

**Universidad de las Ciencias Informáticas**

**Facultad 3**



*Módulo Quejas, Peticiones o Denuncias del  
Sistema de Informatización de la Gestión de  
las Fiscalías fase II.*

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias  
Informáticas

**Autor(es):**

Deisy Leandra Cardoso Grimón

Amalia Aguilera Revuelta

**Tutor:** Ing. Yenier Figueroa Machado

**Co-tutor:** Ing. Annaliet Parra Pérez

La Habana, Curso 2011-2012



*“Lo fundamental es que seamos capaces de hacer cada día algo que perfeccione lo que hicimos el día anterior.”*

A stylized signature or logo in black ink, consisting of several fluid, connected strokes that form a unique, abstract shape. It is positioned in the bottom right corner of the page.



## *Declaración de Autoría*

---

### Declaración de Autoría

Declaramos que Deisy Leandra Cardoso Grimón y Amalia Aguilera Revuelta somos los únicos autores del trabajo de diploma Módulo Quejas, Peticiones o Denuncias del Sistema de Informatización de la Gestión de las Fiscalías fase II y se le reconocen a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2012.

\_\_\_\_\_  
**Firma del autor**

Deisy Leandra Cardoso Grimón

\_\_\_\_\_  
**Firma del autor**

Amalia Aguilera Revuelta

\_\_\_\_\_  
**Firma del tutor**

Ing. Yenier Figueroa Machado





### Datos de Contacto

#### **Datos de tutor:**

**Nombre:** Ing. Yenier Figueroa Machado

Ingeniero en Ciencias Informáticas. 5 años de experiencia laboral. Líder del proyecto Sistema de Informatización de Gestión de las Fiscalías fase II.

**Correo electrónico:** [yfigueroa@uci.cu](mailto:yfigueroa@uci.cu)

#### **Datos de Co-Tutor:**

**Nombre:** Ing. Annaliet Parra Pérez

Ingeniera en Ciencias Informáticas. Labora en el proyecto SIGEF con el rol de Analista Principal. Recién Graduado en Adiestramiento.

**Correo electrónico:** [apperez@uci.cu](mailto:apperez@uci.cu)

#### **Datos del Autor:**

**Nombre:** Deisy Leandra Cardoso Grimón.

Correo electrónico: [dlcardoso@estudiantes.uci.cu](mailto:dlcardoso@estudiantes.uci.cu)

#### **Datos del Autor:**

**Nombre:** Amalia Aguilera Revuelta.

Correo electrónico: [arevuelta@estudiantes.uci.cu](mailto:arevuelta@estudiantes.uci.cu)



# Agradecimientos

## Compartidos

Queremos agradecer a nuestro tutor y cotutora por toda su ayuda en la realización de este trabajo. A la UCI por ser nuestra casa. A todo el equipo de trabajo del proyecto SIGEF II por la valiosa ayuda brindada. A nuestros compañeros por acogernos y soportarnos durante estos 5 años. A los fiscales por brindarnos su atención y ayudarnos a que el trabajo saliera adelante.

A todos muchas gracias.

## Deisy

Me gustaría empezar agradeciendo a las personas más importantes en mi vida, esas que siempre han estado ahí en los buenos y malos momentos, que me han dado todo su apoyo sin esperar nada a cambio:

A mi mamá que es lo más grande que tengo en la vida, por estar a mi lado y poder contar siempre con su comprensión.

A mi papá por siempre haber confiado en mí, apoyarme y ayudarme cuando más lo he necesitado. Por sus consejos y cada minuto que dedicó a mis estudios, para que este sueño se hiciera realidad.

A mi hermanita linda que ha sido como mi mami en este último año.

A mi familia y a mis abuelitos que ocupan un lugar especial en mi corazón y que han hecho muchísimo por mí.

A mi compañera de tesis un especial agradecimiento por su apoyo incondicional.

A todas mis amistades en general por tantos momentos agradables, en especial (Leinita, Lisbecita, Saylita, Maide, Lisi, Yanita, Yazmincilla y Yunier).

A todos a los que mencioné y a los que no, pero en algún momento formaron parte de mi campo de acción. ....Muchas gracias de todo ♥.

## Amalia

A mis padres por todo el amor y la confianza depositada en mí durante toda mi carrera, por estar en cada momento cuando más los necesité, por ser mi motivación, por darme siempre todo su apoyo incondicional, ejemplo y guía; por lo orgullosos que se sienten hoy.

A mi hermano, por todo el apoyo que me ha brindado.

A Damisela y mi sobrina, por marcar momentos felices en mi vida.

A mis abuelas, que siempre han deseado ver este día.

A mis primos por las maldades y a toda mi familia, por ser parte importante en mi vida y por toda su preocupación.

A mi novio por darme todo su amor y cariño.

A mi compañera de tesis, que más que compañera es una amiga.

A todos los profesores que me dedicaron parte de su valioso tiempo con profesionalidad.

A todos mis amigos y compañeros, esos que sin dudas son los mejores que he tenido, los que más he querido y los que extrañaré muchísimo, en especial a Yadiani.

A los que mencioné y a los que se quedaron por mencionar, a todos los que siempre han creído en mí y me han querido incondicionalmente.

Muchas Gracias.



*Dedico este trabajo de diploma a los amores de mi vida, Mis padres, por su cariño, dedicación y confianza aún cuando pensaba que no podía llegar al final. Por sus palabras alentadoras (en los llantos del mes), por estar ahí, quererme y ayudarme en cada momento que viví en esta universidad. . . .*

*Gracias a los dos por haber hecho de mí la personita que hoy soy, su Deisyta felizmente alcanzó el sueño que todos esperábamos y quiere dedicarles el fruto de todo este esfuerzo a ustedes dos.*

*Los Amo.*

**Deisy**

*Quiero dedicar este trabajo a las personas que más quiero en mi vida:*

*A mi mamá, por su apoyo incondicional, por estar siempre pendiente y por todo el esfuerzo en convertirme en una mujer de bien.*

*A mi papá, que tanto ha contribuido en mi formación, por consentirme tanto y apoyar todas mis decisiones. Por todo el esfuerzo de estos últimos años, que me han hecho más fuerte.*

*A los dos por haberme dado la vida, por quererme en todo momento, por darme amor, cariño, confianza, educación y respeto, por darme mucho en lo poco. Aunque no se los diga los adoro.*

**Amalia**



## Resumen

El presente trabajo de diploma tiene como propósito desarrollar el módulo Quejas, Peticiones o Denuncias (QPD) del área Control de la Legalidad en los Establecimientos Penitenciarios (CLEP), que se encarga de controlar y supervisar el proceso de recepción, tramitación, investigación y respuesta que en el orden legal proceda a los planteamientos de los detenidos, acusados en prisión provisional, sancionados y asegurados, así como de sus familiares. (1)

Se realizó un estudio profundo de los procesos relacionados con las quejas, peticiones o denuncias que se llevan a cabo en el departamento CLEP de la Fiscalía General de la República, a partir de los cuales se identificaron las funcionalidades que el sistema debe tener. Se realizó el diseño del sistema y la evaluación de la calidad de los resultados obtenidos. Todas estas actividades se llevaron a cabo para facilitar el trabajo de implementación, poniendo a disposición una guía para que el producto sea confiable y eficiente.

Con la implementación de esta propuesta se espera que los fiscales tengan a su disposición un sistema que cumpla con sus expectativas, dotándolos de una aplicación que mejore sustancialmente el proceso de gestión fiscal, así como el manejo de la información utilizada para su trabajo.

### PALABRAS CLAVE

CLEP, Denuncia, Diseño, Gestión Fiscal, Funcionalidades, Petición, Queja, Sistema de Gestión de Información.



## ÍNDICE DE CONTENIDO

Introducción .....	1
Capítulo 1. Fundamentación teórica. ....	6
1.1. Introducción. ....	6
1.2. Conceptos fundamentales.....	6
1.3. Escenario actual del proceso. ....	7
1.4. Valoración del estado del arte. ....	8
1.5. Formulación de la propuesta de solución. ....	10
1.5.1. Características del proyecto SIGEF II. ....	10
1.5.2. Metodología de desarrollo de software.....	11
1.5.3. Herramientas y tecnologías utilizadas. ....	13
1.5.4. Marco de Trabajo.....	17
1.5.5. Entorno de desarrollo.....	18
1.5.6. Gestor de base de datos. ....	19
1.5.6.1. ORM Doctrine 2.....	20
1.5.7. Conceptualización de las técnicas utilizadas.....	20
1.5.8. Patrones.....	21
1.5.9. Métricas .....	24
1.6. Conclusiones Parciales.....	26
Capítulo 2. Análisis y Diseño. ....	28
2.1. Introducción. ....	28
2.2. Modelado del Negocio.....	28
2.2.1. Resultado de la aplicación de las técnicas de recopilación de información. ....	29
2.2.2. Descripción de subprocesos de negocio. ....	30
2.2.3. Modelo Conceptual. ....	34
2.2.4. Patrones de Negocio.....	34
2.2.5. Validación del Modelado del Negocio.....	35
2.2.6. Conclusiones parciales. ....	36
2.3. Requisitos. ....	36



2.3.1. Resultados de la aplicación de las técnicas de recopilación de información. ....	36
2.3.2. Especificación de requisitos. ....	38
2.3.3. Aplicación de los patrones de CU. ....	39
2.3.4. Validación de requisitos del proceso. ....	42
2.3.5. Conclusiones parciales. ....	46
2.4. Diseño. ....	46
2.4.1. Patrones de Diseño que implementa Symfony. ....	46
2.4.2. Diagrama de Secuencia. ....	47
2.4.3. Modelo lógico de datos. ....	48
2.4.4. Validación del diseño del proceso. ....	51
2.4.5. Conclusiones Parciales. ....	53
Capítulo 3. Implementación y Prueba. ....	54
3.1. Introducción. ....	54
3.2. Modelo de Implementación ....	54
3.2.1. Diagrama de Componentes. ....	54
3.2.2. Modelo de Despliegue. ....	55
3.2.3. Estándares de codificación. ....	56
3.3. Pruebas. ....	58
3.3.1. Prueba de Caja Negra. ....	58
3.3.2. Pruebas de unidad. ....	60
3.3.2.1. Prueba del camino básico. ....	60
3.4. Conclusiones Parciales. ....	63
Conclusiones Generales. ....	64
Recomendaciones. ....	65
Bibliografía. ....	66
Anexos. ....	71



## ÍNDICE DE FIGURAS

Figura Nº 1 Fases y disciplinas de RUP.....	12
Figura Nº 2 Patrón Secuencia.....	22
Figura Nº 3 Patrón Selección Exclusiva.....	22
Figura Nº 4 Patrón Mezcla Múltiple o Serialización.....	23
Figura Nº 5 Diagrama de Procesos de Negocio de las Quejas, Peticiones o Denuncias.....	29
Figura Nº 6 Diagrama de Procesos de Negocio del subproceso Tramitar Queja.....	31
Figura Nº 7 Sesión del diagrama Tramitar Queja, Petición o Denuncia.....	34
Figura Nº 8 Sesión del diagrama Tramitar Queja, Petición o Denuncia.....	35
Figura Nº 9 Sesión del diagrama Tramitar Queja, Petición o Denuncia.....	35
Figura Nº 10 Diagrama de Casos de Uso del Sistema.....	39
Figura Nº 11 Patrón Roles Comunes.....	40
Figura Nº 12 Patrón CRUD Parcial.....	40
Figura Nº 13 Diagrama de Secuencia del caso de uso Depurar Queja, Petición o Denuncia.....	48
Figura Nº 14 Modelo lógico de datos del Módulo Quejas, Peticiones o Denuncias.....	49
Figura Nº 15 Comportamiento de los valores de dependencia.....	53
Figura Nº 16 Diagrama de componentes.....	55
Figura Nº 17 Diagrama de despliegue.....	56
Figura Nº 18 Código fuente del método cargarFormulariosAction.....	62
Figura Nº 19 Grafo de flujo del método cargarFormulariosAction.....	62

## ÍNDICE DE TABLAS

Tabla: 1 Clasificación de las Clases.....	25
Tabla: 2 Descripción de procesos de negocio del sub-proceso Tramitar Queja.....	32
Tabla: 3 Especificación de Requisitos para el requisito Buscar Antecedentes.....	38
Tabla: 4 Especificación de casos de uso para el CU Buscar Documentos.....	40
Tabla: 5 Resultados de la métrica para evaluar los casos de uso.....	43
Tabla: 6 Descripción de las tablas del modelo lógico.....	49
Tabla: 7 Resultados de la Métrica Tamaño Operacional de Clases.....	51
Tabla: 8 Criterios de evaluación de la métrica AC.....	52
Tabla: 9 Cantidad de dependencias por clasificación.....	52
Tabla: 10 Resultados de los atributos de calidad de la métrica AC.....	53
Tabla: 11 Escenario Adicionar Queja, Petición o Denuncia del Caso de uso Gestionar Queja.....	59



## **Introducción**

La introducción masiva de las nuevas Tecnologías de la Información y las Comunicaciones (TIC) han influenciado fuertemente en la estructura y dinámica de los procesos económicos y sociales, redefiniendo aceleradamente las formas de producir, vender y competir, en prácticamente todos los sectores de bienes y servicios; así como en las nuevas formas de interacción y comunicación entre las personas y organismos de la sociedad.

Desde hace algunos años Cuba ha emprendido el reto de la informatización de la sociedad, con el objetivo de elevar la calidad de vida del pueblo y lograr más eficiencia en la economía. La informatización de la sociedad cubana es un proceso de utilización ordenada y masiva de las TIC para satisfacer las necesidades de información y conocimiento de todas las personas y sectores de la sociedad. La Universidad de Ciencias Informáticas (UCI) es un ejemplo de todo el proceso de informatización que se está desarrollando en Cuba. Fue creada como un proyecto de la Revolución, con el fin de formar jóvenes con amplios conocimientos en la rama de la informática, crear grupos de proyectos para brindar servicios informáticos tanto nacionales como para exportación y darle solución de esta manera a diferentes problemas existentes en el país.

La producción de software y la tarea de informatizar las diferentes áreas del país, recae principalmente en los proyectos productivos de la UCI, siendo uno de estos el Sistema de Informatización de la Gestión de las Fiscalías II (SIGEF). Desde el año 2007 la Fiscalía General de la República de Cuba (FGR) está inmersa en la informatización de sus principales procesos como mejora del cumplimiento de su objetivo de controlar y preservar la legalidad para el estricto cumplimiento de la Constitución, las leyes y demás disposiciones legales por parte de los organismos del Estado, entidades económicas, sociales y la población. El proyecto SIGEF fue creado con el objetivo de realizar una aplicación web que mejore la calidad de tramitación, supervisión y control en tiempo de los procesos fiscales cubanos teniendo como alcance obtener una fuerza fiscal más eficiente y con mayor economía en cada uno de sus procesos.

El proyecto SIGEF está compuesto por subsistemas que incluyen los módulos correspondientes a los procesos que representan en la FGR.

Este sistema contempla el área de Control de la Legalidad en los Establecimientos Penitenciarios (CLEP), que es el área de la FGR que controla los aspectos relacionados con las visitas de inspección a los establecimientos penitenciarios, así como las quejas, peticiones o denuncias de los familiares del implicado en un delito, del propio interno o detenido, además de la documentación que se genera en estos procesos. Para la primera fase del proyecto se decidió contemplar, en esta área solamente, el proceso de las visitas de inspección a los establecimientos penitenciarios.



Actualmente llevar a cabo estos procesos necesita una cantidad considerable de fiscales, de los que no dispone el país. Debido a esto, se hace engorroso el trabajo, la realización y el control de todos los procesos requieren más tiempo del esperado ya que diariamente se genera un gran número de documentos, viéndose afectado el análisis de los mismos de manera rápida y efectiva. Por su parte, toda la documentación que requiere la tramitación de la queja se encuentra manuscrita por lo que al paso del tiempo se deteriora, mostrando el contenido en tonos borrosos. Se deduce así que el mayor problema se enfoca en que toda la información está manuscrita, haciéndose difícil la búsqueda inmediata de una documentación determinada. Pasados los cinco años de permanencia de estos documentos en la fiscalía, se procede a la destrucción de los que cuya información, ya no mantiene la relevancia como para su conservación, imposibilitando su acceso en tiempos futuros. Son estos los factores que hacen real la necesidad por parte de la FGR de informatizar los procesos fiscales, en este caso, las quejas, peticiones o denuncias de la población.

Actualmente en la FGR se trabaja con un sistema de quejas que no satisface las necesidades de trabajo de los fiscales en cuanto a trámite de la reclamación. Este sistema solamente da la posibilidad de hacer registros y búsquedas en determinadas materias que finalmente no logran abarcar todo el círculo de trabajo de los fiscales.

Por todo lo antes planteado se necesita crear un mecanismo que facilite el registro de esta información, agilice y haga más efectivo el trabajo de los fiscales a lo largo de todo el territorio nacional. Para ello es menester agregarle nuevas funcionalidades al sistema ya existente (SIGEF I), de manera que se logre dar una solución eficiente a los problemas anteriormente tratados. Teniendo en cuenta que el sistema ya implementado se rige por una arquitectura definida en el proyecto, se preverá que las nuevas funcionalidades logren insertarse perfectamente en el mismo, obteniendo como resultado mejores servicios, haciendo uso de las Tecnologías de la Información y las Comunicaciones (TIC) que juegan un papel protagónico en este campo para la informatización y automatización de la mayoría de las esferas y procesos de la producción y los servicios.

Bajo estas circunstancias es identificado el siguiente **problema de la investigación**: ¿Cómo contribuir a mejorar la gestión del proceso de Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios, de manera que permita garantizar con mayor celeridad, el control de la legalidad y la preservación de la información, sobre la base del estricto cumplimiento de la Constitución?

Este problema se enmarca en el **objeto de estudio**: el proceso de desarrollo de software de gestión.



En función de resolver el problema propuesto se ha definido como **objetivo general**: desarrollar el módulo Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios en el proyecto Sistema de Informatización de la Gestión de las Fiscalías fase II, de manera que permita garantizar con mayor celeridad, el control de la legalidad y la preservación de la información, sobre la base del estricto cumplimiento de la Constitución, enmarcándose el **campo de acción** en: proceso de desarrollo de software de gestión fiscal.

Se define como **idea a defender** el desarrollo del módulo Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios en el proyecto Sistema de Informatización de la Gestión de las Fiscalías fase II, permitirá garantizar con mayor celeridad, el control de la legalidad y la preservación de la información, sobre la base del estricto cumplimiento de la Constitución.

Para darle cumplimiento al objetivo general propuesto se han definido los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico de la investigación.
- ✓ Obtener el Modelo de Diseño.
- ✓ Obtener el Modelo de Implementación.
- ✓ Validar la solución propuesta mediante pruebas.

Para desarrollar el presente trabajo, es necesario definir **tareas de la investigación** que guíen el camino a seguir para dar cumplimiento a los objetivos específicos:

- ✓ Estudio del estado del arte de sistemas informáticos de gestión fiscal.
- ✓ Estudio del proceso de desarrollo de software en el proyecto Sistema de Informatización de la Gestión de las Fiscalías fase II.
- ✓ Estudio de los procesos Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios.
- ✓ Modelado y descripción de los procesos Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios.
- ✓ Identificación y descripción de los requisitos asociados al módulo Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios en el proyecto Sistema de Informatización de la Gestión de las Fiscalías fase II.



- ✓ Realización del diseño del módulo Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios en el proyecto Sistema de Informatización de la Gestión de las Fiscalías fase II.
- ✓ Implementación del módulo Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios en el proyecto Sistema de Informatización de la Gestión de las Fiscalías fase II.
- ✓ Realización del diseño de casos de prueba del módulo Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios en el proyecto Sistema de Informatización de la Gestión de las Fiscalías fase II.
- ✓ Ejecución de pruebas de software al módulo Quejas, Peticiones o Denuncias del área Control de la Legalidad en los Establecimientos Penitenciarios en el proyecto Sistema de Informatización de la Gestión de las Fiscalías fase II.
- ✓ Validación de los resultados obtenidos a través de pruebas de caja negra al sistema para comprobar su correcto funcionamiento.

Para la realización del trabajo fue necesario la utilización de los siguientes métodos y técnicas de investigación:

### **Métodos Teóricos:**

*Histórico-Lógico:* se utilizó para conocer todos los antecedentes que existen sobre el proceso de quejas, peticiones o denuncias dentro de la FGR, específicamente en el departamento del CLEP, mediante la documentación generada en este.

*Analítico-Sintético:* se utilizó para el procesamiento de toda la información relacionada con el tema de investigación, analizando los documentos que permitieran extraer los elementos más significativos relacionados con el objeto de estudio.

*Modelación:* posibilitó la creación de los diferentes diagramas y modelos que ayudaron a un mejor entendimiento de las funcionalidades que debe cumplir el sistema y al estudio de las relaciones entre las mismas.

### **Métodos Empíricos:**

*Entrevista:* permitió realizar un profundo análisis de las necesidades del cliente para obtener información valiosa acerca del negocio y los requisitos en el desarrollo del software.

El presente trabajo de diploma está compuesto por 3 capítulos estructurados de la siguiente manera:



En el [Capítulo 1](#) se realiza la **Fundamentación Teórica** que comprende el estado del arte, la cual incluye un estudio de sistemas existentes para la gestión de Quejas, Peticiones o Denuncias en el ámbito internacional como nacional, se enuncian los principales conceptos relacionados con el tema, así como un estudio de las metodologías, lenguajes y herramientas usadas como apoyo para darle solución al problema planteado.

En el [Capítulo 2](#) se realiza el **Análisis y Diseño** del módulo. En este capítulo se expone el Modelado del Negocio y la descripción de los procesos de negocio mediante los Diagramas de Procesos de Negocio (DPN), donde se aplican métricas y métodos para validar los resultados obtenidos. Se definen los requisitos tanto funcionales y no funcionales, teniendo en cuenta las restricciones o políticas a cumplir por el sistema y su validación aplicando técnicas y métricas para medir la calidad de los mismos. Además se presenta el diseño a través de los diagramas derivados de dicha disciplina, haciendo uso de los patrones adecuados y la validación de todo el proceso utilizando las métricas correspondientes.

En el [Capítulo 3](#) se presenta la **Implementación y Prueba** de la solución. En este capítulo quedan establecidos el diagrama de despliegue y el diagrama de componentes, que conformarán el Modelo de Implementación, además de la descripción detallada de la implementación. Así mismo se exponen las pruebas realizadas a la solución para asegurar el correcto funcionamiento de la aplicación.



## Capítulo 1. Fundamentación teórica.

### 1.1. Introducción.

El presente capítulo aborda los elementos que permiten realizar un correcto análisis y diseño del sistema, sustentados en un estudio de estos elementos y los conceptos fundamentales de algunos de ellos. Se incluyen aspectos relacionados con las tendencias internacionales y las aplicaciones que informatizan el proceso de Quejas, Peticiones o Denuncias de las Fiscalías en el mundo. Se ofrece un estudio de las tecnologías y herramientas de desarrollo de software, además de las herramientas para el modelado de prototipos no funcionales de Interfaz de Usuario (IU). Se presentan patrones para cada una de las disciplinas desarrolladas, y los lenguajes de programación encaminados a la web. Finalmente se hace alusión a las métricas para evaluar la calidad de las disciplinas descritas en dicho capítulo.

### 1.2. Conceptos fundamentales.

Entre los principales conceptos usados en la investigación se encuentran los siguientes:

**FGR:** Órgano del Estado al que corresponde, como objetivo fundamental, el control y la preservación de la legalidad, sobre la base de la vigilancia del estricto cumplimiento de la Constitución, las leyes y demás disposiciones legales, por los organismos del Estado, entidades económicas, sociales y por los ciudadanos; la promoción y el ejercicio de la acción penal pública en representación del Estado. (2)

**Queja:** Inconformidad ciudadana sobre violación de sus derechos constitucionales, garantías legalmente establecidas e infracciones cometidas en actos o disposiciones de organismos y funcionarios del estado y de entidades económicas y sociales. (3)

**Petición:** Solicitud, peticiones de un derecho, que no entraña presunción de violación de la legalidad, y requiere orientación sobre la base de la norma jurídica establecida. (3)

**Promovente:** Persona natural que presenta una queja o petición.

**Impugnación:** Derecho que tiene el promovente y es expresado a través de la acción que realiza el mismo al no estar de acuerdo con la respuesta obtenida tras la tramitación de su queja.

**Legislación:** Son todas las leyes o documentos vigentes que pueden ser consultados para resolver un caso determinado.



## Capítulo 1. Fundamentación Teórica

---

**Legalidad:** Todo lo que es legal, es decir, que está permitido ante la ley.

**Orientación Jurídica:** Consiste en orientar, mediante escrito con el formato de una carta, al quejoso o reclamante sobre como encauzar en el orden legal su pedimento o solicitud, la procedencia o no de su reclamación o asunto planteado, con fundamento en la norma jurídica que así lo establece. (3)

**Traslado Definitivo (TD):** Tipo de traslado que decide darle un fiscal a una queja, petición o denuncia para que sea tramitada en el órgano que le compete.

**Traslado Bajo Control (TBC):** Tipo de traslado que se le da a una queja, petición o denuncia por no ser competencia de la fiscalía en la que se encuentra, la queja es enviada a la fiscalía inmediata inferior bajo indicaciones.

**Respuesta verbal:** Intercambio directo con la persona que posibilita una comprensión mayor y más completa de todas las interrogantes del promovente. (3)

**Solución operativa:** Respuesta a una queja, que mediante la acción del fiscal se resuelve conforme a la pretensión interesada por el promovente, y que no necesariamente implica la elaboración de un pronunciamiento por escrito. (3)

**Pronunciamiento no violatorio:** Cuando el fiscal, luego de practicadas las diligencias del caso, comprueba que no se ha quebrantado la legalidad, ofrecerá respuesta al promovente, con el formato de una carta, debidamente fundamentada, que abarque con un lenguaje sencillo en el orden técnico, los planteamientos formulados. (3)

### 1.3. Escenario actual del proceso.

Entre las instituciones encargadas de atender las Quejas, Peticiones o Denuncias, que en el orden legal formulan los ciudadanos sobre presuntas violaciones de sus derechos se encuentra el área de CLEP perteneciente a la Fiscalía General de la República de Cuba, la que conforme a lo regulado en el artículo 127 de la Constitución es el órgano del Estado al que corresponde, como objetivo fundamental, el control y la preservación de la legalidad sobre la base de la vigilancia del estricto cumplimiento de la Constitución, las leyes y demás disposiciones legales, por los organismos del Estado, entidades económicas y sociales y por los ciudadanos.

En dicha área el fiscal designado atiende, investiga y responde en el plazo de sesenta días las quejas, peticiones o denuncias, que en el orden legal formulan los ciudadanos. (4)



Actualmente los procedimientos por parte de la FGR para tramitar el proceso de quejas, peticiones o denuncias se realiza de forma manual, cada uno de los casos genera un alto volumen de información por lo que se utiliza gran cantidad de tiempo para generar todos los documentos pertinentes. Estos documentos se archivan en las condiciones propicias, pero al paso del tiempo se deterioran, dificultando su uso posterior. Lo que contribuye a la demora de los procesos incluso hasta 3 meses más de lo que se deberían demorar. El envío de expedientes en caso de ser necesario a instancias superiores se realiza por correo tradicional; que suma más tiempo si el mismo es devuelto para que se realicen otras investigaciones. (4)

### 1.4. Valoración del estado del arte.

#### Tendencias Internacionales y Nacionales

Las quejas, peticiones o denuncias de la población sobre distintas situaciones que respectan a violaciones de sus derechos, es un asunto de carácter internacional. En casi todos los lugares del mundo existe un órgano o persona natural que las atiende. Existe un “defensor” en la mayoría de los países capitalistas, que recibe las quejas y las dirige al fiscal correspondiente para que éste investigue y adopte las medidas oportunas, en su caso, para subsanar las deficiencias expuestas. En países como Puerto Rico, Bolivia, Colombia o México, es la Fiscalía General de Estado quien cuenta con servicios de Atención al Ciudadano que tiene entre sus funciones tramitar las quejas formuladas por los ciudadanos por correo electrónico, haciéndolas llegar en su caso al órgano competente para resolverlas.

En Cuba no sucede así en cuanto a muchos aspectos, las quejas no se reciben por correo electrónico ni por vía telefónica, sino que se envían por correo postal o se presentan personalmente en la FGR o cualquier otra fiscalía del país. No existe un defensor del pueblo encargado de estos asuntos, todo el proceso de tramitación de la queja es realizado por el fiscal u órgano según competa, para esto se cuenta con un término de sesenta días desde el momento en punto en que fue recibida. La queja es estudiada y tramitada por el fiscal que corresponda según la materia que trate la misma, obteniéndose como resultado el pronunciamiento redactado por el fiscal que constituirá la respuesta del promovente, el cual tiene derecho a impugnar en caso de no estar de acuerdo con la misma.

#### 1.4.1. Algunas soluciones informáticas existentes relacionadas con las quejas.

El desarrollo por igual de las nuevas tecnologías de la información y las comunicaciones ha permitido transmitir, gestionar y compartir muchos datos, pero el exceso de información hace que se tenga que invertir mucho tiempo en ello, de esta manera surge la necesidad de gestionar la información, que no es más que el proceso de analizar, utilizar, recuperar y almacenar la



## Capítulo 1. Fundamentación Teórica

---

información que se ha obtenido y registrado, para permitir el aprovechamiento de la misma de una manera más eficiente, rápida y organizada.

Debido a esto se hace necesario realizar un estudio de diferentes soluciones informáticas que tratan las quejas, peticiones o denuncias de la población.

### La Rama Judicial de Puerto Rico

Sitio web a través del cual la Rama Judicial de Puerto Rico presta un servicio a todas las personas que acudan a cualquier sala del Tribunal de Primera Instancia o al Tribunal de Apelaciones de Puerto Rico, o a cualquier otra unidad del Poder Judicial, y no sea atendida por los jueces y las juezas, o por el personal del sistema judicial con el fin de informar y orientarlas en cuanto al modo de presentar una queja contra esos funcionarios. De la misma forma, un ciudadano(a) también puede presentar una queja en contra de cualquier abogado, abogada o fiscal. Este sitio web muestra solamente el procedimiento a seguir para presentar la queja, por lo que no refleja completamente el proceso del CLEP, que está destinado hacia el control de la legalidad en los Establecimientos Penitenciarios. (5)

### ComplaintsBoard

Sitio web de Estados Unidos que presta servicios para la presentación de quejas de un ciudadano contra una empresa determinada de cualquier parte del mundo. Da la posibilidad de elegir la categoría de la queja a modo de clasificación logrando obtener la información más organizada. El sitio permite ver las últimas quejas presentadas por lo que se evidencia que los temas más frecuentes son los relacionados con préstamos y robos monetarios. (6)

### Quejas en el funcionamiento del transporte público, bici-taxis y atención a problemas viales

Es un servicio que prestan las Delegaciones en México para informar, orientar, gestionar y canalizar ante la Secretaría de Transportes y Vialidad (SETRAVI) las quejas que presenten los ciudadanos sobre el funcionamiento del transporte público, principalmente microbuses, autobuses y taxis, así como la atención de problemas con bici-taxis y problemas viales o los referentes al reglamento de tránsito. (7)

### Sistema de Control de las Quejas de la Población

Es el sistema que se utiliza actualmente en la FGR desde el año 2005. El mismo automatiza el proceso de recepción de las quejas de la población y se encuentra diseñado solamente para el trabajo en el Dpto. de Protección de los Derechos Ciudadanos (PDC). Brinda facilidades de búsqueda a los fiscales de un asunto determinado. Este sistema no satisface todas las necesidades de los fiscales ya que sólo permite el registro y la búsqueda de quejas en materia penal, laboral, civil y familia y otras dirigidas a la fiscalía, que no contempla el área del CLEP.



# Capítulo 1. Fundamentación Teórica

---

Esta aplicación está desarrollada en el lenguaje de programación Visual Basic 6, usando como gestor de base de datos Microsoft Office Access, como generador de reportes Crystal Reports 8.5 y se generó su instalación con InstallShield Professional 7.0, ambas herramientas privativas por lo que no cumplen con la meta que se propone alcanzar la FGR y el país en general: lograr la independencia tecnológica.

Después de realizado el estudio de diferentes soluciones informáticas en la gestión del trabajo de las fiscalías en Cuba y el resto del mundo, relacionadas con las quejas, se puede plantear que ninguno de los anteriores cumple con las necesidades del sistema de tramitación de quejas peticiones o denuncias de las fiscalías cubanas, ya que dichos sistemas son puramente informativos o carecen de funcionalidades para realizar cualquier tipo de acción referente a quejas presentadas por los ciudadanos en la especialidad del CLEP, por lo que no satisfacen las necesidades del ciudadano ni las del fiscal en los trámites de las mismas.

## **1.5. Formulación de la propuesta de solución.**

### **1.5.1. Características del proyecto SIGEF II.**

Uno de los objetivos de la UCI es la informatización de la sociedad cubana; y como cumplimiento a este se lleva a cabo actualmente el Sistema de Informatización de la Gestión de las Fiscalías II (SIGEF II), que no es más que la continuidad de SIGEF fase I.

Esta nueva fase está dirigida a informatizar los procesos fiscales de todas las áreas existentes en la Fiscalía General de la República de Cuba, que no fueron informatizados en la fase I, para así, crear una solución que condicione un incremento en la calidad de la tramitación de los mismos.

La solución de software propuesta se desarrolla como aplicación web, teniendo en cuenta Linux Apache Postgres PHP (LAPP) como plataforma de desarrollo, ya que la estrategia del país es alcanzar la soberanía tecnológica. Para guiar todo el proceso de desarrollo de software, se hará uso de la guía de desarrollo basada en el Proceso Unificado de Software (RUP) aplicada en la universidad y descrita en el Programa de Mejora. (8)

SIGEF en su fase II está estructurado por varios subsistemas que se corresponden en su mayoría con las diferentes áreas de procesos por las que se compone la Fiscalía General de la República, dentro de estos subsistemas se encuentra el CLEP, que encierra el proceso Quejas, Peticiones o Denuncias el cual es el objetivo fundamental del desarrollo de este trabajo de diploma.

La informatización de este proceso dará solución a una serie de problemas existentes en las fiscalías Cubanas, además traerá consigo una serie de beneficios a distintos niveles:



## Capítulo 1. Fundamentación Teórica

---

A nivel de ciudadanos de la República de Cuba:

- ✓ Rapidez en sus tramitaciones legales.

A nivel de Fiscalía General de la República de Cuba:

- ✓ Disponer de un sistema para gestionar centralmente el proceso.
- ✓ Lograr la supervisión y control en tiempo real de los procesos.
- ✓ Contribuir a la preservación de la información por un período mínimo de 30 años.

A nivel de Fiscales:

- ✓ Disponer de un sistema de ayuda a la toma de decisiones.
- ✓ Aumentar el cumplimiento de los términos establecidos.

A nivel de Trabajadores de las fiscalías:

- ✓ Rapidez en el procesamiento de la información, así como su envío, búsqueda y recepción.

SIGEF fase II de forma general, traerá consigo beneficios significativos para el país, ya que contribuirá a que la FGR de Cuba cumpla de manera eficiente con su objetivo de controlar y preservar el estricto cumplimiento de la legalidad.

Por lo que se hace necesario entonces pensar en la solución, con el objetivo de abarcar el proceso Quejas, Peticiones o Denuncias de la población del departamento de CLEP, que no fue tratado en la primera fase, ya que realizar el análisis y el diseño de este proceso marcará la partida para una buena implementación del sistema, donde se define para su estructura la siguiente metodología, herramientas y lenguajes.

### **1.5.2. Metodología de desarrollo de software.**

Debido a que se tiene como objeto de estudio el proceso de desarrollo de software, es importante definir en qué consiste el mismo: Un proceso de desarrollo de software encierra una serie de actividades y resultados encaminados a producir un software. La metodología de desarrollo de software se encarga de elaborar estrategias; centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega. Su objetivo es elevar la calidad del software a través de un mayor control sobre el proceso. (9)

#### **Rational Unified Process (RUP).**

Desde la fase I, el proyecto SIGEF se viene guiando en la metodología RUP, principalmente por la ventaja de guiar a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo

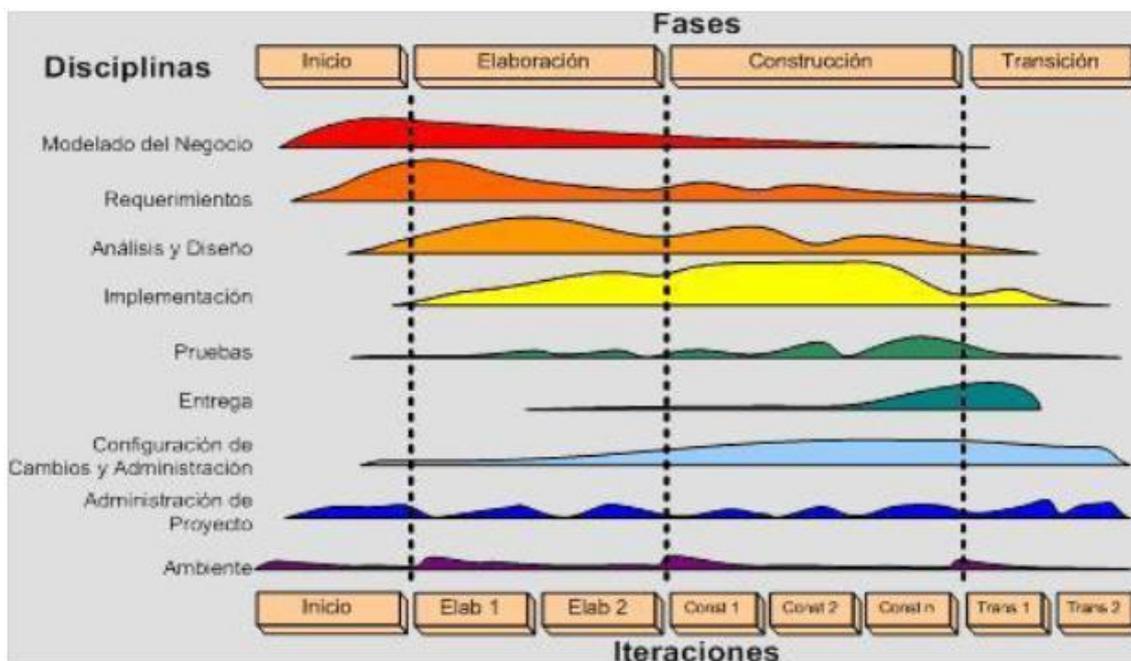


## Capítulo 1. Fundamentación Teórica

controlado, mientras se balancean los requisitos del negocio y los riesgos del proyecto. Éste de manera clara describe los diversos pasos involucrados en la captura de los requisitos, detalla qué entregables producir, cómo desarrollarlos y también provee patrones. (10)

El Proceso Unificado está dirigido por casos de uso, está centrado en la arquitectura y es iterativo e incremental, por lo que es un marco de referencia de procesos que brinda una buena documentación en todo el desarrollo del ciclo de vida del software. (11)

Divide en 4 fases el desarrollo del software: Inicio, donde se describe el negocio y se identifican los casos de uso. Elaboración, aquí se define la arquitectura del sistema y se obtiene una aplicación ejecutable. Construcción, donde se obtiene un producto listo para su utilización y Transición donde finalmente se instala el software en condiciones reales. En cada ciclo se produce una nueva versión del sistema y cada versión es un producto preparado para su entrega. El producto terminado satisface todas las necesidades de los usuarios. (11)



**Figura Nº 1 Fases y disciplinas de RUP.**

(Tomado de: *Proyecto Técnico del Sistema de Informatización de la Gestión de las Fiscalías II*)

Como parte del proceso de desarrollo de software para realizar este trabajo, será utilizado el ciclo de vida definido como parte del Programa de Mejora (PM) que define el Centro de Calidad para Soluciones Informáticas (CALISOFT) y en un modelo de desarrollo iterativo e incremental constituido por cuatro fases: Inicio, Elaboración, Construcción y Transición. Según este ciclo de vida, el proceso de desarrollo incluye las siguientes disciplinas: Estudio Preliminar, Modelado del



Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas Internas, Pruebas de Liberación, Despliegue y Soporte. (8)

Una vez realizado el estudio preliminar correspondiente a la fase de inicio incluida en este ciclo de vida de proyectos, se decide adoptar el mismo para guiar el proceso de desarrollo de software de la segunda fase del proyecto por ser éste una adaptación y por ende, muy similar a la metodología de desarrollo utilizada en la primera fase -RUP-. El uso de esta guía de trabajo, permitirá manejar correctamente proyectos a largo plazo haciendo especial énfasis en la generación de artefactos bien documentados que facilitan la capacitación y transferencia del producto.

El PM está basado en el modelo CMMI<sup>1</sup>, haciendo posible a la UCI obtener una evaluación del nivel 2 de este modelo. Además, esto permite lograr un crecimiento en cuanto a capacidad y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y Software, logrando así una mejora considerable en el ciclo de vida dentro del desarrollo de software, mayor productividad, presupuestos predecibles en los proyectos productivos y mayor calidad en los productos. (8)

### 1.5.3. Herramientas y tecnologías utilizadas.

Una herramienta es un dispositivo o procedimiento que aumenta la capacidad de llevar a cabo determinadas tareas, por ejemplo herramientas de programación, de gestión, matemáticas, entre otras. (12)

Las herramientas de desarrollo de software, llamadas herramientas CASE<sup>2</sup> son un 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. (13)

#### 1.5.3.1. Visual Paradigm 8.0

Esta herramienta principalmente en la Modelación del Negocio y en el Diseño es de gran ayuda a los analistas porque pueden visualizar el flujo central y detallado de cada proceso mediante diagramas, el Visual Paradigm es una de las herramientas CASE más amigables, ya que sus componentes se encuentran relacionados y se puede realizar la creación de diferentes diagramas; además posibilita crear un conjunto amplio de artefactos utilizados con mucha frecuencia durante las disciplinas análisis, diseño, implementación o el despliegue. Esta herramienta permite a múltiples usuarios trabajar sobre el mismo proyecto. (14)

---

<sup>1</sup> CMMI: del inglés Capability Maturity Model Integration (modelo integrado de capacidad y madurez).

<sup>2</sup> CASE: del inglés Computer Aided Software Engineering (ingeniería de software asistida por ordenador).



## Capítulo 1. Fundamentación Teórica

---

Visual Paradigm apoya plenamente la última versión del estándar BPMN<sup>3</sup> y la característica Animación, posibilita a los usuarios ver el flujo de trabajo en acción, dando así una perspectiva más amplia del funcionamiento del negocio.

Por todas las características anteriormente planteadas se escoge esta herramienta para la modelación del módulo Quejas, Peticiones o Denuncias, creando así distintos diagramas útiles para implementar de una forma sencilla y dinámica la propuesta de solución. Así mismo posee todos los recursos para integrarse tanto con el IDE<sup>4</sup> de desarrollo escogido, como con el sistema gestor de base de datos a utilizar.

### 1.5.3.2. Axure Pro 5.5

El sistema desarrollado es una aplicación web, por lo que es necesario una herramienta para crear prototipos y especificaciones muy precisas para páginas web. En cuanto a esto el Axure RP permite componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad, usando anotaciones permite especificar el estado de cada elemento (Propuesto, Aceptado, Incorporado), el beneficio esperado (Crítico, Importante, Útil), el riesgo, la estabilidad, a quién va dirigido y a quién se le asignará la tarea. (15)

Permite la realización de prototipos interactivos, es decir, se pueden simular muchas de las acciones y navegación que el usuario puede realizar en el nuevo sitio web lo que les dará una visión de cómo el sistema debe funcionar. (19)

Teniendo en cuenta las ventajas que brinda Axure se decide utilizar como herramienta para modelar los prototipos de Interfaz de Usuario para el desarrollo del módulo propuesto.

### 1.5.3.3. Lenguajes de Modelado

El uso de un lenguaje de modelado permite hacer representaciones mediante diagramas. Para el desarrollo de las disciplinas se utilizarán los siguientes lenguajes de modelado.

#### 1.5.3.3.1. Business Process Modeling Notation (BPMN)

Para realizar el modelado de negocio se utilizó BPMN como notación que logra ajustarse a las necesidades que presenta el proyecto, para proveer un modelado de procesos que resulte fácilmente entendible para todos los usuarios del negocio, desde los analistas hasta los desarrolladores técnicos, y no por último serán menos importantes los clientes que estarán manejando el proceso de negocio. BPMN define un diagrama de procesos de negocio, basado en

---

<sup>3</sup> BPMN: del inglés Business Process Management Notation (notación para el modelado de procesos de negocio).

<sup>4</sup> IDE: del inglés Integrated Development Environment (entorno integrado de desarrollo)



una técnica de diagramas de flujo que será utilizado para modelar los procesos de negocio. Esta notación de modelado tiene actualmente una alta aceptación en proyectos de software, que cada día crece su utilización. (15)

Se seleccionó el lenguaje de modelado BPMN para modelar los procesos del modulo Quejas, Peticiones o Denuncias por lo anteriormente expuesto y por su integración con Visual Paradigm.

### 1.5.3.3.2. Unified Model Language (UML)

Como lenguaje para el modelado se utilizó UML ya que permite organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan con él. Este lenguaje está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se le conoce como modelo. Un modelo UML describe lo que supuestamente hará el sistema, pero no dice cómo implementarlo. (16)

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar cada una de las partes que comprende el desarrollo de software. Posee formas de modelar conceptos como son los procesos de negocio y funciones de sistema, además de aspectos concretos como lo son escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. (17)

Este lenguaje es usado para la elaboración de los diagramas correspondientes a las disciplinas de diseño e implementación por los elementos antes mencionados.

### 1.5.3.4. Lenguaje de Programación

#### 1.5.3.4.1. Lenguaje del lado del servidor: PHP 5.3

PHP<sup>5</sup> es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica. (18)

Ventajas de utilizar PHP para el desarrollo de páginas Web:

- ✓ Es multiplataforma y libre.
- ✓ Permite las técnicas de programación orientada a objetos.
- ✓ Mejoras de rendimiento.
- ✓ Manejo de excepciones.

---

<sup>5</sup> PHP: del inglés Hypertext Pre-processor.



Se selecciona este lenguaje por las ventajas antes mencionadas y además porque el marco de trabajo seleccionado tiene como requisito trabajar con la versión 5.3 de este lenguaje.

### 1.5.3.4.2. Lenguajes del lado del cliente

Un lenguaje del lado del cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Es necesario tener en cuenta que en dependencia del navegador algunas páginas no se verán bien si el ordenador cliente no tiene instalados los plugin<sup>6</sup> adecuados.

#### Lenguaje JavaScript

JavaScript es un lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML<sup>7</sup>.

Sus características más importantes son:

- ✓ JavaScript es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente.
- ✓ JavaScript es un lenguaje orientado a eventos. Cuando un usuario pincha sobre un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante JavaScript se pueden desarrollar scripts que ejecuten acciones en respuesta a estos eventos.

#### Lenguaje HTML 5

HTML en su versión 5, no sólo trata de incorporar nuevas etiquetas o eliminar otras, sino que supone mejoras en áreas que hasta ahora quedaban fuera del lenguaje y para las que se necesitaba utilizar otras tecnologías.

Los cambios comienzan añadiendo semántica y accesibilidad implícitas, especificando cada detalle y borrando cualquier ambigüedad. Se tiene en cuenta el dinamismo de muchos sitios webs, donde su aspecto y funcionalidad son más semejantes a aplicaciones webs que a documentos. También cuenta con nuevas APIs<sup>8</sup> para interfaz de usuario.

### 1.5.3.5. TortoiseSVN

En el proyecto se trabaja con mucha documentación digital importante por lo que es de gran necesidad trabajar en base a repositorios de ficheros, una herramienta que permita tenerlos

---

<sup>6</sup> Plugin: Aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica y es ejecutada por la aplicación principal.

<sup>7</sup> HTML: del inglés HyperText Markup Language (lenguaje marcado de hipertexto).

<sup>8</sup> API: del inglés Application Programming Interface (interfaz de programación de aplicaciones).



controlados, que permita gestionar la seguridad y preservación de esta; por esto se escoge la herramienta TortoiseSVN el cual es un cliente Subversion y es implementado como una extensión al shell de Windows, además es software libre liberado bajo la licencia GNU GPL.

Entre otras características que brinda por el cual fue seleccionado se encuentran

- Que no existan copias no controladas.
- Que se sepa el histórico de cambios.
- Que se puedan comparar con facilidad.
- Que se puedan recuperar versiones antiguas con facilidad.

Para que esto sea viable, hace falta que alguien monte el repositorio, establezca la estructura, asigne los permisos y proporcione herramientas sencillas para interactuar con el repositorio.

### 1.5.3.5.1. Subversion

Para el control de versiones según las necesidades del proyecto se usa Subversion, es un software libre y se le conoce también como svn por ser el nombre de la herramienta utilizada en la línea de comando.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer porque la calidad del mismo vaya a verse afectada, si se ha hecho un cambio incorrecto a los datos, simplemente deshace ese cambio.

Entre otras ventajas por las que se selecciona esta herramienta están:

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.

### 1.5.4. Marco de Trabajo

#### Symfony 2.0.11

Symfony es un framework PHP basado en la arquitectura MVC (Model-View-Controller). Fue escrito desde un origen para ser utilizado sobre la versión 5 de PHP ya que hace ampliamente uso de la orientación a objetos que caracteriza a esta versión y desde la versión 2.0.11 de Symfony se necesita mínimamente PHP 5.3.



## Capítulo 1. Fundamentación Teórica

---

Symfony es un marco de trabajo que se utiliza para crear aplicaciones PHP, permitiendo aumentar la productividad y calidad de trabajo. Ha sido diseñado para optimizar el desarrollo de las aplicaciones web.

Symfony reutiliza conceptos y desarrollos exitosos de terceros y los integra como librerías para luego ser utilizados. Entre ellos se encuentra uno de los frameworks ORM<sup>9</sup> más importantes dentro de los existentes para PHP llamado Doctrine, el cual es el encargado de la comunicación con la base de datos.

Tiene variadas características como por ejemplo: separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. (19). Este framework ha alcanzado su versión 2.0.11, cuyo objetivo es rapidez y flexibilidad. Entre otras características que se conocen se encuentran:

- ✓ Versátil porque está basado en componentes
- ✓ Útil porque soluciona los retos de la programación web.
- ✓ Buenas prácticas porque coge ideas de otros framework.
- ✓ Flexible pues cada programador puede usar lo que desee.
- ✓ Rendimiento pues exige usar como mínimo PHP 5.3, diferentes archivos de configuración que al final se ejecutan en PHP.
- ✓ Amplia documentación
- ✓ Desaparecen los plugins y es reemplazado por un elemento llamado bundle.

Entre otras ventajas se destaca sobre todo la independencia de módulos que han creado. El propio núcleo del framework está dividido en módulos que tienen una alta cohesión, lo que permite la reutilización de los mismos fuera de un proyecto basado en Symfony 2.

Además, el gran esfuerzo que han hecho en la seguridad del propio framework – utilizando técnicas para evitar inyección SQL<sup>10</sup>– así como la integración de PHP-Unit<sup>11</sup> para la realización de pruebas unitarias. Es un marco de trabajo muy documentado y se publica bajo licencia MIT 22, con la que se puede desarrollar aplicaciones web comerciales, gratuitas y/o de software libre.

### 1.5.5. Entorno de desarrollo.

#### NetBeans 7.0

El IDE NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extender el IDE NetBeans. Es un producto libre y gratuito sin restricciones de uso. (23)

---

<sup>9</sup> ORM: del inglés Object Relational Mapping (Mapeo relacional de objetos).

<sup>10</sup> SQL: del inglés structured query language (lenguaje estructurado de consulta)

<sup>11</sup> PHP-Unit: entorno para realizar pruebas unitarias en el lenguaje de programación PHP.



Entre las novedades principales de esta versión destaca mejoras en el soporte para la nueva sintaxis de Java 7<sup>12</sup>, soporte de HTML 5, y soporte para PHP 5.3. Otras características que presenta son:

- Tienen soporte para las Base de datos de Oracle.
- El equipo de desarrollo del soporte para PHP de NetBeans 7.0 incluye una interesante característica que permite definir tipos de variables en lo comentarios. Esto es útil, por ejemplo, en aquellas situaciones en los que el código de terminación no reconoce el tipo de objeto que se está tratando.
- Tiene una sección de edición de HTML 5.
- Es multiplataforma.

### 1.5.6. Gestor de base de datos.

#### PostgreSQL 9.1

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional basado en el proyecto POSTGRES, de la universidad de Berkeley. Es una derivación libre (Open Source) Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

"PostgreSQL 9.1 provee algunas de las más avanzadas características empresariales de cualquier base de datos de código abierto, y es soportado por una entusiasta e innovativa comunidad con probados casos de éxito. PostgreSQL está muy bien preparada para construir y correr aplicaciones en la nube." palabras de Charles Fan, Sr. VP R&D, VMware.

Ofrece muchas características retirando obstáculos para el despliegue de aplicaciones nuevas o migradas en PostgreSQL. Estas incluyen:

- Replicación Sincrónica: permitiendo alta disponibilidad con consistencia sobre múltiples servidores.
- Regionalización por columna: soportando correctamente el ordenamiento por lenguaje en las bases de datos, tablas o columnas.
- Tablas unlogged: importante incremento del rendimiento para datos efímeros.

---

<sup>12</sup> Java: lenguaje de programación de alto nivel orientado a objetos.



### 1.5.6.1. ORM Doctrine 2.

El marco de trabajo Symfony 2 utiliza Doctrine como mapeador de objetos-relacional escrito en PHP que proporciona una capa de persistencia para objetos PHP. Es una capa de abstracción que se sitúa justo encima de un SGBD<sup>13</sup>.

Necesita un bajo nivel de configuración para empezar un proyecto. Puede generar clases a partir de una base de datos existente y después se puede especificar relaciones y añadir funcionalidad extra a las clases autogeneradas. (20)

Una característica importante de Doctrine es la posibilidad de escribir consultas de base de datos utilizando un dialecto de SQL denominado Doctrine Query Language (DQL) que está inspirado en Hibernate<sup>14</sup>. Además de:

- Soporte para datos jerárquicos;
- Soporte para hooks (métodos que pueden validar o modificar las escrituras y lecturas de la base de datos) y eventos para manejar la lógica de negocio relacionada;
- Herencia;
- Transacciones ACID;
- Diversos comportamientos del modelo (conjuntos anidados, internacionalización, log, índice de búsqueda);
- Una función "compilar" que combina varios archivos PHP del framework en uno solo para evitar el descenso de rendimiento que provoca incluir varios archivos PHP.

### 1.5.7. Conceptualización de las técnicas utilizadas.

Las técnicas que a continuación se muestran fueron definidas una vez realizado el estudio preliminar correspondiente a la fase de inicio del ciclo de vida mencionado.

#### Técnicas para el Modelado del negocio

Observación: es utilizada para capturar visualmente y registrar por escrito el desenvolvimiento de las actividades habituales de los clientes del negocio. (21)

Entrevista: es una técnica que requiere seleccionar bien a los entrevistados para obtener la mayor cantidad de información importante en un período de tiempo limitado. Es muy aceptada y permite acercarse al problema de una manera natural. Abarca cuatro pasos: identificación de los

---

<sup>13</sup> SGBD: Sistema Gestor de Base de Datos.

<sup>14</sup> Hibernate: entorno de trabajo que tiene como objetivo facilitar la persistencia de objetos Java en bases de datos relacionales y al mismo tiempo la consulta de estas bases de datos para obtener objetos.



entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados.

Tormentas de ideas: es una técnica cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. Puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas. Suele estar formada por un número de cuatro a diez participantes, uno de los cuales es el jefe de la sesión, encargado más de comenzar la sesión que de controlarla.

En la disciplina de Requisitos se aplicarán también la combinación de las técnicas anteriores con el objetivo de hacer una correcta captura de requisitos que permita incluir en el sistema lo que realmente será de utilidad para el cliente. Siendo la tormenta de ideas muy eficaz al comienzo del proceso cuando aún los requisitos resultaban difusos.

### 1.5.8. Patrones

En términos generales, un patrón se define como la descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución. (22)

Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada varias veces sin hacerlo de la misma forma. (23)

#### Patrón de Arquitectura

Un patrón de arquitectura de software es un esquema genérico probado para solucionar un problema particular recurrente que surge en un cierto contexto. Este esquema se especifica describiendo los componentes, con sus responsabilidades, relaciones, y las formas en que colaboran. (22)

#### Patrón Modelo-Vista-Controlador (MVC).

Patrón MVC (Modelo-Vista-Controlador) se encarga de separar interfaces, fue creado con Smalltalk-80<sup>15</sup>. El modelo es la capa del dominio, la vista es la capa de presentación y el controlador son los objetos de flujo de trabajo. (22)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Teniendo en cuenta que es una aplicación web la que se desarrolla, se ve el uso de este patrón de llamada y retorno MVC (según

---

<sup>15</sup> Smalltalk-80: Lenguaje de programación reflexivo orientado a objetos, de tipo dinámico.



# Capítulo 1. Fundamentación Teórica

CMU<sup>16</sup>), donde el modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos, la vista transforma esta información obtenida por el modelo en las páginas web a las que acceden los usuarios y el controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista.

## Patrones para Modelación del Negocio.

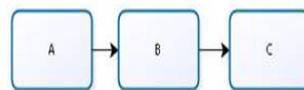
Para desarrollar la disciplina de Modelación de Negocio utilizando BPMN como lenguaje de modelado se hizo uso de determinados patrones estandarizados.

Los patrones a utilizar se encuentran agrupados de la siguiente manera:

### Patrones Básicos de Control de Flujo

#### ✓ Secuencia

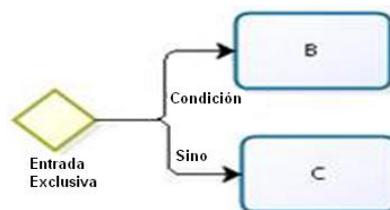
Un diagrama de procesos de negocio ilustra este patrón como una serie de actividades conectadas por flujos de secuencia. Se requiere cuando hay una dependencia entre dos actividades, de tal forma que una actividad no pueda iniciarse antes de que otra haya terminado. (22)



**Figura Nº 2 Patrón Secuencia.**

#### ✓ Selección Exclusiva

Ocurre cuando en un punto del flujo de trabajo se escoge sólo una de varias ramas del proceso, generalmente esta decisión se toma basándose en datos de control del flujo de proceso. (22)



**Figura Nº 3 Patrón Selección Exclusiva.**

### Patrones de Sincronización y Enrutamiento Avanzada

#### ✓ Mezcla Múltiple o Serialización:

<sup>16</sup> CMU: Universidad Carnegie Mellon (en inglés: Carnegie Mellon University, CMU), ubicada en la ciudad de Pittsburgh (Pensilvania) y es uno de los más destacados centros de investigación superior de los Estados Unidos en el área de informática y robótica.



## Capítulo 1. Fundamentación Teórica

La Mezcla Múltiple es un patrón del flujo de trabajo en donde dos o más ramas se unen sin sincronización. Si se activa más de una rama, posiblemente de forma concurrente la actividad que sigue a la serialización se ejecuta una vez por cada rama que ha sido activada. (22)

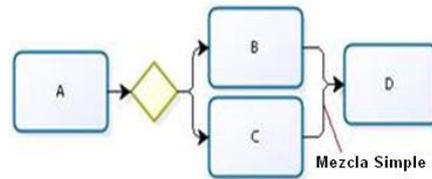


Figura Nº 4 Patrón Mezcla Múltiple o Serialización.

### Patrones de casos de uso

#### CRUD (Creating, Reading, Updating, Deleting)

Es un patrón de estructura que se basa en la fusión de casos de uso simples para formar una unidad conceptual. Este tipo de patrón es denominado Información de CRUD o Administrar Información y es utilizado para modelar distintas operaciones: crear, leer, actualizar y eliminar. Este patrón debe usarse cuando todos los flujos contribuyen al mismo valor del negocio y estos son cortos y simples. (23)

**CRUD Parcial:** patrón alternativo que modela una de las alternativas del caso del uso como un caso de uso separado. Es preferible cuando una de las alternativas del caso del uso es más significativa, grande, o mucho más compleja que las otras alternativas. (23)

**Múltiples actores:** captura la concordancia entre actores manteniendo roles separados.

- ✓ Roles comunes: puede suceder que los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso. (23)

Este patrón brinda la posibilidad de realizar el modelado del sistema de una manera más eficiente. Los patrones de casos de uso juegan un papel importante ya que permiten ahorrar tiempo, coste y se hace más fácil el trabajo.

### Patrones de Diseño.

Para desarrollar la disciplina de Diseño se tuvieron en cuenta determinados patrones los cuales describen formas de solucionar problemas que ocurren de forma frecuente.

Los patrones de diseño constituyen una forma de reutilizar el conocimiento de los expertos en el desarrollo de software. Cada patrón de diseño es una solución a un problema de diseño. (22)



Patrones **GRASP** (Patrones Generales de Software para Asignación de Responsabilidades).

Los patrones GRASP describen un conjunto de principios fundamentales para la asignación de responsabilidades. Dentro de este grupo se encuentran los siguientes patrones: Experto, Creador, Bajo acoplamiento, Alta cohesión y Controlador. (24)

Patrones **GoF** (Gand of Four).

Según su propósito los patrones GOF se clasifican en: creacionales, estructurales y de comportamiento. (25)

El marco de trabajo Symfony 2 trae estos patrones implícitamente, dándole solución a una serie de problemas recurrentes que se presentan, el mismo implementa numerosas funcionalidades que el cliente debe conocer, para así hacer uso de estos, configurándolo según sus necesidades.

## 1.5.9. Métricas

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo del software y los proyectos de mantenimiento. (26)

### Métrica para Requisitos.

En la disciplina Requisitos las métricas brindan información sobre la comprensibilidad y validez de los mismos. (27). Entre estas métricas se encuentra:

#### Estabilidad de los requisitos.

Es necesario lograr una estabilidad en los requisitos para el correcto funcionamiento de los demás flujos de trabajo. El objetivo de esta métrica es medir la estabilidad de los requisitos para asegurar su adecuación antes de pasar al próximo flujo de trabajo. Se considera que los requisitos son estables cuando no existen adiciones o supresiones en ellos que impliquen modificaciones en las funcionalidades principales de la aplicación. (27)

La estabilidad de los requisitos se calcula como:  $ETR = [(RT - RM) / RT] * 100$

Donde:

- ✓ ETR: valor de la estabilidad de los requisitos.
- ✓ RT: total de requisitos definidos.
- ✓ RM: número de requisitos modificados, que se obtienen como la sumatoria de los requisitos insertados, modificados y eliminados.



## Capítulo 1. Fundamentación Teórica

---

Esta métrica ofrece valores entre 0 y 100. El mejor valor de ETR es el más cercano a 100 porque mostrará que no se están realizando cambios sobre los requisitos, son estables y, por tanto, es confiable trabajar el análisis y diseño sobre ellos. (28)

### **Métricas para la Calidad de los Casos de Uso (CU).**

Consiste en la aplicación de preguntas agrupadas bajo categorías relativas a características de calidad, con el fin de comprobar defectos que puedan estar contenidos en los Casos de Uso. (29).

Esta métrica está compuesta por 4 atributos:

*Compleitud:* Grado en que se ha logrado detallar todos los casos de uso relevantes.

*Consistencia:* Grado en que los casos de uso del sistema describen las interacciones adecuadas entre el usuario y el sistema.

*Correctitud:* Grado en que las interacciones actor/sistema soportan adecuadamente el proceso de negocio.

*Complejidad:* Grado de claridad en la presentación de los elementos que describen el contexto y la claridad del sistema.

### **Métricas para diseño.**

Para medir el diseño se utilizaron métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objeto referenciadas por Pressman teniendo en cuenta que este estudio brinda un esquema sencillo de implementación y que a la vez cubre los principales atributos de calidad de software. Las métricas escogidas para la validación del diseño fueron la Métrica orientada a clases: Tamaño Operacional de Clase (TOC) y Acoplamiento entre Clases (AC). (28)

#### Tamaño Operacional de Clase (TOC).

El tamaño general de una clase se puede determinar siguiendo los planteamientos descritos a continuación:

- ✓ El número de atributos (tanto atributos heredados como atributos privados de la instancia) que están encapsulados en la clase.
- ✓ El número total de operaciones (tanto operaciones heredadas como operaciones privadas de la instancia) que están encapsuladas dentro de la clase.

Estos dos valores son sumados de acuerdo a la clase que se analiza y el resultado es tomado como umbrales que luego son comparados en la tabla que se presenta a continuación para determinar el TOC de cada clase.

**Tabla: 1 Clasificación de las Clases.**



Clasificación	Valores de los umbrales
Pequeño	$\leq 20$
Medio	$20 < \leq 30$
Grande	$> 30$

Si existen valores grandes de TOC éstos estarán demostrando que una clase puede tener demasiada responsabilidad, lo cual reduciría la reutilización de la clase y hará complicada la implementación y la prueba. De forma contraria sucede si los valores TOC son de menor valor. Finalmente se calcula los promedios correspondientes a los diferentes valores para tener una estimación general del sistema. (28)

### Acoplamiento entre clases (AC)

CBO<sup>17</sup> de una clase es el número de clases a las cuales una clase está ligada. Existe dependencia entre dos clases cuando una clase usa métodos o variables de la otra clase. Las clases relacionadas por herencia no se tienen en cuenta. (30)

*Propósito:* Chidamber y Kemerer proponen que sea un indicador del esfuerzo necesario para el mantenimiento y en el testeo.

*Guías y comentarios:* Chidamber y Kemerer sugieren que cuanto más acoplamiento se da en una clase, más difícil será reutilizarla. Además, las clases con excesivo acoplamiento dificultan la comprensibilidad y hacen más difícil el mantenimiento por lo que será necesario un mayor esfuerzo y riguroso testeo.

Las clases deberían de ser lo más independientes posible, y aunque siempre es necesaria una dependencia entre clases, cuando ésta es grande, la reutilización puede ser más cara que la reescritura. Al reducir el acoplamiento se reduce la complejidad, se mejora la modularidad y se promueve la encapsulación. (30)

## 1.6. Conclusiones Parciales.

En este capítulo se realizó un estudio del Proceso Quejas, Peticiones o Denuncias en Cuba y de otras soluciones informáticas que se han implantado en otros países, se analizaron las herramientas y lenguajes de modelado que se utilizarán para llevar adelante el módulo Quejas, Peticiones o Denuncias del SIGEF II, teniendo en cuenta además los patrones y métricas necesarios para hacerlo con eficiencia y calidad.

<sup>17</sup> CBO: del inglés Coupling Between Objects (Acoplamiento entre objetos).



## *Capítulo 1. Fundamentación Teórica*

---

Después de tener todos estos elementos se arriba a las siguientes conclusiones:

- El análisis de las soluciones informáticas desarrolladas en otros países permitió concluir que por sus características no son adaptables a las necesidades que presenta la FGR, por lo que es necesario desarrollar una aplicación para ello.
- Las herramientas, tecnologías y lenguajes que serán utilizados facilitarán el trabajo a realizar en cada una de las disciplinas que serán desarrolladas.
- El uso de patrones para el Modelado del Negocio, los Requisitos y el Diseño estudiados anteriormente permitirán obtener una solución más eficiente.
- Las métricas que serán usadas garantizarán la calidad de los resultados que se obtengan en cada una de las disciplinas a desarrollar.



### Capítulo 2. Análisis y Diseño.

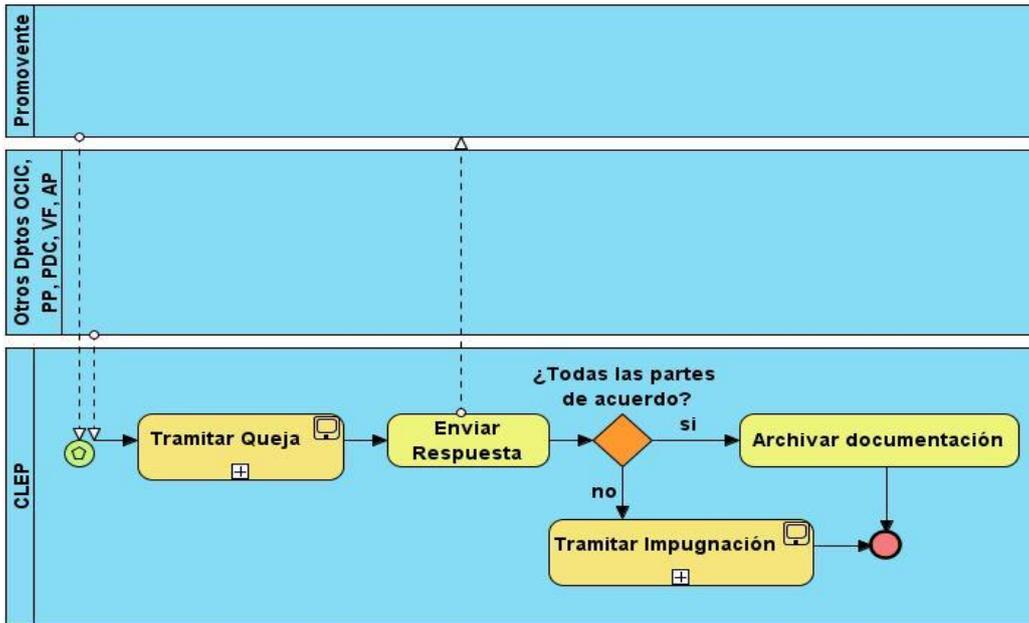
#### 2.1. Introducción.

En este capítulo se realiza el Modelado del Negocio, modelando y describiendo los procesos de negocio mediante los Diagramas de Procesos de Negocio (DPN), donde se aplican métricas y métodos para validar los resultados obtenidos. Se definen los requisitos tanto funcionales y no funcionales, teniendo en cuenta las restricciones o políticas a cumplir por el sistema y su validación aplicando técnicas y métricas para medir la calidad de los requisitos identificados. Además se presenta el diseño a través de los diagramas de clases, una serie de vistas que formarán parte de la interfaz de usuario correspondiente a cada proceso logrando una detallada descripción de la arquitectura.

#### 2.2. Modelado del Negocio

Seguidamente se procede a describir los principales procesos de negocio referentes a Quejas, Peticiones o Denuncias que se llevan a cabo en el departamento CLEP de la FGR. De esta manera se pretende lograr un entendimiento común del negocio por parte de los clientes y los analistas una vez obtenido el modelado con BPMN. Se mostrará un caso muy puntual para ejemplificar el modelado, el resto de los diagramas y sus descripciones se encuentran en el expediente del proyecto. El diagrama que se muestra a continuación representa la tramitación de las quejas de forma general. De aquí se derivan una serie de subprocesos en los que se modelan las especificidades, estos son:

- ✓ Quejas, Peticiones o Denuncias. (*Figura N° 5*).
- ✓ Tramitar Queja, Petición o Denuncia. (*Figura N° 6*).
- ✓ Trasladar Queja, Petición o Denuncia. (*Ver Anexo 1*)
- ✓ Concluir Queja, Petición o Denuncia. (*Ver Anexo 2*).
- ✓ Tramitar Impugnación. (*Ver Anexo 3*).



**Figura Nº 5 Diagrama de Procesos de Negocio de las Quejas, Peticiones o Denuncias.**

En el diagrama mostrado anteriormente se modela cómo ocurre de manera general la tramitación de las Quejas, Peticiones o Denuncias en la FGR. Una vez que llega una queja, petición o denuncia al departamento de CLEP, este se encarga de tramitarla (*Ver descripción del subproceso Tramitar Queja, Petición o Denuncia*), en un término de 60 días hábiles. Luego de concluida la tramitación se procede al envío de respuesta al promovente donde este, si no está conforme con la respuesta dada, impugnará la misma, (*Ver la descripción del subproceso Tramitar Impugnación*), disponiendo de un término de 10 días hábiles a partir de la tramitación para que su queja sea revisada o tramitada nuevamente según corresponda. De lo contrario se realiza la actividad Archivar documentación de la tramitación, donde se archivará toda la documentación referente a la tramitación de las quejas, peticiones o denuncias.

### 2.2.1. Resultado de la aplicación de las técnicas de recopilación de información.

La realización satisfactoria del negocio permite disminuir costos y tardanzas en las entregas de los productos, mejorar la calidad de software y maximizar el agrado de los clientes. En el capítulo 1 se explicó en qué consisten cada una de las técnicas que a continuación se muestran, por lo que en este momento se describen solamente los resultados luego de la aplicación de cada una de ellas:

Observación: Se practicó esta técnica en cada uno de los encuentros con los fiscales para tener constancia de las acciones realizadas en el proceso en cuestión. Fue muy útil para entender el proceso de negocio ya que el analista pudo valorar cómo ocurría este en tiempo real, en este caso, viendo la forma de tramitación de las quejas en la propia FGR.



Tormentas de ideas: Se realizaron talleres con especialistas de la FGR y analistas del proyecto para obtener más información sobre el proceso en cuestión.

Análisis de documentos: Se realizó un estudio de la Metodología de trabajo del departamento del CLEP relacionada con este proceso para lograr un correcto modelado del negocio, para determinar las políticas y condiciones que se deben cumplir para la gestión y control del proceso quejas, peticiones o denuncias.

Entrevistas: Se realizaron 11 entrevistas con los fiscales implicados, mediante una serie de preguntas, con el objetivo de obtener la mayor cantidad posible de información sobre el funcionamiento del negocio. Los resultados de cada una de las entrevistas realizadas quedaron documentados mediante las minutas de reunión, los que fueron usados posteriormente en la documentación referente al proceso.

Una vez aplicada cada una de las técnicas descritas anteriormente se logró identificar que el proceso está compuesto por 4 subprocesos y 55 actividades correspondientes a los mismos, así como los eventos que los generan, las precondiciones, el marco legal en que se desarrollan, y los clientes internos y externos que intervienen. De esta manera se obtuvo finalmente el modelado del negocio del módulo QPD donde se adquirió la información necesaria de todo el ciclo de vida del proceso, viéndose reflejado en los próximos epígrafes.

### **2.2.2. Descripción de subprocesos de negocio.**

La descripción de los procesos de negocio permite detallar en qué consiste cada una de las actividades que se realizan en el proceso. A continuación se muestra el DPN del subproceso Tramitar Queja, Petición o Denuncia y su descripción. Las demás descripciones se encuentran en el expediente del proyecto.

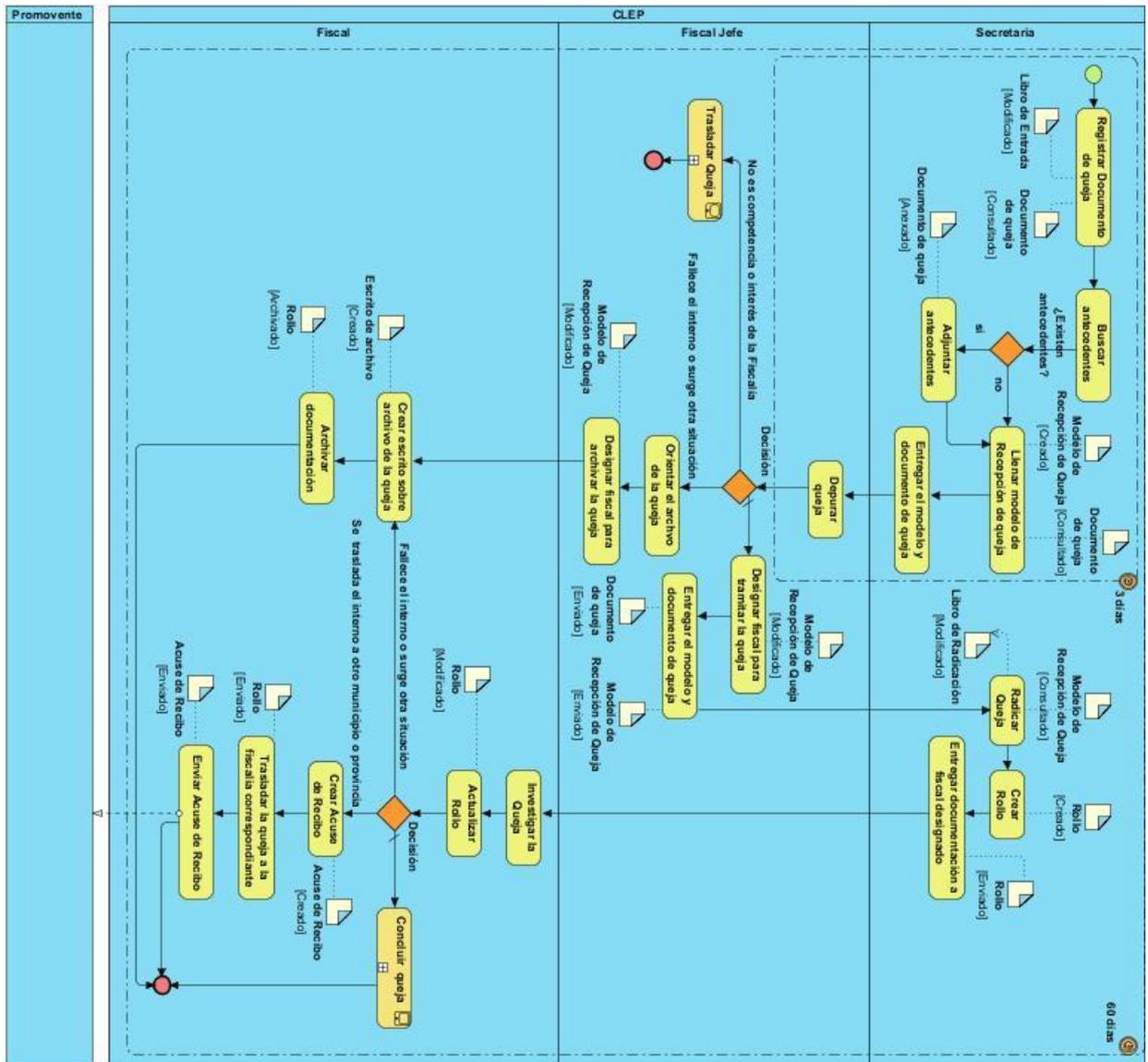


Figura N° 6 Diagrama de Procesos de Negocio del subproceso Tramitar Queja.



Tabla: 2 Descripción de procesos de negocio del sub-proceso Tramitar Queja.

<b>Objetivo</b>	Tramitar e investigar quejas, peticiones y denuncias de la población ante presuntas violaciones de sus derechos.
<b>Evento(s) que lo genera(n)</b>	Llega una queja, petición o denuncia al área de CLEP.
<b>Pre condiciones</b>	Se ha recibido una queja, petición o denuncia relacionada con un interno, detenido o sus familiares.
<b>Marco legal</b>	Constitución de la República. Ley 83/97. Reglamento de la LFGR. Otras.
<b>Clientes internos</b>	N/A
<b>Clientes externos</b>	N/A
<b>Entradas</b>	Queja, Petición o Denuncia.
<b>Flujo de eventos</b>	
<b>Flujo básico Tramitar</b>	
1.	<b>Registrar documento de queja.</b> Al llegar el documento de la queja, petición o denuncia al Dpto. de CLEP la secretaria registra la misma en el Libro de Entrada.
2.	<b>Buscar antecedentes.</b> La secretaria busca antecedentes de la queja, petición o denuncia y el promovente en el Libro de Entrada.
3.	Si existen antecedentes de la queja, petición o denuncia o del promovente ir al flujo alterno 3.a, si no existen antecedentes de la queja, petición o denuncia o del promovente se realiza la actividad <b>Llenar modelo de Recepción de queja.</b> La secretaria llena el Modelo de Recepción de la Queja con los datos pertinentes.
4.	<b>Entregar Modelo y documento de queja.</b> La secretaria entrega el Modelo de Recepción de la Queja y la documentación de la misma al Fiscal Jefe para que la depure.
5.	<b>Depurar queja.</b> El Fiscal Jefe depura la queja, petición o denuncia, donde decide como se procederá según la información que tiene hasta el momento de la queja, petición o denuncia, si esta será tramitada, trasladada o archivada, lo cual queda registrado en el Modelo de Recepción de la Queja. El Fiscal Jefe cuenta con 3 días naturales para depurar la queja, petición o denuncia.
6.	Si la queja, petición o denuncia no es competencia o interés de la Fiscalía el Fiscal Jefe decide trasladar la queja, petición o denuncia ( <i>ver la descripción del <a href="#">subproceso Trasladar Queja</a></i> ). Si se presenta una situación de muerte del interno u otra situación excepcional, ir al flujo alterno 6.a. Si el Fiscal Jefe decide tramitar la queja, petición o denuncia, se realiza la actividad <b>Designar fiscal para tramitar la queja.</b> El Fiscal Jefe designa que fiscal será el encargado de tramitar la queja, petición o denuncia y registra esta información en el Modelo de Recepción de la Queja.
7.	<b>Entregar Modelo y documentación de la queja.</b> El Fiscal Jefe entrega a la secretaria el Modelo de Recepción de la Queja y la documentación de la misma, con la queja, petición o denuncia depurada para que continúe con el procedimiento que le compete.
8.	<b>Radicar queja.</b> La secretaria registra la queja, petición o denuncia en el Libro de Radicación.
9.	<b>Crear Rollo.</b> La secretaria crea el Rollo de la queja, petición o denuncia con la información de la documentación de la queja, los antecedentes si existen y el Modelo de Recepción de la Queja.
10.	<b>Entregar documentación a fiscal designado.</b> La secretaria le entrega toda la documentación correspondiente con la queja, petición o denuncia al fiscal designado en la depuración para que la tramite.
11.	<b>Investigar la Queja.</b> El fiscal investiga y revisa toda la documentación necesaria para



tramitar la queja, petición o denuncia

12. **Actualizar Rollo.** El fiscal incorpora al Rollo toda la documentación que se realiza durante la investigación de la queja, petición o denuncia y el registro de notas donde documenta cada una de las acciones que realizó en la investigación.
13. Si durante la investigación fallece el interno o detenido o surge otra situación excepcional ir al flujo alterno 13.a. Si el interno relacionado con la queja, petición o denuncia es trasladado a otro municipio o provincia ir al flujo alterno 13.b. En otro caso el fiscal continuara el proceso y realiza el subproceso **Concluir queja**, ver descripción del subproceso *Concluir queja*
14. Termina el proceso.

### Poscondiciones

1. Se ha enviado el Modelo de Recepción de la Queja.
2. Se ha consultado el Libro de Entrada.
3. Se ha modificado el Documento Radicación de Queja.
4. Se ha archivado el Rollo.

### Salidas

1. Modelo de Recepción de la Queja.
2. Libro de Entrada.
3. Documento Radicación de Queja.
4. Rollo.

### Flujos alternos

#### 3.a Si existen antecedentes de la queja o del promovente

1. **Adjuntar antecedentes.** La secretaria adjunta los documentos antecedentes de la queja, petición o denuncia o del promovente al documento de la queja, petición o denuncia.
2. Continúa en la actividad **Llenar modelo de Recepción de queja** del paso 3 del flujo básico.

### Poscondiciones

1. Se han adjuntado los documentos antecedentes.

### Salidas

1. Documentos antecedentes.

#### 6.a Si fallece el interno o surge otra situación excepcional

1. **Orientar el archivo de la queja.** El Fiscal Jefe orienta que debe ser archivada la queja, petición o denuncia y da las razones del archivo.
2. **Designar fiscal para ejecutar el archivo de la queja.** El Fiscal Jefe designa el fiscal que realizará el archivo de la queja, petición o denuncia.
3. **Crear Escrito sobre archivo de la queja.** El fiscal designado crea el Escrito donde se recogen las razones del archivo de la queja, petición o denuncia.
4. **Archivar Documentación.** El fiscal archiva toda la documentación que hay de la queja, petición o denuncia hasta el momento junto al escrito.

### Poscondiciones

1. Se ha archivado la documentación de la queja, petición o denuncia.

### Salidas

1. Escrito sobre archivo.

#### 13.a Si durante la investigación fallece el interno o surge otra situación excepcional

1. **Crear Escrito sobre archivo de la queja.** El fiscal designado crea el Escrito donde se recogen las razones del archivo de la queja, petición o denuncia.
2. **Archivar Documentación.** El fiscal incorpora el Escrito sobre archivo en el Rollo y procede al archivo del Rollo.

### Poscondiciones

1. Se ha creado el escrito sobre archivo.
2. Se ha archivado el Rollo.

### Salidas



1. Escrito sobre archivo.

### 13.b Si el interno relacionado con la queja es trasladado a otro municipio o provincia

1. **Crear Acuse de Recibo.** El Fiscal crea el Acuse de Recibo comunicándole al promovente en el mismo, para donde ha sido trasladada su queja, petición o denuncia.
2. **Trasladar la queja a la fiscalía correspondiente.** El Fiscal envía toda la documentación de la queja, petición o denuncia y el Rollo a la fiscalía correspondiente a donde fue trasladado el interno.) La fiscalía a donde fue trasladada la queja, petición o denuncia se encargará de continuar el proceso teniendo en cuenta los días consumidos de la tramitación.

#### Poscondiciones

1. Se ha creado el Acuse de Recibo.
2. Se ha trasladado la queja, petición o denuncia.

#### Salidas

1. Acuse de Recibo.

#### Asuntos pendientes

N/A

### 2.2.3. Modelo Conceptual.

El modelo conceptual es una herramienta muy importante donde no se define ninguna operación sino que se muestran conceptos, relaciones entre ellos y atributos de los mismos. Se representa en UML mediante un grupo de estructura estática (Diagramas de Clases Conceptuales). Se realizó el Modelo Conceptual (Ver Anexo 4) correspondiente al proceso Quejas, Peticiones o Denuncias de la población en la disciplina de Modelado del Negocio y se hizo un refinamiento del mismo en la disciplina de Especificación de Requisitos.

### 2.2.4. Patrones de Negocio.

En el primer capítulo se explicó en qué consistía teóricamente cada uno de los patrones de negocio, que a continuación se ven aplicados a lo largo del modelado del módulo QPD, dándole solución a los problemas que a continuación se presentan:

#### Patrones Básicos de Control de Flujo

##### ✓ *Secuencia*

Cuando ocurre una dependencia entre actividades, como por ejemplo en el subproceso Trasladar Queja, Petición o Denuncia (Ver Anexo 1), para enviar la documentación de la queja, petición o denuncia es necesario haberla creado antes, por lo que se aplica el patrón secuencia.

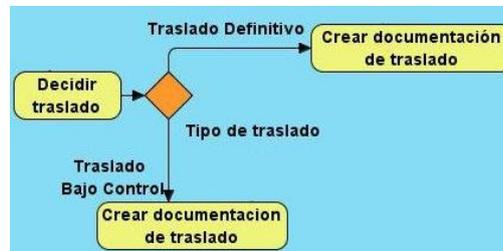


Figura Nº 7 Sesión del diagrama Tramitar Queja, Petición o Denuncia.

##### ✓ *Selección Exclusiva*



En el caso que se ejecuta sólo una rama del proceso, como por ejemplo en el subproceso Trasladar Queja, Petición o Denuncia (Ver Anexo 1), cuando se decide un tipo de traslado de la queja: Definitivo o Bajo Control se aplica el patrón Selección exclusiva como se muestra en la figura siguiente:



**Figura Nº 8 Sesión del diagrama Tramitar Queja, Petición o Denuncia.**

*Patrones de Sincronización y Enrutamiento Avanzada*

✓ *Mezcla Múltiple o Serialización*

En el caso donde se unen dos ramas del proceso, como por ejemplo en el subproceso Trasladar Queja, Petición o Denuncia (Ver Anexo 1), la rama Realizar indicación y la de Modificar o Crear respuesta, siendo cual sea la acción que se realice se envían los resultados a la Fiscalía Inmediata Superior, por lo que se aplica el patrón Mezcla múltiple.



**Figura Nº 9 Sesión del diagrama Tramitar Queja, Petición o Denuncia.**

### 2.2.5. Validación del Modelado del Negocio.

La validación de los artefactos generados en esta disciplina fue posible a través de las Revisiones Técnicas Formales (RTF) y la validación con el cliente. A continuación se muestran los detalles de estas dos formas de validación:

#### RTF

Se realizaron dos RTF al modelado de los procesos de negocio por parte de la dirección del proyecto: Analista Principal, Líder y el equipo de calidad del mismo. Mediante esto fue posible detectar 13 no conformidades en la 1ra iteración y ya la segunda resultó satisfactoria. Con esta



técnica se logra ofrecer a los clientes mayor seguridad logrando que los artefactos cuenten con la calidad requerida desde el punto de vista técnico.

### Validación con el cliente

Se realizaron varios encuentros con los fiscales que atienden las quejas, peticiones o denuncias en las diferentes materias y se analizaron posibles errores que, en cuanto a términos fiscales habían sido mal empleados. Tras una serie de cambios y modificaciones en este sentido, el modelado de los procesos de negocio fue finalmente aceptado por los clientes, obteniendo como artefacto la carta de aceptación de los mismos. (Ver Anexo 5)

### **2.2.6. Conclusiones parciales.**

Con la realización de las actividades correspondientes a esta disciplina, se logró modelar todo el negocio que encierra el proceso quejas, peticiones o denuncias. De esta forma se generaron los artefactos correspondientes: Descripción de Proceso de Negocio, Mapas de Procesos y el Modelo Conceptual los cuales servirán de base para las próximas disciplinas.

## **2.3. Requisitos.**

Esta disciplina tiene como objetivo principal lograr un punto de convergencia entre analistas y clientes de modo que queden identificadas de forma correcta las funcionalidades que debe tener el sistema. Los artefactos generados en el Modelado del Negocio fueron el punto de partida para generar los entregables correspondientes a esta disciplina, entre los que se encuentra la Especificación de Requisitos de Software. Este artefacto será validado a partir de los prototipos y las métricas correspondientes.

Para llevar a cabo esta actividad se aplicó la combinación de las técnicas definidas en el primer capítulo con el objetivo de lograr mejores resultados en este sentido.

### **2.3.1. Resultados de la aplicación de las técnicas de recopilación de información.**

Las técnicas utilizadas para la captura de requisitos en el presente trabajo son Entrevistas, Tormentas de ideas, y la Observación explicadas anteriormente en la disciplina del Negocio; además de:

Sistemas existentes: Mediante el análisis de diferentes soluciones informáticas relacionados con la gestión de las quejas, peticiones o denuncias de los ciudadanos, para así encontrar semejanzas y contradicciones, y tenerlas en cuenta para el sistema a construir. Por un lado, se puede analizar las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada, por



otro lado también es útil analizar las distintas salidas que los sistemas producen (listados, consultas, etc.), porque siempre pueden surgir nuevas ideas sobre la base de estas.

Mapas conceptuales: son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Esta es una técnica muy usada en el levantamiento de requisitos, dada la facilidad de entendimiento que provee para los usuarios. Como resultado de estas técnicas se obtuvieron 26 RNF, 21 RF los cuales agrupados suman un total de 8 casos de uso.

### Requisitos Funcionales obtenidos

RF\_CLEP\_QPD\_ 1 Adicionar Queja

RF\_CLEP\_QPD\_ 2 Actualizar Queja

RF\_CLEP\_QPD\_ 3 Buscar Antecedentes

RF\_CLEP\_QPD\_ 4 Concluir Tramitación

RF\_CLEP\_QPD\_ 5 Depurar Queja

RF\_CLEP\_QPD\_ 6 Tramitar Queja

RF\_CLEP\_QPD\_ 7 Actualizar Impugnación

RF\_CLEP\_QPD\_ 8 Visualizar Detalles

RF\_CLEP\_QPD\_ 9 Vista Previa de Documentos

RF\_CLEP\_QPD\_ 10 Adicionar Archivos Adjuntos

RF\_CLEP\_QPD\_ 11 Eliminar Archivos Adjuntos

RF\_CLEP\_QPD\_ 12 Descargar Archivos Adjuntos

RF\_CLEP\_QPD\_ 13 Buscar personas

RF\_CLEP\_QPD\_ 14 Imprimir Documentos

RF\_CLEP\_QPD\_ 15 Reportar Quejas Recibidas

RF\_CLEP\_QPD\_ 16 Reportar Quejas Tramitadas

RF\_CLEP\_QPD\_ 17 Reportar Quejas Trasladas

RF\_CLEP\_QPD\_ 18 Reportar Quejas Archivadas

RF\_CLEP\_QPD\_ 19 Reportar Quejas Impugnadas

RF\_CLEP\_QPD\_ 20 Reportar Quejas Pendientes



RF\_CLEP\_QPD\_21 Reportar Quejas por Raza y Sexo.

Los RNF son genéricos para el proyecto los cuales se encuentran en el documento 0113\_Especificacion\_de\_Requisitos\_de\_Software.doc en el expediente de proyecto.

### 2.3.2. Especificación de requisitos.

Para la especificación de requisitos se hizo uso de la técnica plantillas o patrones con el objetivo de describir los requisitos de una forma centralizada en la plantilla correspondiente definida por el PM. A continuación se cita un ejemplo muy puntual de especificación de requisitos, en este caso el requisito Buscar Antecedentes. Las demás especificaciones se encuentran en el documento Especificación de Requisitos del Software perteneciente al expediente de proyecto.

**Tabla: 3 Especificación de Requisitos para el requisito Buscar Antecedentes.**

**Prototipo**

#### Búsqueda de Antecedentes

**Criterios de búsqueda**

Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Carnet de Identidad	Provincia	Municipio	<input type="checkbox"/> Promovente
<input type="text"/>	-Seleccione-	-Seleccione-	<input type="button" value="Buscar"/>

**Resultados de la búsqueda**

*Figura 3.1 Buscar Antecedentes.*

Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"><li>Primer nombre</li></ul>	<ul style="list-style-type: none"><li>String</li></ul>	Campo que admite letras y números
<ul style="list-style-type: none"><li>Segundo nombre</li></ul>	<ul style="list-style-type: none"><li>String</li></ul>	Campo que admite letras y números
<ul style="list-style-type: none"><li>Primer apellido</li></ul>	<ul style="list-style-type: none"><li>String</li></ul>	Campo que admite letras y números
<ul style="list-style-type: none"><li>Segundo apellido</li></ul>	<ul style="list-style-type: none"><li>String</li></ul>	Campo que admite letras y números
<ul style="list-style-type: none"><li>Carnet de identidad</li></ul>	<ul style="list-style-type: none"><li>Integer</li></ul>	Campo que sólo admite números.



• Provincia	• String	Campo que admite letras y números
• Municipio	• String	Campo que admite letras y números
<b>Asuntos pendientes</b>	N/A	

Como se había mencionado anteriormente, de la agrupación de estos RF se identificaron los CU del sistema, esto constituye una alternativa para representar las funcionalidades del sistema partiendo de las acciones que realizará el usuario. A continuación se muestra el diagrama de casos de uso correspondiente al proceso Quejas, Peticiones o Denuncias:

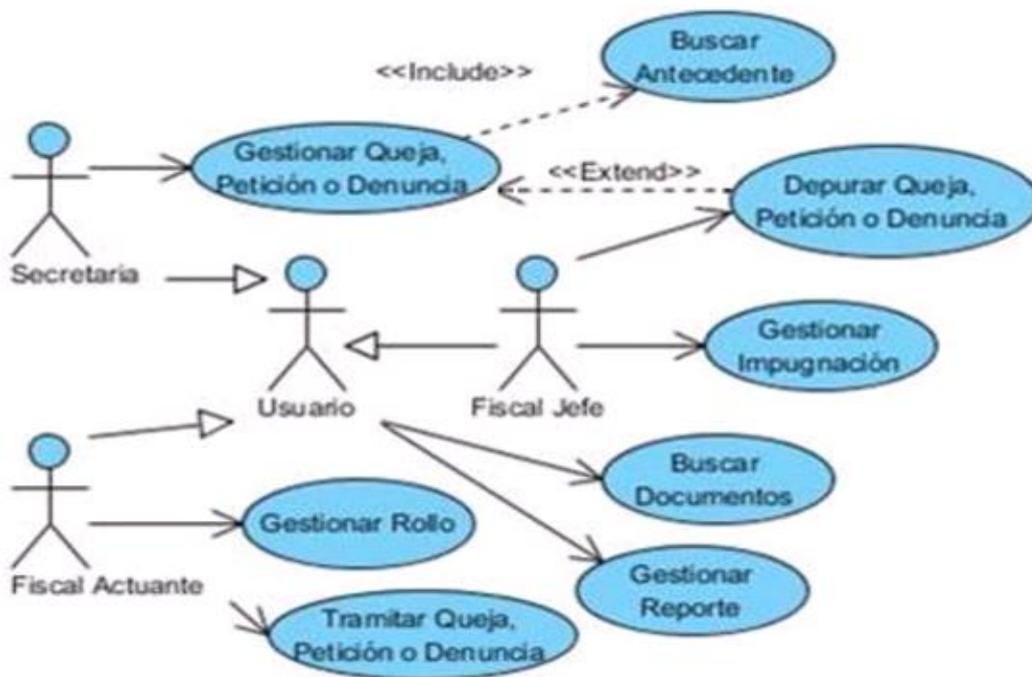


Figura Nº 10 Diagrama de Casos de Uso del Sistema.

### 2.3.3. Aplicación de los patrones de CU.

El patrón Roles Comunes y CRUD Parcial se ven evidenciados en el Diagrama de Casos de Uso del sistema como se muestra en las siguientes figuras:

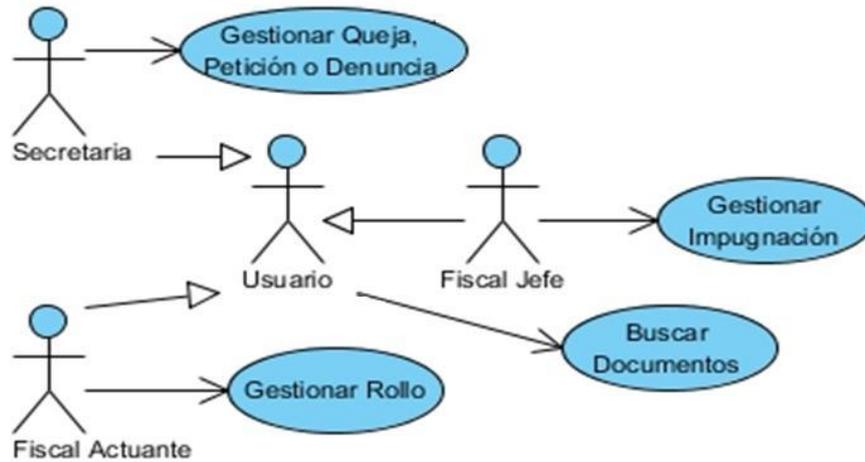


Figura Nº 11 Patrón Roles Comunes.



Figura Nº 12 Patrón CRUD Parcial.

Especificación de casos de uso

Se realiza con el objetivo de describir cada una de las acciones que realiza el sistema tras las opciones que selecciona el usuario (21). Fue utilizada la técnica plantillas o patrones para su elaboración. A continuación se muestra la especificación del CU Buscar Documentos. Las demás especificaciones se encuentran en el documento Especificación de casos de uso perteneciente al expediente de proyecto.

**Tabla: 4 Especificación de casos de uso para el CU Buscar Documentos.**

<b>Objetivo</b>	Buscar un documento que se necesite en cualquier momento con respecto a un determinado caso.	
<b>Actores</b>	Secretaria Fiscal Jefe Fiscal Actuante	
<b>Resumen</b>	Se inicia cuando se desea buscar algún documento, a partir de determinados criterios.	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Auxiliar	
<b>Precondiciones</b>	El actor ha sido autenticado en el sistema.	
<b>Poscondiciones</b>	Encontrar el documento que se desea.	
<b>Flujo de eventos</b>		
<b>Flujo básico &lt; Buscar Documentos &gt;</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción Buscar	



## Capítulo 2. Análisis y Diseño

	Documentos de la interfaz principal del módulo Quejas Peticiones o Denuncias.	
2.		Muestra la interfaz “Buscar Documentos” que contiene los criterios de búsqueda necesarios para localizar un documento. Los resultados de la búsqueda se muestran en el listado inferior de la interfaz. Permite realizar las opciones: <ul style="list-style-type: none"> <li>• Buscar.</li> <li>• Cerrar. Ver Flujo alternativo 2.a</li> </ul> Selecciona la opción Cerrar.
3.	Introduce datos en los criterios de búsqueda que desee y selecciona la opción Buscar.	
4.		Muestra en la interfaz “Buscar Documentos” los resultados de la búsqueda. Permite realizar las acciones: <ul style="list-style-type: none"> <li>• Vista Previa. Ver CU “Vista Previa de Documentos”.</li> </ul> En caso que no existan resultados de la búsqueda. Ver Flujo alternativo 4.a No existen resultados de la búsqueda.
5.		Termina el caso de uso.

### Flujos alternos

#### 2.a <Seleccionar la opción Cerrar >

	Actor	Sistema
1.	Selecciona la opción Cerrar.	
2.		Muestra un mensaje para confirmar si está seguro de cerrar la interfaz. Permite realizar las acciones: <ul style="list-style-type: none"> <li>• Aceptar</li> <li>• Cancelar. Ver flujo alternativo 2.a.1.a</li> </ul> Selecciona la opción Cancelar.
3.	Selecciona la opción Aceptar.	
4.		Regresa a la interfaz principal del módulo QPD.
5.		Termina el caso de uso.

#### 2.a.1.a <Seleccionar la opción Cancelar >

	Actor	Sistema
1.	Selecciona la opción Cancelar.	
2.		Regresa al paso 2 del flujo básico.

#### 4.a <No existen resultados de la búsqueda>

	Actor	Sistema
--	-------	---------



1.	Selecciona la opción Buscar.	
		Muestra un mensaje informando que no existen resultados de la búsqueda.
2.	Selecciona la opción Aceptar.	
3.		Muestra nuevamente la interfaz "Buscar Documentos".
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b>	<b>no</b>	N/A
<b>Asuntos pendientes</b>		N/A

### 2.3.4. Validación de requisitos del proceso.

Como parte de esta fase dentro de la ingeniería de requisitos fue realizada la validación de los mismos a través de las siguientes técnicas:

#### RTF

Se llevó a cabo un conjunto de reuniones con el personal técnico del proyecto, donde se detectaron ambigüedades y omisiones que habían sido cometidas en el proceso de obtención de los requisitos. También se realizó una revisión por el equipo de calidad donde resultaron 2 no conformidades, las cuales fueron corregidas para una segunda revisión.

#### Validación con el cliente

La validación con los clientes fue mediante los prototipos de interfaz de usuario dándole una visión más palpable al cliente de las funcionalidades del sistema. Se revisaron cada uno de los prototipos con los fiscales correspondientes y se corrigieron los errores, resultando de esto el acta de aceptación firmada por el cliente (*Ver Anexo 6*).

#### Aplicación de la métrica Estabilidad de los Requisitos

Es necesario asegurarse de que los requisitos sean lo más estables posibles antes de pasar a la próxima disciplina. Para esto se aplicó la métrica Estabilidad de los Requisitos en una primera iteración en la cual se calculó de la siguiente manera:

$$ETR = [(RT - RM) / RT] * 100$$

Donde:

ETR: valor de la estabilidad de los requisitos.

RT: total de requisitos definidos.



## Capítulo 2. Análisis y Diseño

RM: número de requisitos modificados, que se obtienen como la sumatoria de los requisitos insertados, modificados y eliminados.

Teniendo en cuenta que se identificaron un total de 52 RF, dentro de los cuales 23 resultaron modificados (RM), se calcula:

$$ETR = [(47 - 14) / 47] * 100 = 70.2$$

Luego de aplicar esta métrica, se arrojó como resultado que los requisitos no fueron lo suficientemente estables debido a cambios que se realizaron al comienzo del desarrollo de la fase. Una vez corregidos todos los señalamientos o sugerencias se realizaron nuevamente las validaciones en conjunto con los clientes implicados y los resultados fueron finalmente satisfactorios.

### Métrica para evaluar los casos de uso

Para evaluar la calidad de los casos de uso se utilizó un modelo de métricas orientado a objetos el cual, como se explica en el primer capítulo, tiene en cuenta cuatro atributos: Completitud, Correctitud, Consistencia y Complejidad. Estos atributos se analizan en términos de un conjunto de factores, cada uno de los cuales tiene asociada una métrica. En la siguiente tabla se muestran los resultados obtenidos:

**Tabla: 5 Resultados de la métrica para evaluar los casos de uso.**

Atributo	Factor	Métrica asociada	Valor
Completitud	<b>Factor 1.</b> ¿Han sido definidos todos los roles relevantes de usuario encargados de generar/modificar o consultar información?	<b>Métrica 1:</b> Número de roles relevantes omitidos.  Umbral < 90%	Número de roles relevantes omitidos: 0  <b>Resultado: 100%.</b>
	<b>Factor 2.</b> ¿Se presenta una descripción resumida de todos los casos de uso?	<b>Métrica 2:</b> Número de casos de uso que no tienen descripción resumida.  Umbral < 90%	Número de casos de uso que no tienen descripción resumida: 0  <b>Resultado: 100%.</b>
	<b>Factor 3:</b> ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	<b>Métrica 3:</b> Número de requisitos omitidos por caso de uso.  Umbral < 90%	Número de requisitos omitidos por caso de uso: 0  <b>Resultado: 100%.</b>



## Capítulo 2. Análisis y Diseño

	<b>Factor 4:</b> Todos los casos de uso han sido clasificados de acuerdo a su relevancia: (Crítico, secundario, auxiliar y opcional).	<b>Métrica 5:</b> Número de casos de uso que no han sido clasificados  Umbral < 80%.	Número de casos de uso que no han sido clasificados: 0  <b>Resultado: 100%.</b>
		<b>Umbral &lt; 86%.</b>	Se presenta un 100%.
Consistencia	<b>Factor 5.</b> ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	<b>Métrica 6:</b> Número de casos de uso que tienen un nombre incorrecto.  Umbral < 80%.	Número de casos de uso que tienen un nombre incorrecto: 0  <b>Resultado: 100%.</b>
	<b>Factor 6.</b> ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?	<b>Métrica 7:</b> Grado de adecuación de la descripción del flujo de eventos para un caso de uso.  Umbral < 75%.  <b>Métrica 8:</b> Número de casos de uso no aceptados.  Umbral < 80%.	La descripción se define en el lenguaje del usuario. Se define el responsable de cada acción.  <b>Resultado: 100%.</b>
	<b>Factor 7.</b> ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable?	<b>Métrica 9:</b> Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema.  Umbral < 90%.	Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema: 0  <b>Resultado: 100%.</b>
		<b>Umbral &lt; 55%.</b>	<b>Se presenta un 100%.</b>
Correctitud	<b>Factor 9.</b> ¿Representa el caso de uso requisitos comprensibles por el usuario?	<b>Métrica 10:</b> Número de casos de uso en que los requisitos representados no son comprensibles por el usuario.	Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 0



## Capítulo 2. Análisis y Diseño

		Umbral < 95%.	<b>Resultado: 100%.</b>
	<b>Factor 10.</b> ¿Existe para cada caso de uso de negocio por lo menos un usuario responsable?	<b>Métrica 11:</b> Número de casos de uso que no tienen un usuario responsable.	Número de casos de uso que no tienen un usuario responsable: 0 <b>Resultado: 100%.</b>
	<b>Factor 11.</b> ¿Se ajusta la representación del diagrama de casos de uso de acuerdo a lo normado en la metodología?	<b>Métrica 12:</b> Grado en el que se ajusta el diagrama de casos de uso a la metodología.  Umbral < 90%.	Grado en el que se ajusta el diagrama de casos de uso a la metodología. <b>Resultado: 100%.</b>
	<b>Factor 12.</b> ¿Las interacciones definidas introducen mejoras al proceso actual?	<b>Métrica 13:</b> Número de casos de uso que deben ser modificados para mejorar el proceso actual.  Umbral < 90%.	Número de casos de uso que deben ser modificados para mejorar el proceso actual: 1 <b>Resultado: 94.74%.</b>
		<b>Umbral &lt; 70%.</b>	<b>Se presenta un 98.69%</b>
Complejidad	<b>Factor 13.</b> ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?	<b>Métrica 14:</b> Número de elementos del diagrama que requieren reubicación.  Umbral < 70%.	Número de elementos del diagrama que requieren reubicación: 0 <b>Resultado: 100%.</b>
		<b>Umbral &lt; 70%.</b>	<b>Resultado: 100%.</b>

Al aplicar estas métricas se pudo evaluar los factores completitud, consistencia, correctitud y complejidad del diagrama de casos de uso del sistema. Los resultados arrojados dieron la posibilidad de comprobar que los requisitos que se identificaron son tratados al menos en un caso de uso, con lo que se demuestra que todas las necesidades expresadas por el cliente están presentes. También se verificó que los casos de uso fueron descritos de forma clara, mostrando el flujo alterno separado del flujo básico lo que da una mayor legibilidad de los mismos.



### 2.3.5. Conclusiones parciales.

Luego de haber realizado las actividades correspondientes a esta disciplina se logró entender y realizar una buena identificación de requisitos, llevar un estricto control de los mismos identificando sus inconsistencias y ambigüedades. Se generaron de manera eficiente los artefactos necesarios para continuar con el desarrollo de la siguiente disciplina: Especificación de Requisitos de Software y Especificación de Casos de Uso.

## 2.4. Diseño.

El diseño es uno de los pilares fundamentales en la elaboración del software. El diseño propuesto tiene que cumplir a cabalidad con los requerimientos del sistema ya que es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema de software finalizado (23). Debe ser capaz de facilitar las mejoras del software, tiene que ser entendible por otros profesionales de la especialidad de manera que permita la comprobación del sistema fácilmente.

### 2.4.1. Patrones de Diseño que implementa Symfony.

En el Modelo de Diseño se hizo uso de patrones pertenecientes a los GOF y a los GRASP, para que el diseño tenga un Bajo Acoplamiento, Alta Cohesión y que cada clase experta en algún tipo de información sea la encargada de realizar las operaciones con la misma, logrando con esto un mejor funcionamiento del sistema. (22)

El marco de trabajo Symfony promueve las buenas prácticas de diseño y programación ya que maneja patrones de diseño como Factory, Abstract Factory, Decorator y Singleton pertenecientes a los patrones GOF y que son muy utilizados por parte de los desarrolladores. (22)

A continuación se describen los patrones GOF y GRASP usados en el diseño teniendo en cuenta que Symfony como marco de trabajo los incluye:

#### Patrones GoF

*En la categoría Estructurales:*

- ✓ Composite (Objeto compuesto): Permite tratar objetos compuestos como si fuese uno. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, ya que al poseer todos ellos una interfaz común, se tratan todos de la misma manera.

*En la categoría Creacionales:*



- ✓ *Singleton (solitario)*: su propósito es asegurar que una clase tenga una sola instancia y proveer un punto global de acceso a ella. Es un patrón bastante simple de implementar pero a la vez potente.
- ✓ *Abstract Factory (fábrica abstracta)*: su propósito es definir una interfaz para la creación de familias de objetos relacionados o dependientes sin tener que especificar la clase concreta. La fábrica abstracta se utiliza para la creación de los objetos entidades. Se potencia el encapsulamiento, puesto que se aísla a los clientes de las implementaciones y se obtiene un incremento de la flexibilidad del diseño.

### Patrones Grasp (31)

- ✓ *Creador*: En la clase *RegQuejaController* se encuentran todas las acciones definidas para el módulo en cuestión y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase *RegQuejaController* es "creador" de dichas entidades.
- ✓ *Experto*: Es uno de los más utilizados, puesto que *Symfony* utiliza el ORM *Doctrine* para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.
- ✓ *Alta Cohesión*: *Symfony* permite asignar responsabilidades con una alta cohesión, por ejemplo la clase *Controller* tiene la responsabilidad de definir las varias acciones, es decir está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.
- ✓ *Controlador*: Todas las peticiones web son manejadas por un solo controlador frontal (*app.php*), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.
- ✓ *Bajo Acoplamiento*. La clase *Controller* hereda solamente de *BaseController* para lograr un bajo acoplamiento de clases.

### 2.4.2. Diagrama de Secuencia.

Un diagrama de secuencia muestra una interacción ordenada según la secuencia temporal de eventos, los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes. Tiene como objetivo describir el comportamiento dinámico del sistema de información, haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos. Se pueden colocar etiquetas que pueden ser descripciones de acciones, restricciones



de tiempo, entre otros. A continuación se muestra el diagrama de secuencia para el caso de uso Depurar Queja, Petición o Denuncia.

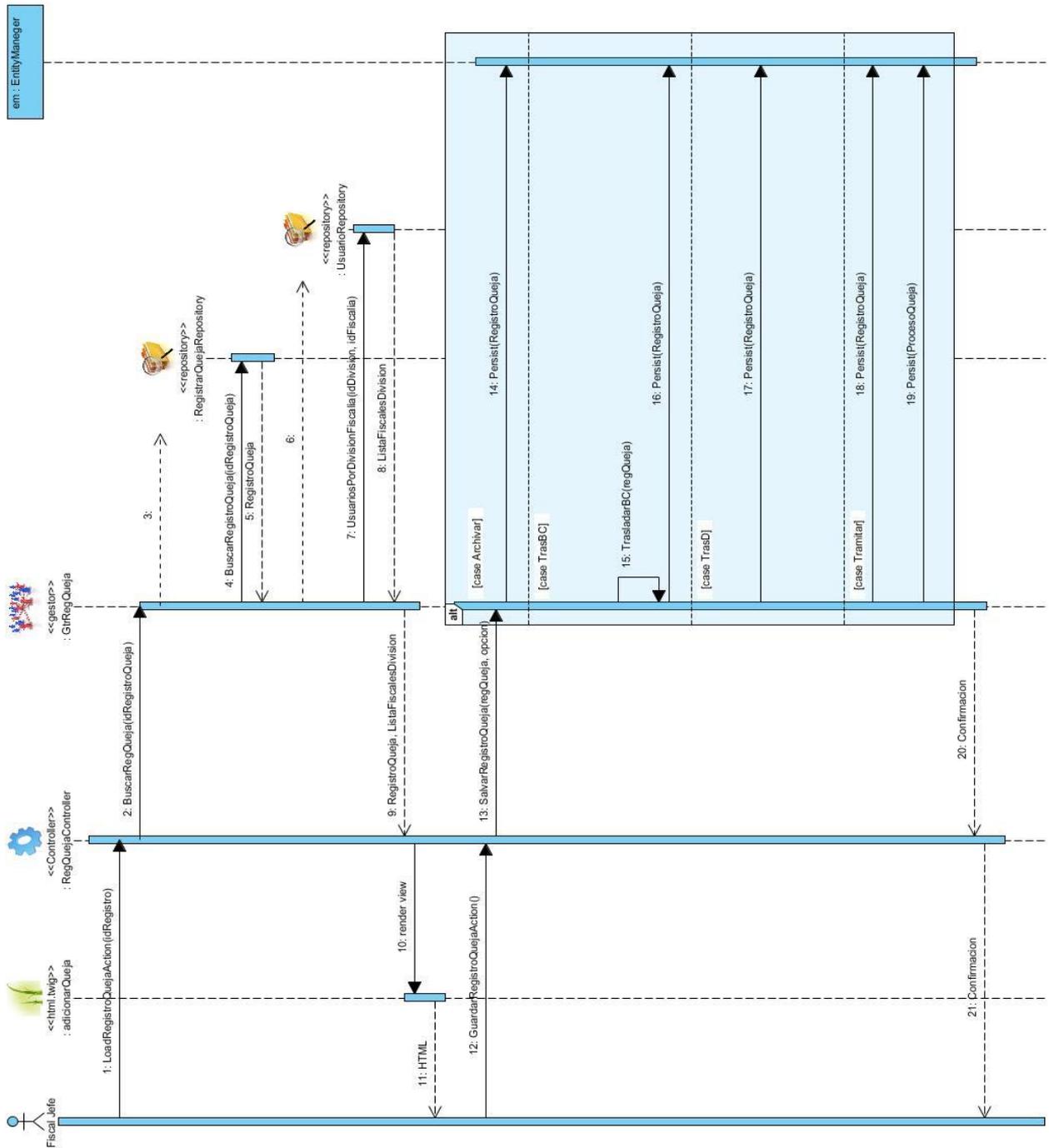


Figura Nº 13 Diagrama de Secuencia del caso de uso Depurar Queja, Petición o Denuncia.

### 2.4.3. Modelo lógico de datos.

El modelo lógico describe de manera abstracta cómo se representan los datos de una aplicación o sistema de información. Este modelo tiene gran importancia ya que define formalmente las estructuras permitidas y las restricciones que se aplican con el fin de representar los datos del



## Capítulo 2. Análisis y Diseño

dominio de la aplicación. Está compuesto por objetos: entidades que existen y se manipulan; atributos, características básicas de dichos objetos y relaciones que son la forma en que se enlazan los objetos entre sí (32). A continuación se muestra el Modelo lógico de datos del módulo Quejas, Peticiones o Denuncias:

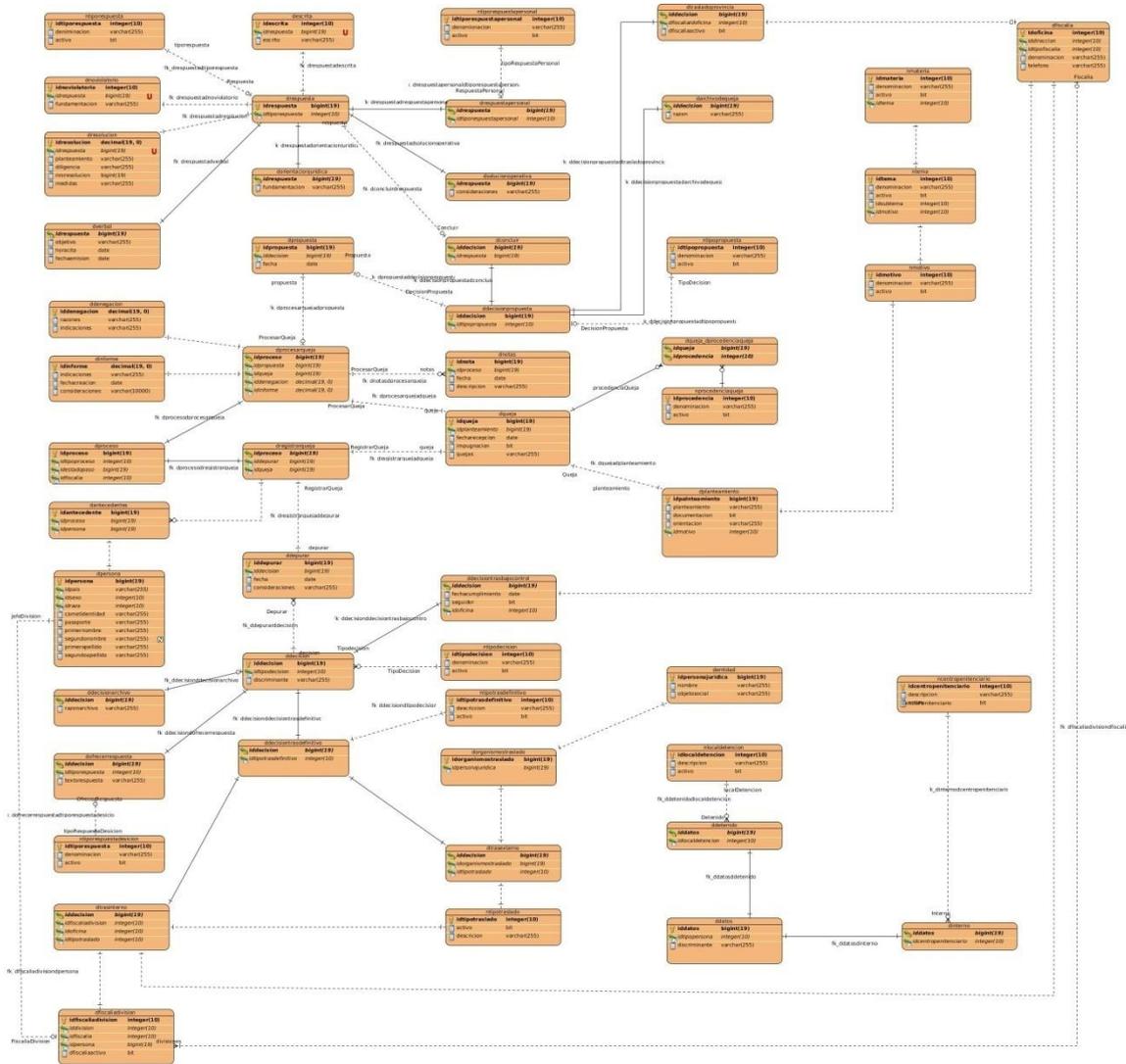


Figura N° 14 Modelo lógico de datos del Módulo Quejas, Peticiones o Denuncias.

Tabla: 6 Descripción de las tablas del modelo lógico.

Nombre de la Tabla	Descripción
dantecedentes	Entidad que contiene la referencia hacia los antecedentes de una persona
darchivodequeja	Archivo guardado para incluir en las revisiones de una queja.
dconcluir	Referencias mediante id para saber la respuesta a una decisión propuesta.
ddecision	Entidad que contiene el id general de una decisión.
ddecisionarchivo	Referencias mediante id para saber la decisión sobre un archivo.
ddecisionpropuesta	Referencias mediante id para saber la decisión propuesta.
ddecisiontrasbajocontrol	Referencias mediante id para saber la decisión de un traspaso bajo un



## Capítulo 2. Análisis y Diseño

	control de alguna entidad fiscal.
ddecisiontrasdefinitivo	Referencias mediante id para saber la decisión de un traspaso definitivo
ddenegacion	Referencia a la denegación mediante razones e indicaciones para procesar una queja.
ddepurar	Contiene la decisión final para procesar una queja.
dentidad	La entidad esta descrita en el modulo de administración, para guardar las entidades fiscales con las cuales se trabaja
descrita	Contiene el escrito que sirve como respuesta a algún tipo de acción.
dfiscalia	La entidad esta descrita en el modulo de administración, contiene los datos de las fiscalías
dfiscaliadivision	La entidad esta descrita en el modulo de administración, contiene la referencia a las divisiones de las fiscalías
dinforme	Contiene el informe utilizado a la hora de procesar una queja.
dnotas	Notas que hacen referencia a un proceso de queja.
dnoviolatorio	Fundamentación no violable para procesar una queja.
dofrecerrespuesta	Contiene la respuesta a una decisión.
dorganismostraslado	Organismo de traslado para un traslado externo.
dorientacionjuridica	Fundamentación jurídica para una respuesta.
dplanteamiento	Datos del planteamiento de una queja.
dprocesarqueja	Referencias a los datos para procesar la queja.
dproceso	La entidad esta descrita en el modulo de administración, contiene la referencia principal a los datos que se describen en los procesos.
dpersona	La entidad esta descrita en el modulo de administración, contiene los datos generales de una persona.
dpropuesta	Referencia a alguna propuesta para procesar alguna queja.
dqueja	Datos generales de una queja.
dqueja_dprocedenciaqueja	Relación de la queja con la procedencia que la misma posee.
dregistrarqueja	Registro oficial de la queja procesada con su última decisión.
dresolucion	Resolución incluida en las respuestas.
dspuesta	Referencia general a la respuesta.
drespuestapersonal	Referencia a un tipo de respuesta para una respuesta.
dsolucionoperativa	Consideraciones de la solución operativa para una respuesta.
dtrasexterno	Datos de un traslado externo.
dtrasinterno	Datos de un traslado interno.
dverbal	Datos agregados para la respuesta.
dinterno	Contiene la referencia de un interno en un centro penitenciario
ddatos	La entidad esta descrita en el modulo de administración, contiene la referencia a los datos del interno
ddetenido	Referencia a la localización del detenido.
nmateria	Materias generales de cualquier tema.
nmotivo	Motivos que se presentan en algún planteamiento.
nprocedenciaqueja	Procedencia de las quejas.
ntema	Temas que poseen los motivos.
ntipodecision	Tipos de decisión a tomar.
ntipopropuesta	Contiene los tipos de propuesta para una decisión de propuesta.
ntiporespuesta	Tipos de respuesta que se pueden dar en una respuesta.
ntiporespuestadesicion	Tipos de respuesta que se pueden dar a una decisión.
ntiporespuestapersonal	Respuestas que puede dar el personal en una respuesta.
ntipotrasdefinitivo	Tipos de traslado definitivos.
ntipotraslado	Tipos de traslados.



nlocaldetencion	locales donde se realiza la detención
ncentropenitenciario	Centros penitenciarios existentes.

### 2.4.4. Validación del diseño del proceso.

La calidad en el diseño es el grado con que las especificaciones de diseño (especificaciones acerca de los distintos componentes del producto) responden a las expectativas del consumidor (33). Cuando se mide la calidad del diseño de software es importante aplicar métricas que arrojen resultados útiles para la medición. El objetivo de aplicar métricas en este sentido es, sencillamente mejorar, pues esto ayuda a la identificación de problemas y a descubrir oportunidades para enriquecer la calidad del sistema y el rendimiento del mismo.

A continuación se muestran los resultados obtenidos tras la aplicación de la métrica tamaño operacional de clases:

✓ Tamaño Operacional de clase (TOC).

Se trabajó con un total de 57 clases para un promedio de 2.4 atributos por clases y 3.9 de operaciones lo que arrojó como resultado que las clases analizadas del diseño son pequeñas en su totalidad de acuerdo a los umbrales que establece esta métrica. Además esta métrica evalúa los atributos de calidad que a continuación se muestran:

**Tabla: 7 Resultados de la Métrica Tamaño Operacional de Clases.**

Cantidad de clases 57	Baja	Media	Alta
Responsabilidad	39%	54%	7%
Complejidad	39%	54%	7%
Reutilización	7%	54%	39%

Esto demuestra que las mismas no tienen grandes responsabilidades por lo que permite la reutilización de clases demostrando que el sistema tendrá una implementación a nivel medio. Por tanto se puede afirmar que los resultados obtenidos por la métrica son positivos.

Acoplamiento entre Clases (AC)

Con la presente métrica se evalúan los siguientes atributos de calidad:

**Responsabilidad:** Responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.



**Complejidad del mantenimiento:** Grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.

**Reutilización:** Grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.

**Cantidad de pruebas:** Número o el grado de esfuerzo para realizar las pruebas de calidad (unidad) del producto (componente, modulo, clase, conjunto de clases) diseñado.

Para los cuales están definidos los siguientes criterios y categorías de evaluación que se muestran en la siguiente tabla:

**Tabla: 8 Criterios de evaluación de la métrica AC.**

Atributo	Categoría	Criterio
<b>Acoplamiento.</b>	Ninguno.	0
	Bajo.	1
	Medio.	2
	Alto.	>2
<b>Complejidad de mantenimiento.</b>	Baja.	$\leq$ Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio
<b>Reutilización.</b>	Baja.	$> 2 \times$ Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$\leq$ Promedio
<b>Cantidad de pruebas.</b>	Baja.	$\leq$ Promedio
	Media.	Entre Promedio y $2 \times$ Promedio
	Alta.	$> 2 \times$ Promedio

Para un total de 44 clases y un promedio de asociaciones de uso de 1.6, se obtiene:

**Tabla: 9 Cantidad de dependencias por clasificación.**

Criterio	Categoría	Cantidad de clases	Promedio
0 dependencias	Muy Bueno	9	20.45454545
1 dependencias	Bueno	18	40.90909091
2 dependencias	Regular	10	22.72727273
3 dependencias	Malo	4	9.090909091
> 3 dependencias	Muy Malo	3	6.818181818
<b>Total</b>		<b>44</b>	<b>100</b>

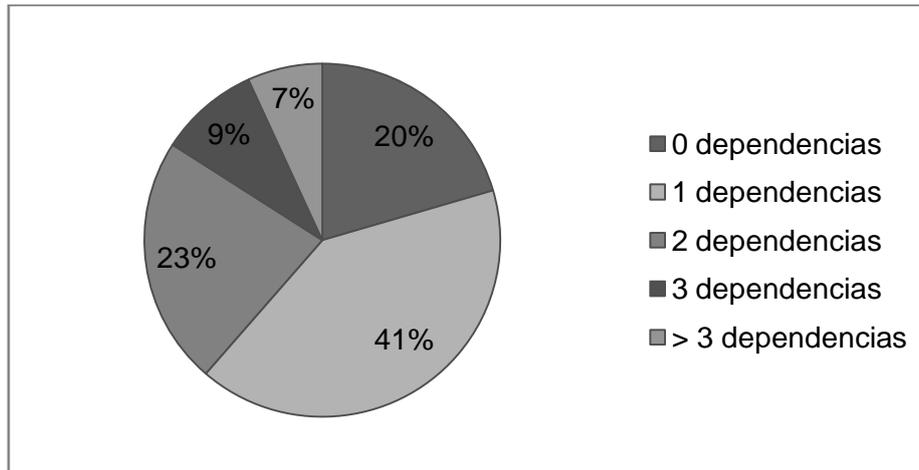


Figura Nº 15 Comportamiento de los valores de dependencia.

Tabla: 10 Resultados de los atributos de calidad de la métrica AC.

Cantidad de clases 44	Ninguno	Baja	Media	Alta
Acoplamiento	20%	40%	22%	15%
Complejidad de mantenimiento		61%	23%	16%
Cantidad de pruebas		61%	23%	16%
Reutilización		16%	23%	61%

Se puede concluir que el diseño propuesto está entre los límites aceptables de calidad, teniendo en cuenta que más de la mitad de las clases (53%) no poseen dependencias respecto a otras. Los atributos de calidad se encuentran en un nivel satisfactorio: el grado de acoplamiento es mínimo, la Complejidad de Mantenimiento, la Cantidad de Pruebas y la Reutilización se comportan favorablemente para un 61%.

### 2.4.5. Conclusiones Parciales.

Al realizar el diseño de los procesos referentes a las Quejas, Peticiones o Denuncias de la población se tiene una visión más profunda de cómo se implementará el sistema. Se obtuvo el Modelo de diseño correspondiente con sus respectivos diagramas de clases y de paquetes logrando una estructura más centralizada de las clases del diseño. Se aplicó la métrica de TOC arrojando resultados satisfactorios y con la calidad requerida.



### Capítulo 3. Implementación y Prueba.

#### 3.1. Introducción.

La implementación comienza luego del resultado del diseño. Gran parte de la arquitectura del sistema es definida durante el diseño, siendo el propósito fundamental de la implementación el desarrollar la arquitectura y el sistema como un todo.

Este capítulo está orientado a la implementación y prueba de la solución. Se hará referencia al modelo de implementación, el cual está formado por los diagramas de componentes y el de despliegue. Además se verificará la calidad del resultado de la implementación, mediante los artefactos generados durante el flujo de trabajo de pruebas, exponiendo los resultados obtenidos por diferentes tipos de pruebas realizadas al módulo.

#### 3.2. Modelo de Implementación

El modelo de Implementación está conformado por los Diagramas de Componentes y de Despliegue, describiendo cómo los elementos del Modelo de Diseño se implementan en términos de componentes, ficheros de código fuente y ejecutables. Describe además cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización, disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y cómo dependen los componentes unos de otros. (34)

##### 3.2.1. Diagrama de Componentes.

Un componente es una parte física de un sistema (módulo, base de datos, programa o ejecutable), es decir, la materialización de una o más clases. El diagrama de componentes representa la forma en que un sistema de software se divide en componentes, mostrando así las dependencias entre dichos componentes. (35).

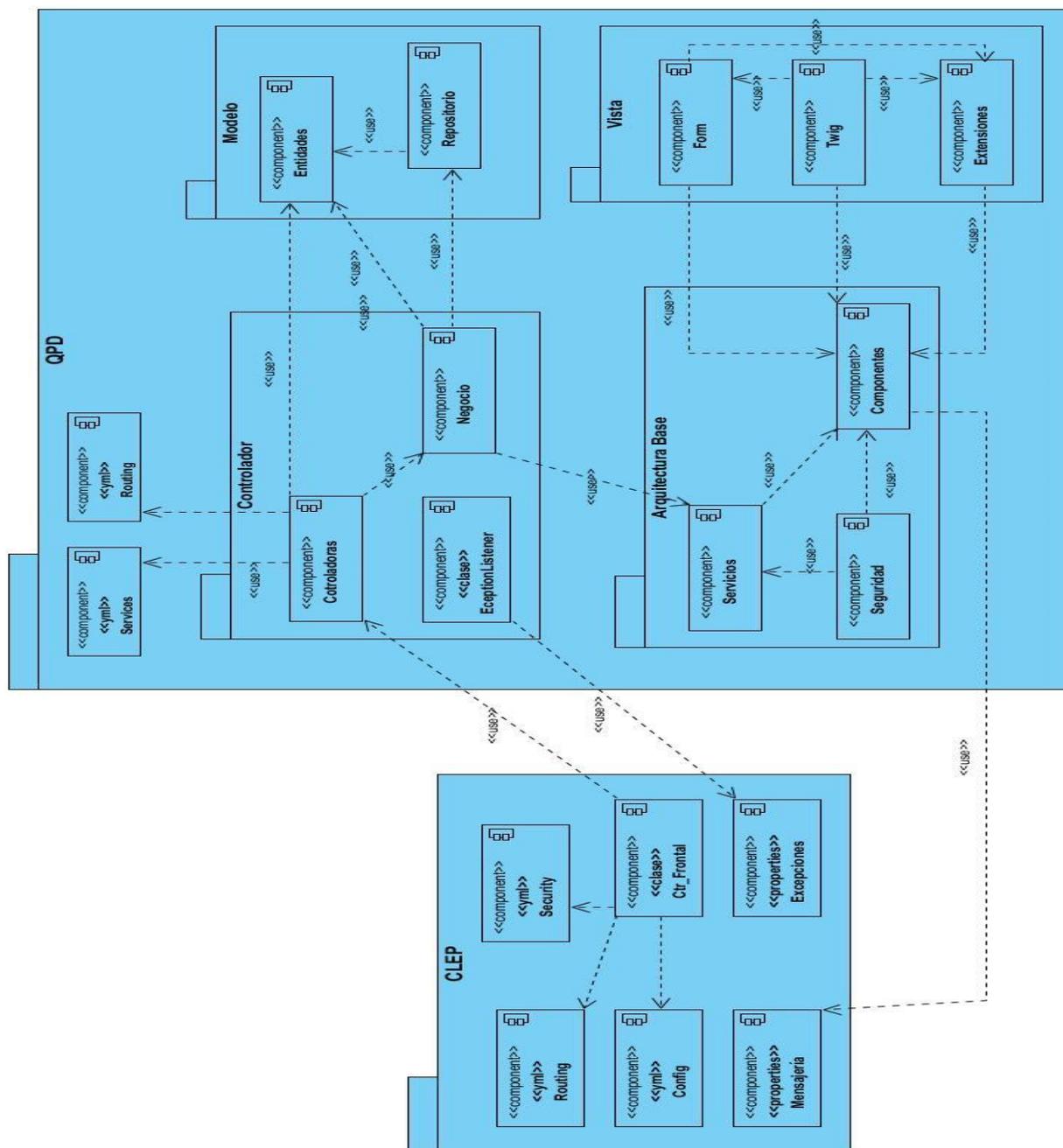


Figura Nº 16 Diagrama de componentes.

### 3.2.2. Modelo de Despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema. El diagrama de despliegue es un tipo de diagrama empleado para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Muestra las relaciones físicas de los distintos nodos que componen un sistema (35). A continuación se muestra el diagrama de despliegue de la solución propuesta:



## Capítulo 3. Implementación y Prueba

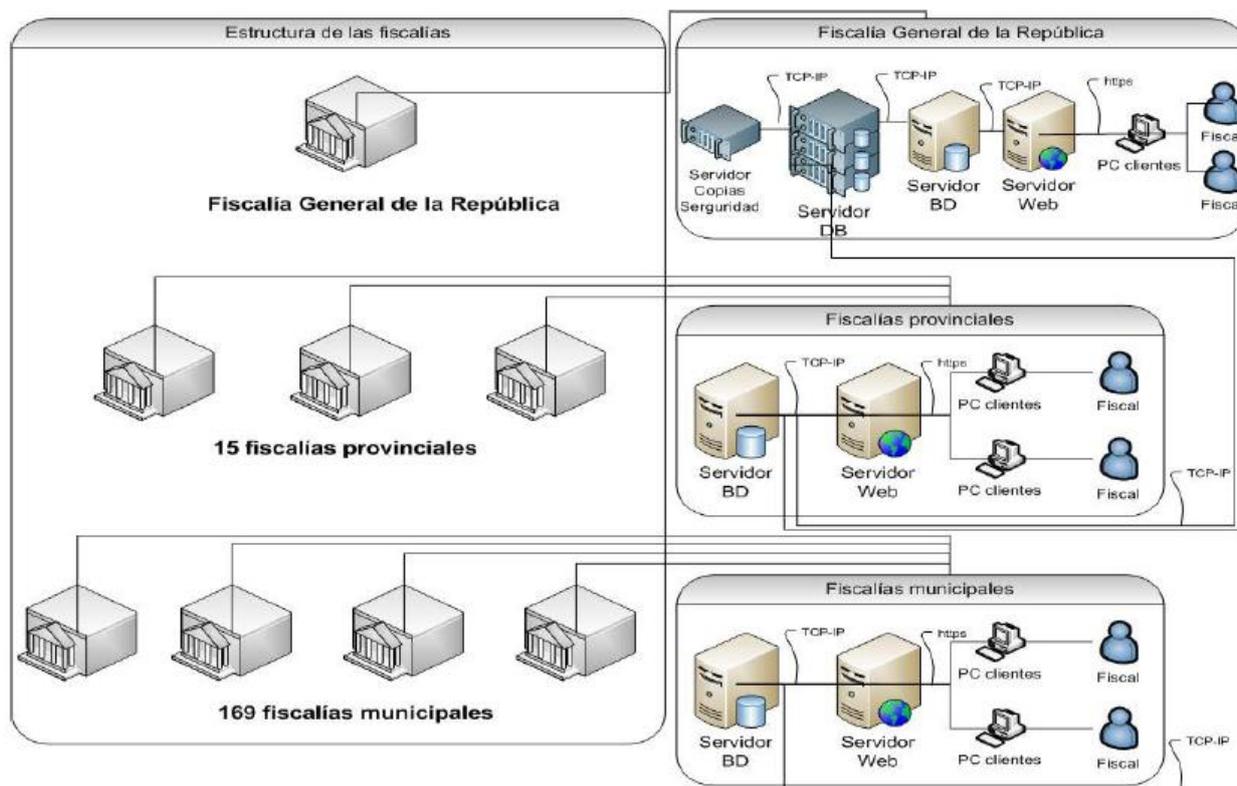


Figura N° 17 Diagrama de despliegue.

### 3.2.3. Estándares de codificación.

Los estándares de codificación son un grupo de reglas que deben seguir los desarrolladores para obtener un código fácil de comprender y de alta calidad, fomentando las buenas prácticas definidas por la Ingeniería de Software. Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armónico, la legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. Posibilita que un equipo de programadores mantenga un código de calidad sobre el que se efectuarán luego revisiones del código. (8).

En el caso del Módulo Quejas, Peticiones o Denuncias, el estándar de codificación fue definido por la dirección del proyecto en sus inicios y se encuentra registrado en el documento "0120\_54 Estándares de codificación para PHP".

A continuación se muestran algunos ejemplos de cómo nombrar los elementos en el módulo QPD:

- Todas las nomenclaturas a utilizar son en español.
- El identificador para las variables, los parámetros y los métodos se definen escribiendo, las palabras simples con la primera letra en minúsculas, y en caso de ser un nombre compuesto usando la notación **Camello**, para el cual existen dos variantes:



## Capítulo 3. Implementación y Prueba

---

- **UpperCamelCase:** las palabras que forman el nombre se escriben juntas y la primera letra de cada una de ellas en mayúscula.
- **LowerCamelCase:** las palabras que forman el nombre se escriben juntas, la primera letra de la primera palabra en minúscula, el resto de las palabras la primera letra en mayúscula.

En el caso del módulo QPD se utilizará la segunda variante para definir los atributos y variables privadas y en otros casos se utilizará la primera variante, aplicando la técnica verbo-sustantivo.

- El bundle del módulo QPD se escribe con las iniciales del nombre seguido de la palabra Bundle (QPDBundle).

Propiedades y Funcionalidades del sistema.

- Las clases persistentes comienzan con las siglas definidas clep.dnombre de la tabla y en el caso de un nomenclador clep.nnombre del nomenclador (clep.dqueja, clep.ntipotraslado).
- Los formularios comienzan con nombre del formulario según su función, seguido de la palabra Type (RegQuejaType.php)
- Los gestores de negocio comienzan con el nombre del gestor seguido de Gtr (RegQuejaGtr.php)
- Para las relaciones entre clases usarán <<use>>.
- Para las relaciones entre clases e interfaces usarán <<realize>>.
- Para las relaciones entre clases y paquetes usarán <<access>>.

Nomenclatura para los componentes de formularios:

- Los botones comienzan con las siglas Btn seguido del nombre de la acción (BtnAdicionar).
- El nombre del formulario está compuesto por el nombre del formulario, éste a su vez puede contener dentro otros formularios, este nombre estaría compuesto por el nombre del primer formulario\_nombre, del segundo y así sucesivamente si hay otros formularios anidados (clep\_qpdbundle\_registrarquejatype\_queja).
- Los campos de texto, campos fecha, componentes de tipo Jscroll, cajas de texto (JComboBox), áreas de texto (JTextArea), labels (JLabel), cuadros de chequeo (JCheckBox), y tablas (JTable), su nombre está compuesto por el nombre del bundle donde se encuentra el nombre del formulario y el nombre del campo. (clep\_qpdbundle\_registrarquejatype\_queja\_fechaRecepcion).



### 3.3. Pruebas.

Al desarrollar sistemas informáticos se corre un alto riesgo de que se produzcan errores producto al fallo humano. Estos pueden ocurrir desde el comienzo del proceso, ya sea en la definición de los objetivos, el diseño, la implementación o en otras disciplinas. Las pruebas de software son un conjunto de herramientas, técnicas y métodos que evalúan el desempeño de un programa. Involucran las operaciones del sistema bajo condiciones controladas y evaluando los resultados, es por eso que la realización de las mismas a los software es un factor de vital importancia. Las pruebas no pueden confirmar la ausencia de errores del software, pero sí demostrar que tiene defectos.

#### 3.3.1. Prueba de Caja Negra

Las pruebas de caja negra son pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene (36).

Las pruebas de caja negra pretenden encontrar los tipos de errores siguientes:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Este método consta de varias técnicas:

- Técnica de la Partición de Equivalencia: divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

La técnica a emplear es la de Partición de Equivalencia, que se basa en una evaluación de las clases de equivalencia para una condición de entrada. Donde una clase de equivalencia representa un conjunto de estados válidos, inválidos o que no aplican, para determinadas condiciones de entrada. Las condiciones de entrada son valores numéricos específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. A continuación se muestra el resultado del



## Capítulo 3. Implementación y Prueba

caso de prueba para el escenario *Adicionar Queja, Petición o Denuncia del caso de uso Gestionar Queja*, las pruebas de los restantes casos de uso, se muestran en el documento 0111\_Diseño de Casos de Prueba Gestionar Queja.

**Tabla: 11 Escenario Adicionar Queja, Petición o Denuncia del Caso de uso Gestionar Queja.**

Escenario	Descripción	Nombres	Apellidos	Sexo	Edad	Raza	No.CI	País	Nacionalidad	Teléfono	Dirección Particular	Respuesta del sistema	Flujo central	
EC 1.1 Adicionar Queja, Petición o Denuncia.	Permite adicionar una queja, petición o denuncia al sistema.	V	V	V	V	V	V	V	V	V	V	Debe adicionar la queja al listado de quejas recibidas del fiscal jefe para que sea depurada.  Debe mostrar un mensaje indicando que existen errores y/o campos obligatorios en blanco al seleccionar la opción Adicionar.	1- Selecciona la opción del Menú Principal/ Registrar Queja o Seleccionar la opción Quejas Recibidas/seleccionar una y Adicionar. 2- Muestra una pantalla con los datos a introducir o a seleccionar de la queja. 3- Selecciona la opción Adicionar. 4- Registra la queja en el listado de quejas recibidas al fiscal jefe para que las depure.	
		Roberta	Mendoza López	F	38	B	73110215236	Cuba	Cubana	8523314	una			
		N/A	V	V	V	V	V	V	V	V	V			
		vacío	Mendoza López	F	38	B	73110215236	Cuba	Cubana	8523314	si			
		V	N/A	V	V	V	V	V	V	V	V			
		Roberta	vacío	F	38	B	73110215236	Cuba	Cubana	8523314	una			
		V	V	N/A	V	V	V	V	V	V	V			
		Roberta	Mendoza López	vacío	38	B	73110215236	Cuba	Cubana	8523314	una			
		V	V	V	N/A	V	V	V	V	V	V			
		Roberta	Mendoza López	F	38	B	73110215236	Cuba	Cubana	8523314	una			
		V	V	V	V	V	N/A	V	V	V	V			
		Roberta	Mendoza López	F	38	B	vacío	73110215236	Cuba	Cubana	8523314			una
		V	V	V	V	V	V	V	V	V	V			
Roberta	Mendoza López	F	38	B	vacío	73110215236	Cuba	Cubana	8523314	una				
V	V	V	V	V	V	V	V	V	V					
Roberta	Mendoza López	F	38	B	73110215236	Cuba	Cubana	8523314	una					
I	V	V	V	V	V	V	V	V	V	V	Debe mostrar un mensaje indicando que el <b>Nombre</b> que intenta Adicionar no es correcto, pues este campo solo admite letras.			
45L-Mar/	Mendoza López	F	38	B	73110215236	Cuba	Cubana	8523314	una					
V	I	V	V	V	V	V	V	V	V	V	Debe mostrar un mensaje indicando que el <b>Apellidos</b> que intenta Adicionar no es correcto, pues este campo solo admite letras.			
Roberta	.Rod.*()	F	38	B	73110215236	Cuba	Cubana	8523314	una					
V	V	V	I	V	V	V	V	V	V	V	Debe mostrar un mensaje indicando que la <b>Edad</b> que intenta Adicionar no es correcta, pues este campo solo admite números.			
Roberta	Mendoza López	F	okis	B	73110215236	Cuba	Cubana	8523314	una					



## Capítulo 3. Implementación y Prueba

V	V	V	V	V	I	V	V	V	V	Debe mostrar un mensaje indicando que el CI que intenta Adicionar no es correcto, pues este campo solo admite números.
Roberta	Mendoza López	F	38	B	Letras y caracteres especiales.	Cuba	Cubana	8523314	una	
V	V	V	V	V	V	V	V	I	V	Debe mostrar un mensaje indicando que el Teléfono que intenta Adicionar no es correcto, pues este campo solo admite números.
Roberta	Mendoza López	F	38	B	73110215236	Cuba	Cubana	Letras y caracteres especiales.	una	

En la primera iteración de aplicación de las pruebas se obtuvo 9 no conformidades lo que representa un 13% y el resto resultaron satisfactorias representando un 87% del total, ya en la segunda iteración se corrigieron las no conformidades y resultaron un 100% satisfactorias.

### 3.3.2. Pruebas de unidad.

Las pruebas unitarias son un procedimiento para validar que una porción del código del sistema funciona apropiadamente de manera aislada. En caso de una programación de procedimientos se entiende como un programa individual, una función o procedimiento, etcétera. (36).

#### 3.3.2.1. Prueba del camino básico.

La técnica camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución.

Para aplicar esta prueba se han definido una serie pasos a seguir que a continuación se describen:

1. **Notación del grafo de flujo:** usando el código como base se realiza la representación del grafo de flujo (*grafo del programa*<sup>18</sup>), mediante una sencilla notación. Cada construcción estructurada tiene su correspondiente símbolo.
  - **Nodo:** cada círculo, denominado nodo, representa una o más sentencias procedimentales.
  - **Arista:** las flechas del grafo de flujo, denominadas aristas, representan flujo de control y son análogas a las flechas del diagrama de flujo.
  - **Región:** Las áreas delimitadas por aristas y nodos se denominan regiones.

<sup>18</sup>Sirve como herramienta útil para ilustrar el método.



## Capítulo 3. Implementación y Prueba

---

2. **Complejidad ciclomática:** es una métrica que proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de *caminos independientes*<sup>19</sup> del conjunto básico de un programa y ofrece un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez. La complejidad ciclomática se calcula de tres formas:
  1. El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
  2. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  se define como:  $V(G)=A-N+2$ , donde  $A$  es el número de aristas del grafo de flujo y  $N$  es el número de nodos del mismo.
  3. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$  también se define como  $V(G)=P+1$ , donde  $P$  es el número de nodos predicado contenidos en el grafo de flujo  $G$ .
3. **Determinar un conjunto básico de caminos linealmente independientes:** El valor de  $V(G)$  brinda el número de caminos linealmente independientes de la estructura de control del programa.
4. **Obtención de casos de prueba:** se realizan los casos de pruebas que forzarán la ejecución de cada camino del conjunto básico.

### Aplicación de la prueba

La siguiente figura es el código que se tomó como base para realizar el procedimiento anteriormente descrito para la técnica de camino base; éste correspondiente al método **cargarFormulariosAction**.

---

<sup>19</sup>Camino del programa que introduce un nuevo conjunto de sentencias de proceso o una nueva condición. Está constituido por al menos una arista que no haya sido recorrida anteriormente a la definición del camino.



## Capítulo 3. Implementación y Prueba

```
public function cargarFormulariosAction()
{
    $idtipodecision = $this->getRequest()->get('idtipodecision');
    $tipodec = $this->getRequest()->get('tipodec');
    $regQueja = new RegistrarQueja();
    $options = array('idtipodecision' => $idtipodecision,
                    'tipoProc' => 1,
                    'tipodec' => $tipodec,
                    'qbTipoDecision' => $this->Resolver('RegQuejaGtr', 'qbListarTipoDecision', array($tipodec)),
                    );

    if($tipodec == 1)
    {
        if($idtipodecision == 1)
        {
            $form = $this->createForm( new RegistrarQuejaType(), $regQueja , $options);
            return $this->render('QPDBundle:RegQueja:formularios\DecisionArchivo.html.twig',
                                array('form' => $form->createView() ));
        }
        else if($idtipodecision == 2)
        {
            $form = $this->createForm( new RegistrarQuejaType(), $regQueja , $options);
            return $this->render('QPDBundle:RegQueja:formularios\TrasBajoControl.html.twig',
                                array('form' => $form->createView()));
        }
    }
    else if($tipodec == 2)
    {
        if($idtipodecision == 4)
        {
            $form = $this->createForm( new RegistrarQuejaType(), $regQueja , $options);
            return $this->render('QPDBundle:RegQueja:formularios\OfrecerRespuesta.html.twig',
                                array('form' => $form->createView()));
        }
        else if($idtipodecision == 5)
        {
            return new Response(null);
        }
    }
}
```

Figura Nº 18 Código fuente del método cargarFormulariosAction.

Se realiza el grafo de flujo partiendo del código tomado:

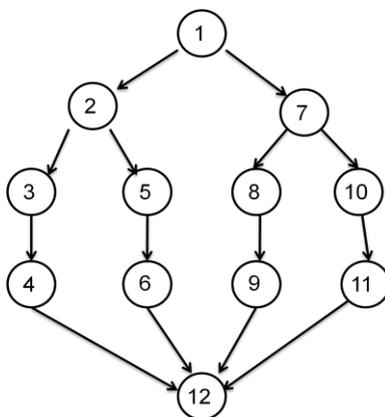


Figura Nº 19 Grafo de flujo del método cargarFormulariosAction.



## Capítulo 3. Implementación y Prueba

---

Luego se calcula la **complejidad ciclomática**:

1.  $V(G) = 4$

2.  $V(G) = A - N + 2 = 14 - 12 + 2 = 4$

3.  $V(G) = P + 1 = 3 + 1 = 4$

El valor de  $V(G)$  brinda el número de **caminos linealmente independientes** de la estructura de control del programa, por lo que se definen los siguientes 4 caminos:

**Camino1:** 1-2-3-4-12

**Camino1:** 1-2-5-6-12

**Camino1:** 1-7-8-9-12

**Camino1:** 1-7-10-11-12

Cada camino independiente es un **caso de prueba** a realizar, de forma que los datos introducidos provoquen que se visiten las sentencias vinculadas a cada nodo del camino.

### Resultados obtenidos

Se llevaron a cabo 21 pruebas correspondientes a las funcionalidades del módulo propuesto como parte de la investigación, de las cuales para una primera iteración 3 resultaron fallidas representando un 14% del total, mientras que el resto resultaron satisfactorias. De estas, 4 están vinculadas a la funcionalidad *cargarFormulariosAction* tomada como ejemplo anteriormente.

En una segunda iteración para un total de 21 pruebas se obtuvo 21 pruebas satisfactorias y 0 fallidas.

### 3.4. Conclusiones Parciales

En el desarrollo del presente capítulo, se valida el sistema desarrollado mediante pruebas de caja negra, realizando casos de prueba a todas las funcionalidades del mismo en dos iteraciones. Este proceso permite detectar la mayor cantidad de no conformidades presentes en el software para así darle solución a los problemas detectados. De esta manera es posible obtener finalmente una aplicación que satisface los requisitos definidos y que cuenta con la calidad requerida.



### Conclusiones Generales

Luego de realizar el presente trabajo se puede concluir que:

- ✓ El estudio y análisis del estado del arte permitió seleccionar herramientas y lenguajes que logran ajustarse a las necesidades del proceso quejas, peticiones o denuncias.
- ✓ Fueron utilizadas varias técnicas y patrones durante las fases desarrolladas que permitieron obtener resultados fiables los cuales se vieron reflejados en cada uno de los artefactos generados.
- ✓ Las técnicas y métricas utilizadas para validar los entregables obtenidos en cada una de las disciplinas, propiciaron la correcta medición de los mismos garantizando así resultados eficientes que lograron contribuir con la rápida implementación del sistema.
- ✓ Mediante el estudio de diferentes técnicas y métodos se logró realizar pruebas a la solución, las cuales demuestran la calidad que presenta el sistema.



### **Recomendaciones**

Como complemento al presente trabajo, se recomienda lo siguiente:

- Realizar otras iteraciones en las que se puedan identificar otras funcionalidades que puedan ser sugeridas por los clientes.
- Dar seguimiento al módulo QPD, al trabajar con los volúmenes de datos reales, ya que hasta el momento sólo ha sido probado con volúmenes de datos inferiores a los del entorno de utilización.



## Bibliografía

1. *Ley de Procedimiento Civil, Administrativo, Laboral y Económico.*
2. *Ley de la Fiscalía General de la República.* La Habana : s.n., 1997.
3. *Indicaciones metodológicas para la atención a la población.* .
4. **Cuba, Fiscalía General de la República de.** *Metodología sobre el Control de la Legalidad en los Establecimientos Penitenciarios, Locales de Detención.* La Habana : s.n.
5. La Rama Judicial de Puerto Rico. [En línea] <http://www.ramajudicial.pr/orientacion/imp-quejas.html>..
6. ComplaintsBoard. [En línea] <http://www.complaintsboard.com/?search=quejas>.
7. Trámites y Servicios. [En línea] [http://www.tramitesyservicios.df.gob.mx/index.php?option=com\\_content&view=article&id=406:quejas-en-el-funcionamiento-del-transporte-publico-bicitaxis-y-atencion-a-problemas-viales&catid=104:quejas-y-demandas-vecinales&Itemid=484](http://www.tramitesyservicios.df.gob.mx/index.php?option=com_content&view=article&id=406:quejas-en-el-funcionamiento-del-transporte-publico-bicitaxis-y-atencion-a-problemas-viales&catid=104:quejas-y-demandas-vecinales&Itemid=484).
8. **Machado, Ing. Yenier Figueroa.** Proyecto Técnico del Sistema de Informatización de la Gestión de las Fiscalías II. [En línea] 2011.
9. **Sommerville, Ian.** *Ingeniería del Software.*
10. **Fernández y Fernández, Carlos Alberto.** El Proceso Unificado de Rational para el Desarrollo de Software. [En línea] [Citado el: 9 de enero de 2012.] <http://www.utm.mx/~caff/doc/EI%20Proceso%20Unificado%20Rational.pdf>..
11. Rational Unified Process (RUP). [En línea] <http://www.lsi.us.es/docencia/get.php?id=697>..
12. Definición ABC. [En línea] [Citado el: 18 de enero de 2012.] <http://www.definicionabc.com/general/herramienta.php>.
13. Scribd. [En línea] <http://es.scribd.com>..
14. Visual Paradigm. [En línea] <http://www.visual-paradigm.com/aboutus/10reasons.jsp>..
15. BPMN and Business Process Management. [aut. libro] J. R. MARTIN OWEN.
16. **Pressman.** *Aprendiendo UML en 24 horas.*



17. *otros, Jacobson y. 2000.*
18. Sitio oficial de PHP. [En línea] <http://www.php.net/>.
19. Symfony. Open-Source PHP Web marco de trabajo . [En línea] <http://www.symfony-project.org/>.
20. Doctrine Documentation. [En línea] [http://www.doctrine-project.org/documentation/manual/1\\_0/en/introduction-to-models](http://www.doctrine-project.org/documentation/manual/1_0/en/introduction-to-models).
21. **M.J. Escalona, N. Koch.** *Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo.* España : s.n.
22. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999. ISBN 970-17-0261-1.
23. **Larman.** 1999.
24. **Prieto, Félix.** Patrones de diseño. [En línea] [http://www.infor.uva.es/~felix/datos/priiii/tr\\_patrones-2x4.pdf](http://www.infor.uva.es/~felix/datos/priiii/tr_patrones-2x4.pdf).
25. Patrones del "Gand of Four". Madrid, España : s.n.
26. **Calero Muñoz, Coral.** Sistema de Información del Ministerio Fiscal. [En línea] [http://eisc.univalle.edu.co/materias/Material\\_Desarrollo\\_Software/Metricas4.pdf](http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/Metricas4.pdf).
27. **Sánchez Fornaris, Maite y Alcantara Rabí, Dayanis.** Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. [En línea] [http://vinculando.org/articulos/sociedad\\_america\\_latina/propuesta\\_guia\\_de\\_medidas\\_para\\_evaluacion\\_sistemas\\_informacion.html](http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html).
28. **Pressman, Roger S.** *Ingeniería de software un enfoque práctico.* 1998.
29. **Bernárdez.** 2004.
30. **Chidamber S. R., Kemerer C. F.** "A metric suite forsz". s.l. : IEEE Transactions on Software Engineering, 1994. pp. 467-493.
31. Symfony users. [En línea] <http://groups.google.com/group/symfony-users/msg/cd94d2ddb2057355>.
32. **Larman, G.** *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* Prentice Hal : s.n., 2004.
33. **Berlinches.** 2004.



34. **Pressman, Roger.** *Ingeniería de software un enfoque práctico.* 2001.
35. *Ayuda de RUP. Rational Unified Process.* s.l. : Copyright (C) IBM Corporation 1987. Versión 7.0.12005.
36. **Pressman, Roger S.** *Ingeniería del Software. Un Enfoque Práctico, 5ta Edición.* s.l. : Editorial McGraw-Hill, 2002. ISBN: 8448132149.
37. *Ley de la Fiscalía General de la República.* .
38. CRAFTWARE. Prototipos de la Interfaz Usuaría con Axure RP Pro. Santiago de Chile. [En línea] <http://www.hotfrog.cl/Products/Axure-RP..>
39. Sistema de Gestión del Capital Humano. [En línea] <http://semanatecnologica.fordes.co.cu..>
40. Scribd. [En línea] <http://www.scribd.com/doc/3708746/Aprendiendo-UML-en-24-horas..>
41. **Limachi Laricano, H.J.** Administración y Control de Equipos Informáticos y Tecnológicos de la Universidad Nacional del Altiplano de Puno. [En línea] <http://www.vision.ime.usp.br/~hlimachi/images/Informe.pdf>.
42. Netbeans. [En línea] <http://netbeans.org..>
43. *Aplicación de Métricas Software en la evaluación de Modelos de Procesos de Negocio.* **Ruiz, Francisco, y otros.** Tampico, México-Ciudad Real, España : s.n.
44. **S. R Chidamber, C. F Kemerer.** *IEEE Transactions on Software Engineering.* "A metric suite for object oriented design". 1994.
45. **Pressman.** *Ingeniería del Software: Un Enfoque Práctico.* Sexta Edición. 2005. pág. 900 .
46. Introducción a Redes. Arquitectura Cliente/Servidor. *Microsoft Windows Server.* [En línea] <http://www.juansa.net/Admin2003/cliser.htm>.
47. *Indicaciones metodológicas para la atención a la población.*
48. *Decreto Ley 99/87. Artículo 1.*
49. **Pereira, Lic. Julio A.Fernández.** *Derecho Procesal Penal. Tema I.* Universidad de La Habana : s.n.
50. Catálogo único de trámites y servicios. [En línea] [http://www.tramitesyservicios.df.gob.mx/index.php?option=com\\_content&view=article&id=406:queja](http://www.tramitesyservicios.df.gob.mx/index.php?option=com_content&view=article&id=406:queja)



s-en-el-funcionamiento-del-transporte-publico-bicitaxis-y-atencion-a-problemas-viales&catid=104:quejas-y-demandas-vecinales&Itemid=484.

51. Calisoft. [En línea] <http://calisoft.uci.cu/index.php/proceso-de-mejora>.

52. Ingeniero Software. [En línea] <http://www.ingenierossoftware.com/calidad/cmm-cmmi.php>.

53. **Pressman**. *Aprendiendo UML en 24 horas*.

54. **Davis**. 1993.

55. **Pressman**. 2005.

56. Ingenieros de Software. [En línea] <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>.

57. **Reynoso, Carlos Billy**. Introducción a la Arquitectura de Software. [En línea] Marzo de 2004 . <http://www.willydev.net/descargas/prev/IntroArq.pdf>.

58. **Kendall**. *Análisis y Diseño de Sistemas*. 1997.

59. **INTERCOM, Grupo**. 1997-2010.

60. *Metodologías Ágiles o formales y robustas*. **Pérez, Dr.C Profesor Auxiliar Pedro Y. Piñero Pérez y Dr.C Profesor Auxiliar Pedro Y. Piñero**.

61. *CMMI-DEV, v1.2 CMU/SEI-2006-TR-008. IEEE 610.12*.

62. **Machado, Ing. Yenier Figueroa**. *Proyecto Técnico del Sistema de Informatización de la Gestión de las Fiscalías II*. 2011.

63. *Área de Proceso de Medición y Análisis de CMMI-DEV v 1.2*. **Calisoft**. Marzo 2010.

64. *Programa de Mejora. Introducción a la Administración de Requisitos. Tema I*. **Calisoft**.

65. *Documento de Arquitectura de Software SIGEF I*.

66. Complaints Board. [En línea] <http://www.complaintsboard.com/?search=quejas>.

67. **Stephen., A. White**. *Process Modeling Notation and workflow Patterns*. .

68. *Conferencia 3. Herramientas para la Administración de Requisitos. Programa de Mejora*.

69. **Robertson, Suzanne y Robertson, James**. *Mastering the Requirements Process*. s.l. : Second Edition, 2006.

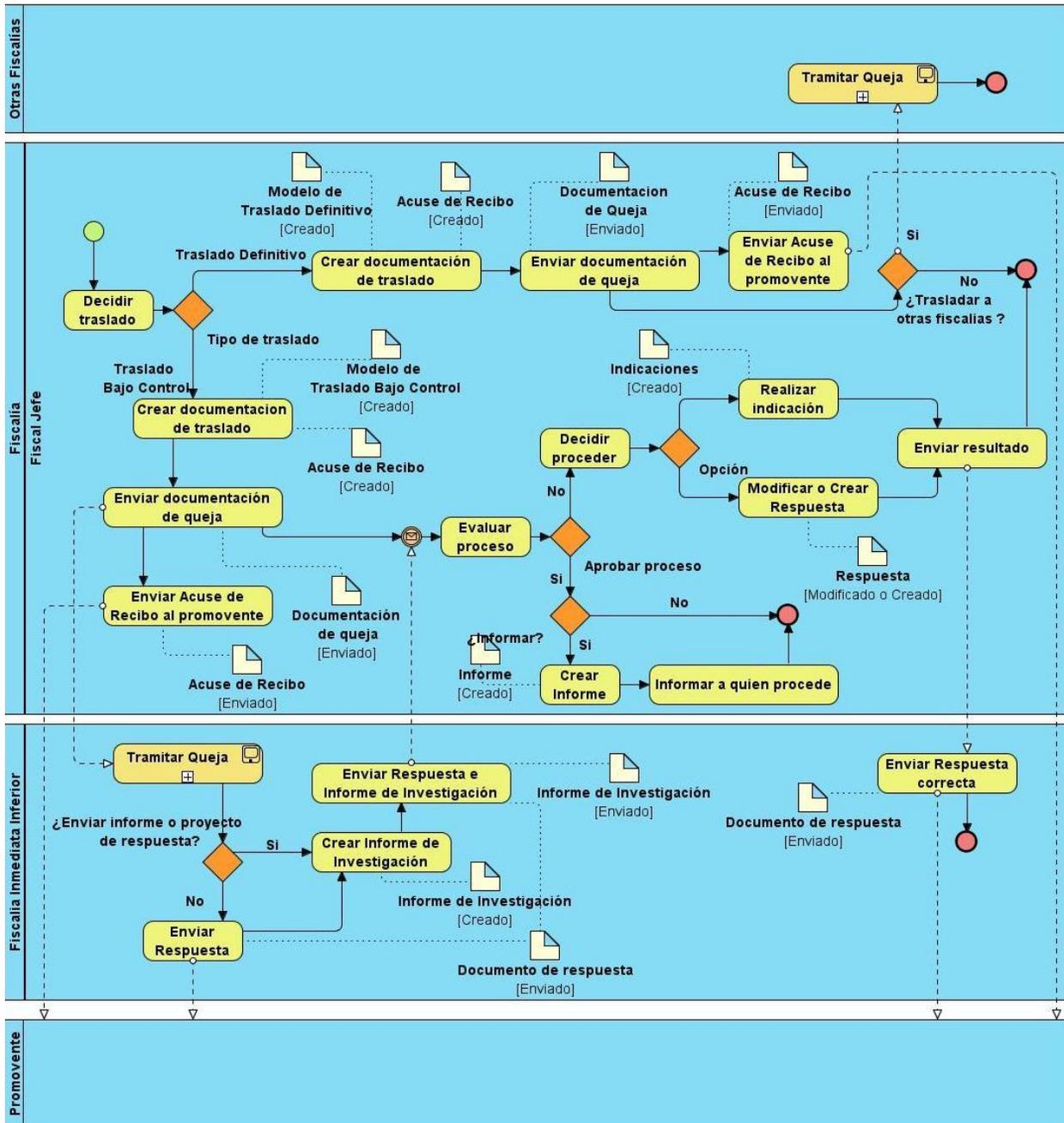


70. **Sommerville., Ian.** *Ingeniería Del Software.* . s.l. : Sexta edición.
71. Aprendiendo UML en 24 horas. [En línea] <http://www.scribd.com/doc/3708746/Aprendiendo-UML-en-24-horas>.
72. **Kotonya, G y Sommerville. I.** *Requirements Engineering: Processes and Techniques.* 2000.
73. —. *I. Requirements Engineering: Processes and Techniques.* 2000.
74. **Overgaard, Gunnar y Palmkvist, Karin.** *Use Cases Patterns and Blueprints.* 2004.
75. **LARMAN C.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* México: PRENTICE HALL : s.n., 1999.
76. **Revista Vinculando.** [En línea] 2003-2011. [http://vinculando.org/articulos/sociedad\\_america\\_latina/propuesta\\_guia\\_de\\_medidas\\_para\\_evaluacion\\_sistemas\\_informacion.html](http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html) .
77. *Ley de la Fiscalía General de la República.*
78. **Gamma.** 1995.
79. **otros, Jacobson y.** 2000.
80. **Roger S. Pressman.** *Ingeniería de Software. Un enfoque práctico.*
81. Buenas tareas. [En línea] <http://www.buenastareas.com/ensayos/Arquitectura-Del-Software/922875.html> .
82. Visual Paradigm. [En línea] <http://www.visual-paradigm.com/aboutus/10reasons.jsp>.
83. **Larman.** 1999.
84. **M.J. Escalona, N. Koch.** *Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo.* España : s.n., 2010.
85. **Frías, Iliana de la Rosa Zayas.** *Análisis del módulo Despacho de Pasajeros, en la Aduana General de la República.* 2010.
86. La Rama Judicial de Puerto Rico. [En línea] <http://www.ramajudicial.pr/orientacion/imp-quejas.html>.



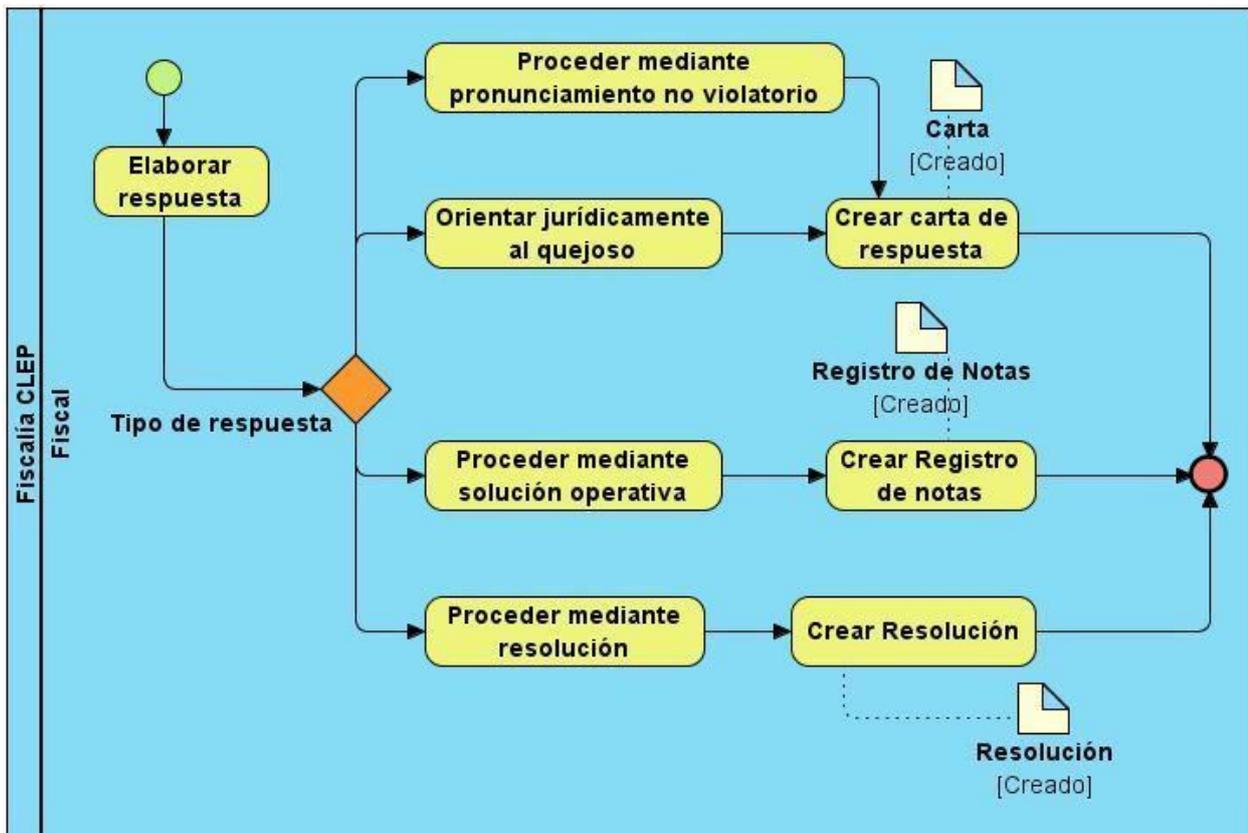
## Anexos

### Anexo # 1. Diagrama de Procesos de Negocio del subproceso Trasladar Queja, Petición o Denuncia.



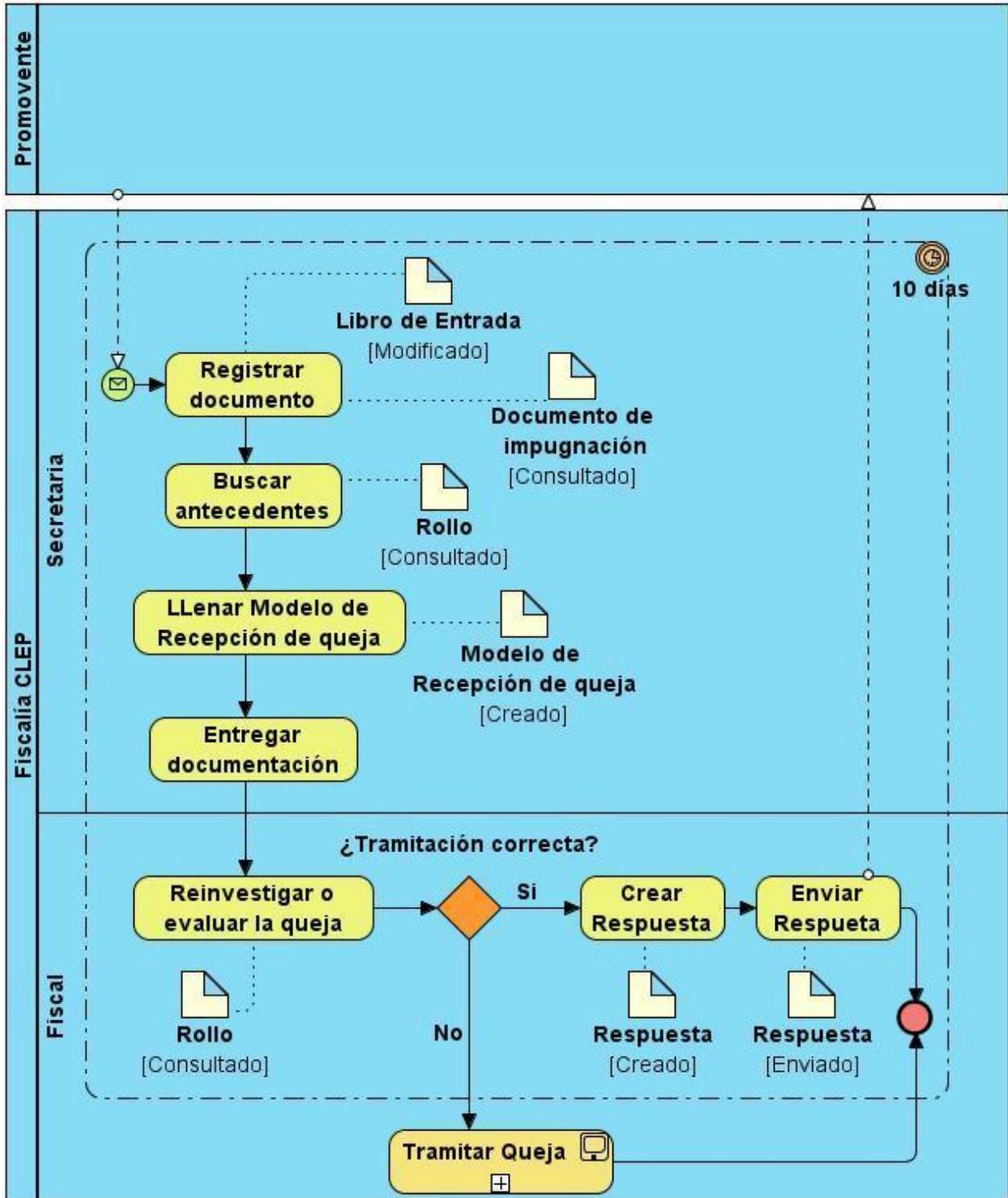


Anexo # 2. Diagrama de Procesos de Negocio del subproceso Concluir Queja, Petición o Denuncia.



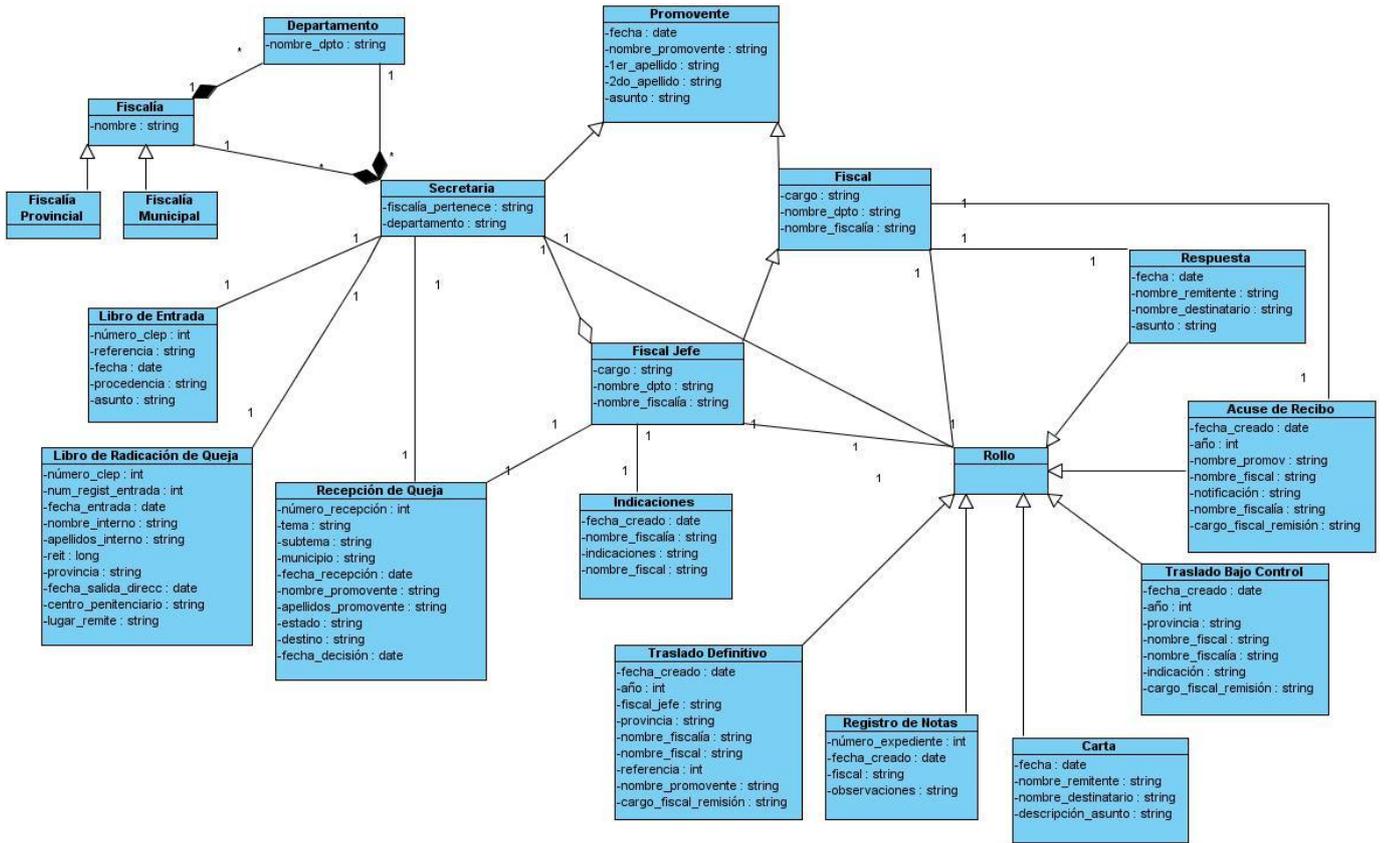


Anexo # 3. Diagrama de Procesos de Negocio del subproceso Tramitar Impugnación.





Anexo # 4. Modelo Conceptual.





Anexo # 5. Acta de aceptación del proceso de negocio Quejas, Peticiones o Denuncias.

**UCI** Universidad de las Ciencias Informáticas

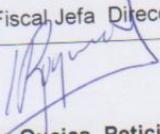
## Acta de aceptación

### ACTA DE ACEPTACIÓN

En cumplimiento del Convenio de colaboración con la Fiscalía General de la República de Cuba y en función de la ejecución del Proyecto: Sistema de Informatización de la Gestión de la Fiscalía (Fase II), se hace entrega del producto que se relaciona a continuación.

Lista de productos que serán aceptados:

- Descripción del proceso de Negocio correspondiente a Quejas, Peticiones y Denuncias del área Control de la Legalidad de los Establecimientos Penitenciarios (incluyen Diagrama de procesos de Negocio).

Entrega	Recibe
<b>Nombre y Apellidos:</b> Yenier Figueroa Machado	<b>Nombre y Apellidos:</b> Hortensia Bonachea Rodríguez
<b>Cargo:</b> Jefe de Proyecto	<b>Cargo:</b> Fiscal Jefa Dirección de CLEP
<b>Firma:</b> 	<b>Firma:</b> 
<b>Comentarios:</b> Una vez examinada la descripción del Negocio Quejas, Peticiones y Denuncias del área Control de la Legalidad de los Establecimientos Penitenciarios, y el Diagrama correspondiente, procede hacer los siguientes señalamientos:	
N/A	

**Representante Parte Suministradora (Centro de Gobierno Electrónico (CEGEL).  
Universidad de las Ciencias Informáticas. La Habana)**

**Nombre y Apellidos:** Yarina Amoroso Fernández

**Cargo:** Directora de Centro

**Firma:**

1



Anexo # 6. Acta de aceptación de los Requisitos del módulo Quejas, Peticiones o Denuncias.



## Acta de aceptación

### ACTA DE ACEPTACIÓN

En cumplimiento del **Convenio de colaboración con la Fiscalía General de la República de Cuba** y en función de la ejecución del **Proyecto: Sistema de Informatización de la Gestión de la Fiscalía (Fase II)**, se hace entrega del producto que se relaciona a continuación.

Lista de productos que serán aceptados:

- Especificación de requisitos de los procesos Quejas, Peticiones y Denuncias de la Dirección de Control de la Legalidad en los Establecimientos Penitenciarios.

**Entrega**

Yenier Figueroa Machado

**Nombre y Apellidos:**

**Cargo:** Jefe de Proyecto

**Firma:**

**Recibe**

Hortensia Bonachea Rodríguez

**Nombre y Apellidos:**

**Cargo:** Fiscal, Dirección de CLEP

**Firma:**

**Comentarios:** Una vez examinada la descripción de los requisitos, se procede hacer los siguientes señalamientos:

**Representante Parte Suministradora (Centro de Gobierno Electrónico (CEGEL).  
Universidad de las Ciencias Informáticas. La Habana)**

**Nombre y Apellidos:** Dailin Benavides Jorge

**Cargo:** Directora de Centro

**Firma:**

Observador independiente ()