

Universidad de las Ciencias Informáticas



Facultad 3

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

Título: Desarrollo del módulo Civil y Familia del Sistema
Informatización de la Gestión de la Fiscalía fase II.

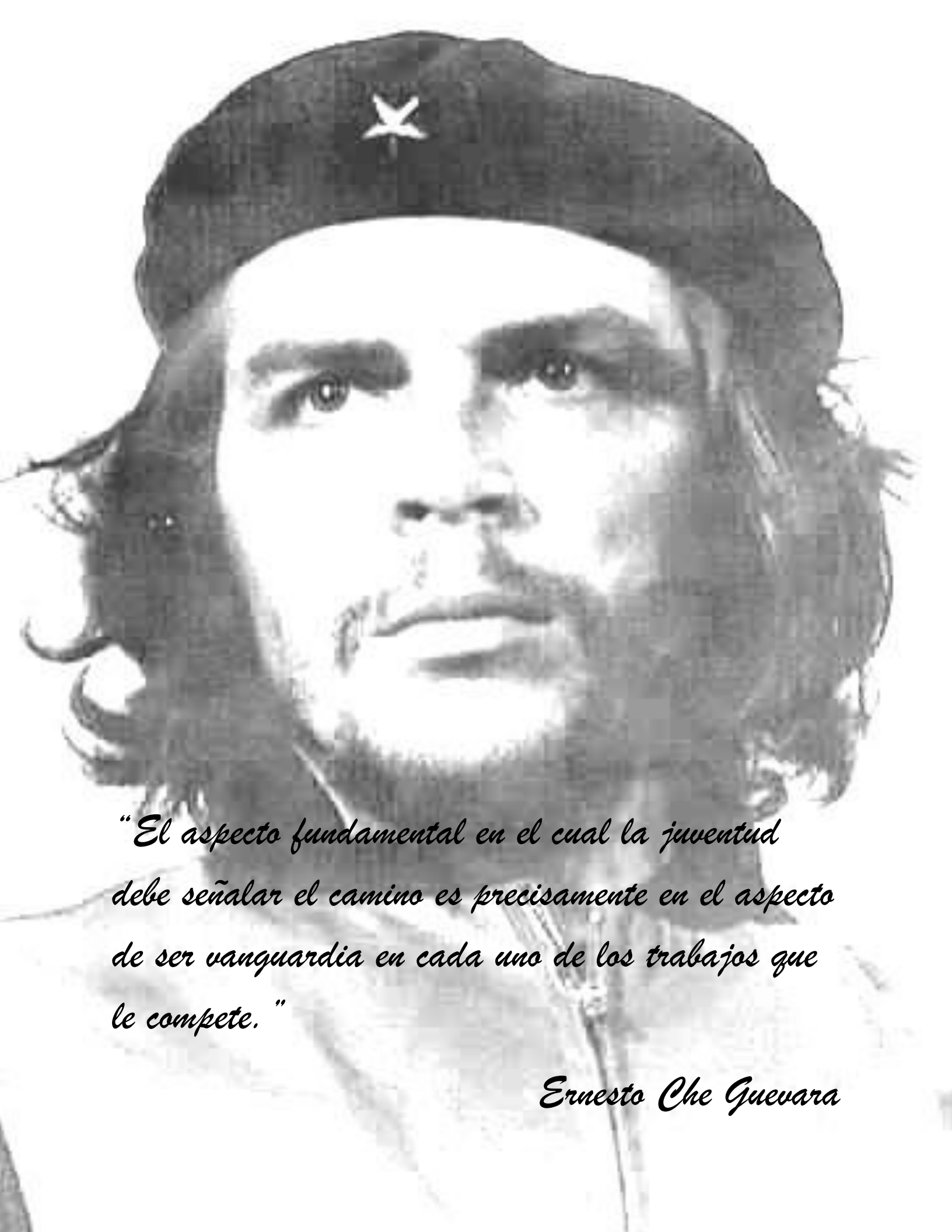
Autores

Daniel Cossio Baró
Luis Angel Pedraye Ochoa

Tutora

Daylenis Ortiz Franco

La Habana, junio de 2013



“El aspecto fundamental en el cual la juventud debe señalar el camino es precisamente en el aspecto de ser vanguardia en cada uno de los trabajos que le compete.”

Ernesto Che Guevara

Declaración de Autoría

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

Daniel Cossio Baró

Firma del autor

Luis Angel Pedraye Ochoa

Firma del autor

Ing. Daylenis Ortiz Franco

Firma de la tutora

Datos de Contacto

Tutora: Ing. Daylenis Ortiz Franco

Correo electrónico: dfranco@uci.cu

Autor: Daniel Cossio Baró

Correo electrónico: dcossio@estudiantes.uci.cu

Autor: Luis Angel Pedraye Ochoa

Correo electrónico: lapedraye@estudiantes.uci.cu

RESUMEN

El propósito fundamental de este trabajo es desarrollar el módulo Civil y Familia del área Protección de los Derechos Ciudadanos, el cual es el encargado de preservar la legalidad respecto al derecho civil, lo que permitirá la informatización de estos importantes procesos dentro de la Fiscalía General de la República. Este trabajo está dividido en tres capítulos para lograr un mejor entendimiento del proceso de desarrollo de software. Entre las principales actividades a realizar se encuentra realizar un análisis sobre la fundamentación teórica, los medios a través de los cuales se llevan a cabo los procesos civiles y familiares, así como el estudio de los lenguajes y herramientas necesarias para desarrollar el software. También se especifican los requisitos funcionales y no funcionales, además, se presentan los diagramas realizados en la disciplina de Diseño y se validan los resultados obtenidos en las disciplinas estudiadas. Finalmente se desarrolla el modelo de implementación y se llevan a cabo las pruebas de caja blanca y caja negra, con el objetivo de lograr la calidad requerida.

Con el desarrollo de esta propuesta se espera que los fiscales tengan a su disposición un sistema que les permita mejorar la realización de los procesos, así como aprovechar el tiempo del que disponen para terminar sus casos y facilitarles el acceso a la información con la que trabajan.

Palabras claves: requisitos, diseño, implementación, procesos civiles y de familia, menor, incapacitado, propietario ausente.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN	1
Capítulo 1. Fundamentación teórica.....	6
1.1 Introducción	6
1.2 Conceptos fundamentales	6
1.3 Comportamiento de los procesos civiles y de familia en Cuba y el mundo.....	7
1.4 Sistemas de gestión en la Informática Jurídica	9
1.5 Modelo de desarrollo de software	11
1.6 Descripción de las herramientas a utilizar	13
1.6.1 Herramienta para el desarrollo de prototipos. Axure 5.5.....	13
1.6.2 Gestor de base de datos. PostgreSQL 9.1	13
1.6.3 Herramientas para el modelado. Visual Paradigm for UML 8.0	14
1.6.4 Lenguaje de Modelado. Unified Modeling Language (UML) 2.0	14
1.6.5 Lenguajes de Programación	15
1.6.6 Entorno de Desarrollo Integrado (IDE). Netbeans 7.3	18
1.6.7 Marco de Trabajo. Symfony 2.1.7	18
1.6.8 Mapeo de objeto relacional (ORM). Doctrine 2.3.2.....	19
1.6.9 Motor de plantillas para PHP. Twig 1.12	20
1.6.10 Herramientas para el control de versiones	20
1.7 Patrones	21
1.7.1 Patrones de arquitectura	21
1.7.2 Patrones de diseño	22
1.8 Métricas	24
1.8.1 Métricas para requisitos	24
1.8.2 Métricas para diseño	25
1.9 Pruebas	26

1.9.1 Pruebas de caja negra	26
1.9.2 Pruebas de caja blanca.....	26
Capítulo 2. Requisitos y Análisis - Diseño.....	28
2.1 Introducción.....	28
2.2 Requisitos.....	28
2.2.1 Técnicas de obtención de requisitos	28
2.2.2 Funcionalidad Adicionar datos del dictamen de Asuntos Notariales.....	30
2.2.3 Técnica de definición de requisitos	31
2.2.5 Requisitos no funcionales	34
2.2.6 Validación de los requisitos.....	35
2.3 Análisis y Diseño	37
2.3.1 Estilos de codificación.....	37
2.3.2 Diagrama de clases persistentes	38
2.3.3 Patrones de diseño empleados	39
2.3.4 Diagrama de secuencia.....	41
2.3.5 Diagrama de clases controladoras y gestoras.....	43
2.3.6 Diagrama de clases repositorio	44
2.3.7 Diagrama de la vista	44
2.3.8 Métricas aplicadas para la validación del diseño	45
Capítulo 3. Implementación y Prueba.	53
3.1 Introducción.....	53
3.2 Implementación	53
3.2.1 Diagrama de componentes	53
3.2.2 Diagrama de despliegue	56
3.3 Pruebas.....	57
3.3.1 Prueba del camino básico	58

3.3.2 Técnica de la partición de equivalencia.....	61
3.3.3 Validación de la propuesta de solución	64
Conclusiones Generales	67
Recomendaciones	68
Bibliografía.....	69
Anexos.....	71
Glosario de términos.....	79

ÍNDICE DE FIGURAS

Figura 1. Ciclo de vida para proyectos.....	12
Figura 2. Modelo conceptual de los procesos civiles y de familia.....	34
Figura 3. Diagrama de clases persistentes para el proceso Asuntos Notariales.....	38
Figura 4. Diagrama de secuencia Adicionar/Actualizar dictamen de Asuntos Notariales.....	42
Figura 5. Diagrama de clases controladoras y gestoras para Asuntos Notariales.	43
Figura 6. Diagrama de clases repositorio para Asuntos Notariales.	44
Figura 7. Diagrama de la vista para Asuntos Notariales.....	45
Figura 8. Resultados obtenidos para el atributo Acoplamiento.....	51
Figura 9. Resultados obtenidos para el atributo Complejidad de mantenimiento.....	51
Figura 10. Resultados obtenidos para el atributo Cantidad de pruebas.	51
Figura 11. Resultados obtenidos para el atributo Reutilización.	52
Figura 12. Diagrama de componentes del módulo Civil y Familia.	55
Figura 13. Diagrama de despliegue del módulo Civil y Familia.	57
Figura 14. Código fuente del método salvarActualizarAsuntosNotarialesAction.....	59
Figura 15. Grafo de flujo de la operación salvarActualizarAsuntosNotarialesAction.	60
Figura 16. Resultados de la aplicación de pruebas unitarias con PHPUnit.....	61

ÍNDICE DE TABLAS

Tabla 1. Especificación del requisito: Adicionar datos del dictamen de AN.	31
Tabla 2. Representación del tamaño operacional de las clases relacionadas con el módulo Civil y Familia.	46
Tabla 3. Resultado de la aplicación de la métrica TOC.	48
Tabla 4. Criterios de evaluación de la métrica AC.	49
Tabla 5. Criterios y categorías obtenidos en la aplicación de la métrica.	50
Tabla 6. Escenario: Adicionar datos del dictamen de AN.	62
Tabla 7. Validación para la variable: control.	65
Tabla 8. Validación para la variable: celeridad.	65

INTRODUCCIÓN

El gran auge que han alcanzado las Tecnologías de la Informática y las Comunicaciones (TIC) en la actualidad ha cambiado los paradigmas y estrategias reconocidos y establecidos por muchos años como válidos. Dentro de las TIC, la industria del software alcanza una posición relevante, por su característica de controlar o hacer accesible, en la mayoría de los casos, los adelantos tecnológicos.

La Universidad de las Ciencias Informáticas (UCI), es uno de los proyectos de la Revolución Cubana y tiene entre sus misiones, informatizar la sociedad, para ello ha creado un grupo de proyectos productivos cuyos resultados evidencian el cumplimiento de tales objetivos. De esta forma, surge el Centro de Gobierno Electrónico (CEGEL), constituido en la facultad 3, el cual tiene la misión de satisfacer necesidades de clientes gubernamentales mediante el desarrollo de productos y servicios de alta confiabilidad, calidad, competitividad, fidelidad y eficiencia, a partir de un personal altamente calificado. Uno de los proyectos que conforman este centro es el Sistema de Informatización de la Gestión de las Fiscalías fase II (SIGEF II), encargado de informatizar los procesos de la Fiscalía General de la República (FGR) a través de una aplicación web que optimice la calidad de la información, la supervisión y el control en tiempo de los procesos fiscales cubanos.

La Fiscalía General de la República es el órgano del Estado al que corresponde, como objetivos fundamentales, 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; y la promoción y el ejercicio de la acción penal pública en representación del Estado. (1)

La FGR está dividida en varias áreas, una de ellas es la de Protección a los Derechos Ciudadanos (PDC) que incluye, entre otros, los procesos civiles y de familia, cuyo objetivo es velar por el cumplimiento de las normas jurídicas y principios que regulan las relaciones personales o patrimoniales entre personas privadas o públicas, tanto naturales como jurídicas, siempre que actúen desprovistas de imperio. (2)

La FGR está dividida en instancias municipales y provinciales. Estas fiscalías constituyen el órgano de dirección del trabajo fiscal en su territorio y su máxima autoridad es el fiscal jefe provincial o municipal correspondiente. (2) Las Fiscalías Municipales en que las necesidades

del servicio así lo determinen, podrán organizarse en secciones, que se integran con varios fiscales, para la atención de esferas de trabajo especializadas. (2)

Un fiscal de Civil y Familia tiene las facultades de asumir la representación de menores de edad, y personas incapacitadas o ausentes que carezcan de representante legal; igualmente tiene participación cuando sus intereses sean contrapuestos a los del menor, el incapacitado o el propietario ausente.

El proyecto SIGEFII está compuesto por subsistemas cuyos módulos corresponden a las áreas representadas en la FGR.

Actualmente la FGR trabaja con el Sistema de Informatización de la Gestión de la Fiscalía fase I, el cual no incluye el área PDC, por lo que no se encuentran informatizados los procesos civiles y de familia, además, los procedimientos para ejecutar los procesos por instancia no se realizan de forma homogénea. Los procesos civiles y de familia generan una alta cantidad de información a procesar, traducida en un gran volumen de documentos a desarrollar de forma manual por los fiscales, a los que se les dedica considerable tiempo y esfuerzo. La documentación y notificaciones de cada instancia se envían por correo electrónico o postal, y los documentos que se considera ya no mantienen relevancia, pasados los cinco años de permanencia en la fiscalía, son destruidos. Esta forma de trabajar provoca: que las consultas a los documentos o rollos legales archivados no sea cómoda para los fiscales, aun cuando los archivos no llegan a colapsar; que sea imposible acceder a un rollo o documento en tiempos posteriores a los 5 años de su creación; que sea complejo controlar y supervisar los procesos en las instancias provinciales y municipales a la hora de obtener reportes estadísticos sobre sus procesos; y que los fiscales deban estar pendientes al correo, verificando el tiempo que les queda para culminar cada caso, porque existen términos que no pueden ser violados y no cuentan con sistemas de alertas que les recuerden las fechas límites.

A partir de la situación problemática identificada se plantea el siguiente **problema a resolver**:

¿Cómo mejorar la gestión del proceso Civil y Familia del área Protección de los Derechos Ciudadanos de la Fiscalía General de la República, de manera que se contribuya al control y celeridad del mismo?

Mejora de la gestión del proceso: se mide a través del control de los procesos y la celeridad en la construcción de los reportes y documentos.

Control: se mide a través de la verificación de los rollos creados en un período, mediante los reportes estadísticos; con la visualización de los rollos creados a diario, a través de los detalles; con el control de los datos que se adicionan al proceso y con la persistencia de la información en períodos superiores a los 5 años.

Celeridad: disminución del tiempo de realización de los reportes, del acceso a estos y de la creación de los documentos, que convergen en mayor velocidad en la toma de decisiones. También se mide a través del acceso a la información, mediante las búsquedas.

Se plantea como **objeto de estudio** el proceso de desarrollo de software de gestión en la informática jurídica.

Para dar respuesta al problema propuesto se traza como **objetivo general**, desarrollar el módulo Civil y Familia del Sistema de Informatización de la Gestión de las Fiscalías fase II, para mejorar la gestión del proceso.

Del objetivo general se desglosan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación mediante un estudio del estado del arte alrededor de los procesos fiscales, técnicas y herramientas de software.
- Realizar la descripción de los requisitos y el diseño del módulo Civil y Familia del Sistema de Informatización de la Gestión de la Fiscalía fase II, mediante el modelo de desarrollo de software establecido.
- Implementar el módulo Civil y Familia del Sistema de Informatización de la Gestión de la Fiscalía fase II, mediante las herramientas y tecnologías seleccionadas.
- Validar los requisitos, el diseño y la implementación de la solución propuesta mediante métricas y pruebas de software.

El objeto de estudio está enmarcado en el **campo de acción** proceso de desarrollo de software de gestión en los procesos fiscales.

La **idea a defender** de la investigación está basada en que si se desarrolla el módulo Civil y Familia del Sistema de Informatización de la Gestión de la Fiscalía fase II, se mejora la gestión del proceso contribuyendo al control y celeridad del mismo.

Como guía a seguir para dar cumplimiento a los objetivos específicos se definen las siguientes **tareas de la investigación** para llevar a cabo el desarrollo del módulo Civil y Familia del Sistema de Informatización de la Gestión de las Fiscalías fase II:

- Análisis de los procesos civiles y de familia del área Protección de los Derechos Ciudadanos.
- Estudio del estado del arte de sistemas de gestión de informática jurídica.
- Estudio del proceso de desarrollo de software que se lleva a cabo en el proyecto.
- Identificación y descripción de los requisitos asociados al módulo.
- Validación de los requisitos asociados al módulo.
- Realización de los diagramas de clases y de secuencia del diseño.
- Validación del diseño desarrollado.
- Implementación del módulo.
- Validación de los resultados obtenidos a través de pruebas al sistema para comprobar su correcto funcionamiento.
- Validación de la propuesta de solución de la investigación frente al problema planteado.

Métodos Teóricos

Histórico-lógico: se emplea para analizar la trayectoria y evolución del diseño, patrones de diseño y metodologías de desarrollo de software, para el desarrollo de los artefactos que proponen los flujos estudiados.

Analítico-sintético: permite el procesamiento de la información y arribar a las conclusiones prácticas y teóricas de la investigación, así como precisar las relaciones para la utilización de artefactos propuestos.

Métodos Empíricos

Entrevista: permite obtener información valiosa sobre las necesidades del cliente para identificar los requisitos en el desarrollo del software.

Estructura del documento

Capítulo 1. Fundamentación teórica. Ofrece el análisis de los sistemas de gestión fiscal desarrollados en la actualidad, además del estudio de los lenguajes y herramientas usadas como apoyo para darle solución al problema planteado.

Capítulo 2. Requisitos y Análisis - Diseño. Se especifican los requisitos funcionales y no funcionales, teniendo en cuenta las restricciones o políticas a cumplir por el sistema. Además se presenta el diseño a través de los diagramas de clases y de interacción, haciendo uso de patrones de diseño. También se validan los requisitos y el diseño a través de técnicas de validación de requisitos y métricas de diseño.

Capítulo 3. Implementación y Prueba. Se desarrolla el modelo de implementación, que incluye las interfaces del sistema, el diagrama de despliegue y el diagrama de componentes, además de la descripción detallada de los paquetes de implementación. Igualmente se lleva a cabo la validación de la implementación a través de pruebas de caja blanca y caja negra que garanticen que el sistema implementado satisface las necesidades del cliente.

Capítulo 1. Fundamentación teórica.

1.1 Introducción

El presente capítulo ofrece un análisis de los sistemas que se encuentran actualmente en el mundo, dedicados al trabajo con los procesos civiles y de familia, a través del cual se toman los elementos que puedan ser de utilidad. Se analizan las herramientas y lenguajes que se utilizarán durante el trabajo. Se presentan los patrones de arquitectura y de diseño a utilizar, así como las métricas para requisitos y diseño, además de las pruebas a emplear.

1.2 Conceptos fundamentales

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

Derecho Civil: es el conjunto de normas jurídicas y principios que regulan las relaciones personales o patrimoniales entre personas privadas o públicas, tanto físicas como jurídicas, de carácter privado y público, siempre que actúen desprovistas de imperio. (2)

El módulo Civil y Familia del subsistema PDC de SIGEF II está compuesto por siete procesos:

Jurisdicción Voluntaria: es la prevención de la Litis. (2)

Revisión Civil: es la revisión de una sentencia firme dictada en otro proceso, por la existencia de causas que la hacen injusta. (2)

Asuntos Notariales: es la protección de la libertad individual en las relaciones personales, patrimoniales y de familia. (2)

Recursos de Apelación y Casación: son recursos que constituyen parte del proceso Civil y Familia, es decir, son una garantía que integran el proceso en caso de inconformidades. (2)

Expropiación Forzosa: se realiza cuando por no existir acuerdo entre la administración y el particular dueño de los bienes de que se trate, se haga necesaria la intervención judicial. (2)

Proceso Contencioso: aquel momento procesal que se produce cuando se entabla la demanda, se traslada la misma al demandado y éste la contesta, fijándose a partir de ese momento los términos del reclamo y las excepciones y defensas opuestas, sobre lo que se producirán prueba si resulta pertinente y sobre lo que el juez deberá decidir. (2)

En Cuba las fiscalías se encuentran organizadas por:

Fiscalía General de la República: es el órgano del Estado al que corresponde, como objetivos fundamentales, 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; y la promoción y el ejercicio de la acción penal pública en representación del Estado. (2)

Fiscalía Provincial: es el órgano de dirección del trabajo fiscal en su territorio, a la cual corresponde además el establecimiento y desarrollo de la colaboración con otras entidades. Está integrada por el fiscal jefe provincial, quien es su máxima autoridad, los vice fiscales jefes provinciales y por varios departamentos, entre los cuales se encuentra el de Protección de los Derechos Ciudadanos. (2)

Fiscalía Municipal: es el órgano de dirección del trabajo fiscal en su territorio, a la cual corresponde además el establecimiento y desarrollo de la colaboración con otras entidades. Está integrada por el fiscal jefe municipal, quien es su máxima autoridad y en su caso por el vice fiscal jefe municipal, los fiscales y el personal administrativo. Las Fiscalías Municipales en que las necesidades del servicio así lo determinen, podrán organizarse en secciones, que se integran con varios fiscales, para la atención de esferas de trabajo especializadas. (2)

1.3 Comportamiento de los procesos civiles y de familia en Cuba y el mundo

El proceso Civil y Familia en Cuba tiene particularidades que lo hacen único, por ejemplo: la jurisdicción civil se ejerce exclusivamente por el Tribunal Municipal Popular y por las salas de lo civil y de lo administrativo del Tribunal Provincial Popular y del Tribunal Supremo Popular. (3) No obstante, la responsabilidad civil derivada de la comisión de delito se reclama conjuntamente con la penal ante los tribunales competentes.

El Código Civil (CC) se inspira en la realidad nacional y en la práctica social de la Revolución, y se aparta de los postulados individualistas que habían caracterizado al derecho civil anteriormente, cada vez más incompatible con el régimen socialista. Refleja los nuevos principios sobre la intervención estatal en las relaciones entre las personas, para tutelar sus intereses en armonía con la conveniencia social. (3)

Las relaciones patrimoniales o vinculadas a ellas que regula el código se establecen entre personas situadas en plano de igualdad, es decir, que le son ajenas las relaciones jurídicas que implican autoridad y subordinación; aunque la voluntad del Estado revolucionario también se revela en las relaciones civiles al imponerles limitaciones como las que prohíben el abuso de los derechos subjetivos que deben estar en todo caso en armonía con los intereses sociales.

En el mundo los procesos civiles cumplen con las mismas funciones, sólo se diferencian en la forma en que son procesados de acuerdo a las características particulares de cada país. También es perceptible que no son encontrados en el resto del mundo los procesos relacionados con la Expropiación Forzosa, pues este se dirige contra los titulares de derechos o intereses económicos directos sobre el bien expropiado, y contra los poseedores y ocupantes legítimos. Esta facultad es particular del Estado cubano. En el caso de que los propietarios o titulares no fueren habidos, o fueren menores o incapacitados sin persona que los representen, el procedimiento se entenderá con el fiscal. (2)

Para Nicaragua, los juicios y sus incidentes tramitados de conformidad con las leyes de los otros estados de América Central, producirán sus efectos en Nicaragua en los casos determinados por la ley. Nicaragua reconoce que las leyes de un estado centroamericano en que un tribunal tiene su asiento, determinan la admisión, apreciación y efectos de la prueba, donde el que apoye su derecho en leyes extranjeras, debe comprobar su existencia en forma auténtica. Además, no se utiliza el término “tutela” respecto a los menores e incapacitados, dentro de la Jurisdicción Voluntaria, sino “guarda”. El juez competente a cuyo conocimiento llegue que alguna persona ha sido declarada por sentencia firme incapacitada para administrar sus bienes, le nombrará guardador, encabezando el expediente con testimonio de dicha sentencia. Los jueces darán a los menores o incapacitados guardadores especiales cuando entre quienes se empeñe la cuestión de competencia tuvieren un superior común. (4)

México tiene disposiciones especiales en su CC que lo hace único, por ejemplo: la parte que pierde debe reembolsar a su contraria las costas del proceso. Las costas del proceso consisten en la suma que, según la apreciación del tribunal y de acuerdo con las disposiciones arancelarias, debió o habría debido desembolsar la parte triunfadora, excluido el gasto de todo acto y forma de defensa considerados superfluos. También incluye elementos como la revisión forzosa, la cual es establecida por la ley respecto de algunas resoluciones judiciales, y tendrá por objeto estudiar el negocio en su integridad, a no ser que la misma ley la restrinja a puntos determinados, para el efecto de confirmar, reformar o revocar la sentencia del inferior. (5)

1.4 Sistemas de gestión en la Informática Jurídica

Fueron analizados diferentes sistemas de gestión de procesos fiscales existentes en el mundo, de los cuales, fueron objeto de análisis los que mostraron características similares a los procesos civiles cubanos.

Sistema Automatizado de Gestión Judicial (SAGJ)

El objetivo primordial de la implementación de este nuevo sistema informático es modernizar al Órgano Judicial, incorporando tecnologías de información y comunicaciones que permitan ofrecer la justicia gratuita, expedita e ininterrumpida que preconiza la Constitución Política de la República de Panamá. (6)

La ventaja con la utilización del SAGJ es que facilita la labor de los juzgadores y por lo tanto, la administración de justicia será más expedita.

Poder Judicial Provincia de San Juan-República Argentina

Sistema de gestión de información para diversos organismos, que permiten las tareas de alta, baja, modificación y consulta de los datos respectivos de cada área específica, pudiendo visualizar asimismo reportes por pantalla u obtener listados clasificados por distintos criterios.

Se encuentra informatizada con red y software de gestión la Corte de Justicia. Dicha red permite el flujo de información entre los ministros y secretarios del tribunal, respetando pautas de seguridad y confidencialidad de la información, estipuladas oportunamente. (7)

Se detallan a continuación aquellos sistemas que se encuentran en operación:

- Mesa de Entradas de la Corte de Justicia Salas I, II y III
- Registro de Juicios Universales (Sucesiones y Concursos)
- Personal
- Mesa de Entradas Única Fuero Civil 1º Instancia
- Mesa de Entradas Única Fuero Comercial 1º Instancia
- Mesa de Entradas Única Fuero Familia 1º Instancia
- Mesa de Entradas Única Fuero Laboral 1º Instancia
- Mesa de Entradas Única Cámara Laboral
- Mesa de Entradas Única Cámara Civil (7)

Fortuny

Esta herramienta informática fue desarrollada en 2003 por el Ministerio de Justicia en colaboración con la Fiscalía General del Estado para facilitar la gestión de los procesos de la jurisdicción civil, penal, contencioso-administrativo y social, además de los asuntos de vigilancia penitenciaria e incapacidades de las fiscalías. Actualmente está siendo utilizada en Aragón, Asturias, Extremadura, Castilla-León, Castilla La Mancha, Murcia, Cantabria, La Rioja y Andalucía. (8)

Hasta el año 2010 se han instalado ocho nuevas versiones de Fortuny, en las que los elementos más relevantes han sido los siguientes:

- La creación de una figura de administrador en cada fiscalía que potencie la autonomía de uso y solución de problemas.
- La instalación de un nuevo módulo para la jurisdicción civil, que viene a sustituir al anterior.
- Mejora de las estadísticas anuales, con incorporación de módulos para las especialidades de violencia de género, violencia doméstica, siniestralidad laboral y seguridad vial.
- La creación de un módulo de intercambio con Minerva de los registros de los procedimientos competencia de los juzgados de instrucción.
- Adaptación del registro a la reforma del Código Penal, que entró en vigor en el mes de diciembre de 2010.
- Creación de un módulo específico para el uso de Fortuny por parte de la Fiscalía de la Audiencia Nacional.
- Creación de módulos específicos para el uso de Fortuny por las Fiscalías de Comunidad Autónoma, especialmente las desdobladas. (8)

Luego de analizar algunas de las soluciones informáticas desarrolladas en el mundo se puede concluir que dichos sistemas no cumplen con las características necesitadas, debido a que en Cuba el uso del software libre se está tomando como alternativa de acuerdo a las políticas de uso de la tecnología y los sistemas analizados son privativos. Además, en el caso del Poder Judicial Provincia de San Juan-República Argentina, incluye sistemas en activo que tratan temas civiles y familiares pero de forma separada, también su enfoque es hacia los tribunales.

En el caso de Fortuny sí facilita la gestión de los procesos judiciales a la Fiscalía, pero su funcionamiento y el tratamiento de la ley es diferente, además de que las herramientas con que fue construido son privativas. Por estas razones estos sistemas no constituyen una solución factible respecto a las necesidades de la dirección de PDC de la FGR, por lo que se hace necesario incorporar el proceso Civil y Familia como nueva funcionalidad del sistema que se está implementando para la informatización de los procesos de la FGR, y no como una solución aislada, que permita la correcta interacción con todos los componentes de la aplicación e interoperabilidad con los demás sistemas de gestión existentes.

1.5 Modelo de desarrollo de software

En el contexto de la ingeniería de software, un proceso no es una receta rígida de cómo construir software. Se trata de un enfoque adaptable que permite a las personas hacer el trabajo (el equipo de software) para elegir el conjunto adecuado de acciones de trabajo y tareas. La intención es siempre la de ofrecer software de una manera oportuna y con la calidad suficiente para satisfacer a aquellos que han auspiciado su creación y a aquellos usuarios finales. (9)

A esta afirmación se adecua un estudio realizado en 114 empresas de desarrollo de software en México que reporta que existe un 29% que no usa modelos o metodologías, porque consideran el desarrollo de software como un proceso creativo que no se puede ajustar a una metodología, mientras que existe un 71% que sí hacen uso de un proceso o metodología que guíen el desarrollo de sus software. De ellos un 45.41% apuestan por metodologías propias, un 41.25% desarrollan bajo metodologías ágiles, mientras que un 13.34%, hacen uso de modelos y normas establecidas como: CMM (Capability Maturity Model for Software), CMMI (Capability Maturity Model Integration), ISO: 9000:2000 (Sistema de Gestión de la Calidad) y PMBOK (Project Management Institute). (10)

Teniendo en cuenta lo anterior y las características del proyecto, se hace uso del modelo de desarrollo que propone el centro CEIGE (Centro de Informatización de la Gestión de Entidades) que constituye un caso de éxito por su aplicación en los 12 proyectos que lo componen. Al igual que las metodologías, propone actividades, roles y artefactos en cada disciplina. Este modelo incorpora los distintos subprocesos dictados por el nivel dos de CMMI establecidos en el Programa de mejora propuesto en la UCI. El centro obtuvo la certificación de CMMI nivel 2 en julio de 2011, reconocida por el SEI (Software Engineering Institute) como aval de la calidad de su proceso de desarrollo. (11)

Las fases que propone este modelo son:



Figura 1. Ciclo de vida para proyectos.

El desarrollo de este trabajo se centrará en las fases de Requisitos, Análisis y Diseño, Implementación, Pruebas internas y las Pruebas de liberación. En la fase de Requisitos se realizarán las actividades necesarias para la identificación, análisis, especificación y validación. Se obtendrá como resultado el documento de especificación de requisitos de software, así como los prototipos no funcionales de interfaz de usuario. En la fase de Análisis y Diseño, se obtendrán los diagramas de clases (persistentes, controladoras y gestoras, repositorio, vista), además de los diagramas de secuencia por cada requisito identificado. En la fase de Implementación se desarrollarán el diagrama de componente y el diagrama de despliegue, además de los ficheros de código. En las fases de Pruebas internas y de liberación, se obtendrá el documento de no conformidades. (12)

Se definió el avance del proyecto sobre la plataforma de desarrollo Linux Apache Postgres PHP, la cual se selecciona tomando como punto de partida los requisitos no funcionales, de manera que se tribute a la política de migración hacia software libre en la que está inmersa la

FGR y todo el país, en la continua búsqueda de la soberanía tecnológica. (12) Las herramientas y otras tecnologías definidas por el equipo de arquitectura del proyecto serán descritas en los epígrafes subsiguientes.

1.6 Descripción de las herramientas a utilizar

1.6.1 Herramienta para el desarrollo de prototipos. Axure 5.5

Es una herramienta que permite desarrollar rápidamente prototipos y especificaciones de software, encaminada a las aplicaciones web. Su grado de especialización es en las anotaciones, permite especificar el estado de cada elemento y el beneficio esperado, el riesgo, la estabilidad, a quién va dirigido y a quién se le asigna la tarea. (13)

El uso de esta herramienta proporciona:

- Construcción de los prototipos de interfaz de usuario.
- Soporte de todos los elementos comunes de prototipos (listas de selección, casillas de texto, entre otras).
- Navegación entre las páginas.
- Generación de prototipos en HTML.
- Generación de especificaciones en DOC. (13)

Esta herramienta es de gran importancia para el proyecto porque 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, y con ello se pueden aportar elementos significativos en la toma de decisiones para lograr la calidad requerida. Además brinda la posibilidad de que el cliente se vaya familiarizando con la aplicación con la cual trabajará en un futuro.

1.6.2 Gestor de base de datos. PostgreSQL 9.1

PostgreSQL es un sistema gestor de base de datos de código abierto, con una comunidad mundial de miles de usuarios y contribuyentes, y docenas de empresas y organizaciones. Además, el conjunto de características que tiene PostgreSQL no sólo rivaliza con sistemas de bases de datos propietarias, sino que los supera en características avanzadas, extensibilidad, seguridad y estabilidad. Esta última actualización de la base de datos de código abierto ofrece

tecnología innovadora, mayor extensibilidad, y nuevas características como replicación sincrónica, método de indexado "K-Nearest Neighbor"(índices sobre "distancia" para consultas rápidas de ubicación y búsquedas de texto), y conectores de datos foráneos. (14)

Dado el prestigio con que cuenta PostgreSQL como sistema gestor de base de datos, su nivel de aceptación entre grandes empresas y sus características, que satisfacen las necesidades de los clientes, fue seleccionado como sistema gestor de base de datos. También le favorece que sea una herramienta de código abierto que brinda estabilidad y confiabilidad al trabajo con la base de datos. Por estas razones se decide utilizar PostgreSQL.

1.6.3 Herramientas para el modelado. Visual Paradigm for UML 8.0

Visual Paradigm es una herramienta CASE (Computer Aided Software Engineering) multiplataforma diseñada para la ayuda al desarrollo de software, porque soporta el ciclo de vida completo del mismo. Permite a los analistas visualizar el flujo central y detallado de cada proceso mediante diagramas. Los diseñadores pueden diseñar una parte del sistema, así como identificar y definir las responsabilidades, operaciones, atributos y relaciones. (15) En cuanto a los programadores, ya cuentan con la estructura base sobre la cual se va a desarrollar el trabajo.

Además de las ventajas que brinda el Visual Paradigm como herramienta, en la universidad se cuenta con la licencia para su uso y está incluida como propuesta del Programa de mejora. Posee todos los recursos para integrarse tanto con el IDE de desarrollo escogido, como con el sistema gestor de base de datos a utilizar.

1.6.4 Lenguaje de modelado. Unified Modeling Language (UML) 2.0

UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Desde su creación se ha convertido en el estándar internacional para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos. Con este lenguaje, se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.(16)

La clave está en 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. El UML proporciona tal organización.

Otra característica del desarrollo de sistemas contemporáneo es reducir el período de desarrollo. Cuando los plazos se encuentran muy cerca uno del otro, es absolutamente necesario contar con un diseño sólido.

La necesidad de diseños sólidos ha traído consigo la creación de una notación de diseño que los analistas, desarrolladores y clientes acepten. Por tanto, en el proyecto se decide emplear esta notación, que fue creada para satisfacer esta necesidad. (16)

1.6.5 Lenguajes de Programación

1.6.5.1 Lenguajes del lado del cliente:

HTML 5. (HyperText Markup Language, versión 5)

HTML 5 pretende proporcionar una plataforma con la que desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores.

Incluye novedades significativas en diversos ámbitos. No sólo se 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. (17) Entre ellas se encuentran:

- Estructura del cuerpo: la mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- Etiquetas para contenido específico: hasta ahora se utilizaba una única etiqueta para incorporar diversos tipos de contenido enriquecido, como animaciones flash o vídeo. Ahora se utilizarán etiquetas específicas para cada tipo de contenido en particular, como audio, vídeo, etc.
- Bases de datos locales: el navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del cliente y a través de un API, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectados a internet.

- Fin de las etiquetas de presentación: todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS. (17)

Es empleado en el proyecto porque establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. También es eficaz sobre los navegadores más utilizados.

CSS 3. Cascading Style Sheets, versión 3

Ofrece nuevas posibilidades para crear impacto con sus diseños, permite el uso de hojas de estilo más diversas para variedades de ocasiones, entre otras características que harán la aplicación más amigable para los usuarios.

La novedad más importante que aporta CSS 3, de cara a los desarrolladores de webs, consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos, que a menudo complicaban el código de las web.

Entre las principales características de CSS3 se encuentran: bordes redondeados, con degradados, imágenes; cajas con sombra; múltiples imágenes de fondo; múltiples columnas; texto con sombra; cajas redimensionables. También incluye mejoras en el uso de la opacidad (pudiéndose fijar a una caja, imagen o texto), selectores y personalizado de fuentes. (18)

Ajax. Asynchronous JavaScript and XML

Ajax no es un nuevo lenguaje de programación, sino una nueva forma de utilizar las normas existentes. Es el arte de intercambiar datos con el servidor y actualizar partes de una página web sin tener que recargar toda la página. Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas para mejorar su uso. (19)

Es importante para el proyecto pues constituye una técnica de desarrollo web para crear aplicaciones interactivas de mucha utilidad para los clientes. Facilita el trabajo y la forma en que se muestra la información es más flexible. También permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página.

Las tecnologías que forman Ajax son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías. (19)

JQuery

JQuery es una biblioteca de JavaScript rápida y concisa que simplifica el recorrido del documento HTML, manejo de eventos, animación, y las interacciones Ajax para el desarrollo web rápido. La gran ventaja de la función de jQuery, es que la página se puede manipular en cuanto se ha cargado su código HTML, mientras que la función de JavaScript espera a que se carguen todos los elementos de la página, incluyendo todas las imágenes. De esta forma, las aplicaciones realizadas con jQuery pueden responder de forma mucho más rápida que las aplicaciones JavaScript tradicionales. (20) Es empleado en el proyecto junto con Ajax para el manejo de las funciones JavaScript y html dinámico.

1.6.5.2 Lenguaje del lado del servidor:

PHP 5.3.8

PHP 5.3.8 supone un lanzamiento de una versión con importantes novedades y nuevas características del lenguaje de programación de libre licencia empleado en la creación de páginas web dinámicas. Entre sus principales características se encuentran:

- Es un lenguaje multiplataforma.
- Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Permite las técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

- Incorpora funciones anónimas y namespaces.
- Se ha añadido soporte para espacios de nombres.
- Se ha añadido soporte para enlaces estáticos en tiempo de ejecución.
- Ahora es posible el acceso dinámico a métodos estáticos.
- Ahora se pueden anidar excepciones.
- Se ha añadido un recolector de basuras para referencias circulares, y está activado por omisión.
- La función mail () ahora soporta registros de correos enviados. (21)

Además de las ventajas que posee este lenguaje de programación, se decide utilizarlo porque en la fase 1 de esta aplicación se trabajó con el mismo, y con ello se logra que no exista algún tipo de incompatibilidad entre ambas aplicaciones.

1.6.6 Entorno de Desarrollo Integrado (IDE). Netbeans 7.3

NetBeans IDE es una herramienta de código abierto diseñada para el desarrollo de aplicaciones. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web y control de versiones. Ofrece un rendimiento significativamente mejorado y experiencia de codificación. También incluye características notables, como el soporte para múltiples marcos de trabajo PHP. (22) Netbeans 7.3 es una herramienta que tiene fuerte integración con PHP y demás herramientas a utilizar en el proyecto, es libre y bien ajustado a las condiciones de desarrollo del centro.

Esta versión del netbeans permite la integración con Symfony 2.0, lo que permite agilizar el proceso de desarrollo, debido a que no es necesario utilizar una consola externa para trabajar con dicho framework (marco de trabajo). Existe gran conocimiento de su uso por parte del equipo de desarrollo, lo que garantiza mayor productividad. Es por esto que se decide utilizar en el proyecto este IDE, debido a que la utilización de otro implicaría tiempo en la capacitación y adaptabilidad del equipo de desarrollo sobre la nueva herramienta, lo que trae consigo cambios en el cronograma de trabajo.

1.6.7 Marco de trabajo. Symfony 2.1.7

Symfony es un marco de trabajo que utiliza PHP y emplea la arquitectura MVC (Model-View-Controller). Se utiliza para crear aplicaciones PHP, permitiendo aumentar la productividad y

calidad de trabajo. Es maduro para el desarrollo web, ha alcanzado su versión 2.1.7, y su objetivo es rapidez y flexibilidad. (23)

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 emplea ideas de otros marcos de trabajo.
- Flexible pues cada programador puede usar lo que desee.

Se selecciona en el proyecto porque entre sus ventajas se destaca sobre todo, la independencia de módulos. El propio núcleo 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. Es un marco de trabajo muy documentado y se publica bajo licencia MIT22, con la que se puede desarrollar aplicaciones web comerciales, gratuitas y/o de software libre. (23)

1.6.8 Mapeo de objeto relacional (ORM). Doctrine 2.3.2

Doctrine 2 es un ORM para PHP 5.3.3 + que proporciona persistencia transparente de objetos PHP. Se utiliza el patrón Data Mapper en el centro, permitiendo una completa separación entre la lógica de dominio/negocio y la persistencia de un sistema de gestión de base de datos relacional. El beneficio de este ORM para el programador es la capacidad de centrarse en la lógica de negocio orientado a objetos y preocuparse por la persistencia sólo como un problema secundario. Esto no significa que la persistencia sea minimizada por Doctrine 2, sin embargo, existen considerables beneficios para la programación orientada a objetos, si se mantienen la persistencia y entidades separadas. (24)

El marco de trabajo Symfony 2 utiliza Doctrine, el cual se sitúa en la parte superior de una poderosa capa de abstracción de base de datos (DBAL por Database Abstraction Layer). Entre sus puntos fuertes destaca su lenguaje DQL (Doctrine Query Language) que está inspirado en el HQL de Hibernate. Para crear el modelo, Doctrine permite dos alternativa, hacer una clase por tabla e indicarle mediante PHP el tipo de datos que se almacena. O bien utilizar un esquema en formato YAML (similar a XML, pero más legible para las personas) (25)

1.6.9 Motor de plantillas para PHP. Twig 1.12

Twig que es un motor de plantillas para PHP muy rápido y eficiente, recomendado por Symfony 2 para crear todas las plantillas de la aplicación, las cuales tienen una sintaxis concisa y muy fáciles de leer y de escribir. Sus principales características son:

- Rápido: Twig compila las plantillas hasta código PHP regular optimizado. El costo general en comparación con código PHP regular se ha reducido al mínimo.
- Seguro: Twig tiene un modo de recinto de seguridad para evaluar el código de plantilla que no es confiable. Esto te permite utilizar Twig como un lenguaje de plantillas para aplicaciones donde los usuarios pueden modificar el diseño de la plantilla.
- Flexible: Twig es alimentado por flexibles analizadores léxico y sintáctico. Esto permite al desarrollador definir sus propias etiquetas y filtros personalizados. (26)

Trae un nuevo enfoque para el uso de computadoras. La información se encuentra organizada de tal manera que sea significativa y útil. Permite realizar más tareas con menos pasos, golpes de teclado y los clics del ratón que con cualquier otra forma. De forma general es simple, brinda apoyo y es eficiente. (27)

Twig proporciona para el proyecto todas las herramientas que se puedan necesitar para organizar la información. Entre sus características más útiles: listas que pueden ser priorizadas, notas, twigboard y varios diarios.

1.6.10 Herramientas para el control de versiones

1.6.10.1 TortoiseSVN

Dada la necesidad de una herramienta que permita tener controlada la documentación digital, gestionar la seguridad y la preservación de esta, se decidió escoger TortoiseSVN, ya que una de sus ventajas es que constituye software libre bajo la licencia GNU GPL (GNU General Public License).

Es un cliente gratuito de código abierto para el sistema de control de versiones Apache Subversion. Esto significa que TortoiseSVN administra archivos y directorios a lo largo del tiempo. Los archivos se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de archivos ordinario, con la excepción de que recuerda todos los cambios que se hayan hecho a sus archivos y directorios. Esto le permite al usuario recuperar

versiones antiguas de sus archivos y examinar la historia de cómo y cuándo cambiaron sus datos, y quién hizo el cambio. (28)

1.6.10.2 Subversion

Es un sistema de control de versiones de código abierto. Fundada en 2000 por CollabNet, Inc., el proyecto Subversion y software ha logrado cierta credibilidad en la última década. Subversion ha tenido una amplia adopción, tanto en el campo de código abierto como en el mundo empresarial. Es desarrollado como un proyecto de la Apache Software Foundation, y como tal, forma parte de una rica comunidad de desarrolladores y usuarios. (29)

Subversion existe para ser universalmente reconocido y adoptado como un sistema de control de versiones centralizado y de código abierto, esta es una de las necesidades del proyecto, que se caracteriza por su fiabilidad como un refugio seguro para los datos valiosos. Es de gran utilidad por la simplicidad en el uso de su modelo, y su capacidad para apoyar las necesidades de una amplia variedad de usuarios y proyectos, desde particulares a las operaciones empresariales a gran escala.

1.7 Patrones

Un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. Los patrones son simplemente una manera de resolver problemas del desarrollo del software, fruto de la experiencia, y por esa misma razón, haber llegado a soluciones similares en algunos casos. (30)

1.7.1 Patrones de arquitectura

Resuelven problemas arquitectónicos, adaptabilidad a requerimientos cambiantes, performance, modularidad y acoplamiento. La solución que plantea es la creación de patrones de llamadas entre objetos, decisiones y criterios arquitectónicos, empaquetado de funcionalidad. Los beneficios de la utilización de dichos patrones van desde la imposición de decisiones tempranas en el desarrollo hasta la reutilización.

Dan una descripción de los elementos y el tipo de relación que tienen, con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus

responsabilidades e interrelaciones y una serie de recomendaciones para organizar los distintos componentes. (30)

Patrón Modelo-Vista-Controlador (MVC).

Separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es la capa del dominio y el controlador representa la lógica de negocio.

- Modelo: esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.
- Vista: este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- Controlador: responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. (30)

Este patrón es utilizado en el proyecto dado que el marco de trabajo empleado Symfony, en su versión 2, permite implementarlo a través de una estructura bien definida.

1.7.2 Patrones de diseño

Un patrón de diseño es una manera más práctica de describir ciertos aspectos de la organización de un programa. Los patrones de diseño proponen una forma de reutilizar la experiencia de los desarrolladores, para ello clasifica y describe formas de solucionar problemas que ocurren de forma frecuente en el desarrollo. Por tanto, están basados en la recopilación del conocimiento de los expertos en desarrollo de software. Es una experiencia real, probada y que funciona. Se agrupan de la siguiente manera: Patrones de Software para Asignación de Responsabilidades, del inglés *General Responsibility Assignment Software Patterns (GRASP)* y patrones de la banda de los cuatro, del inglés *Gang of Four (GOF)*. (30)

1.7.2.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Dentro de este grupo se

encuentran los siguientes patrones: Experto, Creador, Bajo acoplamiento, Alta cohesión y Controlador. (30)

Experto: ¿Quién asumirá la responsabilidad en el caso general? Asignar una responsabilidad al experto en información: la clase que posee la información necesaria para cumplir con la responsabilidad.

Creador: ¿Quién crea?

Asignar a la clase B la responsabilidad de crear una instancia de clase A, si se cumple una de las siguientes condiciones:

1. B contiene A
2. B agrega A
3. B tiene los datos de inicialización de A
4. B registra A
5. B utiliza A muy de cerca

Controlador: ¿Quién administra un evento del sistema? Asignar la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente una de las siguientes opciones:

1. El negocio o la organización global (un controlador de fachada).
2. El "sistema" global (un controlador de fachada).
3. Un ser animado del dominio que realice el trabajo (un controlador de papeles).
4. Una clase artificial (Fabricación Pura) que represente el caso de uso (un controlador de casos de uso).

Bajo Acoplamiento: ¿Cómo dar soporte a poca dependencia y a una mayor reutilización? Asignar las responsabilidades de modo que se mantenga bajo acoplamiento.

Alta Cohesión: ¿Cómo mantener controlable la complejidad? Asignar las responsabilidades de modo que se mantenga una alta cohesión. (30)

1.7.2.2 Patrones GOF

Los patrones de diseño el grupo de GoF clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento.

- **Creacionales:** abstraen el proceso de creación de instancias. ¿Cómo se puede crear un objeto? Habitualmente esto incluye aislar los detalles de la creación del objeto, de forma que su código no dependa de los tipos de objeto que hay y por lo tanto, no tenga que cambiarlo cuando añada un nuevo tipo de objeto. (Object Pool, Abstract Factory, Builder, Factory Method, Prototype, Singleton).
- **Estructurales:** esto afecta a la manera en que los objetos se conectan con otros objetos para asegurar que los cambios del sistema no requieren cambiar esas conexiones. Los patrones estructurales suelen imponerlos las restricciones del proyecto. (Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy).
- **De Comportamiento:** objetos que manejan tipos particulares de acciones dentro de un programa. Estos encapsulan procesos que quiere que se ejecuten, como interpretar un lenguaje, completar una petición, moverse a través de una secuencia (como en un iterador) o implementar un algoritmo. (Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor). (9)

1.8 Métricas

Una forma racional de mejorar cualquier proceso es medir sus atributos, desarrollar un juego de métricas significativas según estos atributos y utilizarlas para proporcionar indicadores que conducirán a una estrategia de mejora.

Se utilizan en el proyecto para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad.

Las métricas de análisis y diseño permiten entender tanto el proceso técnico que se utiliza para desarrollar el software, como el propio software. Al momento de diseñar, el principal objetivo es producir un sistema, aplicación o producto de alta calidad, para lo cual se emplean métodos y herramientas efectivas dentro del contexto de un proceso maduro de desarrollo del software y además, se deben desarrollar mediciones que den como resultado sistemas de alta calidad, con el fin de que el producto sea viable y factible, y para que cumpla con lo que el usuario desea. (31)

1.8.1 Métricas para requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los

errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto. (9)

En la disciplina Requisitos se tuvo en cuenta la métrica Especificidad de los requisitos, que permite tener una medida cuantitativa de la especificidad o falta de ambigüedad en la definición de los requisitos. (9)

Además se emplea la métrica de entendimiento de Estabilidad que verifica que no existan adiciones o supresiones en los requisitos que impliquen modificaciones en las funcionalidades principales de la aplicación. (31)

1.8.2 Métricas para diseño

Para medir el diseño se utilizaron métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objeto, principalmente la métrica C&K (Chidamber y Kemerer 1994) CBO (Coupling between Object Classes), y la métrica L&K (Lorenz y Kidd 1994) TOC (Tamaño Operacional de Clase). Fueron seleccionadas por el proyecto para la validación del diseño dado el nivel de aceptación que tienen en el ambiente donde son aplicadas:

Tamaño Operacional de Clase (TOC): plantea que el tamaño general de una clase se puede determinar a partir de la suma del número de atributos (tanto heredados como privados de la instancia) que están encapsulados en la clase, más el número total de operaciones (tanto heredadas como privadas de la instancia) también encapsuladas. Finalmente, se calculan los promedios correspondientes a los diferentes valores para tener una estimación general del sistema. Si existen valores grandes de TOC, se demuestra que una clase puede tener demasiada responsabilidad, lo cual reduciría su reutilización y complicaría la implementación y las pruebas. (9)

Acoplamiento entre clases (CBO-Coupling between Object Classes)

El CBO es el número de clases asociadas a una clase. Se da 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. Cuanto más acoplamiento se da en una clase, más difícil será reutilizarla. Además, se dificulta la comprensión y se hace más difícil el mantenimiento por lo que será necesario un mayor esfuerzo en la implementación y en las pruebas. (32)

1.9 Pruebas

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de la codificación. La creciente percepción del software como un elemento del sistema y la importancia de los costes asociados a un fallo del propio sistema, están motivando la creación de pruebas minuciosas y bien planificadas. (9)

1.9.1 Pruebas de caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, 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 un resultado correcto, así como la integridad de la información externa se mantiene. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. (9)

1.9.2 Pruebas de caja blanca

La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
- Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

La prueba del camino básico es una técnica de prueba de caja blanca que 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.

Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecute por lo menos una vez cada sentencia del programa. (9)

Tomando en cuenta todos los elementos mencionados en el presente capítulo 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 adicionar el módulo Civil y Familia al sistema que se está implementando.
- 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 Diseño y la Arquitectura 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. Requisitos y Análisis - Diseño.

2.1 Introducción

En este capítulo se especifican los requisitos funcionales y no funcionales, teniendo en cuenta las restricciones o políticas a cumplir por el sistema. Incluye el modelo conceptual, el cual refleja las teorías, plasma las propiedades y establece los principios del diseño del sistema. Además se presenta el diseño a través de los diagramas de clases y de interacción, haciendo uso de patrones de diseño.

2.2 Requisitos

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. (9) En esta disciplina se define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

2.2.1 Técnicas de obtención de requisitos

Entre las técnicas de obtención de requisitos empleadas, se tuvieron en cuenta la entrevista, la tormenta de ideas y la observación de sistemas semejantes. (33) Dado que se realizaron varias entrevistas con los fiscales a cargo de los procesos civiles y de familia del área PDC de la FGR, para conocer la problemática e identificar las necesidades del cliente. Además, se realizaron talleres con especialistas de la FGR y analistas del proyecto para obtener una visión general de la recepción de documentos en el área PDC. También se analizaron las principales características de los sistemas de gestión que fueron estudiados.

Los requisitos funcionales describen las acciones que debe llevar a cabo el producto, sin tomar en consideración ningún tipo de restricción física y especifican el comportamiento de entrada y salida del sistema. (9) Se identificaron 159 requisitos funcionales para el módulo. Ver Anexo 1. A continuación los requisitos que serán implementados, correspondientes a los procesos Expediente Prejudicial, Asuntos Notariales y Expropiación Forzosa, pues estos constituyen los procesos primarios, a partir de los cuales se originan nuevos procesos.

RF_PDC_CF_1	Adicionar Expediente Prejudicial
RF_PDC_CF_2	Actualizar Expediente Prejudicial
RF_PDC_CF_3	Adicionar decisión sobre promoción del asunto
RF_PDC_CF_4	Actualizar decisión sobre promoción de asunto
RF_PDC_CF_5	Visualizar detalles del expediente prejudicial
RF_PDC_CF_6	Listar expedientes prejudiciales
RF_PDC_CF_7	Buscar expediente prejudicial
RF_PDC_CF_8	Adicionar dictamen denegatorio de asuntos que se le interesen al fiscal
RF_PDC_CF_9	Actualizar dictamen denegatorio de asuntos que se le interesen al fiscal
RF_PDC_CF_10	Mostrar vista previa del dictamen denegatorio de asuntos que se le interesen al fiscal
RF_PDC_CF_11	Adicionar comunicación al solicitante
RF_PDC_CF_12	Actualizar comunicación al solicitante
RF_PDC_CF_13	Mostrar vista previa de la comunicación al solicitante
RF_PDC_CF_14	Editar persona menor de edad en el asunto tutela
RF_PDC_CF_15	Editar incapacitado en tutela
RF_PDC_CF_16	Editar padres en tutela
RF_PDC_CF_17	Editar parientes en tutela
RF_PDC_CF_18	Editar Solicitante en tutela
RF_PDC_CF_42	Adicionar rollo Asuntos Notariales
RF_PDC_CF_43	Actualizar rollo Asuntos Notariales
RF_PDC_CF_44	Buscar rollo Asuntos Notariales
RF_PDC_CF_45	Visualizar detalles del rollo Asuntos Notariales
RF_PDC_CF_46	Listar rollos de Asuntos Notariales
RF_PDC_CF_47	Generar vista previa de la Carátula del rollo de Asuntos Notariales por dictamen del fiscal

RF_PDC_CF_48	Generar vista previa de la Carátula del rollo de Asuntos Notariales por representación del fiscal
RF_PDC_CF_49	Adicionar Firma de Asuntos Notariales
RF_PDC_CF_50	Actualizar Firma de Asuntos Notariales
RF_PDC_CF_51	Adicionar Dictamen de Asuntos Notariales
RF_PDC_CF_52	Actualizar Dictamen Asuntos Notariales
RF_PDC_CF_53	Mostar Vista Previa del Dictamen Asuntos Notariales
RF_PDC_CF_54	Adicionar rollo Expropiación Forzosa
RF_PDC_CF_55	Actualizar rollo Expropiación Forzosa
RF_PDC_CF_56	Buscar rollo Expropiación Forzosa
RF_PDC_CF_57	Visualizar detalles del rollo Expropiación Forzosa
RF_PDC_CF_58	Listar rollos de Expropiación Forzosa
RF_PDC_CF_59	Generar vista previa de la Carátula del rollo Expropiación Forzosa
RF_PDC_CF_60	Adicionar Escrito de contestación a la demanda
RF_PDC_CF_61	Actualizar Escrito de contestación a la demanda
RF_PDC_CF_62	Mostrar vista previa del Escrito de contestación a la demanda
RF_PDC_CF_139	Adicionar datos del acta de entrevista
RF_PDC_CF_140	Actualizar datos del acta de entrevista
RF_PDC_CF_141	Listar actas de entrevistas
RF_PDC_CF_142	Eliminar acta de entrevista
RF_PDC_CF_143	Mostrar vista previa del acta de entrevista
RF_PDC_CF_144	Generar reporte
RF_PDC_CF_145	Generar reporte de AN
RF_PDC_CF_146	Generar reporte resumen de AN

2.2.2 Funcionalidad Adicionar datos del dictamen de Asuntos Notariales

Asuntos Notariales es uno de los procesos civiles y de familia que se llevan a cabo solamente en las fiscalías municipales del país. Esta funcionalidad será tomada como ejemplo para

desarrollar la especificación de requisitos de software, así como el diagrama de secuencia y la prueba de caja negra. Los asuntos notariales empiezan cuando el notario informa al fiscal sobre la existencia de un caso. Los asuntos pueden ser relacionados con:

- Firma de Documentos Notariales
- Dictamen de Asuntos Notariales

Si los documentos enviados son los correspondientes al dictamen, el fiscal estudia los documentos y crea el rollo fiscal. Después se crea el dictamen y se lo presenta al notario. De esta forma termina el asunto.

2.2.3 Técnica de definición de requisitos

Como técnica de definición de requisitos se empleó la técnica de plantillas o patrones, la cual tiene por objetivo describir los requisitos mediante el lenguaje natural, pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida. Se pone de manifiesto en el documento especificación de requisitos de software del módulo, de forma que se elimine parte de la ambigüedad del lenguaje natural al estructurar la información, cuanto más estructurada sea ésta, menos ambigüedad ofrece. (33)

La especificación de los requisitos del software se produce en la culminación de la tarea de análisis de requisitos. La función y rendimiento asignados al software como parte de la ingeniería de sistemas se observan estableciendo una completa descripción de la información, una descripción detallada de la función y del comportamiento, una indicación de los requisitos del rendimiento y restricciones del diseño, criterios de validación apropiados y otros datos pertinentes a los requisitos. (9)

Un ejemplo de especificación es la que se muestra en la tabla 1, correspondiente al requisito *RF_PDC_CF_51 Adicionar datos del dictamen de AN*, que incluye el prototipo de interfaz de usuario. La especificación de cada requisito puede ser consultada en el documento 0113_Especificación de Requisitos de Software.

Tabla 1. Especificación del requisito: Adicionar datos del dictamen de AN.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente

RF_P DC_C F_51	Adicionar datos del dictamen de AN	Permite adicionar los datos del dictamen del AN, cuando la actuación es dictaminar.	Baja	Alta
Prototipo				
<div style="display: flex; justify-content: space-between;"> <div data-bbox="300 394 548 451">Fecha de recibida la solicitud 07/14/2007 </div> <div data-bbox="613 394 862 451">Fecha de recibido el expediente 07/14/2007 </div> <div data-bbox="950 394 1247 451">Asunto -Seleccione- </div> </div> <p data-bbox="300 514 893 535">Breve descripción del contenido de la solicitud del abogado al notario</p> <div data-bbox="300 546 1409 688" style="border: 1px solid gray; height: 68px;"></div> <p data-bbox="300 766 381 787">Dictamen</p> <div data-bbox="300 793 565 829" style="border: 1px solid gray; padding: 2px;">-Seleccione- </div> <p data-bbox="300 934 511 955">Consideración del fiscal</p> <div data-bbox="300 966 1409 1108" style="border: 1px solid gray; height: 68px;"></div> <p data-bbox="300 1171 430 1192">Observaciones</p> <div data-bbox="300 1203 1409 1346" style="border: 1px solid gray; height: 68px;"></div> <p data-bbox="300 1396 446 1417">Fecha de emisión</p> <div data-bbox="300 1423 521 1459" style="border: 1px solid gray; padding: 2px;">07/14/2007 </div>				

Figura 51.1 Datos del Dictamen de AN.

Campos	Tipos de Datos	Reglas o Restricciones
Fecha de recibida la solicitud	Fecha	Obligatorio. Menor o igual a la Fecha de registro.
Fecha de recibido el expediente	Fecha	Obligatorio. Menor o igual a la Fecha de registro.

Asunto	Cadena	Obligatorio. Los valores del campo se muestran en el documento <i>NOMENCLADORES_PDC_CF</i> .
Breve descripción del contenido de la solicitud del abogado al notario.	Cadena	Obligatorio.
Dictamen	Cadena	Obligatorio. Los valores del campo se muestran en el documento <i>NOMENCLADORES_PDC_CF</i> .
Condición	Cadena	Obligatorio. Se muestra si el valor del campo Dictamen es Condicionado.
Consideración del fiscal	Cadena	Obligatorio.
Fecha de emisión	Fecha	Obligatorio. Mayor o igual que la Fecha de registro. Por defecto se muestra la fecha actual.
Observaciones	N/A	

Como parte del documento especificación de requisitos de software se desarrolló el modelo conceptual, el cual permite identificar, organizar y realizar razonamientos sobre los componentes y comportamiento del sistema, es la guía para el proceso de diseño del software y puede usarse posteriormente como una referencia para evaluar un diseño particular, razonar sobre la solución realizada y sobre el posible espacio de soluciones. (34). Se representa en UML mediante un grupo de estructuras estáticas.

En la figura que a continuación se presenta se evidencian las diferentes instancias de la FGR, donde se llevan a cabo los procesos que serán implementados. Se muestran además cada uno de los documentos que se generan y los responsables de tomar parte en cada uno de los procesos, en este caso, la secretaria y el propio fiscal. También se representan los

involucrados, ya sea cada una de las personas que intervienen, las entidades o las notarías. Paralelo a cada proceso se puede tomar notas o redactar actas de entrevista.

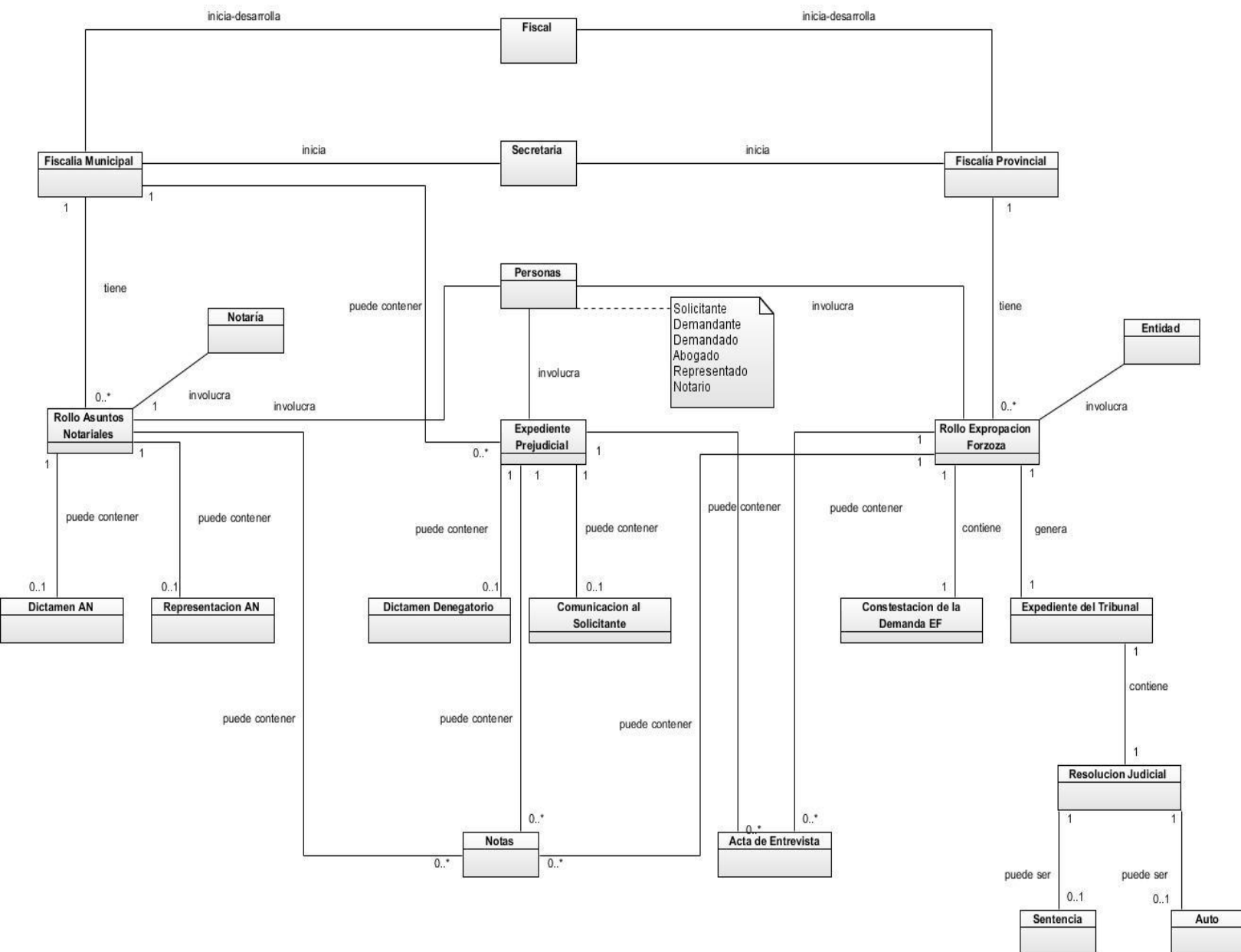


Figura 2. Modelo conceptual de los procesos civiles y de familia.

2.2.5 Requisitos no funcionales

Se especifican las propiedades del sistema que tienen que ver con las características no funcionales como son rendimiento, velocidad, uso de memoria, plataforma, fiabilidad, entre otros. (9)

Los requisitos no funcionales obtenidos fueron agrupados de la siguiente manera: 3 de usabilidad, 5 de confiabilidad, 2 de eficiencia, 2 de soporte, 3 de restricciones diseño, 1 de ayuda del sistema, 3 de interfaz de usuario. Para un total de 19 requisitos no funcionales. Ver documento CEGEL_SIGEFII_0113_ERS_ECS Sección 3.2

2.2.6 Validación de los requisitos

Métricas de requisitos:

Se aplicaron las métricas estabilidad y especificidad para controlar las adiciones y supresiones de requisitos y para verificar que no exista ambigüedad en los mismos respectivamente.

Estabilidad se calcula de la forma:

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

Dado que se identificaron 159 RF, y que de ellos se modificaron 57, se obtiene como resultado

$$ETR = [(159 - 57) / 159] * 100 = 64.15$$

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. Al observar la cifra, se puede concluir que los requisitos identificados no fueron lo suficientemente estables para comenzar las tareas de diseño. Actualmente, erradicada la situación que provocó los cambios de última hora y validados los requisitos por parte del cliente, sin sufrir más cambios, se obtuvo 100% de estabilidad en los requisitos y se logró conformidad entre las partes.

Especificidad, que se calcula como:

$ER = Nui / Nr$, donde

- ER: grado de especificidad de los requisitos.

- Nui: número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.
- Nr: cantidad de requisitos en una especificación, que comprende la suma de los requisitos funcionales y los requisitos no funcionales.
- Los analistas del proyecto son los revisores que se tienen en cuenta para aplicar la métrica.

Se identificaron 159 requisitos funcionales y 19 requisitos no funcionales, en total 178 (Nr). Luego de la revisión de los requisitos por parte de los revisores, 172 requisitos tuvieron interpretaciones idénticas (Nui), ya que seis de los requisitos funcionales mostraron diferencias en cuanto al entendimiento e interpretación por parte de los revisores. Aplicando la fórmula de especificidad se obtiene el siguiente resultado:

$$ER = 172 / 178 = 0.97$$

El valor de ER mostró que la interpretación de los requisitos por parte de los revisores tiene una aceptable consistencia y que la especificación presentó un pequeño grado de ambigüedad, por lo que se procede al refinamiento de los requisitos para la eliminación de dichas ambigüedades. Finalmente, como resultado $ER = 1$, por lo que se puede deducir que el entendimiento de los requisitos es adecuado.

Técnicas de validación

Para validar los requisitos se emplearon como técnicas los prototipos y las revisiones. Los prototipos apoyaron la descripción de los requisitos, al permitirle al usuario hacerse una idea de la estructura de la interfaz del sistema y detectar cualquier problema en la identificación de los mismos. (33) Las revisiones constituyen un aval de calidad. El documento fue entregado al Centro Nacional de Calidad de Software (CALISOFT) para la detección de no conformidades en la documentación. En la primera iteración se detectaron 49 no conformidades, las cuales fueron resultas, permitiendo que el documento se liberara en la segunda iteración.

Validación por parte del cliente: el documento especificación de requisitos de software fue presentado al fiscal a cargo de los procesos civiles y de familia del área PDC de la FGR, para mostrar conformidad o desacuerdo con los requisitos. Finalmente fueron aprobados, y se evidencia a través de la firma del fiscal en la documentación presentada en la FGR.

2.3 Análisis y Diseño

En el diseño se pretende formular los modelos que no solamente se centran en los requisitos funcionales, sino también en los requisitos no funcionales y en el dominio de la solución propuesta. Sin diseño no habría implementación, pues descompone los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo. (35)

2.3.1 Estilos de codificación

- Los nombres deben ser descriptivos y concisos. No usar grandes frases o pequeñas abreviaciones.
- Los nombres de las páginas twig serán en minúscula siempre y underscore, por ejemplo: `adicionar_solicitud`
- Clase: en mayúscula. eje: `DictamenAN`
- Atributos: minúscula siempre y underscore eje: `persona_proceso`
- Funciones: empiezan con minúscula y continúan en mayúscula: `imprimirDatos()`
- Cada atributo debe tener activo sus métodos `getter` y `setter`, además de los generados producto de relaciones. También deben incluir la visibilidad y el tipo de dato.

Métodos de la clase controladora:

- `SalvarActualizar`, para el caso de las páginas que contienen formulario. Se utiliza el mismo método del controlador para mostrar la página (`get`) que para persistir (`post`).
- Para el caso de las páginas iniciales de los procesos, se pone el nombre del proceso en plural, ejemplo: `asuntosNotarialesAction`, `expedientePrejudicialAction`.

Estándares para los demás métodos:

- Se utiliza (`cargar`) para los métodos que cargan por id.
- Se utiliza (`listar`) para los métodos que llenan la información de los grid.

2.3.2 Diagrama de clases persistentes

En el diagrama se representan las clases y sus relaciones. Constituyen parte del diagrama los atributos y las operaciones de las clases, su visibilidad, el nombre y el tipo de dato, así como sus relaciones. A continuación se muestra el diagrama de clases correspondiente al proceso Asuntos Notariales, el cual constituye un fragmento del diagrama de clases persistentes del módulo. Cada proceso en los módulos del proyecto SIGEF II cuenta con una cantidad determinada de documentos, de acuerdo a las características particulares. Es por esta razón que existen clases comunes de las cuales los módulos se apropian de las características para llevar a cabo cada proceso, por ejemplo: las clases Proceso y Documento. En el caso del diagrama mostrado en la figura, la clase principal es RolloAsuntosNotariales, clase que se encuentra relacionada con un Dictamen o una Representación. Las clases Notaría y TipoDictamen constituyen nomencladores, en función de facilitar la ejecución del proceso y la clase nota es común para todos los procesos del módulo.

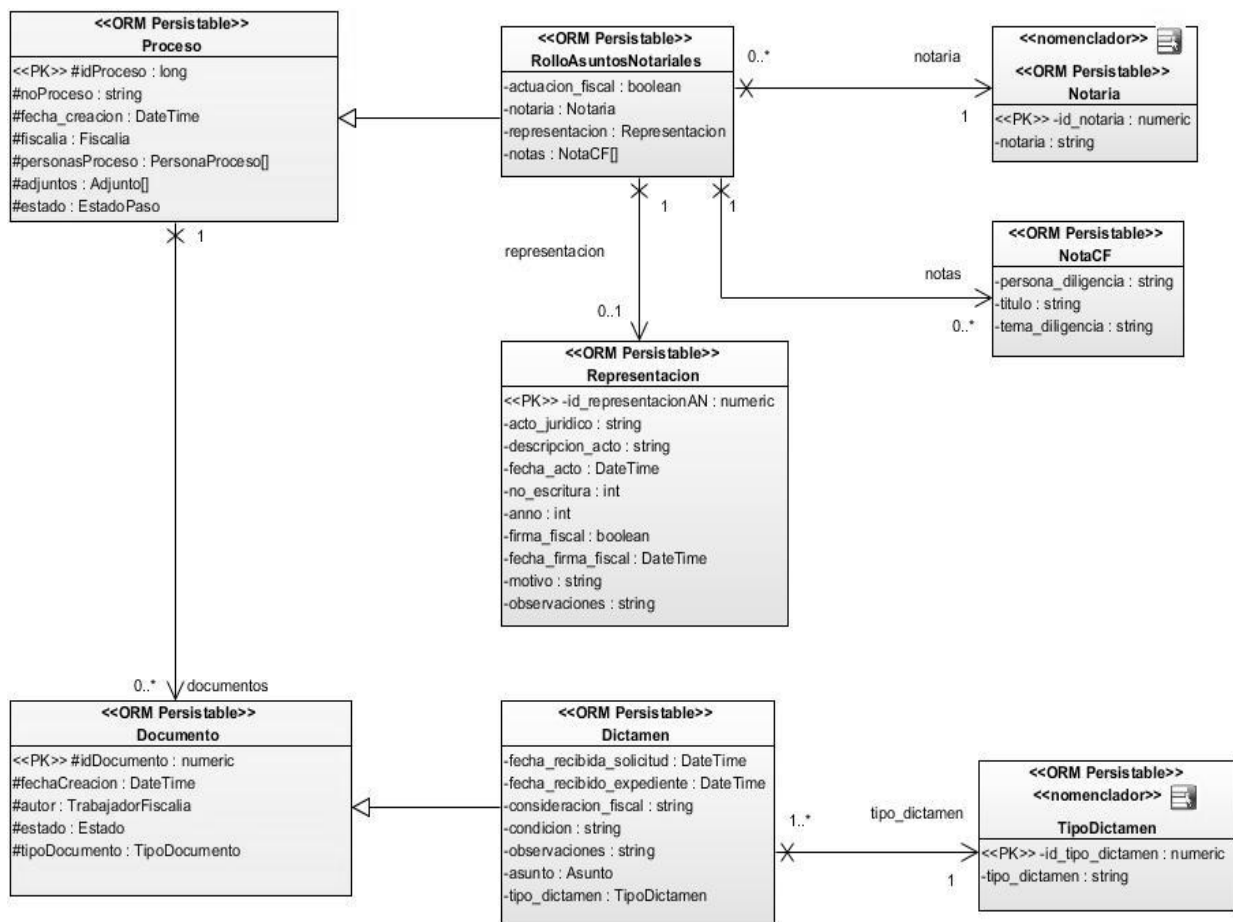


Figura 3. Diagrama de clases persistentes para el proceso Asuntos Notariales.

2.3.3 Patrones de diseño empleados

Los patrones de diseño proporcionan un esquema para refinar los subsistemas o componentes del sistema, o las relaciones entre ellos. Describen estructuras repetitivas de comunicar componentes que resuelven un problema de diseño en un contexto particular. (9) Los patrones GRASP son empleados en su totalidad, mientras que Symfony, como marco de trabajo empleado por el proyecto, utiliza varios patrones pertenecientes a los GOF, de forma que se promuevan las buenas prácticas de diseño y programación, logrando un mejor funcionamiento del sistema.

Patrones GRASP

Experto: este patrón se evidencia, por ejemplo, en la clase `RolloAsuntosNotarialesController`. Esta clase posee toda la información relacionada con los rollos, lo que le permite crear un objeto de la clase `RolloAsuntosNotariales`, convirtiéndola en una experta en la información.

Creador: se puede reconocer en la misma clase `RolloAsuntosNotarialesController`, mediante la creación de la instancia de la clase entidad que contienen los datos del proceso Asuntos Notariales (`RolloAsuntosNotariales`), que permite desarrollar acciones como `salvarActualizarAsuntosNotarialesAction`.

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 del módulo con la URL entrada por el usuario. Este patrón se refleja en el diagrama de componentes. Ver Figura 13. Diagrama de componentes del módulo Civil y Familia.

Alta Cohesión: se encuentra evidenciado por ejemplo en las clases con sufijo `Controller`, las cuales tienen la responsabilidad de definir las acciones para las peticiones propias de un proceso y, paralelo a esto colabora con otras para realizar diferentes operaciones, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas. Ver Figura 5. Diagrama de clases controladoras y gestoras para Asuntos Notariales.

Bajo Acoplamiento: consiste en tener las clases lo menos relacionadas entre sí, para que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión en las

demás. Esta característica permitió potenciar la reutilización y disminuir la dependencia entre las clases. Por ejemplo, la clase `RolloAsuntosNotarialesController` hereda de `ProcesoController` para lograr un bajo acoplamiento entre ellas y así proporcionar que el software sea flexible frente a grandes cambios. Ver Figura 5. Diagrama de clases controladoras y gestoras para Asuntos Notariales.

Patrones GOF

Inyección de dependencias: Se trata de un objeto “inteligente” que conoce las dependencias de los servicios y es capaz de construir cualquier servicio que se le pida y de entregarlo bien configurado, listo para su uso y sin preocupación de instanciar e “inyectar” sus dependencias. Permite estandarizar y centralizar la forma en que se construyen los objetos en la aplicación. (36) Se utilizó en la creación de servicios en la arquitectura base que serán utilizados por todos los módulos del proyecto usando la clase `services.yml`.

Fábrica abstracta: define una interfaz para crear un objeto, pero delega la responsabilidad de instanciarlo a sus subclases y promueve el encapsulamiento de las partes más variables del sistema. Ejemplo de su uso se aprecia en la creación de los objetos formularios en la clase `RolloAsuntosNotarialesController`.

Decorador: aplicado a la generación de vistas, la solución que ofrece dicho patrón es la de añadir funcionalidad adicional a las plantillas. Por ejemplo, añadir el menú y el pie de página a las plantillas que lo requieran, de manera que dichos elementos puedan reutilizarse en distintas plantillas. Literalmente se trata de decorar las plantillas con elementos adicionales reutilizables. El sistema de plantillas twig, está provisto de un mecanismo de herencia gracias al cual la decoración de plantillas resulta de una flexibilidad y versatilidad total. (30) Este patrón se observa en el archivo denominado `layoutCivilFamilia.html.twig` que contiene el layout (plantilla global) del módulo Civil y Familia. El mismo almacena el código HTML que es común para todas las páginas, por lo que cada página que se cree heredará de esta, para no tener que repetir el código, y de esta forma el layout es quien decora la plantilla.

Fachada: se pone de manifiesto en la clase `RolloAsuntosNotarialesGtr`, la misma se encarga de definir un método capaz de salvar o actualizar determinado objeto, dicho método funciona como intermediario entre esta clase y los métodos propios de Symfony, que son los que realmente obtienen todos los datos correspondientes para persistir el objeto en la base de datos.

Observador: permite a los objetos captar dinámicamente las dependencias entre objetos, de tal forma que un objeto notificará a los objetos dependientes de él cuando cambia su estado, siendo actualizados automáticamente. (30) Por ejemplo la clase TrazaListen que es la encargada de registrar eventos como adicionar, modificar y eliminar, ocurridos en las demás clases.

2.3.4 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. (9) A continuación se muestra el diagrama de secuencia para los requisitos Adicionar/Actualizar dictamen de Asuntos Notariales.

En la figura el usuario selecciona la opción Actualizar en la página inicial donde se muestran el listado de todos los rollos del proceso. La acción activa la operación correspondiente en la clase controladora RolloANController, la cual verifica que el proceso se encuentre en edición, obtiene el documento correspondiente y muestra el formulario para que se introduzcan los datos. Se muestra además en el diagrama las acciones que se ejecutan cuando se selecciona la opción “Siguiente”, pues se accede a la interfaz relacionada con la asociación de las personas al documento. En caso de seleccionar la opción “Guardar y Cerrar”, se retorna a la página inicial.

Capítulo 2

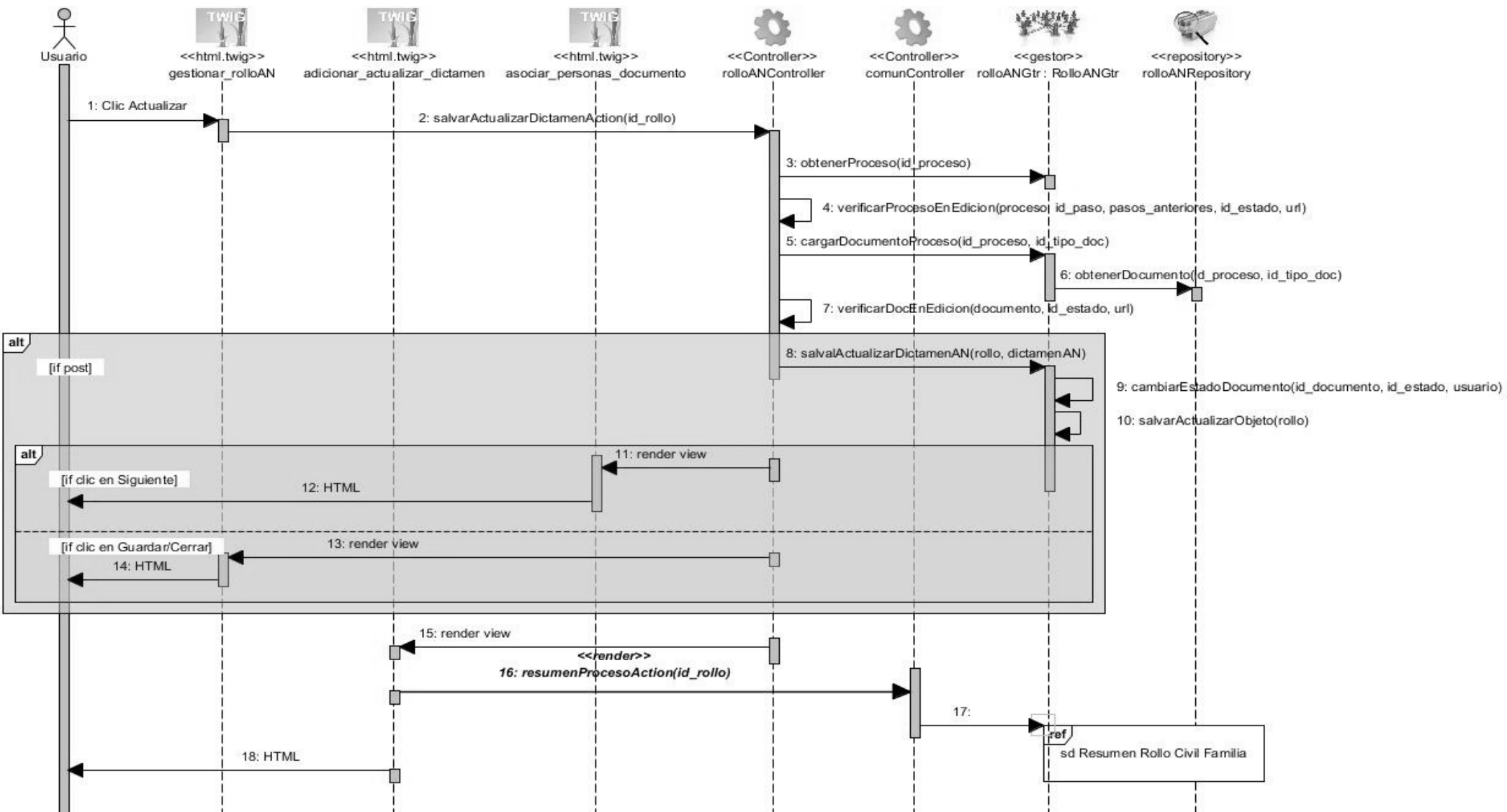


Figura 4. Diagrama de secuencia Adicionar/Actualizar dictamen de Asuntos Notariales.

2.3.5 Diagrama de clases controladoras y gestoras

Las clases controladoras (Controller) coordinan el trabajo de las clases, encapsulan el comportamiento de una funcionalidad, además de realizar las operaciones más complejas. Las clases gestoras (Gtr) son las intermediarias entre la clase controladora y la clase repositorio. Estos tipos de clases se incluyen en el mismo diagrama de acuerdo a la estrecha relación que existe entre ellas a la hora de satisfacer cualquier funcionalidad. En la figura se representan las clases relacionadas con el proceso Asuntos Notariales, donde se evidencia además, la herencia entre las clases del propio módulo con las clases comunes del sistema. (ProcesoController y ProcesoGtr).



Figura 5. Diagrama de clases controladoras y gestoras para Asuntos Notariales.

2.3.6 Diagrama de clases repositorio

Estas clases son las encargadas del acceso a datos, elemento de gran importancia para el marco de trabajo seleccionado por el proyecto. A través de ellas se realizan las consultas. En la figura es mostrada la relación de herencia existente entre la clase del módulo RolloANRepository, y la clase común EntityRepository. También se evidencia la operación a realizar por la clase del módulo, la cual permite obtener todos los rollos de este proceso recepcionados hasta el momento.

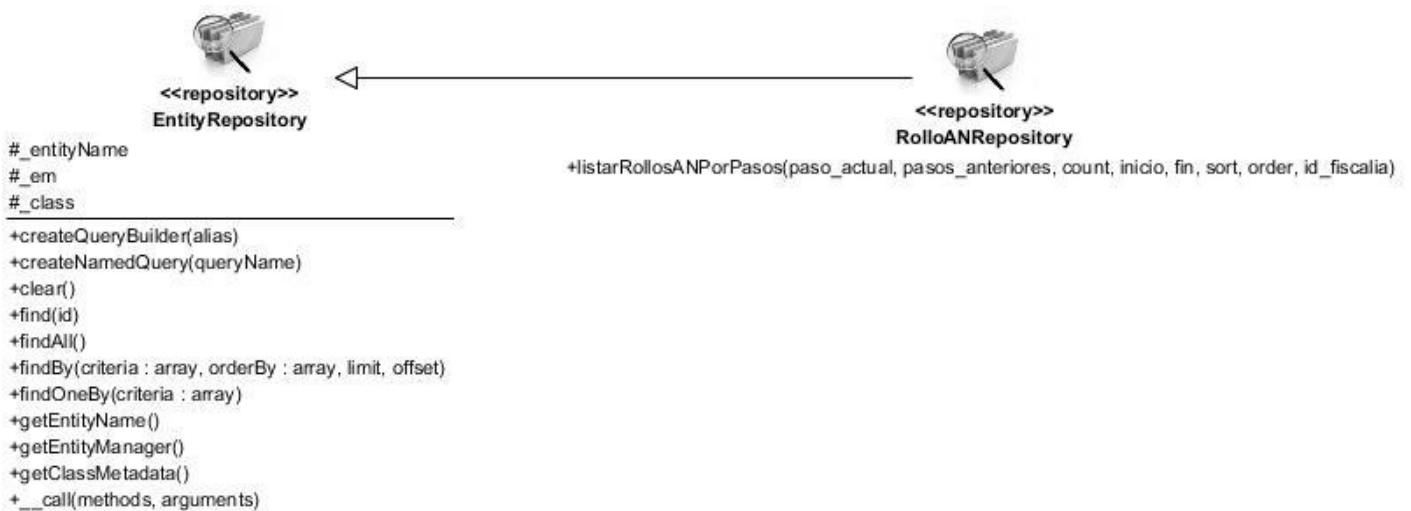


Figura 6. Diagrama de clases repositorio para Asuntos Notariales.

2.3.7 Diagrama de la vista

Este diagrama representa la navegación básica del sistema respecto al proceso Asuntos Notariales, que mediante los estereotipos web Twig, que es un motor de plantillas para PHP recomendado por Symfony 2, muestra las diferentes interfaces de usuario y los formularios que contienen.

En la figura se muestra la página gestionar_rolloAN, donde se obtiene el listado de todos los rollos relacionados con el proceso. Siguiendo la secuencia de interfaces, a través de los botones Adicionar o Actualizar se accede a la interfaz del mismo nombre. Posteriormente se introducen en el sistema las personas relacionadas con el caso y se finaliza el paso inicial con la inserción de los datos adjuntos. El otro flujo representado es para la creación de los documentos, pues mediante la opción Actualizar se accede al Dictamen o a la Representación de acuerdo a la actuación del fiscal, para finalmente concluir en la página asociar_persona_dictamenAN, donde se seleccionan las personas pertenecientes al caso.

Específicamente para la Representación, su flujo termina en la página del mismo nombre, ya que no constituye en documento.

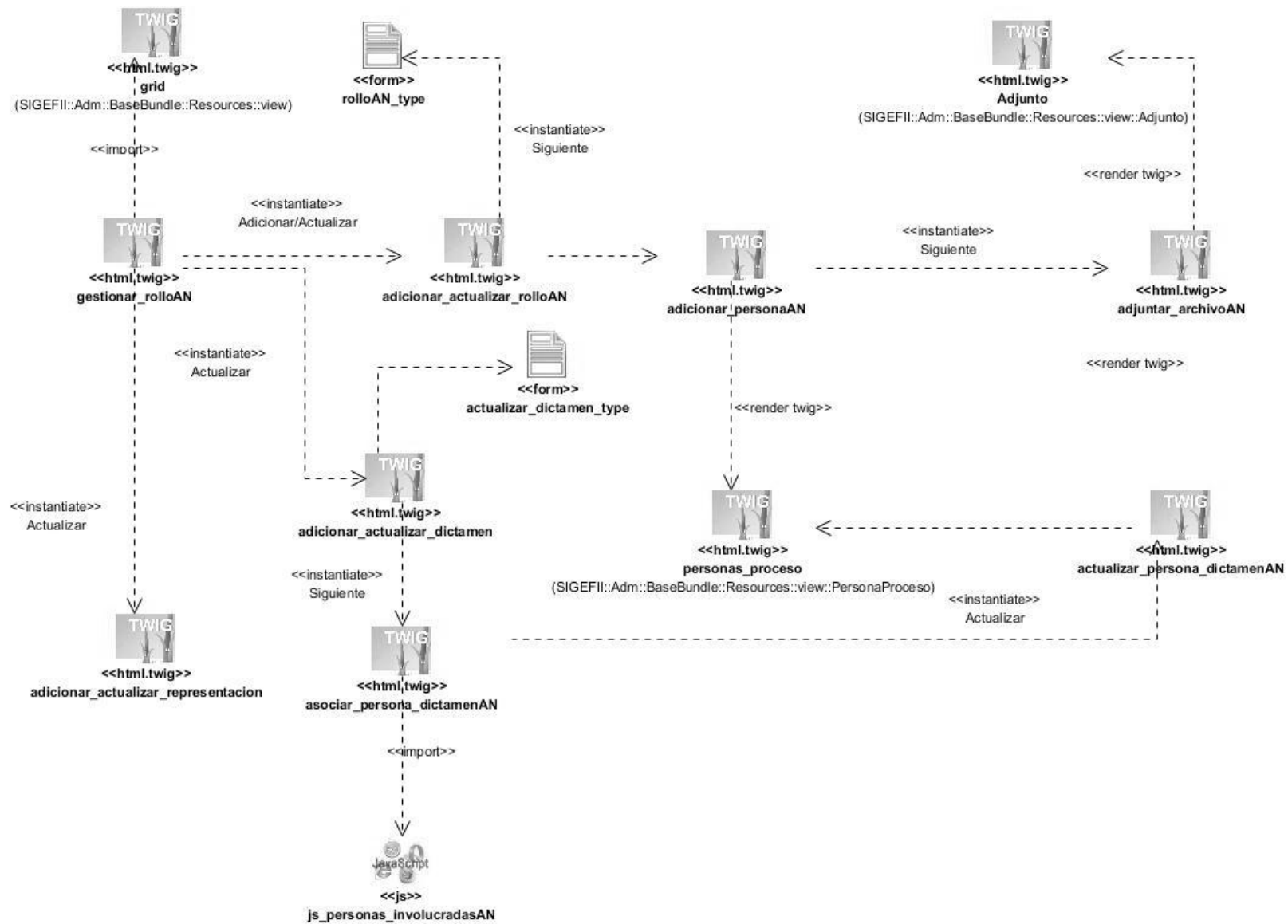


Figura 7. Diagrama de la vista para Asuntos Notariales.

2.3.8 Métricas aplicadas para la validación del diseño

Para validar el diseño del módulo se emplearon las métricas TOC y Acoplamiento entre clases.

- TOC

Los resultados obtenidos después de aplicar esta métrica son los siguientes:

Tabla 2. Representación del tamaño operacional de las clases relacionadas con el módulo Civil y Familia.

No.	Clases	Cantidad de atributos	Cantidad de operaciones	Tamaño
1	RolloAsuntosNotariales	3	6	Medio
2	Proceso	7	14	Medio
3	Documento	5	10	Medio
4	Representacion	10	20	Grande
5	DictamenAN	7	14	Medio
6	TipoDictamen	2	4	Pequeño
7	Notaria	2	4	Pequeño
8	Nota	3	6	Pequeño
9	RolloJurisdiccionVoluntaria	4	8	Medio
10	ProcedenciaJV	3	6	Pequeño
11	EscritoPromoviendoTutela	2	4	Pequeño
12	DatosTutor	7	14	Medio
13	EscritoPromoviendoAsunto	4	8	Medio
14	DictamenJV	3	6	Pequeño
15	DecisionDictamen	3	6	Pequeño
16	RolloExpropiacionForzosa	5	10	Medio
17	ContestacionDemandaEF	6	12	Medio
18	ActaEntrevista	3	6	Pequeño
19	ExpedientePrejudicial	13	26	Grande
20	DecisionEP	2	4	Pequeño

21	MotivoRadicion	2	4	Pequeño
22	Asunto	5	10	Medio
23	TipoAsunto	2	4	Pequeño
24	Datos	2	4	Pequeño
25	AFavorDe	3	6	Pequeño
26	ComunicacionSolicitante	1	2	Pequeño
27	DictamenDenegatorio	3	6	Pequeño
28	TipoProceso	3	6	Pequeño
29	Incapacitado	3	6	Pequeño
30	DatosMenor	1	2	Pequeño
31	DatosPariente	2	4	Pequeño
32	DatosPadreMadre	1	2	Pequeño
33	GradoConsanguineidad	3	6	Pequeño
34	MotivoNoPatriaPotestad	3	6	Pequeño
35	RolloRevisionCivil	4	8	Medio
36	PresentacionDemanda	7	14	Medio
37	ExpedienteOrigen	5	10	Medio
38	ProcedenciaEO	3	6	Pequeño
39	ContestacionDemandaRRC	5	10	Medio
40	ExpedienteTribunal	3	6	Pequeño
41	RolloRecursoCasacion	6	12	Medio
42	DatosRolloRC	3	6	Pequeño

43	EscritoNoRecurrenteRC	4	8	Medio
44	EscritoRecurrenteRC	4	8	Medio
45	ActitudAnteRecurso	3	6	Pequeño
46	Vista	3	6	Pequeño
47	Motivo	5	10	Medio
48	EscritoEstableciendoRC	3	6	Pequeño
49	RolloRecuroApelacion	6	12	Medio
50	DatosRollorRA	3	6	Pequeño
51	EscritoEstableciendoRA	3	6	Pequeño
52	Acapite	3	6	Pequeño
53	FalloTribunal	3	6	Pequeño
54	EscritoNoRecurrenteRA	4	8	Medio
55	EscritoRecurrenteRA	3	6	Pequeño

Se relacionan con el proceso un total de 55 clases, obteniendo un promedio de 3.84 atributos y 7.67 operaciones por clase.

Tabla 3. Resultado de la aplicación de la métrica TOC.

Cantidad de clases: 55	Baja	Media	Alta
Responsabilidad	60%	36.36%	3.63%
Complejidad	60%	36.36%	3.63%
Reutilización	3.63%	36.36%	60%

Luego de haber aplicado esta métrica, se obtuvo como resultado que las clases analizadas del proceso son pequeñas en su gran mayoría. 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

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

Acoplamiento: 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 4. Criterios de evaluación de la métrica AC.

Atributo	Categoría	Criterio
Acoplamiento.	Ninguno.	0
	Bajo.	1
	Medio.	2
	Alto.	>2
Complejidad de mantenimiento.	Baja.	<=Promedio

	Media.	Entre Promedio y $2 * \text{Promedio}$
	Alta.	$> 2 * \text{Promedio}$
Reutilización.	Baja.	$> 2 * \text{Promedio}$
	Media.	Entre Promedio y $2 * \text{Promedio}$
	Alta.	$\leq \text{Promedio}$
Cantidad de pruebas.	Baja.	$\leq \text{Promedio}$
	Media.	Entre Promedio y $2 * \text{Promedio}$
	Alta.	$> 2 * \text{Promedio}$

Para un total de 55 clases y un promedio de asociaciones de uso de 2.236.

Tabla 5. Criterios y categorías obtenidos en la aplicación de la métrica.

Criterio	Categoría	Cantidad de clases	%
0 dependencias	Muy Bueno	5	9.090909091
1 dependencias	Bueno	24	43.63636364
2 dependencias	Regular	9	16.36363636
3 dependencias	Malo	8	14.54545455
> 3 dependencias	Muy Malo	9	16.36363636
Total		55	100

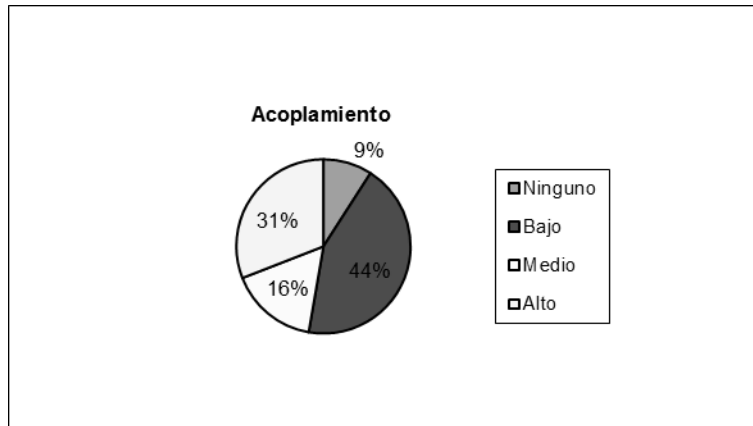


Figura 8. Resultados obtenidos para el atributo Acoplamiento.

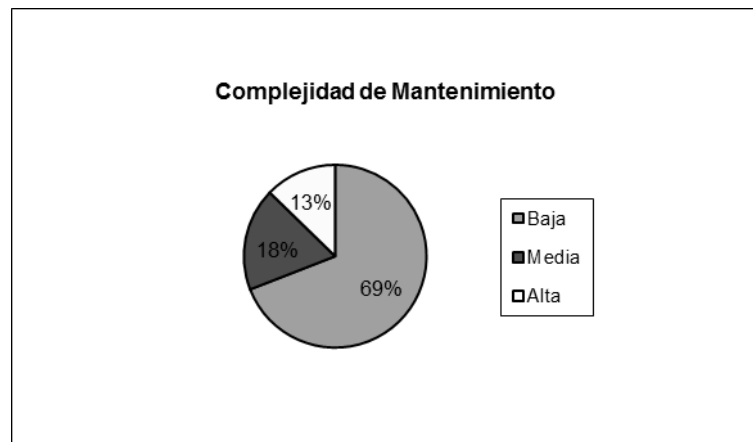


Figura 9. Resultados obtenidos para el atributo Complejidad de mantenimiento.

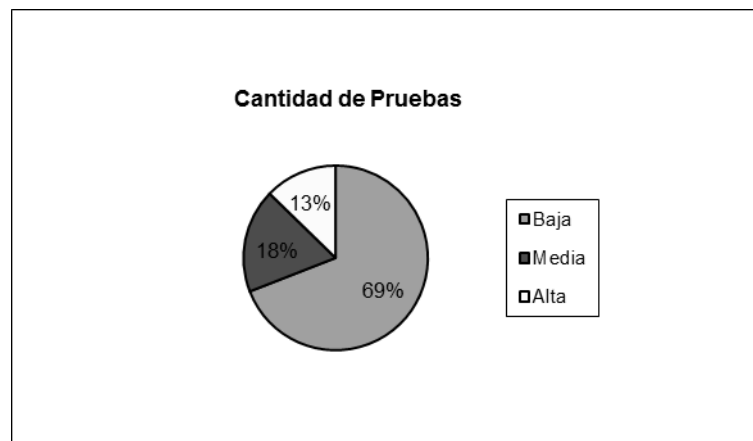


Figura 10. Resultados obtenidos para el atributo Cantidad de pruebas.

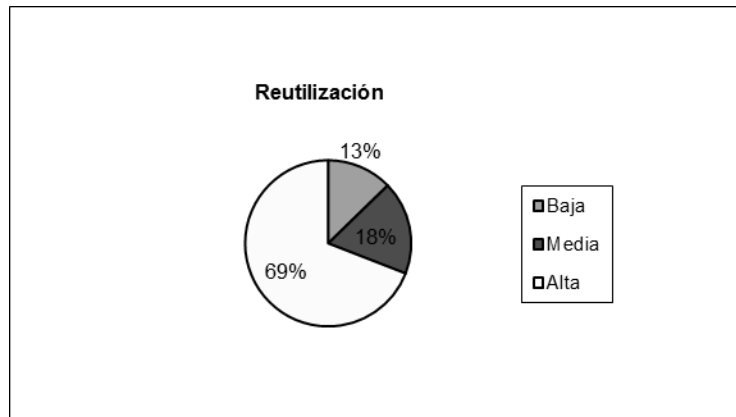


Figura 11. Resultados obtenidos para el atributo Reutilización.

Los elementos medidos a través de las métricas propuestas mostraron que el diseño obtenido se encuentra con una calidad aceptable, pues las dependencias entre clases se comportan de forma favorable y el acoplamiento es en su mayoría bajo, haciendo más fácil reutilizar las clases. También se puede percibir que los demás indicadores dan muestra de los buenos resultados obtenidos del diseño.

En el capítulo se lograron captar las principales funcionalidades del módulo Civil y Familia, identificando de esta manera 159 requisitos funcionales, de los cuales 47 serán implementados, y 19 requisitos no funcionales. El documento de especificación de requisitos de software permitió validar con el cliente los prototipos no funcionales de interfaz. El modelo conceptual permitió conocer los procesos que se desarrollarán posteriormente y sus particularidades. El diseño, patrones, artefactos generados en la disciplina y métricas aplicadas permitirán implementar el sistema de forma eficiente.

Capítulo 3. Implementación y Prueba.

3.1 Introducción

En este capítulo se desarrolla el modelo de implementación que incluye las interfaces del sistema, el diagrama de despliegue y el diagrama de componentes, además de la descripción detallada de los paquetes de implementación. Igualmente se lleva a cabo la validación a través de las pruebas de caja blanca y caja negra de los procesos Expediente Prejudicial, Asuntos Notariales y Expropiación Forzosa, que garanticen que las funcionalidades implementadas cumplan con las necesidades del cliente.

3.2 Implementación

El modelo de implementación es el artefacto que se genera en esta disciplina. Está conformado por el diagrama de componentes y el diagrama de despliegue, describiendo cómo los elementos del modelo de diseño se implementan en términos de componentes, ficheros de código fuente y ejecutables, además de cómo estos se organizan de acuerdo a los nodos específicos en el diagrama de despliegue. Se describe también cómo se organizarán los componentes de acuerdo con los mecanismos de estructuración, disponibles en el entorno de implementación y en el lenguaje de programación utilizado, y cómo dependen los componentes unos de otros. (9) Todo esto teniendo en cuenta la estructura de la FGR, así como la relación que debe existir entre los componentes para un correcto funcionamiento del sistema.

3.2.1 Diagrama de componentes

Modelan la vista estática de un sistema. Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten.

Componente: es la parte modular de un sistema, desplegable y reemplazable que encapsula implementación, un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos (ficheros ejecutables, binarios, etc.). Son las piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas. (9)

En un diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. (9)

La siguiente figura muestra el diagrama de componentes del módulo Civil y Familia. Representa la estructura del proyecto SIGEF II, que contiene el subsistema PDC, al cual pertenece el módulo Civil y Familia. En el subsistema SIGEF II se encuentran archivos de configuración como son: Security, encargado del trabajo de la seguridad del subsistema; archivo Routing, gestiona las rutas del subsistema y el Config, que posee las principales configuraciones de SIGEF II. El componente Ctr_Frontal es el controlador frontal que recibe cada petición realizada al sistema. El paquete Arquitectura Base contiene componentes con funcionalidades comunes para todo el sistema, como son la Seguridad, Servicios y Componentes.

En el subsistema PDC se encuentra el componente Mensajería, que gestiona los mensajes que deben ser mostrados y Excepciones para la captura y tratamiento de las excepciones generadas. Contiene el módulo Civil y Familia el cual se divide en 3 paquetes fundamentales. El paquete Controlador posee el componente Controladoras compuesto por todas las clases controladoras correspondientes a los procesos que se llevan a cabo en el módulo, por ejemplo, RolloAsuntosNotarialesController, encargadas de dar respuesta a las peticiones de los usuarios. Controladoras se relaciona con el componente Negocio, compuesto por las clases gestoras, por ejemplo, RolloAsuntosNotarialesGtr, que se ocupan de gestionar la lógica del negocio. Ambos componentes se relacionan con el componente Entidades que pertenece al paquete Modelo, este contiene las clases entidades del módulo. Entidades es usado por el componente Repositorio encargado de realizar las consultas a la base de datos. El paquete Vista, contiene el componente Form encargado del trabajo con los formularios, el componente Twig que tiene como objetivo el trabajo con las interfaces de usuario y el componente Extensiones que posee los elementos javascript y los css. Routing es el encargado de trabajar con las rutas definidas para el módulo. Services, ofrece los servicios públicos a los que se accede desde cualquier parte del subsistema PDC, en este componente se registran las clases gestoras para acceder a ellas desde los controladores.

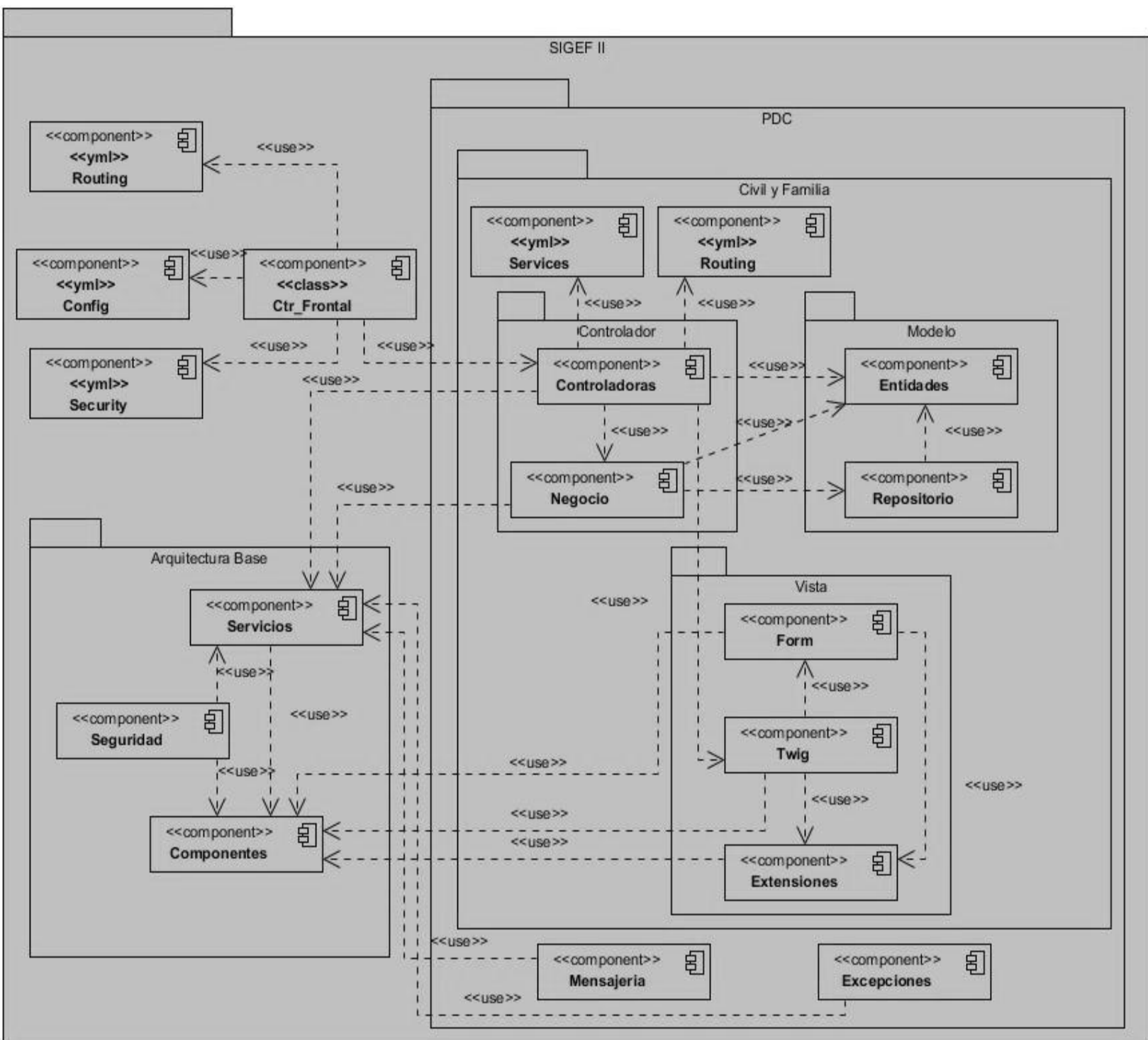


Figura 12. Diagrama de componentes del módulo Civil y Familia.

3.2.2 Diagrama de despliegue

El diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los links de comunicación entre ellos, y las instancias de los componentes y objetos que residen en ellos. El propósito es capturar la configuración de los elementos de procesamiento, y las conexiones entre estos elementos en el sistema. Los elementos que lo componen son:

Procesador: nodo con elementos de procesamiento, al menos un procesador, memoria, y posiblemente otros dispositivos.

Dispositivos: nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.

Conectores: expresa el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo. (9)

En la figura que a continuación se presenta se muestra la distribución de los distintos elementos que componen este diagrama, respecto a la organización con que cuenta la FGR. Entre los procesadores se encuentran los distintos tipos de servidores y las estaciones de trabajo. Como dispositivos, se perciben la impresora y el escáner. Como conector se utiliza el protocolo TCP/IP.

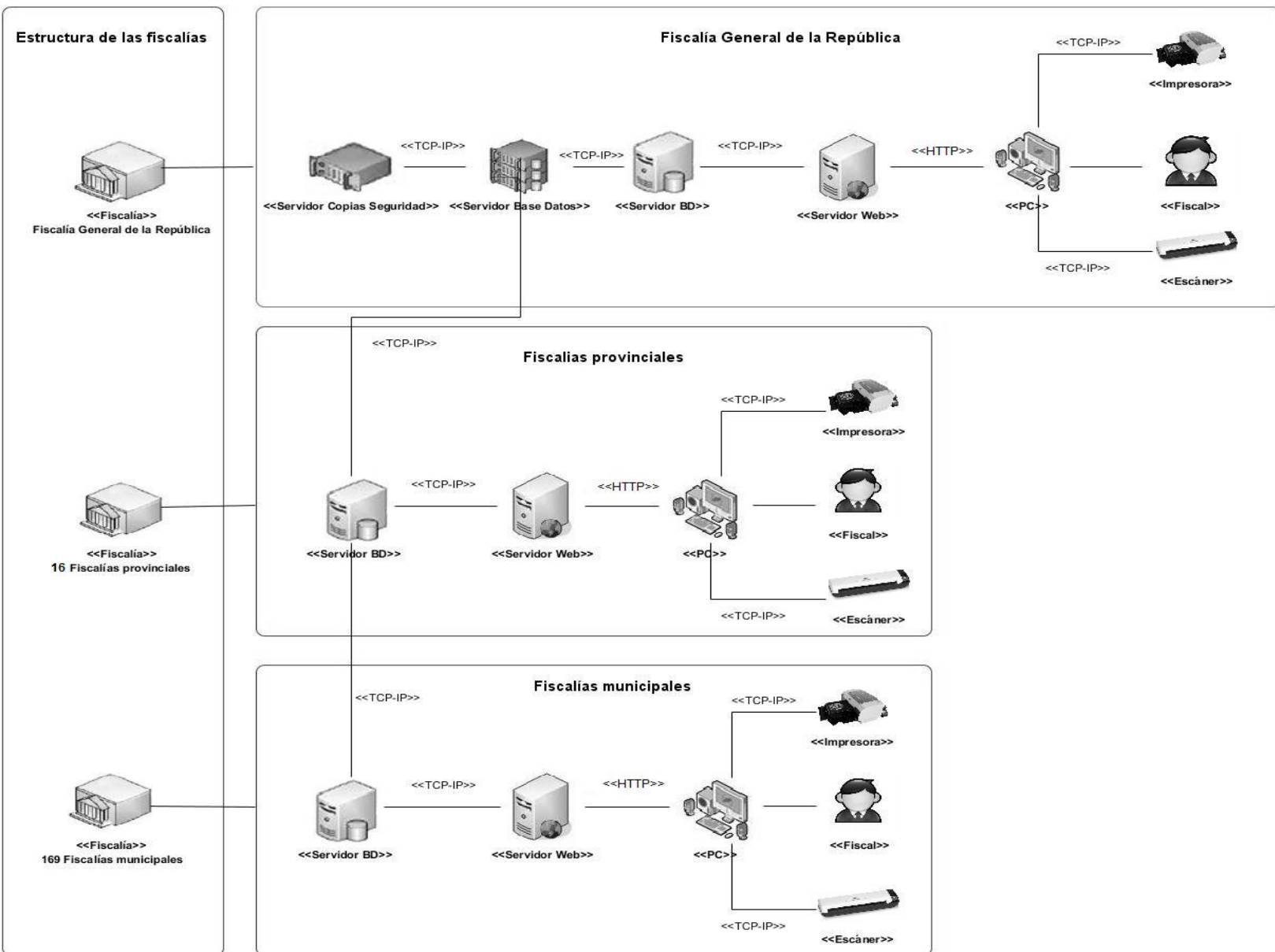


Figura 13. Diagrama de despliegue del módulo Civil y Familia.

3.3 Pruebas

Para la validación de la implementación de los procesos Expediente Prejudicial, Asuntos Notariales y Expropiación Forzosa, del módulo Civil y Familia, se seleccionó la prueba del camino básico, que es una técnica de prueba de caja blanca que permite 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.

También se empleó la prueba de caja negra Técnica de la Partición de Equivalencia, que divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.

3.3.1 Prueba del camino básico

A continuación se presentan los pasos para llevar a cabo la prueba del camino básico:

Notación de grafo de flujo: antes de considerar el método del camino básico se debe introducir una sencilla notación para la representación del flujo de control, denominada grafo de flujo (o grafo del programa). El grafo de flujo representa el flujo de control lógico.

Cada círculo, denominado nodo del grafo de flujo, representa una o más sentencias procedimentales. Un solo nodo puede corresponder a una secuencia de cuadros de proceso y a un rombo de decisión. Las flechas del grafo de flujo, denominadas aristas o enlaces, representan flujo de control y son análogas a las flechas del diagrama de flujo. Una arista debe terminar en un nodo, incluso aunque el nodo no represente ninguna sentencia procedimental. Las áreas delimitadas por aristas y nodos se denominan regiones. (9)

Complejidad ciclomática: la complejidad ciclomática proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez.

Un camino independiente es cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una nueva condición. En términos del grafo de flujo, un camino independiente está constituido por lo menos por una arista que no haya sido recorrida anteriormente a la definición del camino. (9)

La complejidad se puede calcular de tres formas:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- 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.

- 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 . (Cada nodo que contiene una condición se denomina nodo predicado y está caracterizado porque dos o más aristas emergen de él).

Obtención de casos de prueba: el método de prueba de camino básico se puede aplicar a un diseño procedimental detallado o a un código fuente. Usando el código como base, se dibuja el correspondiente grafo de flujo. Posteriormente se determina la complejidad ciclomática del grafo de flujo resultante. Después se determina un conjunto básico de caminos linealmente independientes. Finalmente se preparan los casos de prueba que forzarán la ejecución de cada camino del conjunto básico. (9)

Resultado de aplicar la prueba:

A continuación se presenta el código tomado como base para representar el procedimiento anteriormente explicado. Corresponde al método `salvarActualizarAsuntosNotarialesAction`.

```
public function salvarActualizarAsuntosNotarialesAction($paso, $id_proceso) {
    $peticion = $this->getRequest();
    $proceso = $this->getAsuntosNotarialesGtr()->crearAsuntosNotariales($id_proceso);
    $usuario = $this->getUser();
    $resultado = $this->verificarProcesoEnEdicion($proceso, $this->listaPasosAnteriores($paso), $paso, ConfigUtil::estado_edicion,
        $this->generateUrl('cf_listar_rollo_asuntos_notariales_paso', array('paso' => ConfigUtil::Iniciar_AN)));

    if (!is_null($resultado)) {
        return $resultado;
    }

    $frm = $this->createForm(new AdicionarAsuntosNotarialesType(), $proceso);
    if ($peticion->getMethod() == 'POST') {
        $frm->bind($peticion);
        if ($frm->isValid()) {

            if ($peticion->get(ConfigUtil::btn_siguiente)) {
                $proceso = $this->getAsuntosNotarialesGtr()->salvarActualizarAsuntosNotariales($proceso,
                    ConfigUtil::proceso_asuntos_notariales, $paso, ConfigUtil::estado_edicion,$usuario);
                return $this->redirect($this->generateUrl('cf_persona_proceso_asuntos_notariales', array('paso' => $paso, 'id_proceso' => $proceso->getIdProceso()));
            }
            else if ($peticion->get(ConfigUtil::btn_guardar)) {
                $proceso = $this->getAsuntosNotarialesGtr()->salvarActualizarAsuntosNotariales($proceso, ConfigUtil::proceso_asuntos_notariales,
                    $paso, ConfigUtil::estado_pendiente,$usuario);
                return $this->redirect($this->generateUrl('cf_rollo_asuntos_notariales', array('paso' => ConfigUtil::Iniciar_AN)));
            }
        }
    }
    return $this->render('CivilFamiliaBundle:AsuntosNotariales:adicionar_actualizar_asuntos_notariales.html.twig', array('frm' => $frm->createView(),
        'paso' => $paso, 'id_proceso' => $id_proceso));
}
```

Figura 14. Código fuente del método `salvarActualizarAsuntosNotarialesAction`.

Grafo de flujo a partir del código mostrado:

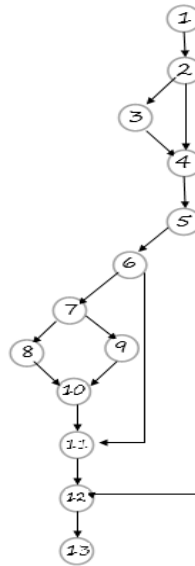


Figura 15. Grafo de flujo de la operación salvarActualizarAsuntosNotarialesAction.

Complejidad ciclomática:

1. $V(G) = 5$
2. $V(G) = A - N + 2$. $16 - 13 + 2 = 5$
3. $V(G) = P + 1$. $4 + 1 = 5$

El valor calculado define el número de caminos linealmente independientes del conjunto básico del programa, por lo que se obtienen 5 caminos:

Camino 1: 1-2-4-5-12-13

Camino 2: 1-2-3-4-5-12-13

Camino 3: 1-2-3-4-5-6-11-12-13

Camino 4: 1-2-3-4-5-6-7-8-10-11-12-13

Camino 5: 1-2-3-4-5-6-7-9-10-11-12-13

Luego de elaborado el grafo de flujo y los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos. Se escogen los datos de forma que

las condiciones de los nodos predicados estén adecuadamente establecidas, con el fin de comprobar cada camino.

Como parte de las pruebas de caja blanca aplicadas al módulo Civil y Familia se hizo empleo del framework PHPUnit, el cual es utilizado para realizar pruebas unitarias y analizar los resultados obtenidos en ellas.

Se realizó el procedimiento para los procesos Expediente Prejudicial, Asuntos Notariales y Expropiación Forzosa, donde se detectaron fallos en 8 de las pruebas en una primera iteración. Posteriormente fueron analizados y resueltos, lo que provocó un nivel de satisfacción aceptable.

A continuación se muestran los resultados de la aplicación de dichas pruebas.

Legend: Low: 0% to 35% Medium: 35% to 70% High: 70% to 100%

	Coverage								
	Lines			Functions / Methods			Classes		
Total		89.78%	3359/3741		97.11%	539/555		100%	90/90
Controller		88.75%	1610/1814		100%	132/132		100%	4/4
Entity		95.93%	684/713		94.66%	284/300		100%	55/55
Form		75.62%	301/398		100%	35/35		100%	25/25
Negocio		93.62%	764/816		100%	88/88		100%	4/4

Figura 16. Resultados de la aplicación de pruebas unitarias con PHPUnit.

Se concluye que el módulo Civil y Familia posee una correcta ejecución de su código y cumple con las funcionalidades requeridas.

3.3.2 Técnica de la partición de equivalencia

La técnica de la partición de equivalencia 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, incorrectos o que no aplican, para determinadas condiciones de entrada. Las condiciones de entrada son valores numéricos específicos, un rango de valores, un conjunto de valores relacionados o una condición lógica. A continuación se observan los resultados del caso de prueba para el escenario “Adicionar datos del dictamen de AN”. El resto de los casos de prueba correspondiente a los procesos objeto de la investigación se muestran en el documento 0122_Diseño de Casos de Prueba basados en requisitos. Módulo Civil y Familia.

Tabla 6. Escenario: Adicionar datos del dictamen de AN.

Escenario	Descripción	Fecha de recibida la solicitud	Fecha de recibido el expediente	Asunto	Breve descripción del contenido de la solicitud del abogado	Dictamen	Consideración del fiscal	Fecha de emisión	Respuesta del Sistema	Flujo central
Adicionar datos del dictamen de AN	Permite adicionar los datos del dictamen de AN, cuando la actuación es dictaminar.	V	V	V	V	V	V	V	El sistema adiciona el documento y lo asocia al rollo.	1- Selecciona la opción Dictaminar del menú principal del proceso. 2- Selecciona un rollo para asociarle un documento. 3- Muestra una pantalla con los datos a introducir. 4- Registra el documento y lo asocia al rollo.
		21/04/07	21/04/07	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Favorable	El fiscal ejerce ...	25/04/07		
		N/A	V	V	V	V	V	V	Debe mostrar un mensaje indicando que existen campos en blanco y que deben ser llenados antes de seleccionar la opción Finalizar.	
		Vacío	21/04/07	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Favorable	El fiscal ejerce ...	25/04/07		
		V	N/A	V	V	V	V	V	El fiscal ejerce ...	
		21/04/07	Vacío	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Favorable	El fiscal ejerce ...	25/04/07		
		V	V	N/A	V	V	V	V	El fiscal ejerce ...	
		21/04/07	21/04/07	Vacío	Que teniendo en cuenta lo dispuesto ...	Favorable	El fiscal ejerce ...	25/04/07		
		V	V	V	N/A	V	V	V	El fiscal ejerce ...	
		21/04/07	21/04/07	Perpetua memoria	Vacío	Favorable	El fiscal ejerce ...	25/04/07		
V	V	V	V	V	N/A	V	V			

		21/04/07	21/04/07	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Vacío	El fiscal ejerce ...	25/04/07	
		V	V	V	V	V	N/A	V	
		21/04/07	21/04/07	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Favorable	Vacío	25/04/07	
		V	V	V	V	V	V	N/A	
		21/04/07	21/04/07	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Favorable	El fiscal ejerce ...	Vacío	
		I	V	V	V	V	V	V	Debe mostrar un mensaje indicando que la fecha de recibida la solicitud no debe ser mayor que la fecha de registro del rollo.
		21/04/58	21/04/07	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Favorable	El fiscal ejerce ...	25/04/07	
		V	I	V	V	V	V	V	
		21/04/07	21/04/58	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Favorable	El fiscal ejerce ...	25/04/07	Debe mostrar un mensaje indicando que la fecha de recibido el

									<p>expediente no debe ser mayor que la fecha de registro del rollo.</p>
	V	V	V	V	V	V	I		<p>Debe mostrar un mensaje indicando que la fecha de emisión debe ser mayor que la fecha de registro del rollo.</p>
	21/04/07	21/04/07	Perpetua memoria	Que teniendo en cuenta lo dispuesto ...	Favorable	El fiscal ejerce ...	21/04/03		

Fueron objeto de pruebas de caja negra los 47 requisitos correspondiente a los procesos Expediente Prejudicial, Asuntos Notariales y Expropiación Forzosa, lo que mostró como resultado en el registro de defectos y dificultades un total de 12 no conformidades detectadas, lo que constituye un 25.53%. En la segunda iteración se corrigieron las no conformidades y se obtuvo un 100% de satisfacción en los casos de prueba.

3.3.3 Validación de la propuesta de solución

Para validar la propuesta de solución se planificó un encuentro entre la Fiscalía Municipal de Playa y el proyecto SIGEF II, pues en esta instancia se desarrollan los procesos civiles y de familia implementados.

Como parte de las actividades que se desarrollaron, se realizó una comparación en cuanto al comportamiento de las variables de la investigación: control y celeridad. Los resultados obtenidos fueron los siguientes:

Tabla 7. Validación para la variable: control

Dimensiones de la variable	Antes	Después
Reportes estadísticos	Los fiscales llenan los datos de la plantilla de reportes con la información encontrada en los libros de control. Esta información puede estar sujeta a errores, teniendo en cuenta que el conteo es manual.	Se muestra la información real de los rollos que se han creado. Tiene como desventaja que aquellos datos que no se hayan introducido en el sistema no serán contados.
Acceso a la información	No se conocen los detalles del proceso si no se tiene el rollo en la mano.	Si se cuenta con los permisos necesarios, se puede acceder a la información del rollo desde cualquier instancia.
Datos generales del proceso. (Personas del proceso, rollo, expediente).	Siempre que sea requerido el dato en el proceso o en procesos posteriores, será escrito nuevamente.	Los rollos y expedientes una vez introducidos se muestran en forma de detalles. Respecto a las personas, una vez en el sistema, serán seleccionadas cuando sea necesario.

Para evaluar la variable celeridad se tuvieron en cuenta los 180 trámites promedio que se realizan en un año en esta fiscalía y los 900 que aproximadamente se ejecutan en 5 años. Al verificar en el sistema la dimensión de la variable, con la misma cantidad de trámites en los mismos períodos, se obtuvieron los resultados que se muestran a continuación.

Tabla 8. Validación para la variable: celeridad

Dimensiones de la variable	Antes		Después
Tiempo de creación de un reporte.	(información del año actual)	Aproximadamente 2 horas	<1 segundos

	(información de hasta 5 años)	Hasta 1 día de trabajo	<2 segundos
Tiempo de redacción y revisión de un documento.	Aproximadamente 15 minutos		Aproximadamente 10 minutos
Tiempo de búsqueda de un rollo.	(información del año actual)	5 a 15 minutos	1 a 5 segundos
	(información de hasta 5 años)	15 a 30 minutos	

A través del desarrollo del capítulo, se logró definir la organización del código, además de implementar los elementos de diseño en términos de los elementos de implementación. También se pudo distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue. Se llevaron a cabo las pruebas de caja blanca, permitiendo verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. Otro de los aspectos tomados en cuenta fue la verificación del sistema desarrollado mediante pruebas de caja negra a las funcionalidades, donde se detectaron un número significativo de no conformidades, a las cuales se les dio solución, obteniendo finalmente una aplicación que satisface los requisitos definidos. Finalmente fue validada la propuesta de solución a través de un encuentro en la Fiscalía Municipal para verificar el cumplimiento del objetivo de la investigación planteado.

Conclusiones Generales

Luego de realizar el presente trabajo se concluye que:

El análisis de las soluciones informáticas desarrolladas en otros países demostró la necesidad de incorporar el módulo Civil y Familia al sistema que se está implementando, y no como una solución aislada.

En la disciplina Requisitos se lograron identificar los requisitos funcionales y no funcionales, teniendo en cuenta las restricciones o políticas a cumplir por el sistema, que deben ser implementadas, identificando de esta manera 159 requisitos funcionales y 19 requisitos no funcionales. El artefacto especificación de requisitos de software permitió validar con el cliente los prototipos no funcionales de interfaz de usuario.

El diseño, mediante el uso de patrones y los diagramas generados, permitió implementar los procesos Expediente Prejudicial, Asuntos Notariales y Expropiación Forzosa de forma eficiente.

El estudio y uso de las métricas fue un elemento clave a la hora de garantizar la calidad de los resultados en cada una de las disciplinas, midiendo el grado de corrección de los artefactos obtenidos y garantizando la continuidad del proceso de desarrollo del software. Las métricas de requisitos lograron una necesaria especificidad y estabilidad en los mismos, y las métricas de diseño permitieron verificar la correcta modelación de los distintos elementos.

La realización de las pruebas de caja negra y caja blanca evaluaron el funcionamiento y rendimiento de los procesos objeto de la investigación, demostrando conformidad con los requisitos.

Se logró validar la propuesta de solución frente al problema planteado, teniendo en cuenta la información obtenida en la fiscalía tomada como muestra, demostrando que existe un mayor control sobre los procesos y se disminuye el tiempo de creación de reportes y documentos, contribuyendo de este modo a la mejora de la gestión de los procesos civiles y de familia.

Recomendaciones

- Implementar los restantes procesos del módulo Civil y Familia de forma que se obtenga un producto completamente funcional.
- Realizar otras iteraciones en las que se puedan identificar otras funcionalidades que puedan ser sugeridas por los clientes o en caso de que en algún momento las leyes puedan ser modificadas.
- Desplegar los procesos implementados para mejorar su gestión en las fiscalías correspondientes.

Bibliografía

1. *Ley de la Fiscalía General de la República*. La Habana : s.n., 2007.
2. **CALDERIO, BLAS ROCA**. *Ley de Procedimiento Civil, Administrativo y Laboral*. La Habana : s.n., 1977.
3. *Código Civil (publicado en la Gaceta Oficial Extraordinaria)*. **PARDO, FLAVIO BRAVO**. 1987, Gaceta Oficial de la República de Cuba.
4. *CÓDIGO DE PROCEDIMIENTO CIVIL DE LA REPÚBLICA DE NICARAGUA*. 2002.
5. *Código Federal de Procedimientos Civiles de los Estados Unidos Mexicanos*. 2008.
6. *Sistema Automatizado de Gestion Judicial de la República de Panamá*. [Online] [Cited: Diciembre 3, 2012.] <http://www.organojudicial.gob.pa/servicios/sistema-automatizado-de-gestion-judicial-sagj-2/>.
7. Poder Judicial Provincia de San Juan-República Argentina. [Online] [Cited: Diciembre 3, 2012.] <http://www.jussanjuan.gov.ar/proyectos.php..>
8. Sistema de Información del Ministerio Fiscal. [Online] [Cited: Diciembre 4, 2012.] http://www.astic.es/sites/default/files/articulosboletic/mono_6_1.pdf.
9. **Pressman, Roger S**. *Ingeniería de Software Un Enfoque Práctico*.
10. *Acerca de la implementación de los modelos de calidad en la construcción de software en México*. **Edna Gutiérrez Gasca, Agustín F. Gutiérrez, Aurora Pérez Rojas, Luis Márquez López**. Ciudad de México : Tecnura, 2010.
11. **Entidades, Centro de In formatización de Gestión de**. *Modelo de desarrollo de software*. 2012.
12. **Machado, Ing. Yenier Figueroa**. *Proyecto Técnico del Sistema de Informatización de la Gestión de las Fiscalías II*. 2011.
13. Axure. [En línea] [Citado el: 5 de Noviembre de 2012.] <http://www.axure.com>.
14. PostgreSQL-es. [Online] [Cited: Noviembre 22, 2012.] <http://www.postgresql.org.es/node/655>.
15. Visual Paradigm for UML. [En línea] 1999. [Citado el: 5 de Noviembre de 2012.] <http://www.visual-paradigm.com>.
16. **Schmuller, Joseph**. *Aprendiendo UML en 24 horas*.
17. desarrolloweb.com. [Online] [Cited: Abril 29, 2013.] <http://www.desarrolloweb.com/articulos/que-es-html5.html>.
18. CSS3.com. [Online] [Cited: Abril 29, 2013.] <http://www.css3.com/>.
19. **Pérez, Javier Eguíluz**. *Introducción a AJAX*. 2007.

20. JQuery. [Online] [Cited: Noviembre 17, 2012.] <http://jquery.com/>.
21. PHP. [Online] [Cited: Abril 29, 2013.] http://php.net/releases/5_3_0.php.
22. NetBeans. [Online] [Cited: Abril 29, 2013.]
<https://netbeans.org/community/news/show/1562.html>.
23. Symfony. Open-Source PHP Web marco de trabajo. [Online] [Cited: Noviembre 17, 2012.]
<http://www.symfony-project.org/>.
24. doctrine. [Online] [Cited: Abril 29, 2013.] <http://docs.doctrine-project.org/en/latest/tutorials/getting-started.html>.
25. Programación en castellano. [Online] [Cited: Diciembre 12, 2012.]
http://www.programacion.com/articulo/conceptos_basicos_de_orm_object_relational_mapping_349.
26. Twig en Español. [Online] [Cited: Abril 29, 2013.] <http://gitnacho.github.io/Twig/>.
27. Twig. [Online] [Cited: Noviembre 17, 2012.] <http://www.twigsoftware.com/home.html>.
28. TortoiseSVN. Un cliente de Subversion para Windows. [Online] [Cited: Noviembre 22, 2012.] http://tortoisesvn.net/docs/release/TortoiseSVN_es/index.html.
29. Subversion. [Online] [Cited: Noviembre 22, 2012.] <http://subversion.apache.org/>.
30. **LARMAN, Craig.** *UML y patrones*. 1999.
31. Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica. [Online] [Cited: Enero 31, 2013.]
http://vinculando.org/articulos/sociedad_america_latina/propuesta_guia_de_medidas_para_evaluacion_sistemas_informacion.html.
32. **Chidamber S. R., Kemerer C. F.** "A metric suite forsz". *s.l. : IEEE Transactions on Software Engineering*. 1994.
33. **Sommerville, Ian.** *Ingeniería del software. Séptima edición*. Madrid : Pearson Addison Wesley, 2005. ISBN: 84-7829-074-5.
34. **Miguel Gea, Fco Luis Gutiérrez, José Luis Garrido, José J. Cañas.** Teorías y Modelos Conceptuales para un Diseño basado en Grupos. [Online] [Cited: Marzo 13, 2013.]
http://lsi.ugr.es/~mgea/invest/articulos/gea_interacc03.pdf.
35. **JACOBSON, Ivar, RUMBAUGH, James and BOOCH, Grady,.** *El proceso unificado de desarrollo*. 2000.
36. symfony.es. [Online] [Cited: Junio 7, 2013.] <http://fabien.potencier.org/article/11/what-is-dependency-injection>.

Anexos

Anexo 1: Lista de requisitos.

RF_PDC_CF_1	Adicionar Expediente Prejudicial
RF_PDC_CF_2	Actualizar Expediente Prejudicial
RF_PDC_CF_3	Adicionar decisión sobre promoción del asunto
RF_PDC_CF_4	Actualizar decisión sobre promoción de asunto
RF_PDC_CF_5	Visualizar detalles del expediente prejudicial
RF_PDC_CF_6	Listar expedientes prejudiciales
RF_PDC_CF_7	Buscar expediente prejudicial
RF_PDC_CF_8	Adicionar dictamen denegatorio de asuntos que se le interesen al fiscal
RF_PDC_CF_9	Actualizar dictamen denegatorio de asuntos que se le interesen al fiscal
RF_PDC_CF_10	Mostrar vista previa del dictamen denegatorio de asuntos que se le interesen al fiscal
RF_PDC_CF_11	Adicionar comunicación al solicitante
RF_PDC_CF_12	Actualizar comunicación al solicitante
RF_PDC_CF_13	Mostrar vista previa de la comunicación al solicitante
RF_PDC_CF_14	Editar persona menor de edad en el asunto tutela
RF_PDC_CF_15	Editar incapacitado en Tutela
RF_PDC_CF_16	Editar padres en tutela
RF_PDC_CF_17	Editar parientes en tutela
RF_PDC_CF_18	Editar Solicitante en Tutela
RF_PDC_CF_19	Adicionar datos generales del rollo Jurisdicción Voluntaria
RF_PDC_CF_20	Adicionar rollo Jurisdicción Voluntaria por promoción de asunto
RF_PDC_CF_21	Adicionar rollo Jurisdicción Voluntaria por dictamen
RF_PDC_CF_22	Actualizar datos generales del rollo Jurisdicción Voluntaria
RF_PDC_CF_23	Actualizar rollo Jurisdicción Voluntaria por dictamen

RF_PDC_CF_24	Buscar rollo Jurisdicción Voluntaria
RF_PDC_CF_25	Visualizar detalles del rollo de Jurisdicción Voluntaria
RF_PDC_CF_26	Listar rollos de Jurisdicción Voluntaria
RF_PDC_CF_27	Generar vista previa de la carátula del rollo de Jurisdicción Voluntaria
RF_PDC_CF_28	Generar vista previa de la carátula del rollo del fiscal en la tutela de Jurisdicción Voluntaria
RF_PDC_CF_29	Adicionar Dictamen Jurisdicción Voluntaria
RF_PDC_CF_30	Actualizar Dictamen Jurisdicción Voluntaria
RF_PDC_CF_31	Mostrar Vista Previa del Dictamen Jurisdicción Voluntaria
RF_PDC_CF_32	Adicionar datos del Escrito promoviendo Tutela
RF_PDC_CF_33	Actualizar datos del Escrito promoviendo Tutela
RF_PDC_CF_34	Mostrar Vista Previa del Escrito promoviendo Tutela de un menor de edad
RF_PDC_CF_35	Mostrar Vista Previa del Escrito promoviendo Tutela de un incapacitado
RF_PDC_CF_36	Adicionar Escrito promoviendo Asunto
RF_PDC_CF_37	Actualizar Escrito promoviendo Asunto
RF_PDC_CF_38	Mostrar Vista Previa del Escrito promoviendo Asunto
RF_PDC_CF_39	Adicionar datos del escrito promoviendo incapacidad
RF_PDC_CF_40	Actualizar datos del Escrito promoviendo Incapacidad
RF_PDC_CF_41	Mostrar Vista Previa del Escrito promoviendo Incapacidad
RF_PDC_CF_42	Adicionar rollo Asuntos Notariales
RF_PDC_CF_43	Actualizar rollo Asuntos Notariales
RF_PDC_CF_44	Buscar rollo Asuntos Notariales
RF_PDC_CF_45	Visualizar detalles del rollo Asuntos Notariales
RF_PDC_CF_46	Listar rollos de Asuntos Notariales
RF_PDC_CF_47	Generar vista previa de la Carátula del rollo de Asuntos Notariales por dictamen del fiscal

RF_PDC_CF_48	Generar vista previa de la Carátula del rollo de Asuntos Notariales por representación del fiscal
RF_PDC_CF_49	Adicionar Firma de Asuntos Notariales
RF_PDC_CF_50	Actualizar Firma de Asuntos Notariales
RF_PDC_CF_51	Adicionar Dictamen de Asuntos Notariales
RF_PDC_CF_52	Actualizar Dictamen Asuntos Notariales
RF_PDC_CF_53	Mostar Vista Previa del Dictamen Asuntos Notariales
RF_PDC_CF_54	Adicionar rollo Expropiación Forzosa
RF_PDC_CF_55	Actualizar rollo Expropiación Forzosa
RF_PDC_CF_56	Buscar rollo Expropiación Forzosa
RF_PDC_CF_57	Visualizar detalles del rollo Expropiación Forzosa
RF_PDC_CF_58	Listar rollos de Expropiación Forzosa
RF_PDC_CF_59	Generar vista previa de la Carátula del rollo Expropiación Forzosa
RF_PDC_CF_60	Adicionar Escrito de contestación a la demanda
RF_PDC_CF_61	Actualizar Escrito de contestación a la demanda
RF_PDC_CF_62	Mostrar vista previa del Escrito de contestación a la demanda
RF_PDC_CF_63	Adicionar rollo de Recursos en la instancia inferior
RF_PDC_CF_64	Actualizar rollo de Recursos en la instancia inferior
RF_PDC_CF_65	Adicionar rollo de Recurso de Apelación en la instancia provincial
RF_PDC_CF_66	Actualizar rollo de Recurso de Apelación en la instancia provincial
RF_PDC_CF_67	Buscar rollo Recurso Apelación en la Fiscalía Municipal
RF_PDC_CF_68	Buscar rollo Recurso Apelación en la Fiscalía Provincial
RF_PDC_CF_69	Visualizar detalles del Rollo Recurso Apelación en la Fiscalía Municipal
RF_PDC_CF_70	Visualizar detalles del Rollo Recurso Apelación en la Fiscalía Provincial
RF_PDC_CF_71	Listar rollos de Recursos de Apelación y Casación
RF_PDC_CF_72	Listar rollos en tiempo de Apelación

RF_PDC_CF_73	Listar información para personería en Recurso de Apelación
RF_PDC_CF_74	Generar Vista Previa de la Carátula del rollo de Recurso Apelación
RF_PDC_CF_75	Adicionar datos del escrito estableciendo Recuso de Apelación
RF_PDC_CF_76	Actualizar datos del escrito estableciendo Recuso de Apelación
RF_PDC_CF_77	Mostrar Vista Previa del Escrito Estableciendo Recuso de Apelación
RF_PDC_CF_78	Adicionar datos de envío a la instancia superior con recurso establecido y actuación no recurrente
RF_PDC_CF_79	Actualizar datos de envío a la instancia superior con recurso establecido y actuación no recurrente
RF_PDC_CF_80	Adicionar Escrito de personería como no recurrente en el recurso de Apelación
RF_PDC_CF_81	Actualizar Escrito de personería como no recurrente en el recurso de Apelación
RF_PDC_CF_82	Mostrar vista previa del escrito de personería como no recurrente en el Recurso de Apelación
RF_PDC_CF_83	Adicionar escrito de personería como recurrente en el Recurso de Apelación
RF_PDC_CF_84	Actualizar escrito de personería como recurrente en el Recurso de Apelación
RF_PDC_CF_85	Mostrar vista previa del escrito de personería como recurrente en el Recurso de Apelación
RF_PDC_CF_86	Mostrar conclusión del proceso en la Fiscalía Municipal
RF_PDC_CF_87	Adicionar datos del rollo Recurso de Casación en la Fiscalía General
RF_PDC_CF_88	Buscar rollo Recurso Casación en la Fiscalía Provincial
RF_PDC_CF_89	Buscar rollo Recurso Casación en la Fiscalía General
RF_PDC_CF_90	Visualizar detalles del Rollo Recurso Casación en la Fiscalía Provincial
RF_PDC_CF_91	Visualizar detalles del Rollo Recurso Casación en la Fiscalía General
RF_PDC_CF_92	Listar rollos de procesos en tiempo para casar sentencia
RF_PDC_CF_93	Listar información para personería en Recurso de Casación

RF_PDC_CF_94	Generar Vista Previa de la Carátula del rollo de Recurso Casación
RF_PDC_CF_95	Adicionar datos del Escrito Estableciendo Recuso de Casación
RF_PDC_CF_96	Actualizar datos del Escrito Estableciendo Recuso de Casación
RF_PDC_CF_97	Adicionar motivo de Casación
RF_PDC_CF_98	Actualizar motivo de Casación
RF_PDC_CF_99	Eliminar motivo de Casación
RF_PDC_CF_100	Listar motivos de Casación
RF_PDC_CF_101	Visualizar detalles de motivos de Casación
RF_PDC_CF_102	Mostrar vista previa del Escrito Estableciendo Recuso de Casación
RF_PDC_CF_103	Adicionar datos del Escrito de personería como recurrente en el recurso de Casación
RF_PDC_CF_104	Actualizar datos del Escrito de personería como recurrente en el recurso de Casación
RF_PDC_CF_105	Mostrar vista previa del Escrito de personería como recurrente en el recurso de Casación
RF_PDC_CF_106	Adicionar datos del Escrito de personería como no recurrente en el recurso de Casación
RF_PDC_CF_107	Actualizar Escrito de personería como no recurrente en el recurso de Casación
RF_PDC_CF_108	Mostrar vista previa del Escrito de personería como no recurrente en el recurso de Casación
RF_PDC_CF_109	Mostrar vista previa del Escrito de Personería del Fiscal, Sosteniendo, Ampliando y Agregando motivo
RF_PDC_CF_110	Adicionar datos de envío a la FGR sin establecer recurso en la instancia inferior
RF_PDC_CF_111	Actualizar datos de envío a la FGR sin establecer recurso en la instancia inferior
RF_PDC_CF_112	Adicionar rollo de Revisión Civil
RF_PDC_CF_113	Actualizar rollo de Revisión Civil

RF_PDC_CF_114	Buscar sentencia origen
RF_PDC_CF_115	Adicionar datos de la sentencia origen
RF_PDC_CF_116	Actualizar datos de la sentencia origen
RF_PDC_CF_117	Buscar rollo de Revisión Civil
RF_PDC_CF_118	Visualizar detalles del rollo de Revisión Civil
RF_PDC_CF_119	Listar rollos de Revisión Civil
RF_PDC_CF_120	Generar vista previa de la carátula del rollo de Revisión Civil
RF_PDC_CF_121	Adicionar escrito de demanda del Fiscal en Revisión Civil
RF_PDC_CF_122	Actualizar escrito de demanda del Fiscal en Revisión Civil
RF_PDC_CF_123	Mostrar vista previa del escrito de demanda del Fiscal en Revisión Civil
RF_PDC_CF_124	Adicionar datos del escrito de contestación a la demanda en Revisión Civil
RF_PDC_CF_125	Actualizar escrito de contestación a la demanda en Revisión Civil
RF_PDC_CF_126	Mostrar vista previa del escrito de demanda del Fiscal en Revisión Civil
RF_PDC_CF_127	Adicionar datos del expediente del Tribunal cuando el fiscal actúa como actor
RF_PDC_CF_128	Actualizar datos del expediente del Tribunal cuando el fiscal actúa como actor
RF_PDC_CF_129	Adicionar datos del expediente del Tribunal cuando el fiscal actúa como recurrente
RF_PDC_CF_130	Actualizar datos del expediente del Tribunal cuando el fiscal actúa como recurrente
RF_PDC_CF_131	Adicionar datos del expediente del Tribunal cuando el fiscal actúa como demandado o No Recurrente
RF_PDC_CF_132	Actualizar datos del expediente del Tribunal cuando el fiscal actúa como demandado o No Recurrente
RF_PDC_CF_133	Adicionar datos del expediente del Tribunal cuando el fiscal actúa emitiendo dictamen

RF_PDC_CF_134	Actualizar datos del expediente del Tribunal cuando el fiscal actúa emitiendo dictamen
RF_PDC_CF_135	Adicionar datos de presentación del documento ante el Tribunal
RF_PDC_CF_136	Actualizar datos de presentación del documento ante el Tribunal
RF_PDC_CF_137	Adicionar datos conclusivos del proceso
RF_PDC_CF_138	Actualizar datos conclusivos del proceso
RF_PDC_CF_139	Adicionar datos del acta de entrevista
RF_PDC_CF_140	Actualizar datos del acta de entrevista
RF_PDC_CF_141	Listar actas de entrevistas
RF_PDC_CF_142	Eliminar acta de entrevista
RF_PDC_CF_143	Mostrar vista previa del acta de entrevista
RF_PDC_CF_144	Generar reporte
RF_PDC_CF_145	Generar reportes de procesos contenciosos y otros procedimientos en los que interviene el fiscal en la FM
RF_PDC_CF_146	Generar reportes de procesos contenciosos y otros procedimientos en los que interviene el fiscal en la FP
RF_PDC_CF_147	Generar reportes de asuntos de JV con participación del fiscal
RF_PDC_CF_148	Generar reportes de tutela
RF_PDC_CF_149	Generar reporte de AN
RF_PDC_CF_150	Generar reporte de expedientes de patria potestad
RF_PDC_CF_151	Generar reportes de RA en los que participó el fiscal
RF_PDC_CF_152	Generar reporte resumen de procesos contenciosos y otros procedimientos en los que interviene el fiscal
RF_PDC_CF_153	Generar reporte resumen de JV y Alimentos
RF_PDC_CF_154	Generar reporte resumen de tutela
RF_PDC_CF_155	Generar reporte resumen de AN
RF_PDC_CF_156	Generar reporte resumen de expedientes de patria potestad

- RF_PDC_CF_157 Generar reporte resumen del total de RA
- RF_PDC_CF_158 Generar reporte resumen del total de RC en los que participó el fiscal
- RF_PDC_CF_159 Generar reporte del total de PRC

Glosario de términos

Proceso: el proceso judicial se iniciará con la presentación de la demanda verbal o escrita.

Asunto: asuntos civiles a los cuales los litigantes se someten expresa o tácitamente.

Rollo fiscal: en los rollos se incluirán todos los escritos que se originen en los procesos por cada uno de los sujetos que tengan el carácter de partes.

Expediente prejudicial: la presentación al fiscal para que promueva o solicite algún asunto ante el Tribunal.

Resolución judicial: cuando resulta procedente emitir un juicio para que se restablezca la legalidad quebrantada. Las resoluciones judiciales se consignarán por escrito en forma de autos y sentencias.

Auto: adoptarán la forma de autos, las resoluciones que decidan incidentes o puntos esenciales que afecten de una manera directa la personería o la competencia; rechacen de plano o decidan la procedencia o improcedencia de la recusación; rechacen el recibimiento a prueba o la admisión de alguna diligencia de ésta; resuelvan los recursos contra las providencias o autos, y las demás que según las leyes o de acuerdo con su naturaleza deban dictarse en forma razonada. Los autos se acordarán dentro de los cinco días siguientes al en que se dé cuenta.

Sentencia: se dictarán en forma de sentencia las resoluciones que pongan fin al proceso en la instancia o en el recurso de casación o apelación, según proceda, o decidan cuestiones o excepciones que impidan entrar en el fondo del asunto.

Dictamen: cuando resulta procedente emitir un juicio o criterio técnico en virtud de consulta formulada o para la evaluación y respuesta de un asunto sometido a su consideración, que no implique ninguna obligación para persona o institución ajena a la Fiscalía.

Acto jurídico: el acto jurídico es una manifestación lícita, de voluntad, expresa o tácita, que produce los efectos dispuestos por la ley, consistentes en la constitución, modificación o extinción de una relación jurídica.

Personería: acto de presentarse el fiscal ante el tribunal popular.

Actor o demandante: el fiscal, al intervenir en el proceso, puede adherirse, ampliar o modificar la pretensión formulada por el actor principal o alegar otras nuevas aunque sin alterar sustancialmente lo que sea objeto del pleito.

Demandado: el fiscal, al intervenir en el proceso, puede alegar cuantas excepciones estime pertinentes.

Representado: el fiscal representa y defiende a los menores, incapacitados y ausentes, hasta que se les provea de tutores, representantes o encargados del cuidado de sus personas y de la defensa de sus bienes y derechos.

Recurrente: el fiscal como parte recurrente deberá exponer brevemente las razones en que fundamente su recurso y proponer las pruebas de que intente valerse, así como pedir la celebración de vista, si lo estima conveniente.

No Recurrente: el fiscal como parte no recurrente podrá adherirse u oponerse al recurso, y en cualquiera de estos casos, podrá presentar sus alegaciones y proponer las pruebas pertinentes ante el Tribunal, en los extremos de la resolución recurrida que le sean desfavorables.