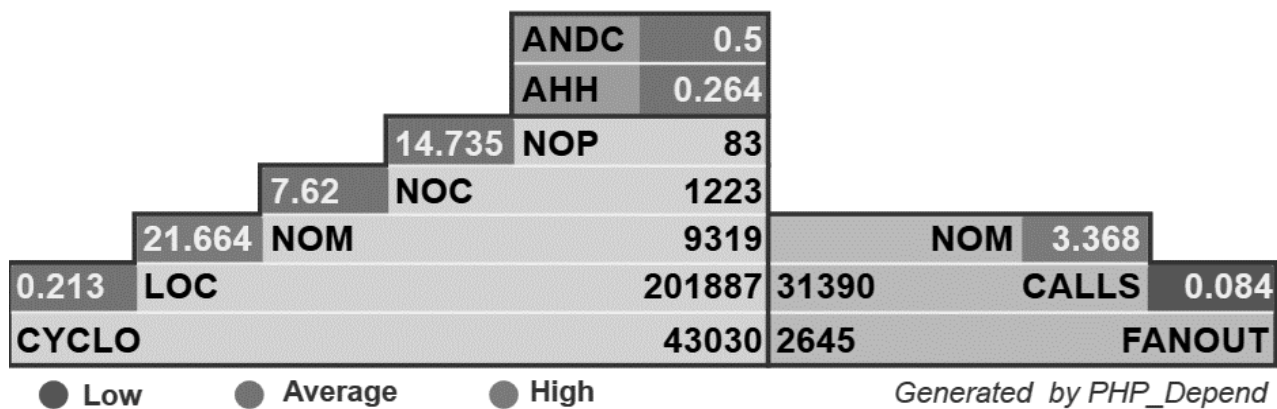


Figura 12: Gráfico piramidal

Fuente: Creación de los autores

<sup>10</sup> Un plugin es aquella aplicación que, en un programa informático, añade una funcionalidad adicional o una nueva característica al software.



Este gráfico se divide en tres partes o categorías: en la parte superior se muestran valores promedios relacionados con la herencia, los valores de la izquierda muestran medidas de la complejidad ciclomática y el tamaño del proyecto, en su parte derecha se refleja el nivel de acoplamiento existente entre clases del módulo.

Para una mayor comprensión del gráfico se muestra a continuación cada una de las categorías con sus respectivas métricas.

### **Herencia:**

- **ANDC:** Promedio de las clases derivadas por clase.
- **AHH:** Promedio de altura de herencia por clase.

### **Complejidad ciclomática y Tamaño de proyecto:**

- **NOP:** Cuenta la cantidad de paquetes.
- **NOC:** Cuenta la cantidad de clases.
- **NOM:** Cuenta la cantidad de métodos.
- **LOC:** Cuenta la cantidad de líneas de código.
- **CYCLO:** Suma la complejidad ciclomática de cada método del proyecto.

### **Acoplamiento:**

- **CALLS:** Cuenta la cantidad de llamadas a métodos diferentes, en caso que el método sea llamado dos veces en la misma clase, se cuenta una vez solamente.
- **FANOUT:** Cuenta el número de clases colaboradoras que utiliza la clase (Reyes, 2015).

Para una mayor comprensión de los valores obtenidos en el gráfico piramidal se ofrece la siguiente tabla donde se pueden apreciar los tres rangos donde pueden agruparse los valores obtenidos.

<b>Métrica</b>	<b>bajo</b>	<b>medio</b>	<b>alto</b>
Cyclo/loc	0.16	0.20	0.24
Loc/nom	7	10	13
Nom/noc	4	7	10
Noc/nop	6	17	26
Calls/nom	2.01	2.62	3.2
Fanout/calls	0.56	0.62	0.68
andc	0.25	0.41	0.57
ahh	0.09	0.21	0.32

**Figura 13: Valores de las métricas evaluadas en el gráfico piramidal**

**Fuente:(Idailis Gutiérrez Reyes 2015)**

Como se puede apreciar en la Figura 12: Gráfico piramidal las métricas referentes a la herencia se encuentran en el nivel medio en relación con la anterior tabla. En cuanto a la complejidad ciclométrica y tamaño de proyecto todas las métricas se encuentran en nivel medio a excepción del *NOM* (cantidad de métodos) que es elevado debido a la complejidad de algunas funcionalidades del sistema y del uso de métodos auxiliares para facilitar la comprensión del código. Denotando esta métrica que el software posee una complejidad moderada y es fácil de entender. Por otra parte, el acoplamiento se comportó bajo para las llamadas de los métodos y alto para el *NOM* por las razones antes expuestas.

A continuación, se muestra el gráfico de Abstracción/Inestabilidad. El mismo hace referencia en su eje vertical al acoplamiento entre paquetes o clases, donde mientras más cercano a uno sea el valor que tome en este eje, mayor inestabilidad tendrá el paquete indicando una mayor dependencia de otros, y en el eje horizontal se muestra la relación entre clases abstractas y clases concretas donde se evidenciará una mayor cantidad de clases e interfaces abstractas si el valor tomado por la abstracción se acerca a uno. La herramienta traza una diagonal entre las dos esquinas, esta línea se llama secuencia principal y representa un promedio entre la abstracción y la inestabilidad, lo que significa que los paquetes cerca de esta línea

tienen una buena mezcla entre estos aspectos y se les llama paquetes equilibrados.

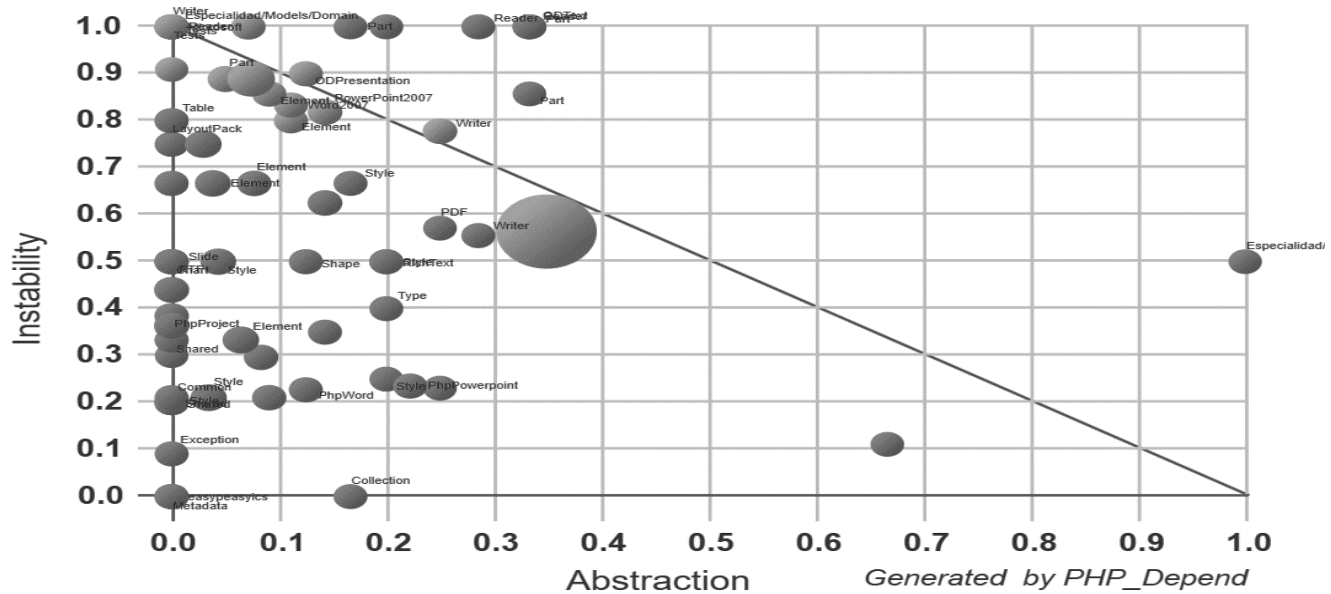


Figura 14: Gráfico de Abstracción/Inestabilidad

Fuente: Creación de los autores

Al analizar este gráfico se puede apreciar que existe poca dependencia entre los paquetes y clases del sistema gracias al uso del patrón de diseño bajo acoplamiento. La mayoría de las clases del sistema son concretas y existe un buen número de paquetes equilibrados.

### 3.4.2 Pruebas de sistema (funcionales)

Las pruebas funcionales tienen por objetivo probar que los sistemas desarrollados cumplen con las funciones específicas para los que han sido creados. Se utiliza el método de caja negra para evaluar y buscar errores de funcionamiento en el sistema (Pressman y Troya, 2013). Para su realización, se diseñaron un grupo de casos de prueba, los cuales tiene como único objetivo, tener la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible. Dichas pruebas por sus características son consideradas además como pruebas unitarias debido a que están dirigidas a las funcionalidades individuales dentro de cada uno de los módulos de la propuesta de solución.

Un diseño de caso de pruebas (DCP) no es más que un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para determinar si el requisito de una aplicación está parcial o completamente satisfecho. Su propósito es especificar una forma de probar el sistema que incluya las entradas, los resultados esperados y las condiciones bajo las que ha de probarse (Pressman y Troya, 2013).

Para confeccionar los casos de prueba de caja negra existen distintas técnicas entre la que se encuentra la de partición de equivalencia, que se basa en identificar las particiones para un sistema o componente. Por cada requerimiento funcional del sistema se generó un caso de prueba donde se recogieron los datos necesarios para probarlo. Hay que tener en cuenta que en el diseño de caso de prueba **V** indica Válido, **I** Inválido y **NA** es irrelevante (no es necesario proporcionar un valor del dato). Las variables significan los valores de entrada de datos para los casos de pruebas en cada escenario; la respuesta del sistema indica su comportamiento y el flujo central, los pasos que se deben ejecutar para completar cada escenario. Para ejecutar la prueba el usuario debía estar autenticado en el sistema.

A continuación, se muestra el DCP para el requisito funcional gestionar incidente específicamente adicionar base legal:

**Tabla 6: DCP Adicionar base legal**

**Fuente: Creación de los autores**

Id del escenario	Escenario	Variable 1	Variable 2	Variable 3	Respuesta del sistema
		Tipo	Denominación	Cuadro de texto	
EP 1.1	Adicionar base legal de forma correcta	V(seleccionar)	V(seleccionar)	V(cadena de caracteres)	Se muestra el mensaje "base legal adicionada"
Ep 1.2	Adicionar base legal de forma incorrecta	I(Vacío)	I(Vacío)	I(Vacío)	Se señala el campo en rojo y se muestra el mensaje: "Este campo es obligatorio.". No se habilita la opción <b>Guardar</b> .
		NA	NA	I(Vacío)	Se señala el campo en rojo y se muestra el mensaje: "Base legal sin cuerpo". No se habilita la opción <b>Guardar</b> .
EP 1.3	Cancelar acción	NA	NA	NA	Se anula la acción

Para consultar los demás DCP visitar Anexo 6: Diseños de casos de pruebas (DCP).

### 3.4.2.1 Resultado de las pruebas de sistema (funcionales)

Al Sistema de Gestión de Ilegalidades de la DPPF de Granma se le aplicaron cuatro iteraciones pertenecientes a las pruebas de sistema (funcionales) haciendo uso del método de caja negra, donde se detectaron un grupo de no conformidades como se muestra a continuación.



**Figura 11: Resultado de las pruebas de caja negra**

**Fuente: Creación de los autores**

Las no conformidades detectadas en cada una de las iteraciones estuvieron distribuidas de la siguiente manera:

**Iteración 1:**

- 20 en el módulo de Ilegalidades de ellas 13 funcionales y 7 no funcionales.
- 10 en el módulo de Inspectores de ellas 5 funcionales y 5 no funcionales.
- 3 en el módulo de Multa de ellas 2 funcionales y 1 no funcional.
- 2 en el resto de la aplicación de tipo no funcionales.

**Iteración 2:**

- 10 en el módulo de Ilegalidades de ellas 7 funcionales y 3 no funcionales.
- 3 en el módulo de Inspectores de ellas 2 funcionales y 1 no funcional.
- 2 en el módulo de Multas 1 de ellas funcional y 1 no funcional.
- 3 en el resto de la aplicación de tipo no funcionales.

### Iteración 3:

- 6 en el módulo de Ilegalidades de ellas 3 funcionales y 3 no funcionales.
- 2 en el resto de la aplicación de tipo no funcionales.

### Iteración 4:

- No hubo no conformidades.

El módulo Ilegalidades fue el elemento que más sufrió cambios en cada una de las iteraciones hasta la 4ta iteración donde no se detectó ninguna no conformidad. La interfaz de la aplicación también sufrió algunos cambios de iteración a iteración quedando al final una interfaz legible y fácil de usar por el usuario. El tiempo de generación de los reportes y el modelo de los mismos también varió en cada una de las iteraciones de las pruebas. El proceder iterativo de esta prueba permitió la identificación correcta de las no conformidades y mejorar el software hasta lograr un producto de calidad que cumpla con las funcionalidades requeridas por el usuario.

### 3.4.3 Pruebas de sistema (carga y estrés)

Para la realización de las pruebas de carga y estrés se emplea el método de Caja Negra haciendo uso de la herramienta *Jmeter*<sup>11</sup>, la misma es una herramienta desarrollada en el lenguaje Java por el proyecto *Apache Software Foundation*. Esta aplicación brinda la posibilidad de realizar pruebas de rendimiento a través de la construcción de peticiones *HTTP* con cantidad de usuarios concurrentes variables, lo que

---

<sup>11</sup> Herramienta que se utiliza para realizar pruebas de rendimiento y resistencia, normalmente contra aplicaciones web. Permite realizar simulaciones de gran carga en el servidor, red o aplicación para comprobar su “fuerza” y para analizar el rendimiento ante diferentes tipos de sobrecarga. Posee una interfaz gráfica que permite la interacción del usuario de una forma muy intuitiva.

posibilita obtener un resumen de los niveles de estrés del sistema y los límites de trabajo del mismo en condiciones extremas. Los requerimientos no funcionales que conforma el basamento de las pruebas realizadas es el siguiente:

**RNF 3.1** - Los tiempos de respuesta y velocidad de procesamiento de la información deben ser rápidos, no mayores de 5 segundos para las actualizaciones y 10 para las recuperaciones excepto algunas consultas complejas.

**RNF 3.2** - El tiempo máximo de espera está en dependencia de las operaciones realizadas, generalmente no deben exceder los 30 segundos.

Las pruebas de rendimiento son realizadas luego de que el sistema está completamente integrado, con el objetivo de observar el comportamiento del sistema bajo una cantidad de peticiones esperadas y comprobar que cumpla determinadas tareas en condiciones particulares de trabajo. Se encargan de demostrar que el sistema satisface sus requerimientos y monitorizar los comportamientos en cuanto a tiempo de respuesta de la petición y otros componentes que se vean afectados por la prueba. Esta prueba fue realizada al Sistema de Gestión de Ilegalidades de la DPPF de Granma en un servidor con las siguientes características: 500 GB de disco duro, Intel I Core 3 a 2.4 GHZ de microprocesador y 8GB de RAM. Para 100 usuarios conectados concurrentemente y se aplicaron 10 iteraciones de la misma. Para ver los resultados de las 10 iteraciones consultar el Anexo 10: Prueba de sistema (carga y estrés). A continuación, se muestran los resultados obtenidos para la 1ra iteración.



Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
3966 /std/std...	100	1621	1835	1892	608	1906	0,00%	51,3/sec	347,7
3967 /std/std...	100	242	74	697	33	874	0,00%	72,8/sec	253,7
3968 /std/std...	100	79	34	105	27	747	0,00%	107,2/sec	218,2
3969 /std/std...	100	34	33	38	29	105	0,00%	456,6/sec	2464,1
3970 /std/std...	100	28	29	32	21	41	0,00%	719,4/sec	962,5
3972 /std/std...	100	26	25	31	22	35	0,00%	819,7/sec	557,1
3971 /std/std...	100	26	26	28	23	31	0,00%	847,5/sec	2410,8
3973 /std/std...	100	22	24	26	16	27	0,00%	952,4/sec	598,0
3974 /std/std...	100	18	17	26	9	28	0,00%	1123,6/sec	954,6
Total	900	233	29	990	9	1906	0,00%	418,8/sec	1118,2

Figura 15: Resultado de la 1ra iteración de la prueba de carga y estrés con la herramienta Jmeter

Fuente: Creación de los autores

Para una mejor comprensión de la tabla, a continuación, se detallan cada uno de los parámetros usados y los umbrales de valores que debe poseer un sistema web.

#### Parámetros usados:

- **Muestras #:** indica la cantidad de usuarios realizando peticiones de manera concurrente.
- **Media:** indica la media de tiempo en milisegundos invertido por una petición.
- **Mediana:** indica la mediana de tiempo en milisegundos invertido por una petición. Significa que el 50% de las peticiones realizadas tardaron menos del valor reflejado.
- **Min:** indica el mínimo de tiempo de ejecución invertido para una petición.
- **Max:** indica el máximo de tiempo de ejecución invertido para una petición.
- **Error:** indica la relación entre el total de peticiones y el número de peticiones que originaron errores.

- **Rendimiento:** hace referencia al número de peticiones que el servidor puede procesar en un segundo.
- **Kb/sec:** rendimiento medido en Kilobytes por segundo.

**Umbrales para sistemas web:**

**Tabla 7: Umbrales para sistemas web**

Fuente:(Pérez Rojas, 2014)

Umbrales para sistemas web	
Nombre del umbral	Valor del umbral
Tiempo medio de respuesta.	5 segundos
Máximo tiempo de respuesta.	10 segundos (para sitio web es 5 segundos)
Máximo tiempo de respuesta para consultas pesadas o descarga de software.	20 segundos
Rendimiento crítico	-20% de las peticiones/segundo
Rendimiento medio.	50% de las peticiones/segundo
Rendimiento máximo.	100% de las peticiones/segundo
Tiempo medio de espera.	5 segundos
Tiempo máximo de espera.	10 segundos

**3.4.3.1 Resultado de la prueba de sistema (carga y estrés)**

En el caso de la figura 15 se realizó 1 iteración de la prueba para 100 usuarios conectados simultáneamente y se puede resumir que, para un total de 100 usuarios y un total de 900 muestras analizadas, el sistema tardó en realizar estas solicitudes en un tiempo medio de 2.33 segundos, la mediana fue de 2.9 segundos. De manera general se puede observar que las peticiones son ejecutadas en tiempos inferiores a 30

segundos en la mayoría de los casos y los tiempos para las recuperaciones excepto algunas consultas complejas como es el caso de la etiqueta 3966 son menores a 10 segundos cumpliendo así con los requisitos no funcionales de eficiencia RNF 3.1 y RNF 3.2. Para todas las muestras de usuarios la ocurrencia de errores se mantiene en 0.0%, lo que quiere decir que todas las peticiones hechas se ejecutan satisfactoriamente. El sistema cuenta con un rendimiento aceptable logrando procesar 418.8 peticiones por segundo excediendo el 50 % de estas, generando un total de 1118,2 KB/segundo.

### **3.4.4 Pruebas de integración**

Las pruebas de integración son una técnica sistemática para construir la arquitectura del software mientras se llevan a cabo pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño.(Pressman y Troya, 2013).

Para el desarrollo de esta prueba se utilizó la técnica de Caja Negra mediante el diseño de casos de prueba de integración. Esta técnica estuvo dirigida a probar la integración de los módulos Ilegalidad, Multa, Inspector y Base Legal con los módulos Persona, Seguridad, Nomencladores y Flujo de Trabajo. La siguiente tabla muestra el caso de prueba de integración realizado entre los módulos Ilegalidad, Multa, Inspector y Base Legal y el módulo Seguridad. Para consultar todos los casos de prueba de integración remitirse al Anexo 11: Casos de prueba de integración (CPI).

Tabla 8: Integración con el módulo Seguridad

Fuente: Creación de los autores

Módulo al cual se integra	Seguridad
<b>Condiciones de Ejecución</b>	El usuario debe autenticarse mediante el módulo Seguridad
<b>Descripción de la prueba</b>	Comprobar que los módulos Ilegalidad, Multa, Base Legal e Inspector no pueden ser accedidos por usuarios no autenticados por el modulo Seguridad.
<b>Entradas/Pasos de ejecución</b>	El usuario se autentica en el sistema
<b>Resultado esperado</b>	Los usuarios tienen acceso a las funcionalidades de acuerdo al rol y sus responsabilidades.
<b>Evaluación</b>	Prueba satisfactoria.

#### 3.4.4.1 Resultado de las pruebas de integración

Luego de aplicar todos los casos de pruebas de integración entre los módulos Ilegalidad, Multa, Inspector y Base Legal con los módulos Persona, Seguridad, Nomencladores y Flujo de Trabajo se pudo observar que todas las evaluaciones fueron satisfactorias. Resultados que indican que el Sistema de Gestión de Ilegalidades de la DPPF de Granma posee una alta integración entre sus módulos tanto los internos como los módulos externos que se reutilizan en la aplicación.

#### 3.4.5 Pruebas de aceptación

Las pruebas de aceptación son básicamente pruebas funcionales sobre el sistema completo, ya que tienen como objetivo obtener la aceptación final del cliente antes de la entrega del producto para su utilización. Su ejecución es facultativa del cliente, y en el caso de que no se realicen explícitamente, se dan por incluidas dentro de las pruebas del sistema. La ejecución de las pruebas de aceptación requiere un entorno de pruebas que represente el entorno de producción (Pressman, 2010). Para el desarrollo de las pruebas se utiliza el método de caja negra, este método se centra en las funcionalidades que se espera obtener de un sistema, es decir, intentan encontrar casos en que el sistema no atiende alguna especificación. En estas pruebas el probador se limita a suministrarle datos al sistema y estudiar las salidas, sin preocuparse de lo

que pueda estar realizando el sistema por dentro. La técnica alfa fue la escogida donde el cliente hace las pruebas al sistema en un ambiente controlado con el desarrollador como observador del usuario, registrando los errores y los problemas de uso. El desarrollo de la prueba fue satisfactorio, quedando como constancia el acta de aceptación del cliente accesible en el Anexo 13: Acta de aceptación del cliente.

### **3.4.6 Resultado de las pruebas de software**

Posterior a la ejecución de todas las pruebas de software pertenecientes a la estrategia de pruebas de software trazada para comprobar la calidad del Sistema de Gestión de Ilegalidades de la DPPF de Granma se puede resumir que:

1. Mediante las pruebas unitarias usando el método de la Caja Blanca con la herramienta Jenkins se pudo apreciar que las métricas referentes a la herencia, complejidad ciclomática y tamaño de proyecto se encuentran en el nivel medio. Evidenciando que el sistema posee una moderada complejidad y es fácil de entender.
2. Mediante las pruebas de sistema (funcionales) usando el método de Caja Negra con la técnica de casos de pruebas se detectaron varias no conformidades en los requisitos funcionales y no funcionales que fueron resueltas en 3 iteraciones del método. Garantizando en la 4ta iteración un producto con calidad que cumple con los requerimientos del cliente.
3. Mediante las pruebas de sistema (carga y estrés) usando el método de Caja Negra con el empleo de la herramienta *Jmeter* se comprobó que el sistema para 100 usuarios conectados concurrentemente y para un total de 900 peticiones obtiene tiempos promedios de respuesta de 2.33 segundos, cumpliendo de esta manera con los requisitos no funcionales de rendimiento. Además, se comprueba que para todas las muestras de usuarios la ocurrencia de errores se mantiene en 0.0%, lo que quiere decir que todas las peticiones hechas se ejecutan satisfactoriamente.
4. Mediante las pruebas de integración usando los casos de prueba de integración se comprobó la alta integración existente entre los módulos del sistema; tanto los módulos internos como los módulos externos que se reutilizan en la aplicación.

### **3.5 Validación de los resultados obtenidos**

A continuación, serán presentados los resultados obtenidos a partir de la evaluación del Sistema de Gestión de Ilegalidades de la DPPF de Granma haciendo uso del método criterio de expertos, desde su concepción teórica, a partir de las respuestas obtenidas de las diferentes secciones del

Anexo 7: Cuestionario #1.

Para la confección del cuestionario se utilizó la técnica *Likert*. Esta técnica permite medir actitudes y conocer el grado de conformidad del encuestado en cuanto al cuestionario de actitudes realizado (Ospina et al, 2003). La entrevista se realizó a 50 expertos de ellos 30 especialistas de la DPPF de Granma y 20 especialistas del IPF; cada uno de estos encuestados tienen más de 5 años de vinculación en procesos de gestión de ilegalidades en urbanismo y vivienda. Los expertos expresan sus valoraciones mediante indicadores como se presentan en la siguiente tabla:

**Tabla 9: Indicadores de la escala Likert**

**Fuente: Creación de los autores**

1	2	3	4	5
Totalmente en desacuerdo(TD)	En desacuerdo(ED)	Ni de acuerdo, ni en desacuerdo(NI)	De acuerdo(DA)	Totalmente de acuerdo(TA)

Los principales aspectos utilizados para aplicar la técnica de *Likert* fueron:

1. Considero que el sistema ahorra tiempo en el proceso de gestión de las ilegalidades.
2. Considero que las ilegalidades registradas en el sistema pueden ser consultadas en cualquier momento.
3. Considero que los gráficos del módulo de Análisis constituyen información visual de importancia para la toma de decisiones.
4. Considero que los reportes que genera el sistema son importantes para la toma de decisiones.
5. Considero que de forma general un usuario del sistema al disponer de toda la información que brinda el sistema obtiene el apoyo necesario para la toma de decisiones.
6. Considero que de forma general el sistema contribuye en el proceso de gestión de las ilegalidades.
7. Considero que me siento satisfecho con las funcionalidades del sistema.

Posteriormente a la realización del cuestionario se procede a la creación de una tabla de valoración de expertos en cuanto al sistema, para agrupar los resultados del cuestionario que se les realizó a los

especialistas elegidos. Con el análisis de los datos anteriormente expuestos se procede a obtener el porcentaje de valoración de los expertos en los aspectos analizados. Una vez calculado los porcentajes de concordancia de las respuestas de los expertos, se calcula el Índice Porcentual (IP), que integra en un solo valor la aceptación de cada planteamiento por los evaluadores mediante la siguiente fórmula:

$$IP = 5(\%)+ 4(\%)+ 3(\%)+ 2(\%)+ 1(\%) / 5$$

**Tabla 10: Valoración de expertos referente al sistema**

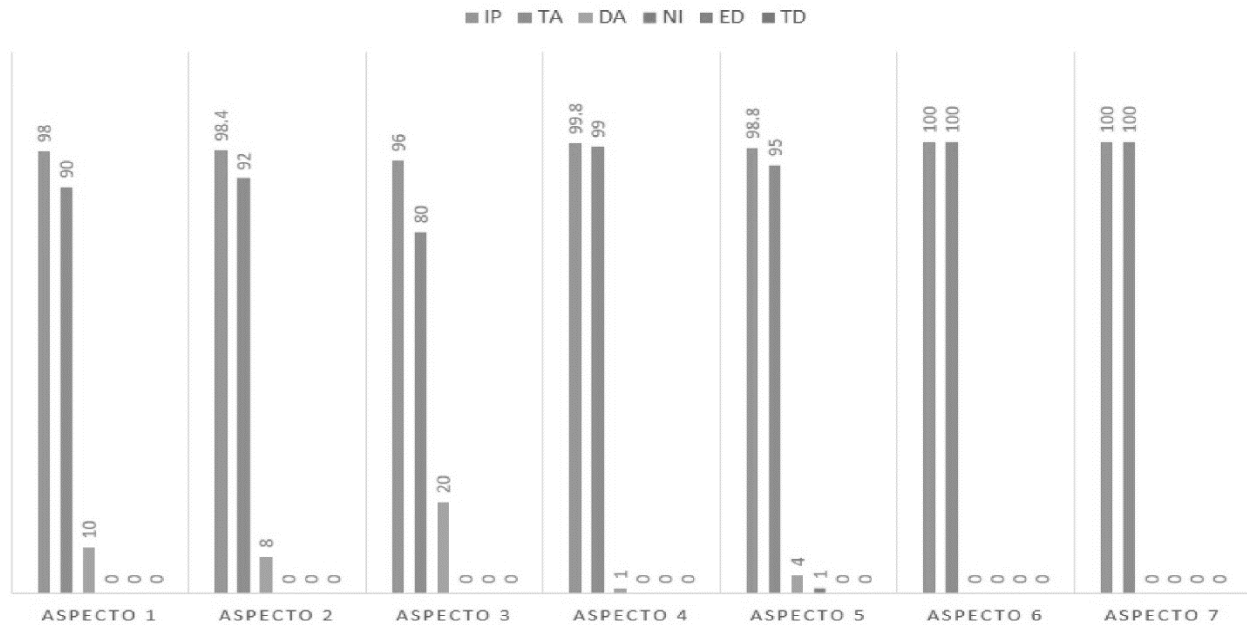
**Fuente: Creación de los autores**

Aspecto	Escala					
	IP	TA	DA	NI	ED	TD
1	98	90	10	0	0	0
2	98.4	92	8	0	0	0
3	96	80	20	0	0	0
4	99.8	99	1	0	0	0
5	98.8	95	4	1	0	0
6	100	100	0	0	0	0
7	100	100	0	0	0	0

Como se evidencia en la tabla 17 el Índice Porcentual por cada una de las preguntas es mayor que 90 lo que indica que los especialistas están satisfechos con las afirmaciones realizadas. Para un mejor entendimiento de la tabla de valoración de expertos en cuanto al sistema, se muestra el siguiente gráfico.



### VALORACIÓN DE LOS EXPERTOS



**Figura 16: Distribución del IP de la encuesta #1**

**Fuente: Creación de los autores**

Se realizó además la encuesta #2 a 100 especialistas e inspectores de la DPPF de Granma para medir la conformidad con el sistema a través de los siguientes aspectos:

1. Nivel de retroalimentación que brinda el sistema en los niveles de IPF.
2. Nivel de satisfacción con el sistema.
3. Calidad de la información gestionada por el sistema.
4. Nivel de subjetividad de la información disponible.
5. Tiempo para la generación de los reportes.

Dicha encuesta arrojó los siguientes resultados:

**Tabla 11: Resultados de la encuesta #2 Sección I**

Fuente: Creación de los autores

Criterio	Escala		
	Alta	Media	Baja
Nivel de retroalimentación que brinda el sistema en los niveles de IPF.	93	7	0
Nivel de satisfacción con el sistema.	98	2	0
Calidad de la información gestionada por el sistema.	90	8	2
Nivel de subjetividad de la información disponible.	0	5	95

**Tabla 12: Resultados de la encuesta #2 Sección II**

Fuente: Creación de los autores

Criterio	Escala		
	20seg	Entre 21 y 40 seg.	Más de 1min.
Tiempo para la generación de los reportes.	93	7	0

Como es apreciable en las tablas anteriores el sistema goza de una alta aceptación por el usuario obteniendo más del 90 % en todos los criterios de satisfacción y un 97% de los usuarios apreciaron la generación de los reportes referentes a las ilegalidades en 20 segundos aproximadamente. Por otra parte, el indicador: nivel de subjetividad arrojó como resultado un porcentaje bajo, evidenciando un mayor nivel de objetividad en la información visualizada.

### **3.5.1 Validación de la hipótesis**

Para la validación de la hipótesis se realizó una encuesta teniendo en cuenta cinco dimensiones para la gestión de las ilegalidades y apoyo a la toma de decisiones: disponibilidad de la información, calidad de la información para la toma de decisiones, usabilidad de información disponible, satisfacción del decisor sobre la calidad de la información y retroalimentación brindada por el sistema entre los diferentes niveles del

Instituto de Planificación Física. El total de los usuarios encuestados así como la distribución de los mismos por cargo puede ser consultada en Anexo 12: Usuarios encuestados. En la siguiente tabla se recogen los resultados en por ciento de la encuesta.

**Tabla 13: Resultados de la encuesta #3**

**Fuente: Creación de los autores**

Dimensiones	Unidad	Cantidad	% Respecto al total
Tiempo de respuesta del sistema	Hasta 20seg	98	98
	Entre 5 y 20 min	2	2
	Más de 1h	0	0
Calidad de la información para la toma de decisiones	Alto [100 % ; 70%]	99	99
	Medio [69% ; 50%]	1	1
	Bajo [49% ; 0%]	0	0
Tiempo para generar reportes de las ilegalidades	Hasta 20seg	98	98
	Entre 5 y 20 min	2	2
	Más de 1h	0	0
Satisfacción del decisor sobre la calidad de la información brindada por el sistema	Alto [100 % ; 70%]	99	99
	Medio [69% ; 50%]	1	1
	Bajo [49% ; 0%]	0	0
Retroalimentación brindada por el sistema entre los diferentes niveles del Instituto de Planificación Física	Alto [100 % ; 70%]	99	99
	Medio [69% ; 50%]	1	1
	Bajo [49% ; 0%]	0	0

Según los datos reflejados en la tabla anterior, se puede apreciar que los cinco indicadores estuvieron por encima del 90% de la puntuación, siendo estos resultados altos, según la escala definida en la introducción. El indicador de medición del tiempo para generar reportes arrojó como resultado que se encuentran en el 98 % para la respuesta del sistema y genera los reportes en un tiempo inferior a los 20s. Los análisis anteriores expresan claramente que el desarrollo del Sistema de Gestión de Ilegalidades en la DPPF de Granma, contribuyó al proceso de gestión de las ilegalidades y al apoyo en la toma de decisiones en dicha entidad, mejorando:

- La retroalimentación entre los diferentes niveles del IPF.
- Tiempo de generación de los reportes referentes a las ilegalidades.
- La disponibilidad de la información para la gestión de las ilegalidades.
- La calidad de la información para la toma de decisiones.
- El proceso de gestión de las ilegalidades.

### 3.6 Conclusiones del capítulo

Con la investigación antes expuesta se concluye que:

- La utilización de estándares de código para la implementación de la propuesta de solución permitió adoptar una estructura homogénea que facilita la comunicación y una menor cantidad de errores; logrando un código más limpio y fácil de mantener.
- Las pruebas realizadas permitieron detectar los errores presentes, corregirlos en el menor tiempo posible y entregar al cliente una aplicación con mayor calidad, seguridad y usabilidad.
- Las encuestas realizadas, así como la técnica de expertos permitió validar el Sistema de Gestión de Ilegalidades de la DPPF de Granma obteniendo resultados relevantes por encima del 90 %, lo cual permitió comprobar que la implementación del sistema responde al objetivo de investigación trazado.

### **Conclusiones generales**

Con la investigación realizada, el diseño y la implementación de una solución informática para la gestión de ilegalidades en la DPPF de Granma se obtuvieron resultados que permiten arribar a las siguientes conclusiones:

La revisión bibliográfica, el estudio de sistemas homólogos y el análisis de antecedentes reflejados en el documento, permitieron identificar conceptos principales a tener en cuenta en un sistema para la gestión de ilegalidades.

Con la aplicación del proceso de desarrollo de software, se logró obtener y especificar los requisitos del sistema y otros artefactos; lo que posibilitó comprender mejor los resultados a obtener en la solución y sirvió de guía para la implementación del sistema.

La utilización de la estrategia de pruebas garantizó evidenciar que el Sistema de Gestión de Ilegalidades de la DPPF de Granma contribuirá a la gestión de ilegalidades y apoyo a la toma de decisiones.

## Recomendaciones

Se recomienda al IPF el despliegue del Sistema de Gestión de Ilegalidades para la DPPF de Granma en todas las provincias del país para lograr una mayor retroalimentación entre los niveles que nutren a dicha entidad y así mejorar la efectividad en el combate contra las ilegalidades de la vivienda y urbanismo.

Para el desarrollo de futuras versiones del Sistema de Gestión de Ilegalidades para la DPPF de Granma se recomienda al centro SATD de la empresa XETID:

- La inclusión de más reportes y análisis estadísticos que el IPF considere necesarios para de esta manera suplir nuevas necesidades que pueda detectar dicha entidad como cliente.
- Incluir un nuevo rol para el trabajo de los consultores del IPF que reportan directamente a la dirección del país. Dichos consultores solo deberían tener acceso a visualizar la información.
- Integrar la solución al Cuadro de Mando Integral de la XETID para realizar otros análisis estadísticos en base a indicadores bases de la gestión de ilegalidades y contribuir así a mejorar la toma de decisiones por parte de los jefes en las máximas instancias del IPF.

### Referencias bibliográficas

- APACHE FOUNDATION, 2017. Welcome! - The Apache HTTP Server Project. [en línea]. [Consulta: 21 noviembre 2017]. Disponible en: <http://httpd.apache.org/>.
- APLICACIONES WEB, 2008. Arquitectura cliente-servidor. [en línea]. 2018 2008. Disponible en: <https://sites.google.com/site/4appweb/tarea/2-4-arquitectura-cliente-servidor>.
- AYMAN NADA, MONA NASR y MARWA SALAH, 2015. *Service oriented approach for decision support systems* [en línea]. S.l.: IEEE. ISBN 978-1-4799-4419-4. Disponible en: <https://ieeexplore.ieee.org/document/7065081/>.
- CAMILO E. SUÁREZ REBAZA, 2009. *Desarrollo de Software Orientado a Objetos* [en línea]. S.l.: s.n. Disponible en: [http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Cuaderno\\_Electronico\\_\\_\\_Desarrollo\\_de\\_Software\\_Orientado\\_a\\_Ojetos.pdf](http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Cuaderno_Electronico___Desarrollo_de_Software_Orientado_a_Ojetos.pdf).
- CARLOS A. GUERRERO, JOHANNA M SUÁREZ y LUZ E GU, 2013. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. . 2013.
- CEO2 SOFTWARE., 2010. Sistemas Clientes Servidor. [en línea], Disponible en: <http://www.co2soft.com.ar/sistemas-cliente-servidor/>.
- CHAI ZHENGMENG y JIANG HAOXIANG, 2012. *A brief review on Decision Support Systems and it's applications* [en línea]. S.l.: IEEE. ISBN 978-1-61284-704-7. Disponible en: <https://ieeexplore.ieee.org/document/6132134/>.
- DANIEL ENRIQUE PÉREZ ROJAS, 2014. *Manual de Usuario de JMeter para la realización de pruebas de rendimiento a las Aplicaciones Web*. S.l.: Centro de Calidad, Estánderes y Seguridad. XETID.
- FRANK BUSCHMANN, REGINE MEUNIER, HANS ROHNERT, PETER SOMMERLAD y MICHAEL STAL, 2013. *Pattern-oriented software architecture (POSA)*. S.l.: s.n.
- GOOGLE, 2017. Material Design with AngularJS. [en línea]. [Consulta: 21 noviembre 2017]. Disponible en: <https://www.airpair.com/angularjs/posts/material-design-with-angularjs>.
- IDAILIS GUTIÉRREZ REYES, 2015. *Manual de instalacion, configuración y administración de Jenkins*. S.l.: Centro de Calidad, Estánderes y Seguridad. XETID.
- IPF, 2015. Misiones y funciones de IPF. [en línea], Disponible en: [http://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAAAAEAMtMSbF1jTAAAUMjS0sjtbLUouLM\\_DxblwMDCwNzAwuQQGZapUt-ckhIQaptWmJOcSoAViX1STUAAAA=WKE](http://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAAAAEAMtMSbF1jTAAAUMjS0sjtbLUouLM_DxblwMDCwNzAwuQQGZapUt-ckhIQaptWmJOcSoAViX1STUAAAA=WKE).

- IPF, 2018. IPF «Ordenamiento Territorial y Urbanismo Equidad y sustentabilidad». [en línea]. Disponible en: <http://www.ipf.cu/es>.
- IVAR JACOBSON, GRADY BOOCH y JAMES RUMBAUGH, 2000. *Proceso Unificado de Desarrollo de Software*. S.l.: s.n. ISBN 74-7829-036-2.
- JAVIER EGUÍLUZ PÉREZ, 2008. *Introducción a JavaScript*. S.l.: s.n.
- LISSA CURBELO OLIVA, LAURA S. ORTEGA RETURETA, YOANNA COLUMBIÉ CISNERO y YUDELKY GONZÁLEZ MILÁN, 2017. Prodesoft v1.5. [en línea]. [Consulta: 20 noviembre 2017]. Disponible en: <https://intranet.xetid.cu/content/prodesoft>.
- MAIKEL JOSÉ RIVERO DORTA, 2016. *AngularJS paso a paso* [en línea]. S.l.: Leanpub. Disponible en: <http://leanpub.com/angularjs-paso-a-paso>.
- MARÍA JESUS LAMARCA LAPUENTE, 2013. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen*. Universidad Complutense de Madrid: s.n.
- Mastermagazine. [en línea], 2012. Disponible en: <http://mastermagazine.info/termino/5234.php>.
- MICHAEL BLAHA, 2010. *Patterns of data modeling*. Segunda Edición. S.l.: CRC Press Taylor & Francis Group. ISBN 1-4398-1989-0.
- MICROSOFT, 2018. *La Guía de Arquitectura Versión 2.0a de Microsoft* [en línea]. S.l.: s.n. Disponible en: url <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
- ORACLE CORPORATION, 2016. The NetBeans Platform. [en línea]. 2016. Disponible en: <https://netbeans.org/downloads>.
- PGADMIN, 2018. *PgAdmin PostgreSQL Tools* [en línea]. S.l.: s.n. Disponible en: <http://www.pgadmin.org/translation/>.
- PMOINFORMATICA, 2016. *7 Técnicas de levantamiento de requerimientos de software* [en línea]. S.l.: s.n. Disponible en: <http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>.
- PMOINFORMATICA, 2017. Pruebas de software. [en línea]. [Consulta: 16 noviembre 2017]. Disponible en: <http://www.pmoinformatica.com/p/pruebas-de-software.html>.
- RAFAEL MARTÍNEZ, 2015. *Portal en español sobre PostgreSQL* [en línea]. PostgreSQL.es: PostgreSQL. Disponible en: <http://www.postgresql.org.es>.
- ROBERT PRESSMAN, 2010. *Software Engineering. A practitioner's approach*. McGraw-Hill, 7ma edición,. S.l.: s.n.



- ROBERTO H. SAMPIERI, CARLOS FERNÁNDEZ COLLADO y PILAR BAPTISTA LUCIO, 2014. *Metodología de la Investigación*. 6ta Edición. México: Mc Craw Hill Education. ISBN 978-1-4562-2396-0.
- RS PRESSMAN y JM TROYA, 2013. *Ingeniería de Software un enfoque práctico*. Séptima edición. Madrid: s.n. ISBN 978-607-15-0314-5.
- STIG SAETHER BAKKEN, ALEXANDER AULBACH, EGON SCHMID, JIM WINSTEAD y LARS TORBEN WILSON, 2001. *PHP Manual* [en línea]. My PHP.net: s.n. Disponible en: <https://secure.php.net/manual/es/preface.php>.
- TOM NIELSEN, [sin fecha]. *Ethics, Aesthetics and Contemporary Urbanism*. S.l.: s.n.
- VISUAL PARADIGM, 2010. *Visual Paradigm. ULM tool, business process modeler and database designer for software development team*. [en línea]. S.l.: s.n. Disponible en: <http://www.visual-paradigm.com>.
- WILSON y LESLIE, 1993. *Comparative Programming Languages*. S.l.: s.n.
- XETID, 2015. Manual de usuario del framework ZEOLIDES v2.2.0. . 2015.

### Bibliografía consultada

1. APACHE FOUNDATION, 2017. Welcome! - The Apache HTTP Server Project. [en línea]. [Consulta: 21 noviembre 2017]. Disponible en: <http://httpd.apache.org/>.
2. APLICACIONES WEB, 2008. Arquitectura cliente-servidor. [en línea]. 2018 2008. Disponible en: <https://sites.google.com/site/4appweb/tarea/2-4-arquitectura-cliente-servidor>.
3. ARMANDO DUANY DANGEL y ECOLINK, 2000. Los sistemas de información en las organizaciones. [en línea], Disponible en: <http://www.econlink.com.ar/sistemas-informacion/definiciones>.
4. ARMANDO SUÁREZ CUETO, BALLESTER, E.G., PATRICIO MARTÍNEZ-BARCO y PALOMA MOREDA, 2006. *Apuntes de Bases de Datos* [en línea]. S.I.: Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos. Disponible en: <http://hdl.handle.net/10045/2990>.
5. BENCHMARK CCM y CREATIVE COMMONS, 2016. Introducción a las Bases de Datos. [en línea]. marzo 2016. Disponible en: <http://es.ccm.net/contents/66-introduccion-bases-de-datos>.
6. CAMILO E. SUÁREZ REBAZA, [sin fecha]. *Desarrollo de Software Orientado a Objetos* [en línea]. S.I.: s.n. Disponible en: [http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Cuaderno\\_Electronico\\_\\_\\_Desarrollo\\_d\\_e\\_Software\\_Orientado\\_a\\_Objetos.pdf](http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Cuaderno_Electronico___Desarrollo_d_e_Software_Orientado_a_Objetos.pdf).
7. CARLOS A. GUERRERO, JOHANNA M SUÁREZ y LUZ E GU, 2013. Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web. . 2013.
8. CARLOS PLATERO, 2014. *Apuntes de Informática Industrial*. S.I.: Dpto.Electrónica, Automática e Informática Industrial.
9. CEO2 SOFTWARE., 2010. Sistemas Clientes Servidor. [en línea], Disponible en: <http://www.co2soft.com.ar/sistemas-cliente-servidor/>.
10. CÉSAR ARTURO GUERRA y BUZZ SG, 2007. Obtención de Requerimientos. Técnicas y Estrategia. ,
11. CLUBENSAYOS, 2014. *Introducción a la Computación. Historias de los lenguajes de computación* [en línea]. S.I.: ClubEnsayos.com. Disponible en:

<https://www.clubensayos.com/Tecnolog%C3%ADa/Introducci%C3%B3n-A-La-Computaci%C3%B3n/1715803.html>.

12. DANIEL ENRIQUE PÉREZ ROJAS, 2014. *Manual de Usuario de JMeter para la realización de pruebas de rendimiento a las Aplicaciones Web*. S.l.: Centro de Calidad, Estánderes y Seguridad. XETID.
13. DELVIS ECHEVERRÍA PEREZ y ARIANNIS ABELLA PAUMIER, 2014. *Testing como Práctica para Evaluar la Eficiencia en Aplicaciones Web*. S.l.: Revista Latinoamericana de Ingeniería de Software.
14. Diccionario de la lengua española. [en línea], 2015. Disponible en: <http://www.wordreference.com/definicion/gesti%C3%B3n>.
15. FRANK BUSCHMANN, REGINE MEUNIER, HANS ROHNERT, PETER SOMMERLAD y MICHAEL STAL, [sin fecha]. *Pattern-oriented software architecture (POSA)*. S.l.: s.n.
16. GOOGLE, 2017. Material Design with AngularJS. [en línea]. [Consulta: 21 noviembre 2017]. Disponible en: <https://www.airpair.com/angularjs/posts/material-design-with-angularjs>.
17. IDAILIS GUTIÉRREZ REYES, 2015. *Manual de instalación, configuración y administración de Jenkins*. S.l.: Centro de Calidad, Estánderes y Seguridad. XETID.
18. Inspección Urbanística. [en línea], 2017. 2017. Disponible en: [http://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAAAAEAMtMSbF1jTAAUMjS0sjtbLUouLM\\_DxblwMDCwNzAwuQQGZapUt-ckhIQaptWmJOcSoAViX1STUAAAA=WKE](http://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAAAAEAMtMSbF1jTAAUMjS0sjtbLUouLM_DxblwMDCwNzAwuQQGZapUt-ckhIQaptWmJOcSoAViX1STUAAAA=WKE).
19. IPF, 2015. Misiones y funciones de IPF. [en línea], Disponible en: [http://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAAAAEAMtMSbF1jTAAUMjS0sjtbLUouLM\\_DxblwMDCwNzAwuQQGZapUt-ckhIQaptWmJOcSoAViX1STUAAAA=WKE](http://guiasjuridicas.wolterskluwer.es/Content/Documento.aspx?params=H4sIAAAAAAAAAEAMtMSbF1jTAAUMjS0sjtbLUouLM_DxblwMDCwNzAwuQQGZapUt-ckhIQaptWmJOcSoAViX1STUAAAA=WKE).
20. IPF, 2018. Dirección Provincial de Planificación Física, Las Tunas. [en línea]. [Consulta: 29 enero 2018]. Disponible en: <http://dppfltu.cubava.cu/2017/06/23/planificacion-fisica-insertandose-en-la-informatizacion-de-la-sociedad-cubana-implementacion-del-siplaf/>.
21. JOAQUIN GUIRAL HERRANDO, RAFAEL LAPIEDRA ALCAMÍ y JOAQUÍN GUIRAL HERRANDO, [sin fecha]. *Introducción a la gestión de sistemas de información en la empresa*. Universitat Jaume. S.l.: s.n.

22. JUAN A LÓPEZ, CARLOS J FUERTES, ALFONSO PEDRIZA y MÓNICA CITORES, 2017. *LOCALGIS 3 Mejorando la Gestión Municipal Georreferenciada*. MINETUR. España: s.n.
23. KYOCERA, [sin fecha]. Los 6 principales tipos de sistemas de información. [en línea], Disponible en: <https://smarterworkspaces.kyocera.es/blog/los-6-principales-tipos-sistemas-informacion/>.
24. LARMAN CRAIG, 1999. *UML y Patrones*. New York : PRENTICE HALL: s.n. ISBN 970-17-0261-1.
25. LEONARDO FERNÁNDEZ, DIEGO GARAY, MIRTA LAPSENSON, CECILIA LARIVERA y ESTANISLAO SIMONETTI, 2011. *Sistemas de Información Geográfica para el ordenamiento territorial*. Buenos Aires: s.n.
26. LIBROSWEB, 2006. *Introducción al JavaScript* [en línea]. LibrosWeb.es: s.n. Disponible en: <http://librosweb.es/libro/javascript>.
27. LISSA CURBELO OLIVA, LAURA S. ORTEGA RETURETA, YOANNA COLUMBIÉ CISNERO y YUDELKY GONZÁLEZ MILÁN, 2017. Prodesoft v1.5. [en línea]. [Consulta: 20 noviembre 2017]. Disponible en: <https://intranet.xetid.cu/content/prodesoft>.
28. MAIKEL JOSÉ RIVERO DORTA, 2016. *AngularJS paso a paso* [en línea]. S.I.: Leanpub. Disponible en: <http://leanpub.com/angularjs-paso-a-paso>.
29. MARÍA JESUS LAMARCA LAPUENTE, 2013. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen*. Universidad Complutense de Madrid: s.n.
30. Mastermagazine. [en línea], 2012. Disponible en: <http://mastermagazine.info/termino/5234.php>.
31. MICHAEL BLAHA, 2010. *Patterns of data modeling*. Segunda Edición. S.I.: CRC Press Taylor & Francis Group. ISBN 1-4398-1989-0.
32. MICROSOFT, 2018a. *La Guía de Arquitectura Versión 2.0a de Microsoft* [en línea]. S.I.: s.n. Disponible en: url <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
33. MICROSOFT, 2018b. Revisiones de código y estándares de codificación. [en línea]. 2018. Disponible en: <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.

34. MICROSOFT, [sin fecha]. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. S.l.: s.n.
35. NOEL BONFILIO PINEDA JAIMES, 2000. *El Uso de los Sistemas de Información Geográfica en el Ordenamiento Territorial Municipal*. Toluca, México: Facultad de Geografía. UAEM.
36. ORACLE CORPORATION, 2016. The NetBeans Platform. [en línea]. 2016. Disponible en: <https://netbeans.org/downloads>.
37. PABLO LÁZARO, 2013. ¿Qué es AngularJS? Una breve introducción. [en línea]. 20 mayo 2013. Disponible en: <http://pablolazarodev.blogspot.com/2013/05/que-es-angularjs-una-breve-introduccion.html>.
38. PMOINFORMATICA, 2016. *7 Técnicas de levantamiento de requerimientos de software* [en línea]. S.l.: s.n. Disponible en: <http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>.
39. PMOINFORMATICA, 2017. Pruebas de software. [en línea]. [Consulta: 16 noviembre 2017]. Disponible en: <http://www.pmoinformatica.com/p/pruebas-de-software.html>.
40. ¿Qué son los Sistemas de Apoyo a la Toma de Decisiones (DDS)? [. [en línea], 2017. [Consulta: 16 noviembre 2017]. Disponible en: <http://pertutatis.cat/que-son-los-sistemas-de-apoyo-a-la-toma-de-decisiones-dds/>.
41. RAFAEL MARTÍNEZ, 2015. *Portal en español sobre PostgreSQL* [en línea]. PostgreSQL.es: PostgreSQL. Disponible en: <http://www.postgresql.org.es>.
42. ROBERT PRESSMAN, 2010. *Software Engineering. A practitioner's approach*. McGraw-Hill, 7ma edición,. S.l.: s.n.
43. ROBERTO H. SAMPIERI, CARLOS FERNÁNDEZ COLLADO y PILAR BAPTISTA LUCIO, 2014. *Metodología de la Investigación*. 6ta Edición. México: Mc Craw Hill Education. ISBN 978-1-4562-2396-0.
44. RS PRESSMAN y JM TROYA, 2013. *Ingeniería de Software un enfoque práctico*. Séptima edición. Madrid: s.n. ISBN 978-607-15-0314-5.

45. STIG SAETHER BAKKEN, ALEXANDER AULBACH, EGON SCHMID, JIM WINSTEAD y LARS TORBEN WILSON, 2001. *PHP Manual* [en línea]. My PHP.net: s.n. Disponible en: <https://secure.php.net/manual/es/preface.php>.
46. *Todo sobre los mapas de proceso* [en línea], 2018. S.l.: s.n. Disponible en: <https://www.lucidchart.com/.../all-about-business-process-mapping-flow-charts-and-di..>
47. VISUAL PARADIGM, 2010. *Visual Paradigm. ULM tool, business process modeler and database designer for software development team.* [en línea]. S.l.: s.n. Disponible en: <http://www.visual-paradigm.com>.
48. XETID, 2015. Manual de usuario del framework ZEOLIDES v2.2.0. . 2015.

Anexos

Anexo 1: Diagramas de requisitos funcionales (DRF) y descripción de requisitos funcionales (DERF).

Fuente: Creación de los autores

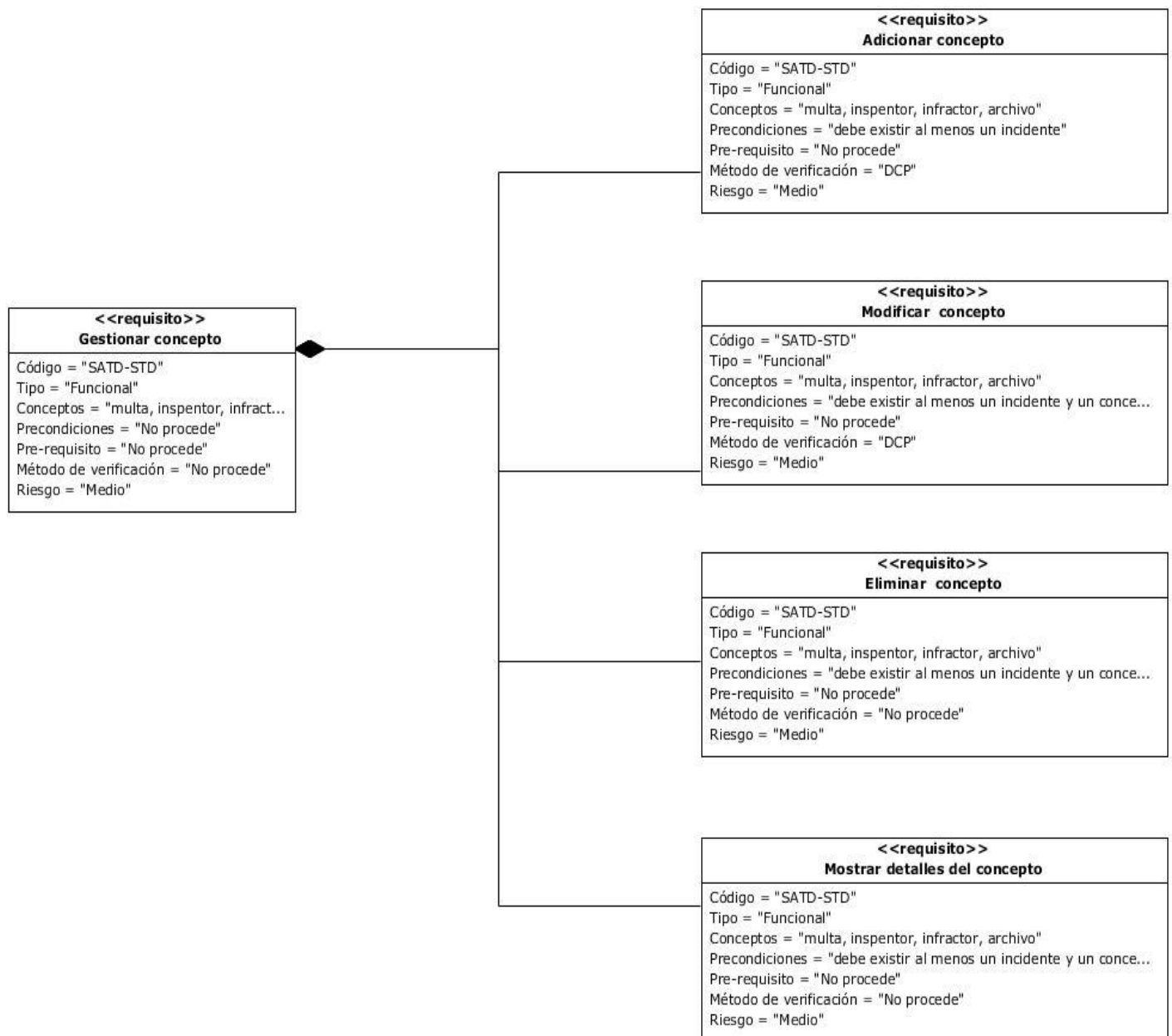


Figura 17: DRF Gestionar concepto

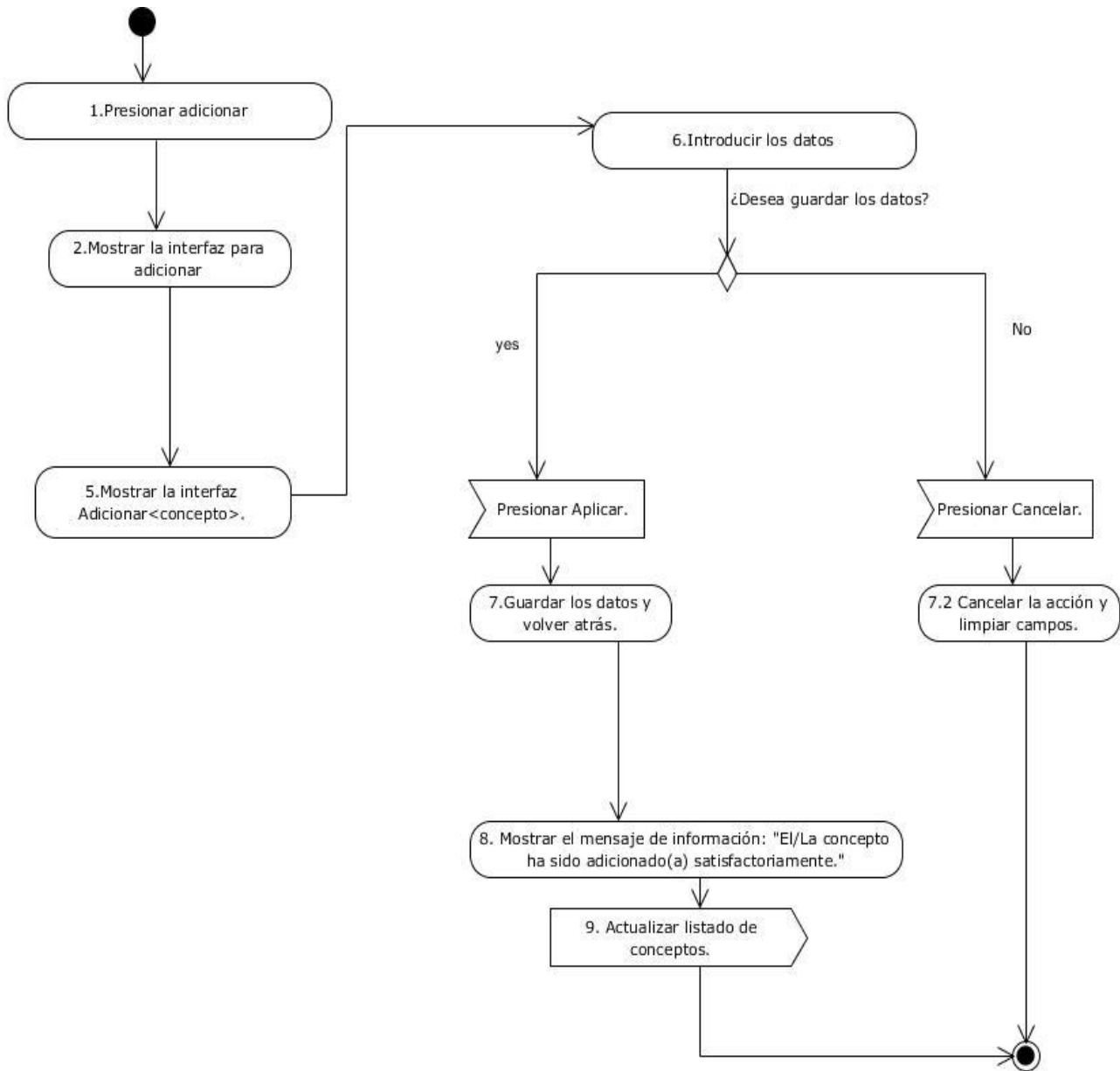


Figura 18: DERF Adicionar concepto



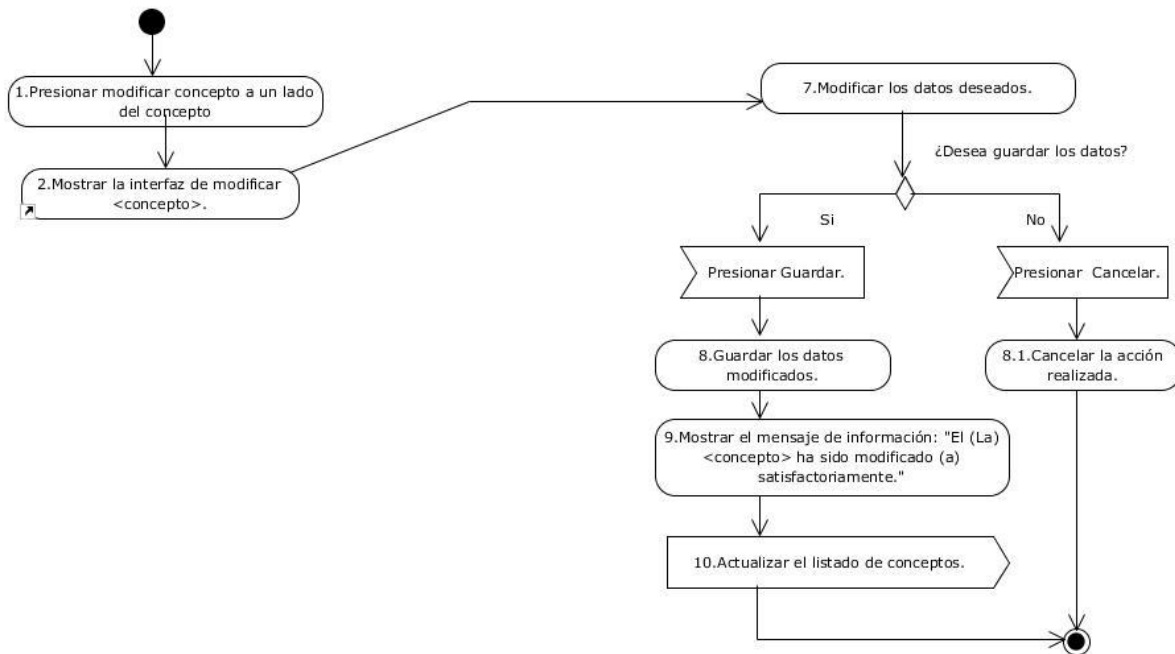


Figura 19: DERF Modificar concepto

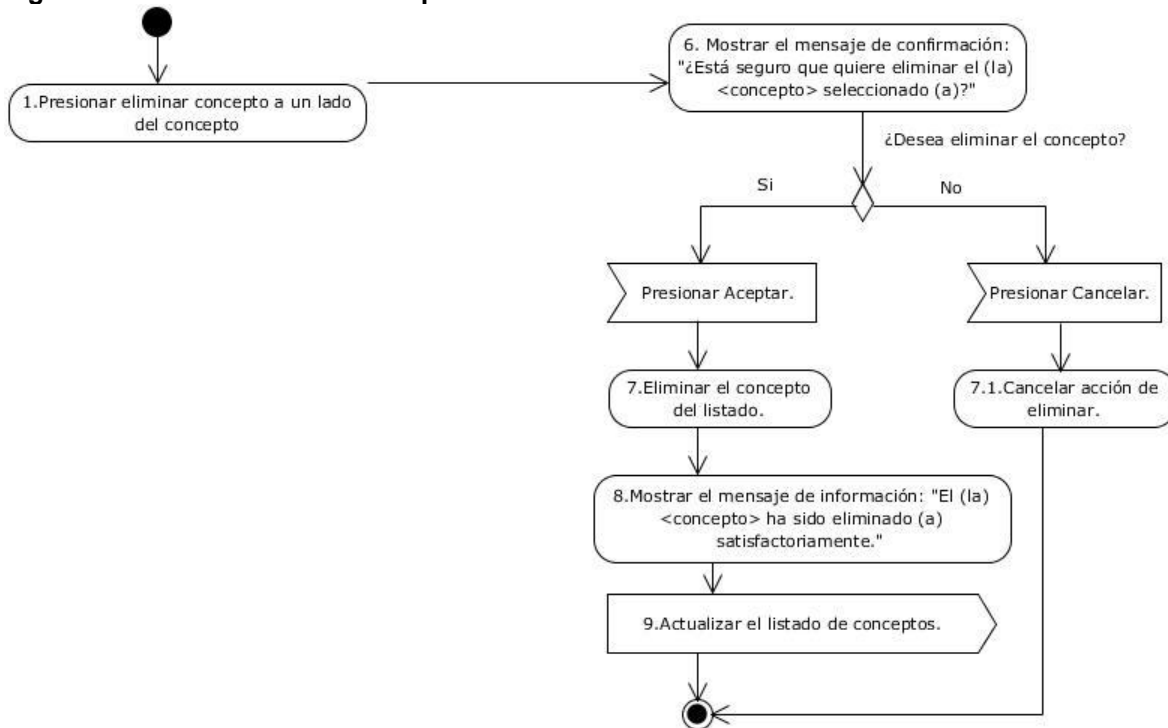


Figura 20: DERF Eliminar concepto

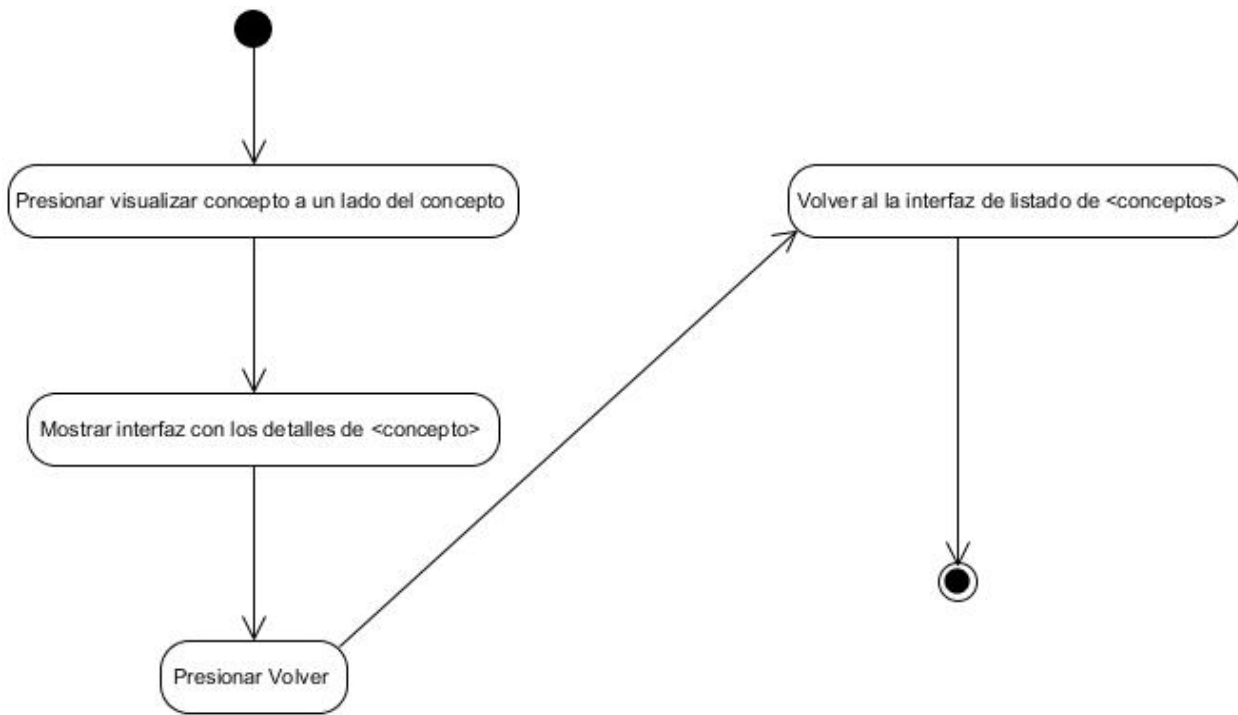


Figura 21: DRP Mostrar concepto

<b>&lt;&lt;requisito&gt;&gt; Asociar concepto</b>
Código = "SATD-STD-RS"
Tipo = "Funcional"
Conceptos = "infractor, base legal"
Precondiciones = "debe existir al menos un incidente"
Pre-requisito = "Adicionar ilegalidad"
Método de verificación = "DCP"
Riesgo = "Medio"

Figura 22: DRF Asociar concepto

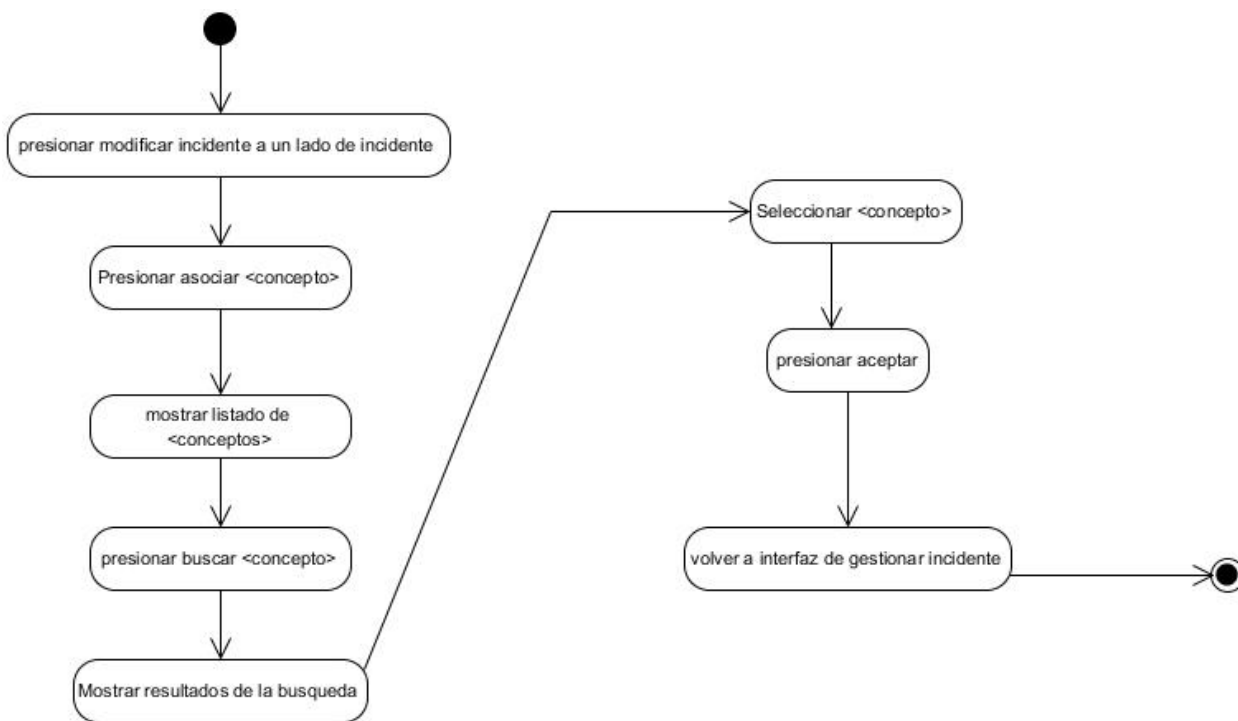


Figura 23: DERF Asociar concepto

<b>&lt;&lt;requisito&gt;&gt; buscar concepto</b>
Código = "SATD-STD-RS" Tipo = "Funcional" Conceptos = "incidente, inspector, multa infractor" Precondiciones = "debe existir al menos un conce..." Pre-requisito = "No procede" Método de verificación = "DCP" Riesgo = "Medio"

Figura 24: DRF Buscar concepto

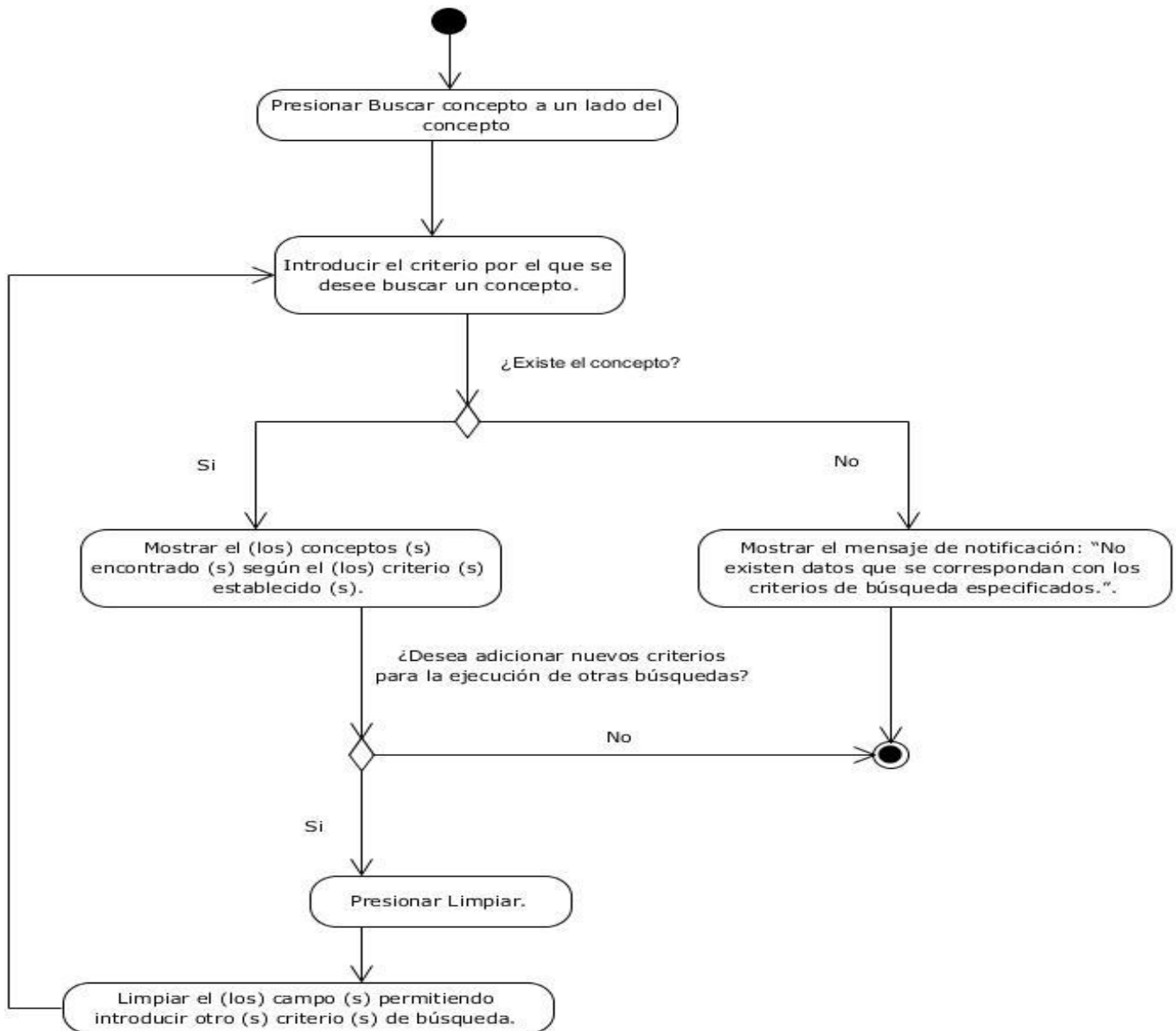


Figura 25: DERF Buscar concepto

<b>&lt;&lt;requisito&gt;&gt; Requirement</b>
Código = "SATD-STD-RS"
Tipo = "Funcional"
Conceptos = "listado de inspectores, listado de multas, reportes"
Precondiciones = "debe existir al menos un concepto"
Pre-requisito = "No procede"
Método de verificación = "DCP"
Riesgo = "Medio"

Figura 26: DRF Exportar concepto

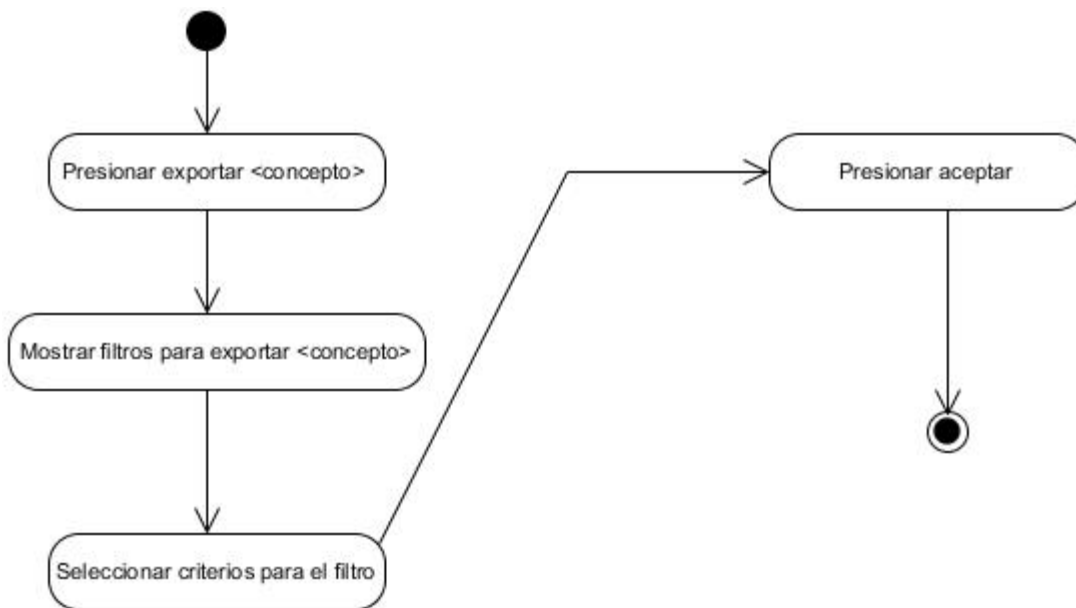


Figura 27: DERF Exportar concepto

Anexo 2 Prototipos de interfaz de usuario (PIU).

**Datos primarios**

<b>Categoría*</b> <input type="text" value="Categoría d..."/>	<b>Tipo de ilegalidad*</b> <input type="text" value="Tipo de ilegali..."/>	<b>Cuál</b> <input type="text"/>	<b>Estado*</b> <input type="text" value="Registrada"/>
<b>Fecha de detección*</b> <input type="text" value="10/31/2016"/>	<b>Fecha de cierre*</b> <input type="text" value="10/31/2016"/>	<b>Prioridad*</b> <input type="text" value="Media"/>	<b>Tipo de violación</b> <input type="text" value="Seleccione..."/>
<b>Inspector</b> <input type="text" value="Inspector:C..."/>	<b>En suelo de</b> <input type="text" value="Seleccione..."/>	<b>Obligación de hacer</b> <input type="text" value="Seleccione..."/>	<input type="checkbox"/> <b>PIEI</b>
<b>Consejo popular</b> <input type="text" value="Lowrey"/> Florencia/Ciego de Avila		<b>Dirección</b> <input type="text" value="Dirección"/>	

**Infractores**

Nombre y apellidos	Dirección	Construye
Alain Pérez González	Calle A % 21 y 23, Vedado, C.Habana	<input checked="" type="checkbox"/>
Carmen Rojas Sánchez	Esquina 23 y G, Vedado, C. Habana	<input type="checkbox"/>

**Multas**

Matriz	Fecha imposición	Violación:Cuantía	Infractor	Pagada	Opciones
15478	21/07/2017	272 8-c 500	Pedro Gonzalez	No	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

**Archivos**

**Observaciones**

<b>Participantes</b> <input type="text"/>	<b>Aseguramientos</b> <input type="text"/>	<b>Observaciones</b> <input type="text"/>
--	---	--

Figura 28: PIU Adicionar incidente  
 Fuente: Creación de los autores.

**Nota: los PIU para asociar conceptos están contenidos en el PIU de adicionar incidente a través de los botones adicionar multa, adicionar inspector y el campo inspector.**

Modificar contraseña

Contraseña anterior

Contraseña nueva

Confirmar contraseña

Cancelar    Aceptar

**Figura 29: PIU modificar contraseña**  
Fuente: Creación de los autores.

Matriz, Fecha de imposición, Infractor

Matriz

Fecha de imposición

Infractor

Cancelar    Limpiar    Buscar

**Figura 30: PIU Buscar multa**  
Fuente: Creación de los autores.



Listado de multas						
Matriz	Ilegalidad : Obligación de hacer	Fecha imposición	Violación:Cuantía	Infractor	Pagada	
<input type="checkbox"/>	15478	Asentamientos y barrios ilegal...	21/07/2017	272 8-c 500	Pedro Gonzalez	No

Página 1 Cantidad por página 5 1-5 de 8

Esto es un modal con todos los campos

Nueva multa

Matriz *	Fecha de imposición *	Cuantía *	Base legal *
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="272 8-c"/>
Infractor *	Inspector *	Procedencia	<input type="checkbox"/> Se cobró
<input type="text"/>	<input type="text" value="754878: Ramón Gar"/>	<input type="text" value="Procede"/>	
<input type="checkbox"/> Se reclamó			
Observaciones			
<div style="border: 1px solid #ccc; height: 40px;"></div>			
<input type="button" value="Cancelar"/> <input type="button" value="Aceptar"/>			

Modal para adicionar base legal

Seleccionar base legal

Base legal <ul style="list-style-type: none"> <li>[-] Ley 272                         <ul style="list-style-type: none"> <li>[-] Capítulo I                                 <ul style="list-style-type: none"> <li>● Inciso a</li> <li>● Capítulo II</li> </ul> </li> </ul> </li> </ul>	Contenido <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">                     El inciso a plantea.....                 </div>
<input type="button" value="Cancelar"/> <input type="button" value="Aceptar"/>	

Se Muestra en lugar del modal de multas (cambia).

**Figura 31: PIU Gestionar multa**  
**Fuente: Creación de los autores.**

Buscar ciudadano

Datos generales


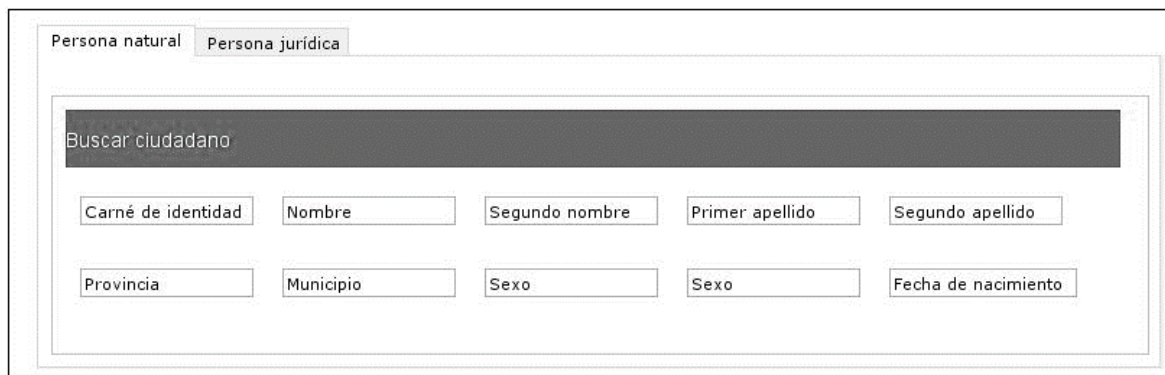
Clave \* 
 Municipio \* 
 Fecha inicio \* 
 Fecha fin

Observaciones

Buscar ciudadano

CI	Nombre(s) y Apellidos	Dirección
88062936635	Pepe Pérez Díaz	# 3 % Reineiro y Calle Real, Holguín, Holguín

Figura 32: PIU Gestionar inspector  
Fuente: Creación de los autores.



**Figura 33: PIU Gestionar infractor**  
**Fuente: Creación de los autores.**



**Figura 34: PIU Exportar listado de multas**  
**Fuente: Creación de los autores.**



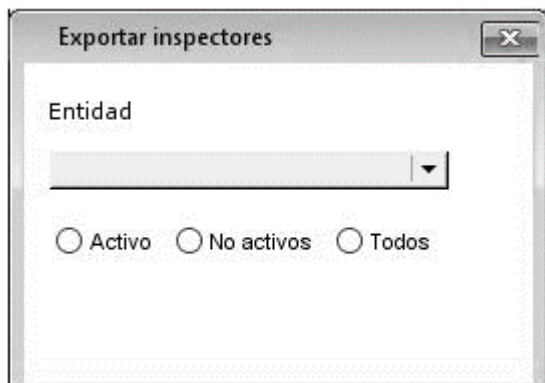
Exportar reporte

Entidad

Fecha desde

Fecha hasta

**Figura 35: PIU Exportar reporte**  
Fuente: Creación de los autores.



Exportar inspectores

Entidad

Activo  No activos  Todos

**Figura 36: PIU Exportar listado de inspectores**  
Fuente: Creación de los autores.

Anexo 3: Diagramas Entidad Relación.

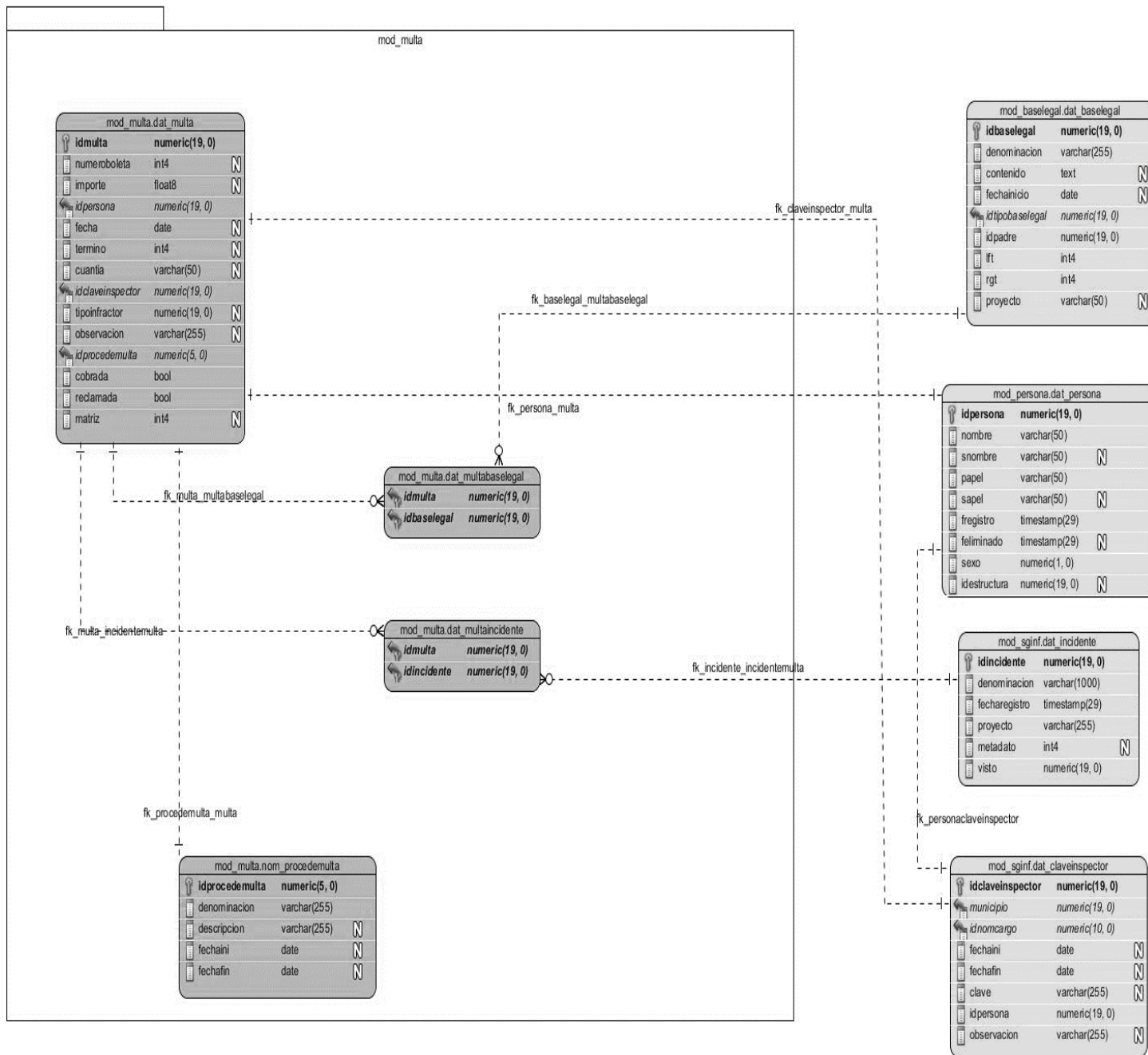


Figura 37 - Diagrama entidad-relación mod\_multa  
Fuente: Creación de los autores.

Anexo 4: Diagramas de clase del diseño con estereotipos web (DCD).

Fuente: Creación de los autores.

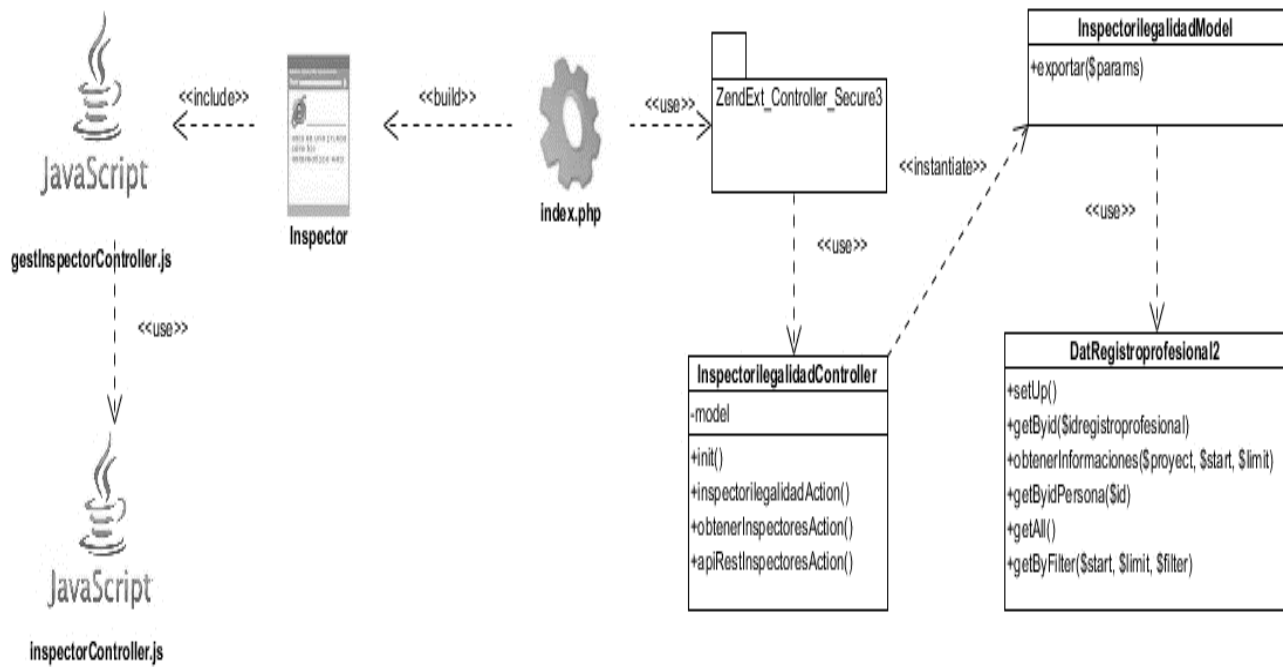


Figura 38: DCD Exportar listado de inspectores

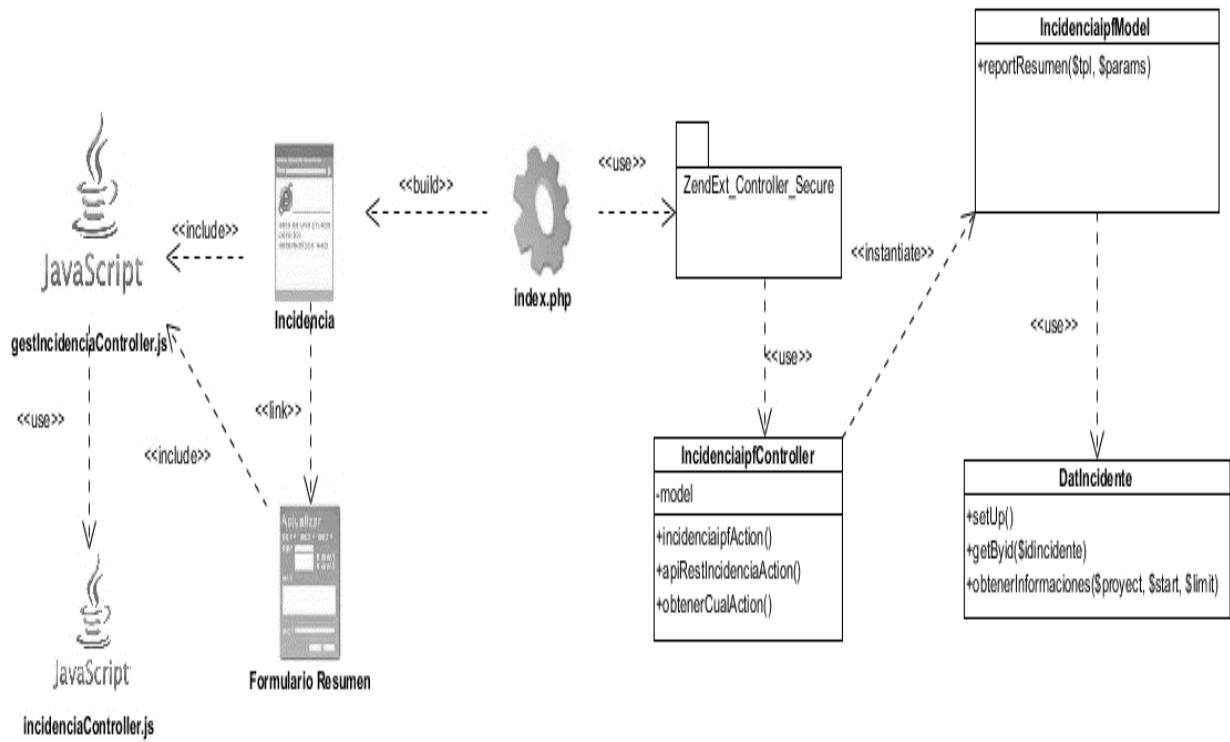


Figura 39: DCD Exportar reporte resumen.

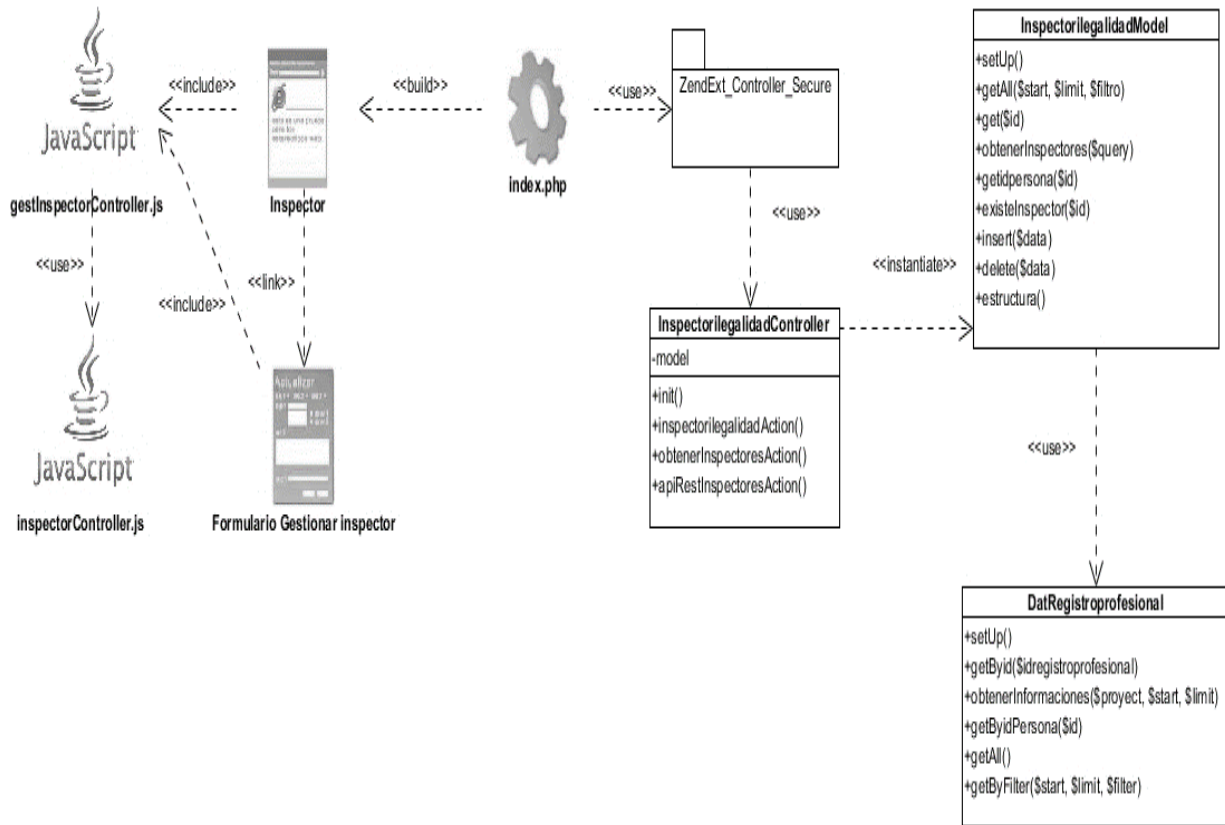


Figura 40: Gestionar inspector.



**Anexo 5: Interfaces de usuario (IU).**

**Fuente: Creación de los autores.**

Listado de ilegalidades									
<input type="checkbox"/>	Ilegalidad	Consejo popular	Infraactor	Fecha de detección	Plazo	Obligación de hacer			
<input type="checkbox"/>	Sistema electrico En inmuebles de personas jurídicas, en suelo Estatal	Jiguaní Sur	Emp. Eléctrica (OBE)	03/01/2017	-430	Legalización			
<input type="checkbox"/>	Vivienda En inmuebles de personas naturales, en suelo Privado	Campechuela 2	Amauris Batista Baldoquín	04/01/2017	-426	Paralización			
<input type="checkbox"/>	Vivienda En inmuebles de personas naturales, en suelo Privado	LAS CAOBAS	Minerva Quijala Peña	06/01/2017	-426	Demolición			
<input type="checkbox"/>	Vivienda En inmuebles de personas naturales, en suelo Privado	Rosa La Bayamesa	Mileysis Paneque Dominguez	06/01/2017	-426	Paralización			
<input type="checkbox"/>	Vivienda En inmuebles de personas naturales, en suelo Privado	SAO GRANDE	Daniel Cespedes Linares	04/01/2017	-426	Restitución			
<input type="checkbox"/>	Corral En inmuebles de personas naturales, en suelo Privado	El Horno	-	05/01/2017	-425	Demolición			

**Figura 41: IU Listado de ilegalidades**



Adicionando Base legal ✕

Tipo \*

Denominación \*

ACEPTAR

Figura 42: IU Adicionar base legal



Nuevo infractor

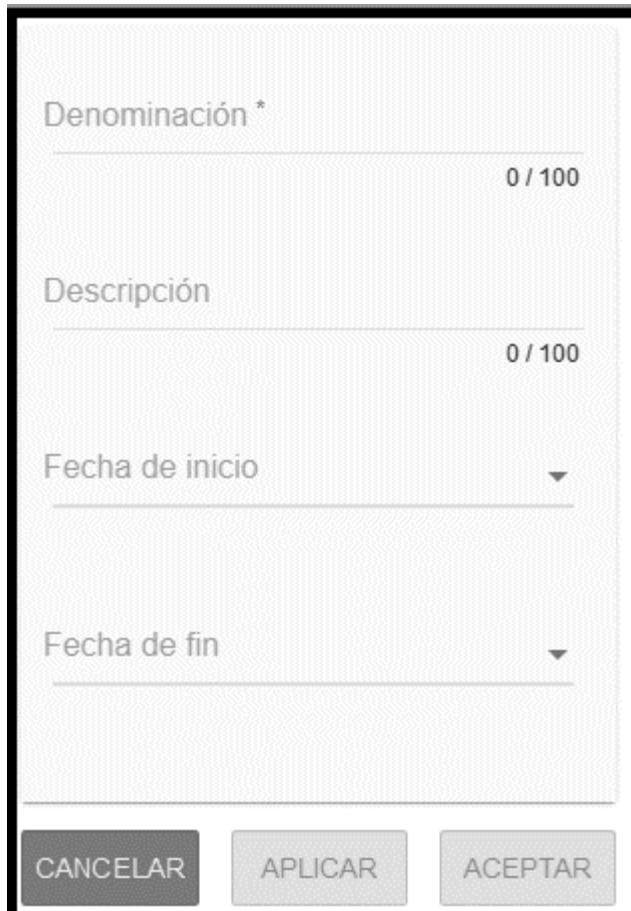
Infractor \*

Modo en que vi...

Construye

CANCELAR APLICAR ACEPTAR

Figura 43: IU Adicionar infractor



Denominación \*

0 / 100

Descripción

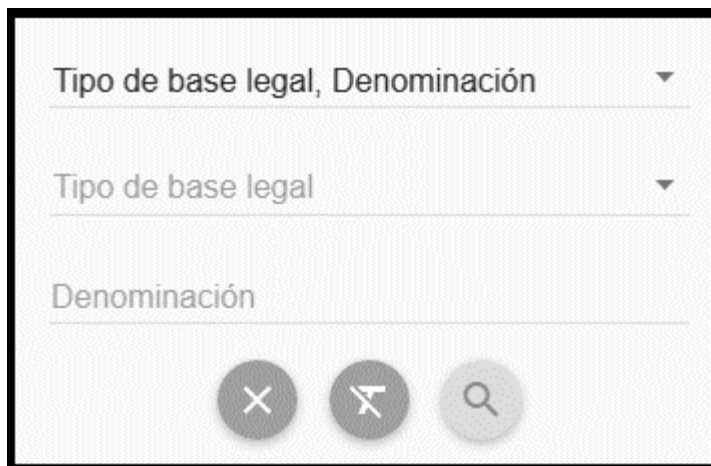
0 / 100

Fecha de inicio ▼

Fecha de fin ▼

CANCELAR APLICAR ACEPTAR

Figura 44: IU Adicionar nomenclador



The screenshot shows a search interface with three input fields and three action buttons. The first field is labeled "Tipo de base legal, Denominación" and has a dropdown arrow. The second field is labeled "Tipo de base legal" and also has a dropdown arrow. The third field is labeled "Denominación". Below the fields are three circular buttons: the first contains an 'X' (clear), the second contains a crossed-out 'X' (cancel), and the third contains a magnifying glass (search).

Figura 45: IU Buscar concepto



Figura 46: IU Flujo de trabajo

Anexo 6: Diseños de casos de pruebas (DCP).

Id del escenario	Escenario	Variable 1	Variable 2	Variable 3	Variab le 4	Variab le 5	Variab le 6	Variab le 7	Variab le 8	Variab le 9	Respuesta del sistema	
		Denominación	Tipo de ilegalidad	Consejo popular	Prioridad	Inspector	Estado	Detección	Cierre	Dirección		
EP 1.1	Adicionar ilegalidad de forma correcta.	V(selección)	V(selección)	V(selección)	V(Selección)	V(Selección)	V(Selección)	V(Selección)	V(Selección)	V(Selección)	V(menos de 225 caracteres)	Se guardan los datos y se muestra el mensaje: <i>“El ilegalidad ha sido adicionado satisfactoriamente.”</i> .
EP 1.2	Adicionar ilegalidad de forma incorrecta.	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	NA	Se señala el campo en rojo y se muestra el mensaje: <i>“Este campo es obligatorio.”</i> . No se habilita la opción <b>Guardar</b> .

		NA	NA	NA	NA	NA	NA	NA	NA	I(Má s de 255 caractere s)	Se muestra el campo en rojo y el mensaje: "Solo admite hasta 255 caracteres."
<b>EP 1.3</b>	Can celar acci ón.	NA	NA	NA	NA	NA	NA		NA		Se anulan las accion

**Tabla 14: DCP Adicionar ilegalidad**

Fuente: Creación de los autores.

Id del escenario	Escenario	Variable	Respuesta del sistema
		Botón eliminar	
EP 1.1	Eliminar Ilegalidad de forma correcta	NA	Se muestra el mensaje "Ilegalidad eliminada"
Ep 1.2	Cancelar acción	NA	Se anula la acción

**Tabla 15: DCP Eliminar ilegalidad**

Fuente: Creación de los autores.

Id del escenario	Escenario	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6	Variable 7	Variable 8	Respuesta del sistema	
		Denominación	Tipo de ilegalidad	Consejo popular	Inspector	Estado	Detección	Cierre	Dirección		
EP 1.1	Modificar ilegalidad de forma correcta	V(selección)	V(selección)	V(selección)	V(selección)	V(selección)	V(selección)	V(selección)	V(selección)	V(menos de 225 caracteres)	Se muestra el mensaje "Ilegalidad eliminada"
Ep 1.2	Modificar ilegalidad de forma incorrecta	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Se señala el campo en rojo y se muestra el mensaje: "Este campo es obligatorio.". No se habilita la opción <b>Guardar</b> .
EP 1.3	Cancelar acción	NA	NA	NA	NA	NA	NA	NA	NA	NA	Se anula la acción

Tabla 16: DCP Modificar ilegalidad

Fuente: Creación de los autores.

Id del escenario	Escenario	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6	Variable 7	Respuesta del sistema
		CI	Clave	Cargo	Municipio	Fecha Inicio	Fecha fin	Observación	
EP 1.1	Adicionar ilegalidad de forma correcta	V(11 dígitos)	V(8 caracteres)	V(seleccionar)	V(seleccionar)	V(seleccionar)	V(seleccionar)	V(225 caracteres)	Se muestra el mensaje "inspector adicionado"
EP 1.2	Adicionar inspector de forma incorrecta	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	I(Vacío)	Se señala el campo en rojo y se muestra el mensaje: "Este campo es obligatorio.". No se habilita la opción <b>Guardar.</b>
		I(menos de 11 dígitos)	NA	NA	NA	NA	NA	NA	NA
EP 1.3	Cancelar acción	NA	NA	NA	NA	NA	NA	NA	Se anula la acción

Tabla 17: DCP Adicionar inspector

Fuente: Creación de los autores.



Id del escenario	Escenario	Variable 1	Variable 2	Variable 3	Respuesta del sistema
		Tipo	Denominación	Cuadro de texto	
EP 1.1	Modificar base legal de forma correcta	V(seleccionar)	V(seleccionar)	V(cadena de caracteres)	Se muestra el mensaje "base legal adicionada"
Ep 1.2	Modificar base legal de forma incorrecta	I(Vacío)	I(Vacío)	I(Vacío)	Se señala el campo en rojo y se muestra el mensaje: "Este campo es obligatorio.". No se habilita la opción <b>Guardar</b> .
		NA	NA	I(Vacío)	Se señala el campo en rojo y se muestra el mensaje: "Base legal sin cuerpo". No se habilita la opción <b>Guardar</b> .
EP 1.3	Cancelar acción	NA	NA	NA	Se anula la acción

Tabla 18: DCP Modificar base legal

Fuente: Creación de los autores.

Id del escenario	Escenario	Variable	
		Botón eliminar	Respuesta del sistema
EP 1.1	Eliminar inspector de forma correcta	NA	Se muestra el mensaje "Inspector eliminado"
Ep 1.2	Cancelar acción	NA	Se anula la acción

**Tabla 19: DCP Eliminar inspector**

Fuente: Creación de los autores.

Id del escenario	Escenario	Variable 1	Variable 2	Variable 3	Respuesta del sistema
		Entidad	Fecha inicio	Fecha fin	
EP 1.1	Exportar multa de forma correcta	V(seleccionar)	V(seleccionar)	V(seleccionar)	Se muestra una ventana de descarga
Ep 1.2	Exportar multa de forma incorrecta	I(Vacío)	I(Vacío)	I(Vacío)	Se señala el campo en rojo y se muestra el mensaje: "Este campo es obligatorio.". No se habilita la opción <b>Exportar</b> .
EP 1.3	Cancelar acción	NA	NA	NA	Se anula la acción

**Tabla 20: DCP Exportar multa**

Fuente: Creación de los autores.

Id del escenario	Escenario	Variable 1	Variable 2	Variable 3	Respuesta del sistema
		Entidad	Fecha inicio	Fecha fin	
EP 1.1	Exportar reporte de forma correcta	V(seleccionar)	V(seleccionar)	V(seleccionar)	Se muestra una ventana de descarga
Ep 1.2	Exportar reporte de forma incorrecta	I(Vacío)	I(Vacío)	I(Vacío)	Se señala el campo en rojo y se muestra el mensaje: "Este campo es obligatorio.". No se habilita la opción <b>Exportar</b> .
EP 1.3	Cancelar acción	NA	NA	NA	Se anula la acción

Tabla 21: DCP Exportar reporte

Fuente: Creación de los autores.



**Anexo 7: Cuestionario #1**

Cuestionario realizado al panel de expertos en la gestión de ilegalidades de urbanismo y vivienda del IPF.

**Evaluación del Sistema de Gestión de Ilegalidades de la DPPF de Granma. XETID.**

**Personas que deben estar presentes:** especialistas del IPF.

Encuesta totalmente anónima realizada con el objetivo de validar el Sistema de Gestión de Ilegalidades de la DPPF de Granma.

**Sección I.**

Al reconocer en usted, un experimentado y prestigioso especialista en el campo de la Gestión de Ilegalidades de urbanismo y vivienda, profundo conocedor de las temáticas relacionadas con la toma de decisiones, así como su gestión eficaz a través del uso de sistemas de información. Si está Usted de acuerdo con ofrecernos su valiosa ayuda, se necesita antes de consultarlo determinar su coeficiente de competencia en el tema, a los efectos de reforzar la validez del resultado de la consulta. Por esa razón, se necesita responda las siguientes preguntas de la forma más objetiva que le sea posible.

**Marque con una cruz (X) en la tabla siguiente, el valor que se corresponde con el grado de conocimiento que usted posee sobre la temática “Gestión de Ilegalidades de urbanismo y vivienda” (considere la escala presentada ascendente, es decir, el conocimiento sobre el tema va creciendo desde 0 hasta 10).**

Escala	0	1	2	3	4	5	6	7	8	9	10
Grado de conocimiento											

**Sección II:**

Realice una autovaloración del grado de influencia, conocimiento y dominio que posee de cada una de las temáticas en Gestión de Ilegalidades de urbanismo y vivienda que se presentan a continuación:

**Marque con una cruz (X) según corresponda en A (Alto), M (Medio) y B (Bajo).**

Temáticas por área de conocimiento	Grado de conocimiento de cada una de las temáticas		
	A	M	B
Registro de ilegalidades			
Control de ilegalidades			
Detección de ilegalidades			
Asociación de multa			
Base legal			
Generación de reportes referentes a las ilegalidades			

**Sección III:**

Evaluación del Sistema de Gestión de Ilegalidades de la DPPF de Granma.

**Marque con una cruz (X) el valor que se corresponde con el grado de acuerdo que usted posee sobre las afirmaciones (1- Totalmente en desacuerdo(TD), 2- En desacuerdo(ED), 3- Ni de acuerdo, ni en desacuerdo(NI), 4- De acuerdo(DA), 5- Totalmente de acuerdo(TA)) de la siguiente tabla:**

Afirmación	Opción				
	1	2	3	4	5
Considero que el sistema ahorra tiempo en el proceso de gestión de las ilegalidades					x
Considero que las ilegalidades registradas en el sistema pueden ser consultadas en cualquier momento.					x
Considero que los gráficos del módulo de análisis constituyen información visual de importancia para la toma de decisiones.					x
Considero que los reportes que genera el sistema son importantes para la toma de decisiones.					x
Considero que de forma general un usuario del sistema al disponer de toda la información que brinda el sistema obtiene el apoyo necesario para la toma de decisiones.					x
Considero que de forma general el sistema contribuye en el proceso de gestión de las ilegalidades					x

Gracias por su tiempo.

### Anexo 8: Cuestionario #2

El siguiente cuestionario está dirigido a los usuarios del Sistema de Gestión de Ilegalidades de la DPPF de Granma con el objetivo de conocer el grado de satisfacción con el mismo.

Lea cuidadosamente cada una de las interrogantes que aparecen y responda según corresponda.

GRACIAS POR SU COLABORACION.

#### Sección I:

Marque con una X el nivel de cumplimiento según corresponda en la siguiente tabla:



Criterio	Escala		
	Alta	Media	Baja
Nivel de retroalimentación que brinda el sistema en los niveles de IPF.			
Nivel de satisfacción con el sistema.			
Calidad de la información gestionada por el sistema.			
Nivel de subjetividad de la información disponible.			

**Sección II:**

Marque con una X el tiempo aproximado que demora el sistema para la generación de informes referentes a las ilegalidades:

Hasta 20seg  Entre 21 y 40 seg. Aproximado  Más de 1min. Aproximado

**Sección III:**

Sugerencias para el equipo de desarrollo del sistema.

1-) \_\_\_\_\_

2-) \_\_\_\_\_

3-) \_\_\_\_\_

Anexo 9: Modelos de reportes generados (MRG)

Fuente: (IPF, 2018).

ENTIDAD:					DESDE:					
Municipio	1	2	3	4	5	6	7	8	9	
	No.	Medida Aplicada	Artículo del Decreto No. 272 aplicado	Descripción de la Ilegalidad	Plazo para erradicar	Autoridad que aplicó la medida	Nombre y Dirección de la persona natural infractora	Entidad infractora organismo al que pertenece y dirección	Observaciones	fecha de detección

Tabla 22: MRG Instrucción #2

Total de ilegalidades en el territorio	Clasificación de las ilegalidades						Asentamientos y barrios ilegales.	
	En inmuebles de personas jurídicas	En inmuebles de personas naturales		En Terrenos bajo responsabilidad de personas jurídicas		En terrenos no estatales	Cantidad de asentamientos y barrios	Cantidad de viviendas
		Viviendas	Otras obras	Viviendas	Otras obras			

Tabla 23: MRG Resumen



Ilegalidades en la zona costera y de protección, diferenciando las ilegalidades en la duna

N O	Ilegalidades	Infractores (persona jurídica o natural)	Estatus legal		Residencia		Tipo de acción	Plazo	Responsable	Año de construcción	Participantes	Aseguramiento
			Legal	Ilegal	Permanente	Veraño						

Tabla 24: MRG Resumen por categoría

NO	Entidades	Total de Ilegalidades	Inmuebles Viviendas	Otras Obras	
				Inmuebles	No Inmuebles
	<b>Total</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

Tabla 25: MRG Resumen por organismo

No	Entidades	Organ	Total Ilegalidades	Solucionar en el Año 2018												2018	2019	2020	2021
				1er			2do			3ro			4to						
				E	F	M	A	M	J	J	A	S	O	N	D				
<b>Total</b>																			

Tabla 26: MRG Resumen de ilegalidades por resolver en el trimestre

**Anexo 10: Prueba de sistema (carga y estrés).**

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
116 /std/std_...	1000	677	625	979	121	1822	0,00%	20,6/sec	30,4
117 /std/std_...	1000	647	695	842	91	1119	0,00%	20,5/sec	31,4
118 /std/std_...	1000	655	701	853	88	1098	0,00%	20,7/sec	34,0
123 /std/std_...	1000	94	31	317	3	718	0,00%	21,0/sec	142,6
127 /favicon.ico	1000	48	26	100	1	497	0,00%	21,4/sec	6,0
128 /std/std_...	1000	34	26	69	2	351	0,00%	21,6/sec	48,1
129 /std/std_...	1000	34	27	69	3	360	0,00%	21,7/sec	144,4
130 /std/std_...	1000	51	26	117	2	595	0,00%	21,7/sec	60,2
131 /std/std_...	1000	30	24	47	2	475	0,00%	21,7/sec	19,3
132 /std/std_...	1000	28	24	43	1	330	0,00%	21,7/sec	19,9
133 /std/std_...	1000	29	25	60	3	330	0,00%	21,7/sec	105,2
134 /std/std_...	1000	29	24	66	2	296	0,00%	21,7/sec	19,8
135 /std/std_...	1000	29	24	60	2	401	0,00%	21,6/sec	52,3
136 /std/std_...	1000	30	24	64	2	329	0,00%	21,7/sec	57,4
137 /std/std_...	1000	39	24	85	2	343	0,00%	21,7/sec	44,1
138 /std/std_...	1000	28	22	36	2	330	0,00%	21,7/sec	34,8
139 /std/std_...	1000	26	22	37	2	307	0,00%	21,7/sec	26,6
140 /std/std_...	1000	25	22	38	2	198	0,00%	21,7/sec	47,1
141 /std/std_...	1000	24	22	39	1	253	0,00%	21,7/sec	20,2
142 /std/std_...	1000	25	23	39	2	197	0,00%	21,7/sec	19,0
143 /std/std_...	1000	28	26	43	3	252	0,00%	21,7/sec	117,1
144 /std/std_...	1000	38	28	83	3	280	0,00%	21,7/sec	98,7
145 /std/std_...	1000	26	25	37	2	266	0,00%	21,7/sec	37,8
146 /std/std_...	1000	25	24	36	2	265	0,00%	21,7/sec	29,0
148 /std/std_...	1000	25	24	37	2	284	0,00%	21,7/sec	61,7
149 /std/std_...	1000	25	22	37	2	418	0,00%	21,7/sec	14,8
150 /std/std_...	1000	25	21	38	2	418	0,00%	21,7/sec	13,6
147 /std/std_...	1000	29	22	44	2	415	0,00%	21,7/sec	35,0
151 /std/std_...	1000	42	23	100	2	413	0,00%	21,7/sec	18,5
152 /std/std_...	1000	640	655	949	59	1218	82,70%	21,7/sec	13,0
153 /std/std_...	1000	56	9	232	2	481	0,00%	21,8/sec	53,6
161 /success...	1000	1360	0	9030	0	10007	100,00%	18,0/sec	29,9
Total	32000	153	26	531	0	10007	5,71%	530,5/sec	1147,4

**Tabla 27: Resultado de la prueba de carga y estrés de la herramienta Jmeter**

Fuente: Creación de los autores.

**Anexo 11: Casos de prueba de integración (CPI).**

**Fuente: Creación de los autores.**

<b>Módulo al cual se integra</b>	<b>Seguridad</b>
<b>Condiciones de Ejecución</b>	El usuario debe autenticarse mediante el módulo Seguridad
<b>Descripción de la prueba</b>	Comprobar que los módulos Ilegalidad, Multa, Base Legal e Inspector no pueden ser accedidos por usuarios no autenticados por el modulo Seguridad.
<b>Entradas/Pasos de ejecución</b>	El usuario se autentica en el sistema
<b>Resultado esperado</b>	Los usuarios tienen acceso a las funcionalidades de acuerdo al rol y sus responsabilidades.
<b>Evaluación</b>	Prueba satisfactoria.

**Tabla 28: CPI Módulo Seguridad**

<b>Módulo al cual se integra</b>	<b>Flujo de Trabajo</b>
<b>Condiciones de Ejecución</b>	Se debe seleccionar una ilegalidad, inspector, base legal o multa.
<b>Descripción de la prueba</b>	Comprobar que el módulo Flujo de Trabajo registra la vida útil de una ilegalidad, multa, base legal e inspector en el sistema.
<b>Entradas/Pasos de ejecución</b>	El usuario selecciona una ilegalidad, inspector, base legal o multa para visualizar su flujo de trabajo.
<b>Resultado esperado</b>	Los usuarios pueden visualizar el flujo de trabajo de una ilegalidad, multa, base legal e inspector en el sistema.
<b>Evaluación</b>	Prueba satisfactoria.

**Tabla 29: CPI Módulo Flujo de Trabajo**

<b>Módulo al cual se integra</b>	<b>Persona</b>
<b>Condiciones de Ejecución</b>	Se debe contar con el nombre y apellidos o el carnet de identidad de la persona.
<b>Descripción de la prueba</b>	Comprobar que el módulo Persona devuelve los datos de la persona correctamente al hacerle la petición desde los módulos Ilegalidad, Multa, Inspector y Base Legal.
<b>Entradas/Pasos de ejecución</b>	El usuario selecciona introduce los datos de la persona.
<b>Resultado esperado</b>	Los usuarios pueden visualizar los datos de la persona y asociarla a los módulo Ilegalidad, Multa, Inspector y Base Legal.
<b>Evaluación</b>	Prueba satisfactoria.

Tabla 30: CPI Módulo Persona

<b>Módulo al cual se integra</b>	<b>Nomenclador</b>
<b>Condiciones de Ejecución</b>	Se debe seleccionar un nomenclador.
<b>Descripción de la prueba</b>	Comprobar que el módulo Nomenclador permite la gestión de los nomencladores asociados a los módulos Ilegalidad, Multa, Inspector y Base Legal.
<b>Entradas/Pasos de ejecución</b>	El usuario gestiona los nomencladores del sistema.
<b>Resultado esperado</b>	Los usuarios pueden gestionar los nomencladores del sistema.
<b>Evaluación</b>	Prueba satisfactoria.

Tabla 31: CPI Módulo Nomenclador

Anexo 12: Usuarios encuestados

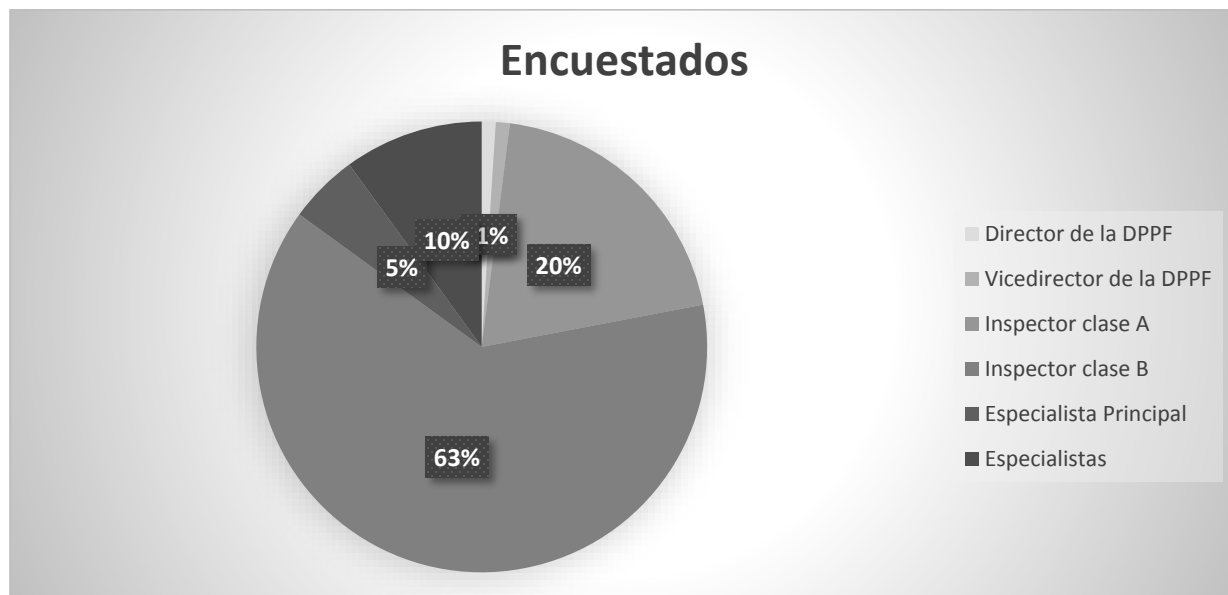


Figura 47: Representación en por ciento de los usuarios encuestados

Fuente: Creación de los autores.

**Anexo 13: Acta de aceptación del cliente**



Dirección: Calle 296ª e/ ave. 207 y 203,  
 municipio Boyeros, La Habana, Cuba  
 Teléfonos: (07) 837 3600 / 837 3613  
 Correo electrónico: xetid@xetid.cu

**Acta de aceptación**

En la Habana, a los 25 días del mes de mayo del año 2018

El sistema de Gestión de Ilegalidades desarrollado de conjunto por el Instituto de Planificación Física, la Dirección Provincial de Planificación Física de Granma y la Empresa de Tecnologías de la Información para la Defensa en forma abreviada XETID, actualmente se encuentra disponible en la nube de XETID, con contrato número 023-N-CV-17, el cual es necesario renovar, se certifica por el departamento de control, que es accesible por la dirección: <https://qi-ipf.xetid.cu> con todos los requerimientos de seguridad y disponibilidad acorde a las normas, está disponible desde cualquier parte del país con una conexión a la red de ETCSA, y paulatinamente se está extendiendo a otras provincias como Mayabeque, Artemisa, Holguín, entre otras.

De la misma manera el contrato de soporte número 039-N-CV-17 que cubre la respuesta a los imprevistos que surjan en el sistema, la corrección de errores y la incorporación de algunas funcionalidades nuevas, es necesario renovar y se certifica que actualmente se emplea, se pretende desplegar en el próximo año en al menos 6 provincias, con posibilidades de muchas más.

Y para que así conste, se extiende la presente Acta en dos (2) ejemplares, rubricados por Las Partes.

Por la PARTE CLIENTE

*[Handwritten signature]*  
 \_\_\_\_\_  
 Sixto Hoddero Rodríguez  
 Jefe Depto Informática  


Por la PARTE PRESTADOR

*[Handwritten signature]*  
 \_\_\_\_\_  
 Jefe de Proyecto  
 Yailen Pozo Laserra

Figura 48: Acta de aceptación del cliente  
 Fuente: SATD