

Universidad de las Ciencias Informáticas
“Facultad 15”



Análisis del Módulo Ordinario del Proyecto
Sistema de Gestión Fiscal

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

Autor: Maylin Marin Galeano.

Tutor: Ing. Maylen Castillo Prieto.

Junio, 2010
Curso 2009 - 2010



...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo de la producción del hombre del mañana. Y en esta producción, en esta dirección está comprendida la producción de sí mismos....

Declaración de Autoría

Declaro que soy la única autora de este trabajo y autorizo al Departamento de Especialidades [Facultad 15], de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los 29 días del mes de Junio del año 2010.

Autor: Maylin Marin Galeano

Tutor: Ing. Maylen Castillo Prieto

DEDICATORIA

Dedicado especialmente a:

Mis padres y abuelas por todo el amor, cariño y apoyo que me brindan. Por ser un ejemplo de sacrificio y esfuerzo, sin su guía no hubiese llegado hasta aquí, ustedes se merecen esto y mucho más.

Mi hermanita Evelyn que te sirva de ejemplo para esforzarte y llegar a ser una buena profesional.

A Gustavo por ayudarme y estar siempre conmigo en estos cinco años de universidad, por ser mi novio y amigo.

Toda mi familia, que tanto se preocupa por mí.

Maylin

AGRADECIMIENTOS

Especialmente a:

A nuestro Comandante en Jefe Fidel Castro, por ser el gran motor impulsor, en la realización de un sueño, la educación en Cuba.

A la UCI, por influir notablemente en mi formación como profesional y en la madurez de mis acciones.

Mis padres por ser lo más grande del mundo.

Mi hermanita y abuela por quererme tanto.

Mi familia por siempre confiar en mí.

Mis compañeros, los nuevos y los viejos, de cinco años de estudio y esfuerzo, por brindarme su amistad.

Todos los profesores que siempre me ayudaron en estos cinco años de universidad.

Mi tutora por brindarme su ayuda.

A todos, los que de una forma u otra, han colaborado con la realización de este trabajo.

RESUMEN

El presente trabajo está centrado específicamente en el análisis del módulo Ordinario del proyecto Sistema de Gestión Fiscal. El mismo tiene como fin traducir las necesidades del cliente en un lenguaje entendible por los desarrolladores en el módulo Ordinario del proyecto Sistema de Gestión Fiscal. Para cumplir el objetivo general trazado, se desarrollaron artefactos propios de los flujos Modelado del Negocio y Requerimientos como: Diagrama de casos de uso del negocio, Modelo de objetos, Diagramas de Actividades, Diagrama de casos de uso del sistema.

Se realiza un estudio de las Metodologías de Desarrollo, Lenguajes de Modelado, de las Herramientas CASE y de las Herramientas para el modelado de prototipos de interfaz de usuario a utilizar para el desarrollo del trabajo. Por último se evalúa el trabajo realizado mediante métricas dirigidas a medir la calidad de la especificación de requisitos y de los casos de uso del sistema, igualmente se utilizan técnicas como las Revisiones, Listas de Chequeo y Prototipos no funcionales de interfaz, obteniéndose resultados positivos.

El documento consta de tres capítulos. El primero aborda el estudio del estado del arte del contenido teórico que se debe conocer para su realización. En el capítulo dos se realiza una caracterización del negocio, describiendo las principales reglas, actores y trabajadores del mismo. Se presentan los requisitos que debe cumplir la aplicación y se exponen los artefactos resultantes del módulo propuesto y por último en el tercer capítulo se validan los resultados obtenidos en la investigación

Palabras claves: Proceso Ordinario, Requisitos, Casos de Uso, Ingeniería de Requisitos, Metodologías, Herramientas CASE, Lenguajes Modelado y Métricas.

ÍNDICE DE CONTENIDOS

| | |
|--|------------|
| DEDICATORIA | I |
| AGRADECIMIENTOS | II |
| RESUMEN | III |
| INTRODUCCIÓN | 1 |
| | |
| CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA | 5 |
| 1.1 Introducción | 5 |
| 1.2 Sistemas de Gestión Fiscal en el Mundo | 5 |
| 1.3 Sistemas de Gestión Fiscal en Cuba | 8 |
| 1.4 Metodologías de Desarrollo de Software | 8 |
| 1.4.1 Justificación de la metodología seleccionada..... | 12 |
| 1.5 Lenguajes de Modelado | 13 |
| 1.5.1 Justificación del lenguaje de modelado seleccionado | 14 |
| 1.6 Herramientas CASE (Computer Aided Software Engineering, CASE) | 15 |
| 1.6.1 Justificación de la herramienta CASE seleccionada..... | 17 |
| 1.7 Herramientas de modelado de Prototipos de interfaz de usuario | 17 |
| 1.7.1 Justificación de la Herramienta de modelado de Prototipos de interfaz de usuario | 19 |
| 1.8 Ingeniería de Requisitos | 19 |
| 1.8.1 Definición de requisito de software..... | 19 |
| 1.8.2 Tipos de Requisitos..... | 20 |
| 1.8.3 Etapas de desarrollo de la ingeniería de requisitos | 21 |
| 1.8.4 Técnicas para la captura de requisitos | 22 |
| 1.8.5 Técnicas para la validación de requisitos | 25 |
| 1.9 Métricas | 26 |
| 1.10 Patrones de casos de uso | 27 |
| 1.11 Conclusiones parciales | 29 |
| | |
| CAPÍTULO II: MODELADO DEL NEGOCIO Y SISTEMA | 30 |
| 2.1 Introducción | 30 |
| 2.2 Modelado del negocio | 30 |
| 2.3 Modelado del Negocio del Proceso Ordinario | 30 |
| 2.3.1 Descripción del Negocio | 30 |
| 2.3.2 Actores del negocio..... | 32 |
| 2.3.3 Trabajadores del negocio..... | 33 |
| 2.3.4 Casos de uso del negocio | 34 |

| | |
|---|-----------|
| 2.3.5 Modelo de casos uso del negocio | 35 |
| 2.3.6 Descripción Textual de los casos de uso del negocio | 36 |
| 2.3.7 Reglas del negocio..... | 41 |
| 2.4 Requisitos | 42 |
| 2.4.1 Actores del sistema..... | 46 |
| 2.4.2 Casos de uso del sistema | 47 |
| 2.4.3 Modelo de casos de uso del Sistema..... | 48 |
| 3.2.4 Descripción Textual de los casos de uso del sistema..... | 48 |
| 2.5 Conclusiones parciales..... | 51 |
| | |
| CAPÍTULO III: VALIDACIÓN | 52 |
| 3.1 Introducción..... | 52 |
| 3.1.3 Revisiones | 52 |
| 3.1.2 Listas de Chequeo | 52 |
| 3.1.3 Prototipos no funcionales | 52 |
| 3.2 Métricas..... | 53 |
| 3.3 Conclusiones parciales..... | 59 |
| CONCLUSIONES GENERALES..... | 60 |
| RECOMENDACIONES..... | 61 |
| REFERENCIAS BIBLIOGRÁFICAS | 62 |
| GLOSARIO DE TÉRMINOS..... | 64 |
| ANEXOS..... | 65 |

INTRODUCCIÓN

En la actualidad las Tecnologías de la Información y las Comunicaciones (TIC) se han convertido en una herramienta poderosa e indispensable en el desarrollo social y económico de cualquier país, vinculadas de forma directa en la informatización y automatización de la mayoría de las esferas, procesos de la producción y los servicios. El concepto de las TIC nace con la convergencia tecnológica de la electrónica, el software y las infraestructuras de las telecomunicaciones. Las TIC proveen herramientas que ofrecen la posibilidad de encontrar soluciones novedosas ante los desafíos sociales de la era moderna.

Desde hace algunos años Cuba ha emprendido el reto de la informatización de la sociedad, con el objetivo de elevar la calidad de vida del pueblo y lograr más eficiencia en la economía. La informatización de la sociedad cubana es un proceso de utilización ordenada y masiva de las TIC para satisfacer las necesidades de información y conocimiento de todas las personas y sectores de la sociedad.

Como parte de este esfuerzo nació la Universidad de las Ciencias Informáticas (UCI) en el curso 2002-2003 y que juega un papel importante en el desarrollo de la Industria Cubana del Software y en la materialización de los proyectos asociados al programa cubano de informatización de la sociedad.

La UCI, como universidad de nuevo tipo, tiene un novedoso modelo de formación de profesionales comprometidos con su país, que combina el estudio con la producción y la investigación. A esta universidad que se encamina a la producción de software y servicios informáticos, se le asignó la tarea de informatizar el sector jurídico, específicamente la Fiscalía General de la República (FGR), como parte del empeño del gobierno cubano de superar las deficiencias que presenta en la actualidad.

La Fiscalía General de la República de Cuba es el órgano del estado al que corresponde, como objetivo fundamental, el control, la preservación de la legalidad, la promoción y el ejercicio de la acción penal pública en representación del estado [FGR, 2008]. La vigilancia y el estricto cumplimiento de la Constitución y las normas legislativas, por parte del estado y el pueblo, constituyen el sustento de este órgano, estructurado de la forma siguiente:

- Fiscalía General de la República
- Fiscalías Provinciales
- Fiscalías Municipales
- Fiscalía Militar

Los procesos fundamentales llevados a cabo en la FGR son:

1. Procesos Penales (PP)
2. Verificación Fiscal (VF)
3. Protección a los Derechos Ciudadanos (PDC)
4. Gestión de Cuadros y Personal de Apoyo (GCPA)
5. Relaciones Internacionales (RI)
6. Herramientas Comunes a Todas las Áreas (HCTA)
7. Control de la Legalidad en Establecimientos Penitenciarios (CLEP)
8. Dirección General de Control (DGC)

En las Fiscalías de todo el país se gestionan y procesan elevados volúmenes de información. Los costos en concepto de tiempo, esfuerzo y recursos humanos destinados a la realización de determinados procesos, resultan elevados. Generalmente los Fiscales, Secretarías y demás implicados en el acceso y control de la información jurídica, intervienen en más de un caso a la vez, deben mantenerse pendientes del límite de sus períodos de tramitación, así como de las violaciones en las que pueden incurrir. Además, el grueso de la información se manipula en formato duro, es decir, consumiendo gran cantidad de papel, propenso a pérdidas o demoras causantes de incumplimientos en la resolución de los casos.

Actualmente la FGR cuenta con un software que, orientado a lograr mayor rapidez en la toma de decisiones, eficiencia y conformidad por parte de los clientes, pretende automatizar todas las áreas de trabajo. El mismo no satisface las necesidades de la institución y lejos de agilizar los procesos, se obtienen resultados que contribuyen al detrimento del prestigio alcanzado a altas esferas.

Es por esto que en el 2008, se realizan convenios entre la Fiscalía General de la República y la Universidad de las Ciencias Informáticas (UCI) y es creado el proyecto Sistema de Gestión Fiscal (SGF).

SGF surge con el objetivo de automatizar los procesos relativos a la Fiscalía General de la República de Cuba, en aras de garantizar mayor economía, profesionalidad, celeridad, control y preservación de la legalidad. Consta con una estructura de 8 subsistemas y 40 módulos, entre los cuales se encuentra el Módulo Ordinario. Este módulo forma parte del subsistema Procesos Penales y engloba los delitos sancionables a más de un año de privación de libertad o multas que excedan las 300 cuotas.

La mayoría de las acciones que se desarrollan como actividades fundamentales del módulo, se encuentran relacionadas con el ejercicio de la acción penal. Las mismas son complejas y su ejecución requiere de un grado de precisión elevado, pues intervienen directamente en procesos como el de toma de decisiones, tramitación de expedientes delictivos y juicios. De ahí que se le

conceda vital importancia a su informatización, a las herramientas y procedimientos utilizados en cada fase.

Para lograr el correcto desarrollo del módulo Ordinario, se considera necesaria la realización de una clara interpretación del flujo de eventos a informatizar, por parte de los analistas, pues el sistema de trabajo en el proyecto SGF resulta ineficiente, presentando, errores en los datos, flujo lento de información, mala organización, poco entendimiento entre la partes y pérdida de información que traen como consecuencias errores e inexactitudes en los procesos legales, incumplimiento de fechas y un levantamiento de requisitos poco exitoso.

Para que el producto sea fiable, los desarrolladores del proyecto SGF, principales encargados de materializar las necesidades del cliente, deben dominar y tener total entendimiento de los procesos que se implementarán en etapas posteriores, ya que dicho módulo contiene muchas especificaciones legales que lo hacen más complejo. De esta forma se reducirán los costos en concepto de tiempo invertido, recursos humanos, reingeniería y mantenimiento.

La situación presentada anteriormente, conduce al siguiente **Problema Científico**: ¿Cómo traducir las necesidades del cliente en un lenguaje entendible por los desarrolladores en el módulo Ordinario del proyecto Sistema de Gestión Fiscal?

El **Objeto de Estudio** lo constituye el Proceso de Desarrollo de Software, del cual se deriva como **Campo de acción**: Ingeniería de Requisitos.

Para dar solución al problema se define como **Objetivo General**: Generar los artefactos modelo del negocio, levantamiento de requisitos y modelo del sistema que garanticen el trabajo de los desarrolladores del módulo Ordinario.

Toda la investigación se encuentra sustentada sobre la **Hipótesis** de que: Si se realiza la elicitación, análisis, especificación y validación de los requisitos se podrán traducir las necesidades del cliente en un lenguaje entendible por los desarrolladores en el módulo Ordinario del proyecto Sistema de Gestión Fiscal.

Con el propósito de dar respuesta al objetivo planteado, se trazan las siguientes **tareas**:

- Elaborar el marco teórico para la justificación del trabajo.
- Identificar los procesos fundamentales que se desarrollan en el módulo Ordinario, con el objetivo de identificar las necesidades del cliente.
- Seleccionar patrones de casos de uso necesario para el desarrollo del presente trabajo.
- Realizar el Modelado del Negocio.
- Definir Requisitos del Sistema.

- Realizar el Modelado del Sistema.

Los métodos investigativos empleados son:

Métodos teóricos:

- **Histórico-Lógico:** Se emplea para el análisis y evolución de la Ingeniería de Requisitos, para la investigación de la metodología a utilizar, herramientas y patrones casos de uso necesarios para el transcurso del presente trabajo.
- **Analítico-Sintético:** Se emplea para el análisis del objeto de estudio, buscando los elementos más importantes.
- **Modelación:** Se emplea para modelar las abstracciones que explican la realidad, a través de herramientas.

Métodos empíricos:

- **Entrevista:** Para adquirir información valiosa sobre cómo se lleva a cabo el proceso Ordinario en la FGR y cómo se espera que funcione la aplicación web.

El documento se encuentra estructurado en tres capítulos:

Capítulo I: Fundamentación Teórica: En este capítulo se realiza un estudio del estado del arte de los principales Sistemas de Gestión Fiscal en el mundo y de la Ingeniería de Requisitos, situación que permitió conocer su importancia. Se expone una caracterización de diferentes metodologías, herramientas y lenguajes de modelado posibles a emplear para desarrollar los artefactos, se presenta una breve descripción de los patrones de casos de uso necesarios para el desarrollo del trabajo y de las métricas a utilizar para lograr una mejor calidad.

Capítulo II: Modelado del Negocio y Sistema: En este capítulo se realiza una descripción del negocio que permite identificar los actores, trabajadores y reglas del negocio. Se explica el proceso de la captura de requisitos, empleando las técnicas necesarias, permitiendo identificar los requisitos (funcionales y no funcionales) que debe cumplir la aplicación y por último se exponen los artefactos resultantes del módulo Ordinario.

Capítulo III: Validación: En este capítulo se realiza la validación de los prototipos de interfaz con el cliente y se aplican técnicas de validación y métricas orientadas a la calidad de las especificaciones de los requisitos, casos de usos y sus diagramas.

Este documento posee, además, Conclusiones por cada capítulo, Conclusiones generales, Recomendaciones, Referencias Bibliográficas, Glosario de Términos y Anexos.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza un estudio del estado del arte de los principales Sistemas de Gestión Fiscal en el mundo y de la Ingeniería de Requisitos, situación que permitió conocer su importancia. Se expone una caracterización de diferentes metodologías, herramientas y lenguajes de modelado posibles a emplear para desarrollar los artefactos, se presenta una breve descripción de los patrones de casos de uso necesarios para el desarrollo del trabajo y de las métricas a utilizar para lograr una mejor calidad.

1.2 Sistemas de Gestión Fiscal en el Mundo

Cada sistema legal cuenta con particularidades u ordenamientos jurídicos, relacionados con las características de un país y su contexto social. Si se tiene en cuenta, que existen elementos comunes en la mayoría de estos sistemas, entonces resulta imprescindible realizar un estudio del arte que aborde las tendencias actuales.

A continuación se exponen ejemplos de Sistemas de Gestión consultados:

Infolex es un sistema encargado de gestionar la actividad legal de un Despacho Jurídico. Desarrollado por la empresa española Jurisoft (Sistemas de Informática Jurídica), provee herramientas de Gestión para la Abogacía y la Procura, básicamente enfocadas al trabajo con:

- Expedientes Judiciales y Extrajudiciales.
- Seguimiento y Gestión Documental.
- Agenda integrada con Expedientes.
- Integración con Lexnet (Acceso Online de clientes y seguimiento de asuntos), Infolex Acceso Remoto (Soluciones para delegaciones y para trabajar en modo remoto desde cualquier lugar).
- Desarrollos a medida (Versiones estandarizadas).
- Control de llamadas.
- Minutación y Facturación.
- Contabilidad y Tributación.

Además consta de un módulo de Servicios Jurídicos, orientado al cumplimiento de tareas relacionadas con:

- Expedientes Mercantiles (Compraventa, alquileres.etc.)
- Expedientes Consultivos.

- Gestión de Expedientes Administrativos (Para Servicios Jurídicos de la Administración).
- Registro de Entrada y Salida.
- Control de Escrituras.
- Desarrollos para Compañías de Seguros.
- Módulo de Recobros y Gestión de Embargos.
- Control Económico de Externos.
- Informes Económicos y de Rentabilidad.
- Integración con Baremos de Minutación de Compañías.
- Gestión de Expedientes Administrativos (Para Servicios Jurídicos de la Administración).

A pesar de lo antes expuesto, el uso de Infolex en la FGR no es factible, debido a que el sistema ha sido desarrollado usando software propietario. El pago de licencias, actualizaciones y mantenimiento resulta demasiado costoso. Además de que no permite la realización de modificaciones al producto en aras de satisfacer las necesidades del cliente, dadas las especificidades de su sistema legal.

RedAbogacía es una plataforma de servicios seguros que permite la interoperabilidad entre los distintos Colegios de Abogados, sus colegiados y la Administración Pública. Como herramienta de seguridad, para el acceso e identificación en RedAbogacía se utilizan los certificados digitales de la Autoridad de Certificación de la Abogacía (ACA) [CGAE, 2006].

El uso de este sistema ahorra costes de desplazamientos, dado el uso de las comunicaciones electrónicas y mejora la calidad de los servicios que se prestan a los colegiados. Se ponen a disposición de los abogados españoles un creciente número de servicios telemáticos, entre los que se encuentran:

- Servicio de pases a prisiones.
- Buromail: Correo electrónico seguro.
- CIP: Servicio de comunicaciones de intervención profesional.
- Oficina Postal Virtual, en colaboración con Correos y Telégrafos.
- Presentación de escritos a los Juzgados, a través de la integración entre RedAbogacía y la plataforma LexNet del Ministerio de Justicia.
- Luris et Legis: acceso gratuito a la Base de Datos de Códigos de La Ley y reducción en precio en las Bases de Datos de Jurisprudencia y Legislación.
- Acceso al Censo General de Letrados.
- Campus Virtual (Servicio de formación online accesible con certificado digital).
- Gestión Documental, acceso seguro a la documentación del Consejo General de la Abogacía Española (CGAE) según permisos específicos.

Esta plataforma no satisface todas las expectativas de un SGF cubano orientado a mejorar el proceso de toma de decisiones, aunque sí las necesidades relacionadas con certificación digital y envío de paquetes e información entre instituciones.

Favorece la escalabilidad del producto, pero a la vez se convierte en una fuente generadora de elevados costos en concepto de soporte y mantenimiento.

El Sistema Integrado de Gestión de la Abogacía (SIGA) es un sistema de gestión de recursos diseñado por el Instituto Tecnológico del Consejo General de la Abogacía Española (IT-CGAE), que integra un conjunto de servicios personalizados, con el objetivo de agilizar los trámites y diligencias más habituales, realizados a través de Internet.

Ofrece una mayor rapidez y seguridad en el intercambio de información entre abogados, Colegios de Abogados, Consejos Autonómicos y CGAE, así como una mejora en los métodos de trabajo y una importante reducción de costes al compartir recursos. Brinda acceso al Colegio Virtual y acciones como: suscripción a servicios, consulta de recibos o facturas, justificación de las actuaciones del turno de oficio utilizando la firma electrónica [CGAE, 2006].

SIGA se encuentra enfocado hacia los procedimientos llevados a cabo por los abogados y obvia la mayoría de los aspectos claves de un SGF. Por esto su uso se considera poco factible.

Fiscal.es es una aplicación web desarrollada en España, con el objetivo de abordar los diversos condicionantes normativos, sistemas de gestión procesal, así como la integración de los mismos. Actualmente sus áreas de actuación están enmarcadas en la Intranet de la Fiscalía General del Estado (FGE), la página web destinada al ciudadano común y demás actividades de gestión jurisdiccional relacionadas. Los modelos de sistemas de gestión aplicados son:

- Aplicaciones judiciales únicas con usuarios diversos: la Fiscalía cliente.
- Aplicaciones de Fiscalía independientes de las judiciales.

Garantiza una base centralizada de procedimientos, el acceso y control de todas las fiscalías a los registros, bases de datos, sistemas y aplicaciones informáticas de ámbito nacional gestionados por el Ministerio de Justicia. También la conexión del FGE y órganos centrales con las Fiscalías y de éstas entre sí [Fiscal.es, 2009].

Se considera a Fiscal.es un modelo de aplicación que satisface en algunos aspectos las necesidades de la FGR. El sistema judicial de la República de Cuba sustenta sus bases en el Derecho Continental Europeo, del que tomó la mayor parte de sus instituciones legales, por lo que la correspondencia entre procesos e intereses en áreas de trabajo de la FGE y FGR, resulta notable.

Fiscal.es cuenta con un conjunto de interfaces, control de procesos, sistemas de manejo, tratamiento de la información que podrían resultar provechosos, aunque tiene como inconvenientes que ha sido desarrollado utilizando como gestor de Base de Datos a Oracles (software propietario), además de un Framework perteneciente a la Oficina Judicial, protegido bajo licencias y derechos de autor restringidos.

1.3 Sistemas de Gestión Fiscal en Cuba

SOFTLEX surge en el seno de la Sociedad Cubana de Derecho e Informática, de la mano del equipo de desarrollo integrado por: Gladys María Franco, Leonardo Valdés Franco y Marbelis Piloto Corzo. Producto que brinda servicios de tratamiento documental, almacenamiento y consulta de legislaciones agrupadas en listas organizadas por: emisión de la gaceta, institución que genera la norma, número de la normativa, tipo de normativa y página de la Gaceta. También brinda la posibilidad de contar con un índice referativo que recoge las referencias de la actividad jurídica en un período de tiempo determinado [Softlex, 2006].

SOFTLEX/Data - DataLex surge como complemento de la aplicación antes expuesta y tiene como características fundamentales: el trabajo con herramientas de almacenamiento de información, el registro de disposiciones jurídicas que incluye un graficador mínimo y recoge los indicadores para el control de normas emitidas por la Gaceta y disposiciones internas generadas por el usuario (cartas circulares, resoluciones, instrucciones u otras). También permite el registro y control de contratos nacionales y de importación y es ajustada a las necesidades del usuario a partir de una plantilla inicial. Cuenta con un diccionario con más de 2600 términos jurídicos presentados en orden alfabético y opciones de búsqueda de términos [Softlex, 2006].

Actualmente no existe un sistema orientado a agilizar el proceso de toma de decisiones y gestión de información en la FGR. Las soluciones expuestas anteriormente no responden objetivamente a los problemas existentes en las Fiscalías, ni reducen la demanda o el tiempo de ejecución de las actividades que se llevan a cabo en dichas instituciones. Es por esto que se considera necesaria la creación de un software de mayor envergadura y profundidad, que cumpla con estándares de desarrollo (Software Libre) y satisfaga las necesidades de los usuarios finales.

1.4 Metodologías de Desarrollo de Software

Un proceso de software detallado y completo suele denominarse “Metodología”. Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucradas, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, entre otras.

Las metodologías de desarrollo de software consisten en fases o etapas, descompuestas en subfases, módulos, pasos; estrategia que beneficia a los desarrolladores durante la elección de las técnicas a utilizar en cada estado del proyecto, facilitando la planificación, gestión, control y evaluación de los proyectos.

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, entre otras).

Debido a la importancia de las metodologías como guías en el desarrollo de software, se han desarrollado varias de estas, por lo que se hace difícil a la hora de escoger una de ellas. Por esto, es necesario realizar un estudio de las mismas para poder determinar cuál es la que se ajusta al proyecto.

Proceso Unificado de Racional (*Rational Unified Process, RUP*)

El proceso unificado de desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental [Jacobson, Booch, & Rumbaugh, 2000].

- Captura varias de las mejores prácticas en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones.
- Es una guía de cómo utilizar de manera efectiva UML.
- Provee a cada miembro de un equipo un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo.
- Crea y mantiene modelos, en lugar de enfocarse en la producción de una gran cantidad de papeles de documentación.

RUP se caracteriza por dividir el ciclo de vida de la producción del software en 4 fases:

- 1. Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- 2. Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

3. **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios *release* del producto que han pasado las pruebas. Se ponen estos *release* a consideración de un subconjunto de usuarios.
4. **Transición:** El *release* ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Las fases antes mencionadas se llevan a cabo desarrollando 9 flujos de trabajo principales, los 6 primeros son conocidos como **flujos de ingeniería** y los 3 últimos como de **apoyo o soporte**:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y Diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán, la ubicación en ellos y la estructura de capas de la aplicación.
- **Instalación:** Produce *release* del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización, concurrente de elementos, control de versiones, entre otros.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Microsoft Solutions Framework (MSF)

Microsoft® Solutions Framework es un marco de trabajo de referencia para construir e implantar sistemas empresariales distribuidos basados en herramientas y tecnologías de Microsoft. MSF comprende un conjunto de modelos, conceptos y guías que contribuyen a alinear los objetivos de negocio y tecnológicos, reducir los costos de la utilización de nuevas tecnologías y asegurar el éxito en la implantación de las tecnologías Microsoft. MSF es el resultado de las experiencias de diferentes áreas de Microsoft con proyectos exitosos [Microsoft Corporation, 2000].

Los cinco modelos de MSF son:

- Modelo de Arquitectura Empresarial de MSF (*Enterprise Architecture Model*).
- Modelo de Aplicaciones de MSF (*Application Model*).
- Modelo de Equipos de Trabajo de MSF (*Team Model*).
- Modelo de Procesos de MSF (*Process Model*).
- Proceso de Diseño de Soluciones con Componentes (*Designing Component Solutions - DCS*).

Posee 5 fases que pueden ser descritas de la siguiente forma:

1. Visión:

- El equipo debe tener una visión clara de lo que quiere lograr para el cliente.
- Especificación funcional del proyecto.
- Plan de gestión de riesgos.
- Plan maestro del proyecto.

2. Planificación:

- En esta fase el equipo prepara la especificación funcional y los planes de trabajo, al igual que los costos y calendarios.

3. Desarrollo:

- Tiene que contener el código fuente y el ejecutable.
- Así como también los instaladores y la documentación de apoyo como son los manuales de usuario.
- Especificaciones funcionales completas y casos de pruebas.

4. Estabilización:

- Dejar el sistema establecido ya que no se sabe si el ambiente real está listo para ir a operación.
- Código fuente y ejecutable al igual que documentación, deben estar disponibles.

5. Utilización:

- Después del despliegue, se les pregunta a los usuarios que tan satisfechos están y que tan contentos están con el producto.

Programación Extrema (*Extreme Programming, XP*)

La programación extrema (XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y

coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico [Beck, 1999].

Los objetivos de XP son muy simples: **“La satisfacción del cliente”**. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, debemos responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. Otro objetivo es **“Potenciar al máximo el trabajo en grupo”**. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

Scrum

Scrum es una metodología de desarrollo ágil o ligera desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, indicada para proyectos con un rápido cambio de requisitos. Las principales características que la definen es que el desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento del producto que se muestra al cliente. Otra cuestión peculiar de Scrum son las reuniones a lo largo del proyecto. De todas ellas la más importante se efectúa diariamente durante 15 minutos por parte del equipo de desarrollo para coordinar e integrar el trabajo [Larman, 2003].

1.4.1 Justificación de la metodología seleccionada

Después de realizar un estudio de las principales metodologías para el desarrollo del software por parte del equipo de desarrollo del proyecto, de acuerdo a las características y la complejidad del proyecto, se decidió trabajar con RUP.

RUP es una metodología de desarrollo de software robusta que se ajusta de forma ideal para un proyecto de grandes dimensiones como es el proyecto SGF que se centra en un eficiente control del desarrollo y del personal generando una gran cantidad de documentación para el equipo y los clientes, por lo que las metodologías ágiles no son las más indicadas, pues se enfocan más en captar con rapidez las necesidades iniciales para proceder de inmediato con la implementación, lo cual no es factible para un software de esta envergadura, compuesto por tantos módulos, pues esto provocaría fallas más adelante por un inadecuado levantamiento de requisitos.

Otra razón por la que se selecciona RUP como metodología a usar, es que el proyecto estará compuesto por gran cantidad de personal, que variará con el paso del tiempo, producto de la dinámica de la universidad, y será necesario generar la mayor cantidad de documentación posible para poder capacitar al nuevo personal, además de contribuir con una alta reusabilidad entre los módulos.

1.5 Lenguajes de Modelado

Un proceso de negocio es un conjunto de actividades relacionadas dentro de una organización que tienen como objetivo conseguir un determinado resultado. Los modelos de procesos de negocio se emplean para mejorar la comunicación tanto entre el analista y el desarrollador como entre el analista y el cliente [Pérez J. D].

Lenguaje de Modelado Unificado (UML)

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar [Jacobson, Booch, & Rumbaugh, 2000].

UML cuenta con varios tipos de diagramas:

- **Diagramas de estructura:** Enfatizan en los elementos que deben existir en el sistema modelado.
- **Diagramas de comportamiento:** Enfatizan en lo que debe suceder en el sistema modelado.
- **Diagramas de Implementación:** Muestra la estructura del código (Diagrama de componentes) y la estructura del sistema en ejecución (Diagrama de ejecución).

Métodos Integrados de Definición (IDEF)

IDEF es una técnica de modelado de sistemas usando una estructura gráfica específica. Abarca desde la modelación de la información hasta el análisis y diseño orientado a objetos.

IDEF0 es una técnica de modelación concebida para representar de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades. Un modelo IDEF0 se compone de una serie jerárquica de diagramas que permiten mediante niveles de detalle, describir las funciones especificadas en el nivel superior. En las vistas superiores del modelo la interacción entre las actividades representadas permite visualizar los procesos fundamentales que sustentan la organización. Los

elementos gráficos utilizados para la construcción de los diagramas IDEF0 son cuadros y flechas.

IDEF1 utilizado como modelo de representación y estructuración de la información.

IDEF2 utilizado para representar modelos que varían con el tiempo.

IDEF3 es una técnica de modelación para representar el flujo de trabajo de un proceso, así como sus objetos participantes a partir de la descripción dada por un experto. Permite documentar a nivel de detalle un proceso, facilitando su análisis a través de la identificación y captura del conocimiento crítico del mismo. Los componentes fundamentales que emplea IDEF3 en su representación son: unidad de trabajo, ligas, conexiones y referencias.

Notación de Modelado de Procesos de Negocio (BPMN)

La Notación de Modelado de Procesos de Negocio (*Business Process Modeling Notation, BPMN*) es un nuevo estándar de modelado de procesos de negocio. Esta notación está diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes (*White*). BPMN define un diagrama de procesos de negocio (BPD, *Business Process Diagrams*), basado en una técnica de diagramas de flujo.

BPD está conformado por un conjunto de elementos gráficos. Las cuatro categorías básicas de estos elementos son: objetos de flujo, objetos de conexión, calles y artefactos [OMG, 2006].

Los objetos de flujo tienen tres elementos centrales: Evento, Actividad y Decisión. Los mismos se conectan en un diagrama para crear el esqueleto básico de la estructura de un proceso de negocio.

Existen tres objetos de conexión: Flujo de secuencia (*Sequence flow*), Flujo de Mensaje (*Message flow*) y Asociación (*Association*). Las calles son construidas teniendo en cuenta dos categorías diferentes, *pool* y *lane*. *Pool* representa un participante en un proceso y también actúa como un contenedor gráfico para separar un conjunto de actividades de otro *pool*. *Lane* es una sub-partición dentro de un *pool* y es utilizado para organizar y categorizar actividades. La versión actual de BPMN predefine sólo tres tipos de artefactos: Objeto de datos (*Data object*), grupo y comentario.

1.5.1 Justificación del lenguaje de modelado seleccionado

Después de realizar un estudio de varios lenguajes de modelado se adopta UML para la realización de este trabajo. UML es un lenguaje de modelado que se ajusta para proyectos de software complejos y de gran envergadura, desarrollados por equipos de trabajo también grandes, donde es imprescindible disponer de una amplia cantidad de documentación y artefactos generados a lo largo del desarrollo, por lo que el mismo se adapta perfectamente a las condiciones y necesidades que presenta el proyecto SGF. Teniendo en cuenta además que fue desarrollado junto con la

metodología RUP, por lo que responde a todas sus necesidades, se combinan como la elección correcta del equipo de desarrollo.

1.6 Herramientas CASE

Se puede definir a las Herramientas CASE (*Computer Aided Software Engineering*) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Ingeniería de Software Asistida por Computadora es un tipo de ingeniería de software en la que se intenta aumentar la eficacia de sus procesos, al soportar la realización de las tareas con el uso de tecnologías [Pérez, 1999].

El empleo de Herramientas Case permite integrar el proceso de ciclo de vida:

- Análisis de datos y procesos integrados mediante un repositorio.
- Generación de interfaces entre el análisis y el diseño.
- Generación del código a partir del diseño.
- Control de mantenimiento.

Rational Rose

Rational Rose (RR) es una herramienta desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson) que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los 42 componentes, transición a los usuarios y certificación de las distintas fases. Proporciona un lenguaje común de modelado que facilita la creación de software con calidad.

Rational Rose es una herramienta CASE de software propietario. El costo de una licencia para un usuario único es de 4,741 euros contando los valores agregados que significa el mantenimiento por 12 meses.

Características Principales

- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++.
- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Soporte de ingeniería directa y/o inversa para algunos de los conceptos más comunes de Java 1.5.
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo-código configurables.
- Soporte Enterprise Java Beans™ 2.0.

- Capacidad de análisis de calidad de código.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones Web.

Visual Paradigm

Visual Paradigm (VP) es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y documentación. También proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones. Visual Paradigm es una herramienta multiplataforma de modelado visual para UML y una herramienta CASE muy potente y fácil de utilizar.

Aporta a los desarrolladores de software una plataforma de desarrollo puntera para construir aplicaciones de calidad. Además aporta también una excelente interoperabilidad con otras herramientas CASE y muchos de los entornos IDE líderes del mercado. Permite la captura de requisitos, análisis, diseño e implementación, también proporciona características tales como generación del código y generación de informes. Permite dibujar todos los tipos de diagramas de clases. Apoya los estándares más recientes de las notaciones de UML. Incorpora el soporte para trabajo en equipo, permitiendo que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. Permite realizar también ingeniería tanto directa como inversa.

Características Principales

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio generando un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa en su versión profesional, e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse en los principales IDE.
- Disponibilidad en múltiples plataformas (Windows, Linux, etc.).

Enterprise Architect

Enterprise Architect (EA) es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener.

Características Principales

- Soporta los 13 diagramas de UML 2.1.
- La Arquitectura Dirigida por Modelos permite transformar elementos simples del modelo en complejos.
- Realizar transformaciones para DDL, Java, C#, EJB, XSD.
- Puede generar código fuente C++, Java, C#, VB.Net, Visual Basic, Delphi, PHP, Python y Action Script.
- Soporte para Corba también disponible como "plug-in" libre.
- Plug-ins para vincular EA a Visual Studio.NET o Eclipse.
- Ingeniería Inversa para muchos de los sistemas populares DBMS, incluyendo Oracle, SQL Server, My SQL, Access, Postgree SQL y otros.
- Disponible solamente para plataformas Windows.

1.6.1 Justificación de la herramienta CASE seleccionada

Después de realizar un estudio de las principales herramientas CASE existentes y en concordancia con una de las prioridades fundamentales a las que se sujeta el desarrollo de un Sistema de Gestión Fiscal para la FGR que es hacer uso de software libre se decidió utilizar Visual Paradigm.

Se toma esta elección principalmente por ser una herramienta multiplataforma, facilidad que pocas herramientas CASE brindan, pero además es robusta, de fácil uso y que permite realizar la ingeniería directa en su versión profesional, e inversa. También se sustenta la elección de la misma en el hecho de que la UCI cuenta con la licencia para el uso de dicha herramienta.

1.7 Herramientas de modelado de Prototipos de interfaz de usuario

Los prototipos de interfaz tienen muchas utilidades y aplicaciones, ya que permiten que se puedan verificar los requerimientos del producto antes de que se inicie su desarrollo, fijar presentaciones en las que se puede solicitar la aprobación de la maqueta del producto a los interlocutores del cliente o del equipo de desarrollo, así como: especificar el contenido, planificar las fases del desarrollo o detectar errores de diseño antes de iniciar el proceso de producción.

Axure RP

Axure RP es una aplicación ideal para crear prototipos y especificaciones muy precisas para páginas web. Es una herramienta especializada en la tarea, así que cuenta con todo lo que se puede necesitar para crear los prototipos de forma eficiente.

Permite componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad. Con el uso de anotaciones permite especificar el estado de cada elemento (Propuesto, Aceptado, Incorporado), el beneficio esperado (Crítico, Importante, Útil), el riesgo, la estabilidad, a quién va dirigido y a quién se le asignará la tarea. Integra un módulo de control de versiones que permite distribuir tareas y gestionar proyectos dentro de un equipo. Una de las principales limitaciones que presenta esta aplicación es que solo funciona en los siguientes sistemas operativos: Windows 98/98SE/Me/2000/XP [Gómez].

Entre las ventajas que provee están:

- Poseer una serie de objetos predefinidos que cubren la mayoría de las necesidades que puede tener el analista a la hora de integrar elementos en un prototipo.
- Facilitar la creación de plantillas.
- Cada elemento del prototipo puede ser comentado y definido y estos comentarios se pueden exportar a Word.
- Organización de prototipos en forma de árbol y exportación de los mismos a formato HTML y de imagen, facilitando la interacción entre prototipos y la presentación de los mismos a los clientes.

Microsoft Office Visio Professional

Microsoft Office Visio Professional es una herramienta desarrollada para brindar una manera más descriptiva y eficiente de presentar información. La visualización, el análisis y la comunicación de información importante son imprescindibles para todo profesional de cualquier área.

Microsoft Office Visio Professional posibilita organizar toda esta información con diferentes plantillas, diagramas de flujos de datos, diagramas de redes, modelos de bases de datos, diagramas de software y muchos otros más. Los esquemas generados con Office Visio pueden ser utilizados para optimizar procesos empresariales, realizar el seguimiento de proyectos y recursos ó simplemente para mostrar de manera eficaz la información de un trabajo realizado. Posee una poderosa interfaz con múltiples opciones para el diseño de métodos propios de organización de información y también ofrece prototipos prediseñados con gran robustez y eficiencia [Microsoft Office Online].

1.7.1 Justificación de la Herramienta de modelado de Prototipos de interfaz de usuario

Después de realizar un estudio de las principales herramientas para el modelado de prototipos de interfaz de usuario el Jefe del Proyecto SGF decidió seleccionar Axure RP ya que es una excelente herramienta dirigida al prototipado de aplicaciones web. Consiste básicamente en una zona de dibujo a la que podemos arrastrar los distintos elementos de la interfaz (botones, cuadros de texto, entre otros). Podemos establecer algunos de esos elementos como parte de una plantilla general y reutilizarlos en cada pantalla concreta, añadir anotaciones, asignarlos a distintos miembros del equipo, establecer acciones para los mismos. En general, es todo bastante intuitivo y muy sencillo de utilizar. Este software permite alcanzar una mejor percepción del funcionamiento futuro que tendrá la aplicación posibilitando visualizar cómo quedará el sitio web que fue diseñado mediante prototipos.

1.8 Ingeniería de Requisitos

La Ingeniería de Requisitos es el uso sistemático de procedimientos, técnicas, lenguajes y herramientas para obtener con un coste reducido el análisis, documentación, evolución continua de las necesidades del usuario y la especificación del comportamiento externo que satisfaga las necesidades del usuario [Pressman R. S., 2005].

La Ingeniería de Requerimientos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software [Pressman, 2006].

La Ingeniería de Requerimientos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema [Sommerville, 2005].

1.8.1 Definición de requisito de software

Se presenta a continuación la definición existente en el glosario del Instituto de Ingenieros Electricistas y Electrónicos (IEEE) de lo que es un “Requerimiento”:

- “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo” [Std 610.12-1900, IEEE].
- “Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal” [Std 610.12-1900, IEEE].

1.8.2 Tipos de Requisitos

Los requerimientos de software pueden dividirse en 2 clasificaciones: requerimientos funcionales y requerimientos no funcionales y este último se dividen en categorías.

Requisitos Funcionales (RF): Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física.

Por lo que se describen mejor a través del modelo de casos de uso y los casos de uso como tal. Por lo tanto los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto.

Requisitos no Funcionales (RNF): Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, conociéndose que el mismo cumple con la toda la funcionalidad requerida. Las propiedades no funcionales como, cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Las categorías de los Requisitos no Funcionales son:

Requisitos de Software: Se refiere al software que debe tener, por ejemplo el sistema operativo que se debe utilizar.

Requisitos de Hardware: Se refiere a los elementos de hardware que deben tener para el correcto funcionamiento de un sistema.

Restricciones en el diseño y la implementación: Especifica o restringe la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

Requisitos de apariencia o interfaz externa: Describe la apariencia del producto. Es importante destacar que no se trata del diseño de la interfaz en detalle sino que especifican cómo se pretende que sea la interfaz externa del producto. También pueden ser necesidades de cumplir con normas estándares como por ejemplo las interfaces de sistemas tipo Windows/Apple/Motif o con los estándares de la empresa para la cual se esté desarrollando el software.

Requisitos de Seguridad: Se refiere al manejo de la seguridad. Pueden conducir a riesgos grandes, si no se gestionan correctamente. Abarcan tres aspectos:

- Confidencialidad: que la información esté protegida de accesos no autorizados.
- Integridad: que la información esté segura de cambios corruptos manejados sin autorización.
- Disponibilidad: que la información esté disponible y según el nivel de acceso que tenga cada usuario.

Requisitos de Usabilidad: Describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello deben revisarse las especificaciones de los perfiles de usuarios y las clasificaciones de sus niveles de experiencia.

Requisitos de Soporte: Abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de este, así como lograr su mejoramiento progresivo y evolución en el tiempo.

1.8.3 Etapas de desarrollo de la ingeniería de requisitos

1. Elicitación: Captura o identificación de requerimientos

El proceso identificación de requisitos tiene lugar una vez que se ha hecho un estudio de viabilidad del sistema. Este proceso tiene como principal objetivo guiar el desarrollo de software hacia el sistema correcto, definiendo objetivos generales concretos de manera tal que tanto el negocio como sus actores se beneficien [Escalona, Jose María; Koch, Nora, 2002].

Durante el proceso de obtención de los requerimientos juega un papel esencial el cliente, que se convierte en un miembro más del equipo de proyecto por lo que el resultado de la captura de requerimientos debe ser escrito en su lenguaje.

2. Análisis y negociación de requerimientos

En esta etapa los requisitos se agrupan por categorías y se organizan en subconjuntos, se estudia cada requisito en relación con el resto, se examinan los requisitos en su **consistencia**, **completitud**, y **ambigüedad**, y se clasifican y ordenan en base a las necesidades y prioridades de los clientes. Los riesgos asociados con cada requisito serán identificados y analizados. Se efectuarán estimaciones del esfuerzo de desarrollo que se utilizan para valorar el impacto de cada requisito en el costo del proyecto y en el plazo de entrega. Utilizando un procedimiento iterativo, se irán eliminando, combinando y/o modificando requisitos y así conseguir satisfacer los objetivos propuestos [Pressman, 2005].

3. Especificación o documentación de requerimientos

El objetivo en esta etapa es obtener una especificación de los requisitos que ya han sido analizados y negociados con los clientes, la cual puede desarrollarse en un documento escrito, un modelo

gráfico, un modelo matemático formal, un prototipo, o una combinación de lo anteriormente citado [Pressman, 2005].

Se sugiere que debe desarrollarse una plantilla estándar, que así se conseguirán requisitos que sean presentados de una forma más consistente y más comprensible. No obstante en muchas ocasiones es necesario buscar la flexibilidad cuando una especificación va a ser desarrollada [Sommerville, 1997].

4. Validación de requerimientos

La validación de requisitos examina la Especificación de Requisitos de Software (ERS) 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. Además se comprueba que la ERS se ajusta a las necesidades del cliente [Pressman, 2005].

1.8.4 Técnicas para la captura de requisitos

Existen un gran número de técnicas que se emplean en todo el mundo para realizar la captura de los requisitos. A continuación se describirán algunas de estas técnicas.

Entrevistas

Es una de las técnicas de elicitación más usadas y ocupan un lugar preponderante en consideración con el tiempo y el objetivo que tienen, consisten en establecer un canal de comunicación directa entre las personas destinatarias del sistema y el equipo de desarrollo. Por lo general, son la mayor fuente de información del analista.

La entrevista es una forma de conversación, no de interrogación. Durante la entrevista el analista no se limita solo a informarse de los puntos que le interesan, sino que además aprovecha la oportunidad para explicar su trabajo a los usuarios y crear un clima sociológico favorable. Las entrevistas se pueden realizar sobre la base de un cuestionario rígido o de una guía más o menos detallada que las orienta hacia puntos bien definidos.

En esta técnica se pueden identificar tres fases: la preparación, la realización y el análisis de la información obtenida.

El método de entrevistas para obtener información tiene las siguientes ventajas.

- Permite a los analistas presentar sus necesidades de forma directa y verificar en las respuestas recibidas, si las preguntas realizadas fueron interpretadas correctamente.
- Es una oportunidad que tiene el analista para conocer el grado de aceptación o resistencia que existe entre los usuarios hacia el sistema que se desea diseñar.

Cuestionario

Constituye la única forma posible de poder relacionarse con un gran número de personas para conocer varios aspectos del sistema, sin tener que estar presente. Consiste en un conjunto de preguntas presentadas a un grupo de personas enfocadas a su respuesta. La forma de la pregunta puede influir en las respuestas, por lo que hay que planearlas cuidadosamente.

Las preguntas suelen distinguirse en dos categorías: abiertas y cerradas. Las preguntas abiertas permiten que los encuestados respondan con su propia terminología. Generalmente estas son más reveladoras, ya que los interrogados no están limitados en sus respuestas. Son especialmente útiles en la etapa exploratoria de la investigación, cuando el analista busca penetrar en el pensamiento del encuestado.

Existen también inconvenientes en los cuestionarios como son:

- Muchos usuarios que pueden ofrecer una buena cantidad de información, se resisten o la dan en poca cantidad cuando se trata de suministrarla por escrito.
- Los entrevistados pueden objetar muchas preguntas, interpretarlas a su forma o no tomarlas en serio.
- Es difícil diseñar cuestionarios que aseguren obtener exactamente toda la información que se desea.

Tormenta de ideas

Fue el primero de estos métodos que se desarrolló. Permite generar gran cantidad de ideas en breve tiempo. Se desarrolla con un grupo de expertos, se expone un problema a los presentes, o se les envía un memorándum previo. Las ideas se generan y exponen por los asistentes en forma clara y precisa, evitando discursos, sin que medie ninguna crítica o evaluación de éstas, por descabelladas que pudieran parecer.

En esta atmósfera no crítica, las personas se sienten libres para decir lo que piensan y estas ideas, aún en el caso de que no tuvieran valor, pueden dar origen a otras por asociación. Las ideas se recogen y listan en pliegos de papel que se mantienen a la vista de todos. Estas ideas se valoran posteriormente.

Grabaciones de video y de audio

Básicamente existen dos formas de utilizar las grabaciones: como registro y apoyo de las entrevistas y para analizar algún proceso en particular. En cuanto a su función de apoyo, es importante por cuanto permite centrar la atención en la entrevista en sí, en vez de distraerse tomando notas de todo lo que se dice. Esta técnica permite analizar los temas con más detenimiento y con una visión más global.

Cuando se trata de analizar algún proceso en particular, su ayuda es inestimable (sobre todo las filmaciones de video) ya que permite ver y analizar en detalle ese proceso la cantidad de veces que sea necesario. Además que filmando el lugar de trabajo, se está capturando el proceso de trabajo, lo que evita que imponga expectativas y preferencias.

Es conveniente comenzar las grabaciones con preguntas poco importantes que sirvan para "relajar el ambiente", ya que el entrevistado puede ponerse nervioso durante los primeros minutos de grabación. Se les debe dar tiempo a las personas para que se sientan cómodas con la idea de ser grabados o filmados.

Prototipos

Un cliente, a menudo, define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, proceso o salida. En esta y en otras muchas situaciones, la construcción de prototipos puede ofrecer un mejor enfoque. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar [Pressman, 2005].

El desarrollo del prototipo comienza con la captura de requisitos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requisitos que son conocidos y señalan áreas en las que será necesaria la profundización en las definiciones. Luego de esto, tiene lugar un "diseño rápido". El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario. El diseño rápido lleva a la construcción de un prototipo [Arias Chaves, Edición Digital. 2007].

La presentación de prototipos de interfaz al cliente, ayuda al mismo a establecer claramente sus requisitos y por parte de los desarrolladores les permite validar la corrección de la especificación, mejorar el producto, examinar la viabilidad y usabilidad de la aplicación.

Sistemas existentes

Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, podemos analizar las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada, por otro lado también es útil analizar las distintas salidas que los sistemas producen (listados, consultas, etc.), porque siempre pueden surgir nuevas ideas sobre la base de estas.

Casos de Uso

Los casos de uso son una técnica para especificar el comportamiento de un sistema. Los casos de uso permiten entonces describir la posible secuencia de interacciones entre el sistema y uno o más actores, en respuesta a un estímulo inicial proveniente de un actor, es una descripción de un

conjunto de escenarios, cada uno de ellos comenzado con un evento inicial desde un actor hacia el sistema. Los requerimientos funcionales se pueden expresar con casos de uso.

Según el autor Sommerville, los casos de uso son una técnica que se basa en escenarios para la obtención de requerimientos. Actualmente, se han convertido en una característica fundamental de la notación UML (Lenguaje de modelado unificado), que se utiliza para describir modelos de sistemas orientados a objetos.

1.8.5 Técnicas para la validación de requisitos

Existen un gran número de técnicas que se emplean en todo el mundo para la validación de los requisitos. A continuación se describirán algunas de estas técnicas.

Revisiones

Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida, pero es muy fructífera pues se obtiene una documentación clara para clientes y desarrolladores.

Listas de Chequeo

Las Listas de Chequeo son listados de preguntas, en forma de cuestionario, que sirven para verificar el grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado. Son útiles para el chequeo de que todo se hizo correctamente, así que siempre que se desee o se estime necesario se puede usar una lista de chequeo [Lodoño, 2008].

Las preguntas, en forma de cuestionario, sirven como una guía que obliga a quien las contesta a reflexionar sobre el nivel de acatamiento de determinados requisitos.

Las siguientes preguntas representan un pequeño subconjunto de las que pueden plantearse:

1. ¿Está el requisito claramente definido?, ¿Puede interpretarse mal?
2. ¿Está identificado el origen del requisito, (por ejemplo: persona, norma, documento)?
3. ¿Qué otros requisitos hacen referencia al requisito estudiado?, ¿Están claramente identificados por medio de una matriz de referencias cruzadas o por cualquier otro mecanismo?
4. ¿El requisito incumple alguna restricción definida?
5. ¿Se puede localizar el requisito en el conjunto de objetivos del sistema?
6. ¿El requisito es verificable?, Si es así, ¿Podemos efectuar pruebas para verificar el requisito?

Las preguntas planteadas en la lista de comprobación ayudan a asegurar que el equipo de validación dispone de lo necesario para realizar una revisión completa de cada requisito.

En la Universidad de las Ciencias Informáticas las Listas de Chequeo se utilizan dándole una evaluación a las preguntas de uno o cero, donde el valor de uno se da cuando la respuesta es negativa y cero cuando es lo contrario. En la Lista de Chequeo se describen una serie de no conformidades que se les da un valor de significativas o no, según el caso, donde a cada una se especifica los requerimientos que están implicados.

Prototipos no funcionales

Los prototipos permiten validar los requisitos identificados que, sin tener la totalidad de la funcionalidad del sistema, permiten al usuario hacerse una idea de la estructura de la interfaz, lo que permite a los desarrolladores conseguir una importante retroalimentación en cuanto a si el sistema diseñado en función de los requisitos le permite al usuario hacer su trabajo de manera eficiente y efectiva. Un prototipo puede revelar errores u omisiones en los requisitos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.

1.9 Métricas

El IEEE define las métricas como “medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” [Pressman, 2005].

Hay varias razones para medir un producto:

- Para indicar la calidad del producto.
- Para evaluar la productividad de las personas que desarrollan el producto.
- Para evaluar los beneficios en términos de productividad y calidad.
- Para establecer una línea de base para la estimación.

Métricas de la calidad de la especificación de los requisitos

Algunos autores proponen una lista de características que pueden emplearse para valorar la calidad del modelo del sistema y la correspondiente especificación de requisitos: especificidad (ausencia de ambigüedad), complejidad, corrección, comprensión, entre otras. Aunque estas características parecen ser cualitativas, se pueden representar usando una o más métricas.

Para determinar la especificidad de los requisitos se emplea una métrica basada en la consistencia de la interpretación de los revisores para cada requisito, para esto es necesario conocer el total de los requisitos n_r dado por: $n_r = n_f + n_{nf}$

Donde n_f son los requisitos funcionales y n_{nf} los requisitos no funcionales. Luego de tener este dato se puede conocer la especificidad planteando: $Q = n_{ui}/n_r$

Donde n_{ui} es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas, señalando que mientras más cerca esté de 1 el valor de Q, menor será la ambigüedad [Pressman, 2005].

Métricas para en el diagrama de caso de uso del sistema

La medición es fundamental para cualquier disciplina de ingeniería, y la ingeniería del software no es una excepción. La medición permite tener una visión más profunda proporcionando un mecanismo para la evaluación objetiva.

Para medir la calidad de la funcionalidad del diagrama de casos de uso del sistema, se utiliza un modelo de métricas orientado a objetos. Éste consta de cuatro atributos los cuales son: Consistencia, Correctitud, Completitud y Complejidad.

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

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

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

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

1.10 Patrones de casos de uso

Los patrones son soluciones simples compuestos por una pareja de problema / solución con un nombre, que (estandariza) buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Algunos de estos son:

Concordancia (*Commonality*)

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.

- Concordancia: Re-uso (*Commonality: Reuse*)

Consta de 3 casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. De manera que deben existir al menos dos de ellos.

- Concordancia: Adición (*Commonality: Addition*)

En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema.

Extensión Concreta (*Concrete Extension*)

Extensión Concreta es un patrón de estructura. Consiste en dos casos de uso y una relación de extensión entre ellos. El caso de uso extendido es concreto, esto quiere decir que, este puede ser instanciado por sí solo, así como extender el caso de uso base. El caso de uso base puede ser concreto o abstracto. Es aplicable cuando un flujo de un caso de uso puede extender el flujo de otro caso de uso, así como ser ejecutado por sí solo.

Inclusión Concreta (*Concrete Inclusion*)

Inclusión Concreta es un patrón de estructura. Consiste en dos casos de uso y una relación de inclusión entre el caso de uso base y el caso de uso incluido. Este último puede ser instanciado por sí solo. El caso de uso base puede ser concreto o abstracto. Se utiliza este patrón cuando un flujo de datos puede ser incluido en el flujo de otro caso de uso y también puede ejecutarse por sí solo.

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

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

- CRUD: Completo

Este patrón consta de un caso de uso, llamado **Información CRUD** o **Gestionar información** modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.

- CRUD: Parcial

Este patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras.

Múltiples actores (*Multiple Actors*)

Captura la concordancia entre actores manteniendo roles separados.

- Múltiples actores: Roles comunes (Common Roles)

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

- Múltiples actores: Roles Diferentes (Distinct Roles)

Captura la concordancia entre actores manteniendo roles separados. Consiste de un caso de uso y por lo menos dos actores. Es utilizado cuando dos actores juegan diferentes roles en un caso de uso, o sea, interactúan de forma diferente con el caso de uso.

1.11 Conclusiones parciales

Se realizó un estudio de los principales Sistemas de Gestión Fiscal existentes en el mundo, situación que permitió comprender su importancia y la necesidad de desarrollar un Sistema de Gestión Fiscal para el país con sus propias características, de forma que se optimicen los procesos en las Fiscalías del país y que se logre solucionar el problema que inició esta investigación. Se realizó un estudio que permitió fundamentar el uso de:

- El Proceso Unificado Racional como metodología de desarrollo de software con UML como lenguaje de modelado para la solución del presente trabajo.
- Visual Paradigm como herramienta para modelar el sistema.
- Axure RP como herramienta para modelar los prototipos de interfaz de usuario.
- Las técnicas para capturar los requisitos del sistema y las técnicas para validación de los mismo, de manera que se obtengan los conocimientos necesarios y se puedan elegir cuales deben ejecutarse para la realización de este trabajo.
- Los patrones de casos de uso como mecanismos de ayuda para los desarrolladores, ya que facilitan una solución ya probada a un problema que se pueda presentar.

CAPÍTULO II: MODELADO DEL NEGOCIO Y SISTEMA

2.1 Introducción

En el presente capítulo se realiza una descripción del negocio que permite identificar los actores, trabajadores y reglas del negocio. Se explica el proceso de la captura de requisitos, empleando las técnicas necesarias, permitiendo identificar los requisitos que debe cumplir la aplicación y por último se exponen los artefactos resultantes del análisis de los procesos fundamentales del módulo Ordinario.

2.2 Modelado del negocio

El modelado del negocio es un flujo de trabajo para comprender los procesos del negocio de la organización. Los propósitos que se persiguen al realizarse el modelado del negocio, son: entender la estructura y la dinámica de la organización, entender los problemas actuales e identificar mejoras potenciales, asegurarse de que los clientes, usuarios finales y desarrolladores tengan una idea común de la organización y derivar los requerimientos del sistema a partir del modelo de casos de uso del negocio que se obtenga.

2.3 Modelado del Negocio del Proceso Ordinario

2.3.1 Descripción del Negocio

El Proceso Ordinario comienza cuando es formulada una denuncia en el Órgano de Instrucción o en la PNR, luego de formulada la denuncia, el Instructor correspondiente puede tomar tres decisiones con respecto a la misma: solicitar la Tramitación de Archivo Denuncia, inicio de la tramitación de un Expediente de Fase Preparatoria o inicio del Expediente por Procedimiento Abreviado. Cualquiera de estas decisiones que el Instructor adopte debe enviarla a la Fiscalía conjunto a las actuaciones para que sea ratificada o no por el Fiscal.

Cuando el Instructor solicita el Archivo Denuncia, la Secretaria, que es la responsable de la recepción y despacho de toda la documentación en la Fiscalía además de registrarla en los controles establecidos, remite al Fiscal que es el responsable de tomar la decisión correspondiente. El Fiscal realiza un estudio de las actuaciones recibidas, si decide revocar la decisión manda a iniciar la tramitación de un Expediente de Fase Preparatoria, de lo contrario emite el Modelo P-2 para ratificarla, en ambos casos el Fiscal se la envía al Instructor a través de la Secretaria.

El Instructor, una vez recibida la misma, es el responsable de notificarle al Denunciante la decisión adoptada, donde el Denunciante, de no estar de acuerdo con la decisión de Archivo Denuncia recurre en queja ante el Instructor. El Instructor envía a la Fiscalía el recurso de queja, la Secretaria lo recibe y remite al Fiscal el cual, haciendo un nuevo estudio de las actuaciones puede aceptar o denegar el recurso. De aceptarlo manda al Instructor a iniciar un Expediente de Fase Preparatoria, cualquiera de las decisiones que tome el Fiscal es inapelable.

Cuando el Instructor toma la decisión de iniciar la tramitación de un Expediente de Fase Preparatoria (delito donde la sanción sea de más de un año de Privación de Libertad o Multa de más de 300 cuotas) la envía a la Fiscalía junto al expediente y a las actuaciones, donde la Secretaria recibe estos documentos y lo remite al Fiscal. El Fiscal estudia las actuaciones correspondientes y mediante un auto fundado puede decidir si se realiza la tramitación por Procedimiento Ordinario o Abreviado.

El inicio de un expediente por Procedimiento Ordinario cuenta con un término de 60 días para su tramitación y el Instructor, siempre con supervisión y control del Fiscal, realiza las diligencias necesarias. Durante su tramitación el Fiscal puede decidir la Acumulación o Desglose de Actuaciones así como la abreviación del proceso.

La Acumulación de Actuaciones es cuando existe otro expediente donde el mismo acusado forma parte del proceso, esto conlleva a que el Fiscal pueda unir las actuaciones de ambos expedientes.

El Desglose de Actuaciones es cuando en un mismo expediente existen varios acusados los cuales no pertenecen a la misma provincia, municipio o es mejor llevarlo por procesos diferentes, esto le brinda la posibilidad al Fiscal de desglosar las actuaciones e abrir un expediente para cada uno de ellos.

La Abreviación del Expediente es cuando se hallen confesos los acusados, se tengan todas las pruebas y diligencias realizadas por lo que no es necesario tener un término de 60 días para la tramitación, sino que se puede hacer en un plazo más corto.

Cuando el término de los 60 días es excedido, el Instructor debe solicitar prórroga a su Jefe Inmediato Superior, la que puede alcanzar hasta los 180 días de tramitación, si el Instructor no concluye la tramitación del expediente en ese término, el Jefe Inmediato Superior se lo comunica al Delegado del MININT de la Provincia, el cual le solicita la prórroga al Fiscal General, y este puede: conceder un nuevo término de carácter extraordinario por el tiempo que estime necesario para la conclusión del Expediente dictando auto fundado o no conceder la prórroga y disponer la conclusión del mismo. Cualquiera de estas decisiones que tome el Fiscal General es inapelable.

Cuando se inicia un expediente por Procedimiento Abreviado (es cuando el autor sea detenido en el momento de estar cometiendo el delito o siendo evidente el hecho y la participación en él del acusado y este se halle confeso) se cuenta con un término de 20 días y el Fiscal, al igual que el Instructor tienen las mismas obligaciones y responsabilidades que en el Procedimiento Ordinario. El Instructor, de no cumplir con el término correspondiente, el Fiscal mediante un auto fundado debe desabreviar el expediente y seguir su tramitación por Procedimiento Ordinario.

Ante la apertura o tramitación de cada procedimiento, al acusado se le implanta una medida cautelar para asegurarlo. Frente a esta medida impuesta, el Defensor puede recurrir en queja ante el Instructor, este último puede acceder a su modificación o denegarla, en caso de denegarla y que

el Defensor esté inconforme, vuelve a recurrir en queja ante el Instructor pero debe ser enviada al Fiscal para que decida. El Fiscal puede decidir modificar esta medida o dejarla impuesta, esta decisión es inapelable.

La Policía cuando inicia un Expediente Investigativo (hecho donde se desconocen los autores o estos no han sido habidos), mantendrá en curso el expediente en el tiempo que lo considere necesario. El Fiscal ejerce las mismas facultades que le confiere la ley para el control de los procesos penales y revisará los Expedientes Investigativos de competencia municipal o provincia indicando las diligencias o acciones que estime procedente en un término prudencial. La Policía en todos los casos y en aquellos en que se indicaron diligencias por el Fiscal, cumplidas las mismas, dispondrá el archivo provisional de las Actuaciones, lo que comunicará al Fiscal y al Denunciante si lo hubiere o de existir autores identificados y éstos no sean habidos en un término de 72 horas, previa requisitoria del acusado, dispondrá el archivo provisional de las Actuaciones lo que requerirá de la aprobación del Fiscal. El Fiscal, al recibir la comunicación del archivo provisional procede a estudiar las Actuaciones indicándole las acciones a realizar. La Policía cumplirá lo dispuesto por el Fiscal.

Para llevar a cabo todo el proceso, el Fiscal emite una serie de autos fundados ante cada decisión tomada que legalizan todo el trabajo realizado.

2.3.2 Actores del negocio

Un actor, es una entidad externa del sistema que de alguna manera participa en la historia del caso de uso. Por lo general estimula el sistema con eventos de entrada o recibe algo de él.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados [Rational, 2003].

Son actores candidatos del negocio:

- Socios
- Proveedores
- Autoridades (legales, reguladoras, etc.)
- Propietarios, si no están dentro del negocio que se modela.
- Sistemas de información externos al negocio.
- Otras partes del negocio si este es grande y esas partes no están dentro del campo de acción del negocio que se modela.

Los actores definidos en el Proceso Ordinario se detallan a continuación:

| Actor | Descripción |
|---|--|
| Instructor (MININT, DSE, Fiscalía) | Es la persona encargada de solicitar la tramitación de la denuncia y expediente en fase preparatoria. |
| Denunciante | Es la persona que presenta una denuncia en la PNR o recurso de queja contra la decisión de archivo denuncia del Fiscal. |
| Defensor | Es la persona encargada de solicitar la modificación de medida cautelar, la práctica de pruebas y establecer recurso de queja cuando estas solicitudes le son denegadas. |
| Recurrente | Es una generalización de los actores Defensor y Denunciante, es el que presenta el recurso de queja al Instructor y ejecuta lo dispuesto por el Fiscal. |
| Policía | Es el responsable de iniciar el expediente investigativo por un hecho presuntamente delictivo en que se desconocen los autores o estos no han sido habidos. |

Tabla No. 1 Actores del negocio

2.3.3 Trabajadores del negocio

Un trabajador del negocio es una abstracción de una persona o grupo de personas, una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio [Rational, 2003].

Los trabajadores definidos en el Proceso Ordinario se detallan a continuación:

| Trabajador | Descripción |
|---|---|
| Secretaria | Es la persona encargada de recibir y enviar las denuncias y expedientes a los Órganos de Instrucción o Tribunales correspondientes, además de llenar los controles existentes en la Fiscalía. |
| Fiscal (Designado, Jefe del Designado y General) | Es la persona encargada de la toma de decisiones en los procesos ordinario y abreviado, además de llevar el control de los términos. |
| Superior Jerárquico | Es el responsable de resolver el Recurso de Queja por No Ha lugar a Modificación de Medida Cautelar. |
| Jefe Inmediato Superior | Es al que se le solicita prórroga cuando se excede más de 60 días en la tramitación del expediente en fase preparatoria. |
| Delegado del MININT | Es el responsable de solicitar prórroga cuando no se concluye la tramitación del expediente en fase preparatoria en un término de 180 días. |
| Fiscal Militar | Es el encargado de recibir el expediente en fase preparatoria y hacer un escrito en el que hace constar su desinterés. |
| Tribunal | Es el responsable de resolver los pedimentos del Fiscal y tramitar los recursos interpuestos por este contra sus decisiones. |

Tabla No. 2 Trabajadores del negocio

2.3.4 Casos de uso del negocio

Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera, y que emplean los recursos de la organización para dar resultados en apoyo a sus objetivos.

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables [Jacobson, Booch, & Rumbaugh, 2000].

2.3.5 Modelo de casos uso del negocio

Un modelo de casos de uso del negocio describe los procesos del negocio de una empresa en términos de casos de uso del negocio y actores del negocio que se corresponden con los procesos del negocio y los clientes respectivamente. Al igual que el modelo de casos de uso para un sistema de software, el modelo de casos de uso del negocio presenta un sistema (en este caso, el negocio), desde la perspectiva de su uso y esquematiza como proporciona valor a sus usuarios (en este caso, sus clientes y socios) [Jacobson, Booch, & Rumbaugh, 2000].

Un proceso es todo aquello que hacemos que puede ser concebido dentro de un procedimiento estándar. Un proceso del negocio es el conjunto estructurado de las actividades que han sido diseñadas para producir un resultado específico para un cliente o el mercado. Debe haber un enfoque a la lógica del negocio de dicho proceso, desde la perspectiva del producto.

A continuación se presenta el **Diagrama de Casos de uso del negocio del módulo Ordinario**:

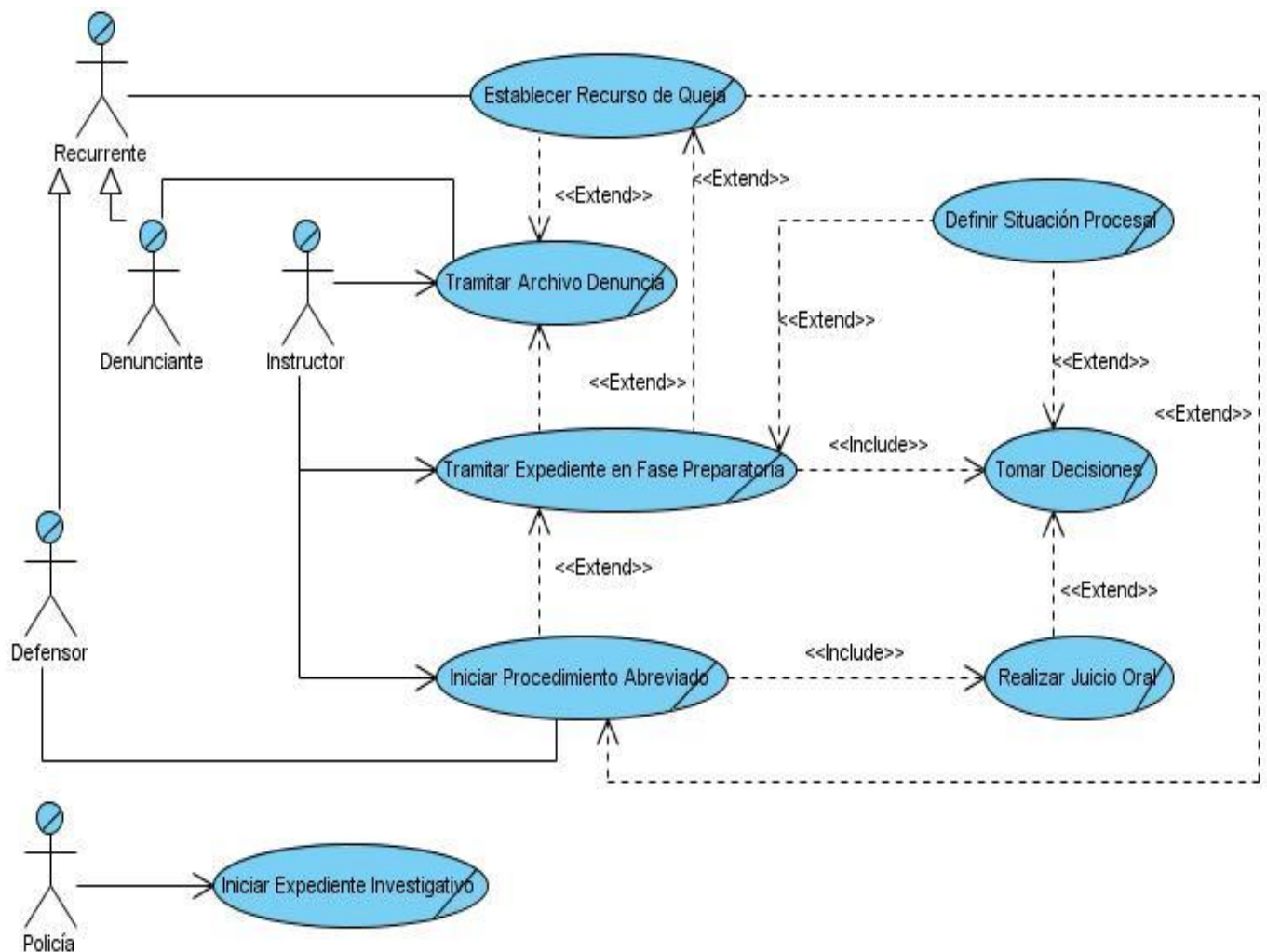


Fig. 1 Diagrama de Casos de uso del negocio

2.3.6 Descripción Textual de los casos de uso del negocio

A continuación se expone la realización de los casos de uso del negocio donde se detallan los actores y trabajadores que en él participan, además de un resumen del caso de uso. El mismo se organiza en 8 casos de uso.

| CU - Tramitar Archivo Denuncia | |
|---------------------------------------|--|
| Actores: | Instructor (inicia) y Denunciante. |
| Trabajadores: | Fiscal y Secretaria. |
| Resumen: | El caso de uso se inicia cuando el Instructor, recibida la Denuncia formulada ante él o la PNR, toma la decisión de archivarla la que remite al Fiscal para que la ratifique o no. El Denunciante puede recurrir en queja ante el Fiscal si no está de acuerdo con la decisión tomada. |
| Precondición: | Ha sido realizada una denuncia en la unidad de la PNR correspondiente. |

Tabla No. 3 Descripción del CU Tramitar Archivo Denuncia (véase documento Modelo del negocio del Proceso Ordinario).

| CU - Tramitar Expediente en Fase Preparatoria (EFP) | |
|--|--|
| Actores: | Instructor (inicia) |
| Trabajadores: | Fiscal, Secretaria, Jefe Inmediato Superior y Delegado del MININT. |
| Resumen: | El caso de uso se inicia cuando el Instructor, recibida la Denuncia formulada ante él o por la PNR, toma la decisión de iniciar la tramitación de un expediente de fase preparatoria. Luego remite las Actuaciones al Fiscal para que este proceda según lo establecido en la LPP. |
| Precondición: | Ha sido realizada una Denuncia en la unidad de la PNR por un delito donde la sanción sea de más de un año de privación de libertad o multa de más de 300 cuotas. |

Tabla No. 4 Descripción del CU Tramitar Expediente en Fase Preparatoria (véase documento Modelo del negocio del Proceso Ordinario).

| CU - Iniciar Procedimiento Abreviado | |
|---|--|
| Actores: | Instructor (inicia) y Defensor. |
| Trabajadores: | Fiscal, Superior Jerárquico, Secretaria y Tribunal. |
| Resumen: | El caso de uso comienza cuando recibida la Denuncia por el Instructor este solicita al Fiscal iniciar el EFP por el Procedimiento Abreviado o en |

| | |
|----------------------|---|
| | cualquier momento de la tramitación del Expediente el Fiscal decide continuarlo por este procedimiento. |
| Precondición: | <p>El autor sea detenido en el momento de estar cometiendo el delito o como consecuencia inmediata de su persecución luego de cometido este o habiendo eludido el autor identificado la persecución sea detenido dentro de las 72 horas siguientes.</p> <p>Siendo evidente el hecho y la participación en él del acusado este se halle confeso.</p> <p>En los casos de delitos sancionables de 1 a 8 años de privación de libertad.</p> |

Tabla No. 5 Descripción del CU Iniciar Procedimiento Abreviado (véase documento Modelo del negocio del Proceso Ordinario).

| CU - Tomar Decisiones <<include>> | |
|--|--|
| Actores: | - |
| Trabajadores: | Fiscal, Secretaria, Fiscal Militar y Tribunal |
| Resumen: | El caso de uso se inicia cuando el Instructor concluye el EFP y lo presenta al Fiscal para que tome las decisiones correspondientes. |
| Precondición: | El EFP tiene que estar concluido por el Instructor y presentado a la Fiscalía. |

Tabla No. 6 Descripción del CU Tomar Decisiones (véase documento Modelo del negocio del Proceso Ordinario).

| CU - Realizar Juicio Oral <<include>> | |
|--|--|
| Actores: | - |
| Trabajadores: | Fiscal, Secretaria y Tribunal. |
| Resumen: | El caso de uso se inicia cuando se remite el Expediente al Tribunal para su examen, radicación de la causa y apertura a juicio oral. |
| Precondición: | Debe haberse presentado el Expediente al Tribunal con el escrito de Conclusiones Provisionales del Fiscal solicitando la apertura de la causa a juicio oral. |

Tabla No. 7 Descripción del CU Realizar Juicio Oral (véase documento Modelo del negocio del Proceso Ordinario).

| CU - Iniciar Expediente Investigativo | |
|--|---|
| Actores: | Policía (inicia) |
| Trabajadores: | Fiscal y Secretaria |
| Resumen: | El caso de uso comienza cuando la Policía inicia un Expediente Investigativo por un hecho delictivo en el cual se desconocen los autores o estos no han sido habidos. |
| Precondición: | Ha sido realizada una Denuncia en la unidad de la PNR por un delito donde se desconocen los autores o estos no han sido habidos. |

Tabla No. 8 Descripción del CU Iniciar Expediente Investigativo (véase documento Modelo del negocio del Proceso Ordinario).

| CU - Establecer Recurso de Queja <<extend>> | |
|--|---|
| Actores: | - |
| Trabajadores: | Superior Jerárquico, Fiscal y Secretaria. |
| Resumen: | El caso de uso se inicia cuando el Recurrente no está de acuerdo con las decisiones tomadas por el Fiscal y este presenta un Recurso de Queja ante el Instructor. |
| Precondición: | Tiene que haberse adoptado por el Instructor o el Fiscal una resolución (auto) que pueda causar perjuicio irreparable (artículo 53 de la LPP). |

Tabla No. 9 Descripción del CU Establecer Recurso de Queja (véase documento Modelo del negocio del Proceso Ordinario).

| CU - Definir Situación Procesal <<extend>> | |
|---|---|
| Actores: | - |
| Trabajadores: | Fiscal, Superior Jerárquico y Secretaria. |
| Resumen: | El caso de uso comienza cuando el Instructor durante la tramitación del EFP solicita al Fiscal la imposición de la medida cautelar de prisión provisional, la modificación de la medida cautelar de fianza en efectivo no prestada por la de prisión provisional o su parecer sobre la procedencia de remitir al acusado al centro penitenciario por no haber |

| | |
|----------------------|---|
| | hecho efectiva la fianza. El Fiscal en el trámite de calificación puede decretar la aplicación de cualquier medida cautelar si esta no se hubiera aplicado antes, así como modificar o dejar sin efecto la que se hubiere aplicado. |
| Precondición: | El EFP tiene que estar en tramitación o concluido y en el trámite de calificación del Fiscal. |

Tabla No. 10 Descripción del CU Definir Situación Procesal (véase documento Modelo del negocio del Proceso Ordinario).

A partir de estas descripciones se realizan **Modelos de Objetos y Diagramas de actividades** (véase documento *Modelo del negocio del Proceso Ordinario*).

Un **diagrama de actividades** es un diagrama diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o un proceso. Es una extensión de un diagrama de estados, pero el diagrama de actividades resalta precisamente a las actividades. A cada actividad se le representa con un rectángulo con las esquinas redondeadas, está dividido en calles y muestra como se utilizan las entidades que apoyan la realización de los procesos de negocio.

Un **modelo de objetos** del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio [Jacobson, Booch, & Rumbaugh, 2000].

Una entidad del negocio representa algo, como una factura, que los trabajadores toman, inspeccionan manipulan, producen o utilizan en un caso del negocio [Jacobson, Booch, & Rumbaugh, 2000].

A continuación se presenta el Modelo de Objetos general del Proceso Ordinario:

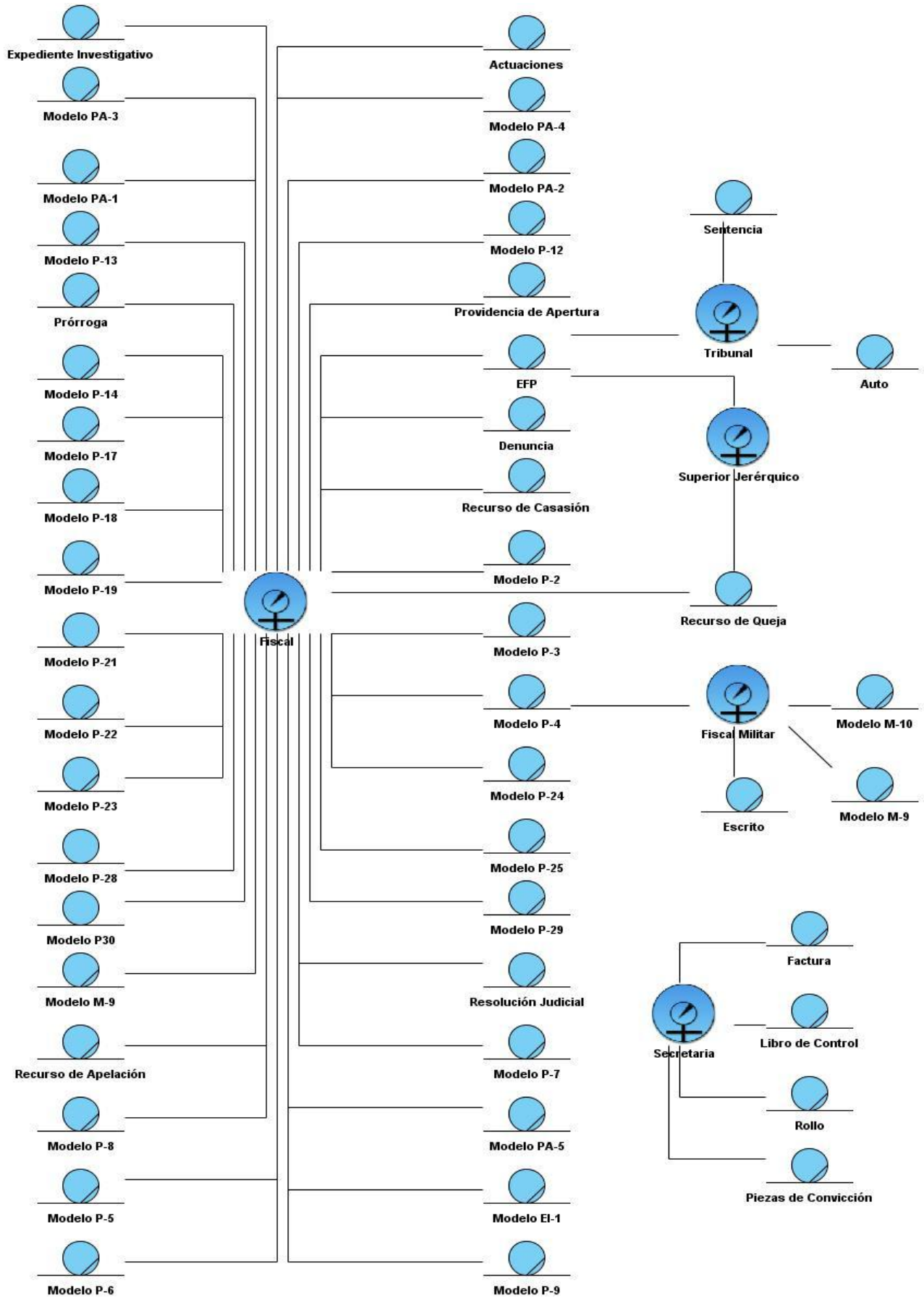


Fig. 5 Modelo de Objetos

2.3.7 Reglas del negocio

Las reglas del negocio describen políticas o condiciones que se deben cumplir, es decir que en alguna medida rigen el negocio a desarrollar.

Las reglas identificadas son:

Reglas de estructura:

- **Términos:**

1. Para archivar la denuncia se cuenta con un término de 10 días.
2. Para evaluar la decisión propuesta por el Instructor el Fiscal cuenta con un término de 72 horas.
3. Para evaluar la decisión propuesta por el Instructor el Fiscal cuenta también con un término de 7 días.
4. Para establecer recurso de queja por no estar de acuerdo con la decisión de archivo denuncia el Denunciante cuenta con un término de 3 días.
5. Para enviar el Expediente en Fase Preparatoria al Fiscal de existir acusado detenido o solicitar procedimiento abreviado, el Instructor cuenta con un término de 3 días.
6. La tramitación de un Expediente en Fase Preparatoria no puede exceder los de 60 días.
7. Si la tramitación de un expediente en fase preparatoria excede los 60, el Instructor solicita prórroga, el cual puede alcanzar hasta los 180 días de tramitación.
8. Para iniciar un expediente en fase preparatoria por procedimiento abreviado, el Instructor cuenta con un término de 20 días.
9. Para revisar un Expediente en Fase Preparatoria y la situación procesal de los acusados, el Fiscal cuenta con un término de 5 días.
10. El Instructor cuenta con un término de 20 días para cumplir las diligencias dispuestas por el Fiscal en la tramitación del expediente.
11. Para establecer recurso de súplica con las formalidades establecidas, el Fiscal cuenta con un término de 3 días.
12. Para examinar las actuaciones con la solicitud del Instructor, el Fiscal cuenta con un término de 72 horas.
13. Para recurrir en queja, el Defensor cuenta con un término de 48 horas.
14. Para aceptar o no la práctica total o parcial de las pruebas propuestas, el Fiscal cuenta con un término de 48 horas.
15. El Expediente en Fase Preparatoria de estar completo el Fiscal cuenta con un término de 72 horas para evacuar conclusiones.
16. Para decidir si procede continuar la tramitación por Procedimiento Abreviado aperturando la causa a juicio oral, el Tribunal cuenta con un término de 3 días.
17. Para la práctica de diligencias indicadas, el Instructor cuenta con un término improrrogable de 20 días.
18. Para establecer recurso de apelación contra las sentencias dictadas por los TMP en

procedimiento ordinario, el Fiscal cuenta con un término de 5 días.

19. Para establecer recurso de apelación contra la sentencia dictada en juicio seguido por el Procedimiento Abreviado ante el TPP competente, el Fiscal cuenta con un término de 5 días a partir de la fecha de la notificación de la sentencia.
20. Para establecer recurso de casación contra las sentencias dictadas por los TPP en procedimiento ordinario, el Fiscal cuenta con un término de 10 días.
21. Para disponer el archivo provisional de las actuaciones lo que requerirá de la aprobación del Fiscal, el Instructor cuenta con un término de 72 horas.
22. Para resolver el Recurso de Queja, el Superior Jerárquico cuenta con un término que no exceda de 3 días siguientes a su recibo en el Procedimiento Ordinario.
23. Para resolver el Recurso de Queja, el Superior Jerárquico cuenta con un término que no exceda de 48 horas siguientes a su recibo en el Procedimiento Abreviado.
24. Para la ratificación de la decisión o para la revocación de la decisión de Archivo Denuncia, el Fiscal cuenta con un término de 3 días.
25. Para imponer medida cautelar de prisión provisional o cualquiera de las que establece la ley o disponer la libertad del acusado o modificar la medida cautelar de fianza en efectivo no depositada por la de prisión provisional o mantener la fianza en efectivo dispuesta por el Instructor disponiendo la remisión del acusado al centro penitenciario hasta tanto haga efectiva la fianza, dejando constancia de esta decisión en la caratula del Expediente en Fase Preparatoria, el Fiscal cuenta con un término de 72 horas.

Reglas generales:

1. Todas las interacciones de la Fiscalía con los órganos (PNR y Tribunal) debe tener una factura donde se especifiquen la fecha con los documentos entregados y recibidos.
2. No se pueden violar los términos del trámite de los procesos.
3. Todos los controles existentes en la Fiscalía deben estar actualizados.

2.4 Requisitos

La finalidad de la disciplina de requisitos es:

- Establecer y mantener un acuerdo con los clientes y otros interesados acerca de lo que debe hacer el sistema.
- Proporcionar desarrolladores de sistema con un buen conocimiento de los requisitos del sistema.
- Definir los límites del sistema (delimitarlo).
- Proporcionar una base para planificar el contenido técnico de las iteraciones.
- Proporcionar una base para la estimación del coste y del tiempo en que desarrollar el sistema.

- Definir una interfaz de usuario para el sistema, centrándose en las necesidades y los objetivos de los usuarios.

Para alcanzar esos objetivos, es importante, en primer lugar, comprender la definición y el ámbito del problema que se intenta resolver con el sistema. Los interesados se identifican y las solicitudes de los interesados se obtienen, se reúnen y se analizan.

A partir de ahí se desarrollan los productos de trabajo de los requisitos para describir completamente el sistema (qué va a hacer el sistema) en un esfuerzo que percibe a todos los interesados, incluidos los clientes y los usuarios potenciales, como fuentes importantes de información (además de los requisitos del sistema) [Rational, 2003].

Para la captura de requisitos se utilizaron varias de las técnicas que definen las etapas de elicitación como son:

Entrevistas: Se entrevistaron a los Fiscales que participan en el Proceso Ordinario, donde a través de la misma se pudo capturar de forma directa las necesidades que ellos querían.

Sistemas existentes: A través del software que tienen actualmente en la Fiscalía se analizaron las interfaces de usuario donde se pudo capturar características que sirvió para identificar requisitos del Módulo Ordinario.

Prototipos: A través de los prototipos se pudo ofrecer un mejor enfoque donde se logró capturar requisitos que con las técnicas utilizadas anteriormente no se identificaron.

Después de capturado los requisitos se procedió al **análisis y negociación** de los mismo, donde se examinaron los requisitos en su **consistencia, completitud, y ambigüedad** y se clasificaron u ordenaron en base a las necesidades y prioridades de los clientes.

Requisitos Funcionales

A continuación se muestran los requisitos funcionales que se identificaron en el módulo Ordinario:

RF_PP_OR_1. Crear Expediente

RF_PP_OR_2. Guardar Expediente

RF_PP_OR_3. Editar Expediente

RF_PP_OR_4. Imprimir Portada del Expediente

RF_PP_OR_5. Añadir Denuncia

RF_PP_OR_6. Buscar Expediente

RF_PP_OR_7. Buscar Acusado

RF_PP_OR_8. Guardar Acusado

- RF_PP_OR_9. Imprimir Acusado
- RF_PP_OR_10. Actualizar Historial
- RF_PP_OR_11. Imprimir Historial
- RF_PP_OR_12. Guardar Documento
- RF_PP_OR_13. Abrir Documento
- RF_PP_OR_14. Imprimir Documento
- RF_PP_OR_15. Añadir Documento
- RF_PP_OR_16. Crear Indicación
- RF_PP_OR_17. Guardar Indicación
- RF_PP_OR_18. Abrir Indicación
- RF_PP_OR_19. Abrir Historial
- RF_PP_OR_20. Imprimir Historial
- RF_PP_OR_21. Mostrar Reporte
- RF_PP_OR_22. Imprimir Reporte

A continuación se procedió a la etapa de **especificación o documentación de los requisitos**, los mismos fueron recogidos y especificados en una plantilla estándar que propuso el proyecto SGF (*véase documento Especificación de Requisitos*).

Más adelante se hará una validación de los requisitos mediante técnicas y métricas para medir la corrección de los mismos (*véase cap. III*).

Requisitos no Funcionales

Los requisitos no funcionales definidos en el Proceso Ordinario se describen a continuación:

Hardware

Cliente

- RNF 1. Computadora cliente de 128 Mb de memoria RAM o superior.
- RNF 2. Computadora cliente de 40 Gb de disco duro o superior.
- RNF 3. Pentium a 200 MHz de velocidad de procesamiento o superior.
- RNF 4. Tarjeta de red.
- RNF 5. El sistema tiene que interactuar con dispositivos de impresión (IMPRESORA HP 1018 LASERJET).
- RNF 6. El sistema tiene que interactuar con dispositivos de escaneo (SCANNER CANON LIDE 90).

Servidores

- RNF 7. Se debe contar con un servidor que brinde las siguientes funciones integradas: DNS, DHCP, Firewall.
- RNF 8. Se debe contar con un servidor de correo independiente Mdaemon.

- RNF 9.** Se debe contar con un servidor independiente LDAP.
- RNF 10.** Se debe contar con un servidor Postgree y FTP.
- RNF 11.** Se debe contar con un servidor independiente de Salva.
- RNF 12.** Quad-Core Intel® Xeon® E5430 (2.66 GHz, 80 Watts, 1333 FSB).
- RNF 13.** Número de procesadores 1.
- RNF 14.** Máximo de procesadores 2.
- RNF15.** Máximo de Memoria 32GB.
- RNF 16.** Número de usuarios 200.

Software

Cliente

- RNF 17.** La PC inicialmente debe estar virgen.
- RNF 18.** La PC no debe contener ninguno de los programas con los cuales se desarrollo la aplicación.
- RNF 19.** Si la aplicación contiene base de datos debe tener instalado Postgree 8.3.
- RNF 20.** Debe poseer un cliente de servidor Mdaemon.
- RNF 21.** Debe tener el sistema operativo Debian 4.

Servidor

- RNF 22.** Debe poseer servicio de directorio activo.
- RNF 23.** Debe poseer FTPD PROT.

Seguridad:

- RNF 24.** Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
- RNF 25.** El sistema debe mantener en todo momento la seguridad de la información asegurando la autenticidad de la misma.
- RNF 26.** La seguridad se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.
- RNF 27.** El software brindará solamente aquellas funcionalidades que competen a la Unidad Ejecutora donde este implantado.
- RNF 28.** El sistema mantendrá en todo momento las trazas que se corresponden con las diferentes situaciones críticas que se puedan ocurrir.

Usabilidad

- RNF 29.** El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.
- RNF 30.** Existirán servidores locales con capacidad necesaria para el procesamiento de las solicitudes del conjunto de aplicaciones de las diferentes oficinas.

RNF 31. Las aplicaciones siempre solicitarán los datos a través del servidor local.

RNF 32. Desde cada servidor local se establecerá la conexión con servidores centrales para mantener la actualización de los datos en ambos sentidos.

RNF 33. El tiempo de entrenamiento requerido para que usuarios normales y avanzados sean productivos operando el sistema es de 15 días.

RNF 34. Debe poseer una interfaz agradable para el cliente.

Restricciones de Diseño

RNF 35. El lenguaje de programación es PHP 5.2.0.

RNF 36. El framework de desarrollo es Symfony 1.0.17.

RNF 37. La herramienta IDE de desarrollo utilizada será Eclipse 3.3.1.1+PDT.

RNF 38. La herramienta case utilizada es el Visual Paradigm 6.0.

RNF 39. Se utilizará el patrón de arquitectura en capas.

RNF 40. La herramienta gestor de base de datos es el PostgreSQL 8.3.

Requerimientos de Apariencia o Interfaz Externa

RNF 41. El sistema tiene que ofrecer una interfaz amigable, fácil de operar.

RNF 42. El sistema tiene que mantener la línea de diseño establecida para la institución que mantiene la uniformidad y representatividad de la misma.

RNF 43. Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para que los usuarios puedan utilizar el sistema.

Disponibilidad

RNF 44. El sistema estará disponible en días y horarios laborables.

Soporte

RNF 45. Se le debe dar un mantenimiento periódico al servidor de Base de Datos.

Seguridad

RNF 46. El sistema permitirá la autenticación de usuarios.

RNF 47. El sistema almacenará de forma cifrada la información confidencial generada.

RNF 48. El sistema realizará salvadas periódicas de la información contenida en la base de datos.

2.4.1 Actores del sistema

Cada trabajador del negocio (inclusive si fuera un sistema ya existente) que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema [Jacobson, Booch, & Rumbaugh, 2000].

Los actores del sistema:

- No son parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

Los actores del sistema definidos en el Proceso Ordinario se detallan a continuación:

| Actor | Descripción |
|------------------------|--|
| Gestor Común | Es una generalización entre los actores Fiscal y Secretaria, es la persona encargada de acceder a la interfaz y crear uno o varios expedientes, emitir documentos legales sobre el expediente, consultar las indicaciones, además de realizar las búsquedas de los expedientes y acusados. |
| Controlador | Es una generalización del Fiscal Jefe Municipal, Fiscal Jefe Dpto. Proceso Penal, Fiscal Jefe Provincial, Fiscal Jefe Dir. Proceso Penal, Vice Fiscales Provinciales, Vice Fiscal General y Fiscal General, es la persona encargada de hacerle indicaciones deseadas a un expediente durante la tramitación del mismo o del juicio oral. |
| Fiscal Superior | Es una generalización del Fiscal Jefe Municipal, Fiscal Jefe Dpto. Proceso Penal, Fiscal Jefe Dpto. DGC, Fiscal Jefe Provincial, Vice Fiscales Provinciales, Fiscal Jefe Dir. Proceso Penal, Fiscal Jefe Dir. DGC, Vice Fiscal General y Fiscal General, es la persona encargada de ver los reportes del proceso penal. |

Tabla No. 11 Actores del sistema

2.4.2 Casos de uso del sistema

Los casos de uso del sistema son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. De manera más precisa un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia [Jacobson, Booch, & Rumbaugh, 2000].

Por tanto un caso de uso es una especificación. Especifica el comportamiento de cosas dinámicas, en este caso, instancias de los casos de uso.

2.4.3 Modelo de casos de uso del Sistema

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas.

A continuación se presenta el **Diagrama de Casos de uso del sistema del módulo Ordinario:**

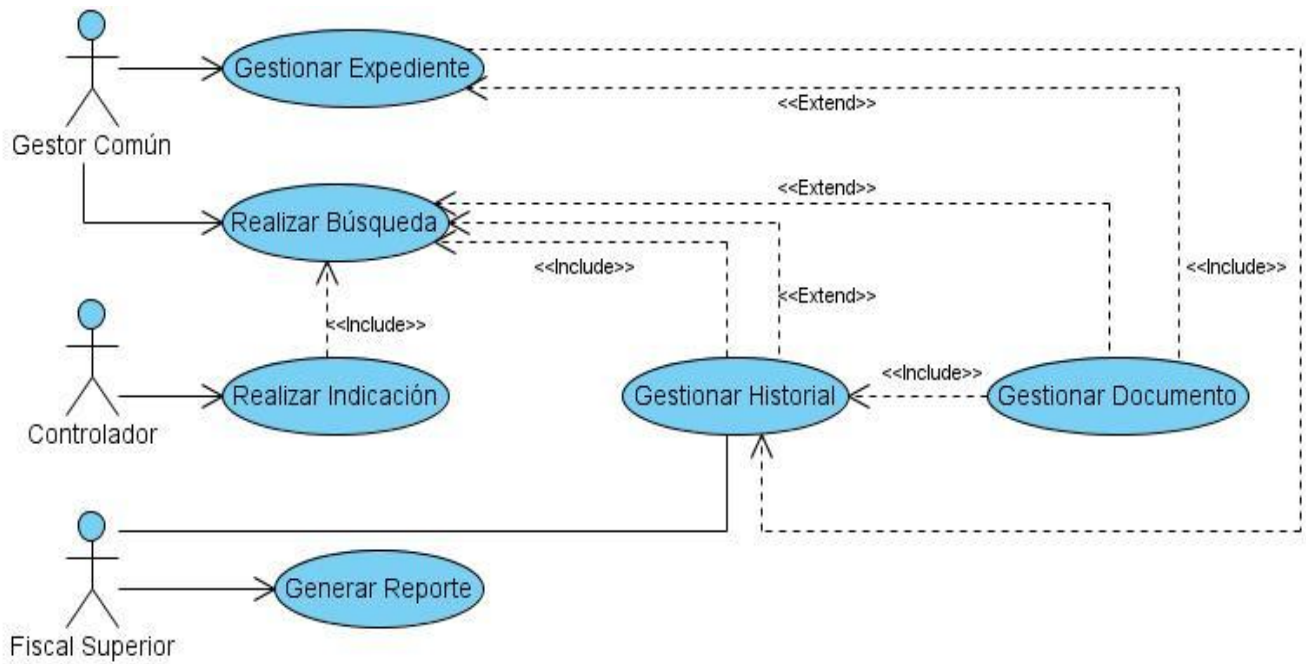


Fig. 6 Diagrama de Casos de uso del sistema

3.2.4 Descripción Textual de los casos de uso del sistema

A continuación se expone la realización de los casos de uso del sistema por prioridad, además de un resumen del caso de uso, precondiciones y sus referencias. El mismo se organiza en 4 casos de uso.

| CUS - Gestionar Expediente | |
|----------------------------|--|
| Actores: | Gestor Común (inicia) |
| Resumen: | El caso de uso comienza cuando el Gestor Común accede a la interfaz Nuevo para introducir un nuevo expediente en el sistema, donde puede imprimir el expediente, emitir documentos legales y demás acciones, el caso de uso termina cuando el Gestor Común no desea crear más expedientes. |
| Precondiciones: | Debe estar autenticado con los permisos necesarios. |
| Referencias: | RF_PP_OR_1, RF_PP_OR_2, RF_PP_OR_4, RF_PP_OR_5, RF_PP_OR_8. |

| | |
|-------------------|----------------|
| Prioridad: | Crítico |
|-------------------|----------------|

Tabla No. 12 Descripción Textual del CU Crear Expediente (véase documento Modelo de casos de uso del sistema del Proceso Ordinario)

| CUS - Realizar Búsqueda | |
|--------------------------------|--|
| Actores: | Gestor Común (inicia) |
| Resumen: | El caso de uso comienza cuando el Gestor Común accede a la interfaz Realizar Búsqueda con el fin de buscar un Expediente o Acusado, el caso de uso termina cuando el Gestor Común no desea realizar más ninguna otra búsqueda. |
| Precondiciones: | Debe estar autenticado con los permisos necesarios. |
| Referencias: | RF_PP_OR_2, RF_PP_OR_3, RF_PP_OR_4, RF_PP_OR_5, RF_PP_OR_6, RF_PP_OR_7, RF_PP_OR_9, RF_PP_OR_11, RF_PP_OR_15. |
| Prioridad: | Crítico |

Tabla No. 13 Descripción Textual del CUS Realizar Búsquedas (véase documento Modelo de casos de uso del sistema del Proceso Ordinario)

| CUS - Gestionar Documento <<include>> | |
|--|---|
| Actores: | - |
| Resumen: | El caso de uso comienza cuando se crea un expediente y el Consultor decide emitir un documento legal sobre un expediente, termina cuando el Consultor no desea realiza mas ninguna emisión sobre el expediente. |
| Precondiciones: | Debe haberse creado un expediente. |
| Referencias: | RF_PP_OR_12, RF_PP_OR_14. |
| Prioridad: | Secundario |

Tabla No. 14 Descripción Textual del CUS Gestionar Documento (véase documento Modelo de casos de uso del sistema del Proceso Ordinario)

| CUS - Gestionar Historial <<include>> | |
|--|---|
| Actores: | - |
| Resumen: | El caso de uso consiste en la actualización del historial, el sistema actualiza automáticamente el historial por crear/editar un expediente y por la emisión de |

| | |
|------------------------|--|
| | un documento, el caso de uso termina cuando no se crea o se emita mas ningún expediente o documento respectivamente. |
| Precondiciones: | Debe haberse creado o editado un expediente. |
| Referencias: | RF_PP_OR_10. |
| Prioridad: | Secundario |

Tabla No. 15 Descripción Textual del CUS Gestionar Historial (véase documento Modelo de casos de uso del sistema del Proceso Ordinario)

| CUS - Realizar Indicación | |
|---------------------------|---|
| Actores: | Controlador (inicia), Fiscal |
| Resumen: | El caso de uso comienza cuando el Controlador realiza indicaciones a un expediente, para eso primero deberá buscar el expediente, el caso de uso termina cuando el Controlador realice las indicaciones necesarias y guarde las mismas. |
| Precondiciones: | Debe estar autenticado. |
| Referencias: | RF_PP_OR_16, RF_PP_OR_17, RF_PP_OR_13. |
| Prioridad: | Secundario |

Tabla No. 16 Descripción Textual del CUS Realizar Indicación (véase documento Modelo de casos de uso del sistema del Proceso Ordinario)

| CUS - Generar Reporte | |
|------------------------|--|
| Actores: | Fiscal Superior (inicia). |
| Resumen: | El caso de uso comienza cuando el Fiscal Superior accede a la interfaz Estadísticas para imprimir reportes, el caso de uso termina cuando el Fiscal no desea generar más ningún reporte. |
| Precondiciones: | Debe estar autenticado. |
| Referencias: | RF_PP_OR_19, RF_PP_OR_20. |
| Prioridad: | Auxiliar |

Tabla No. 17 Descripción Textual del CUS Gestionar Reporte (véase documento Modelo de casos de uso del sistema del Proceso Ordinario)

2.5 Conclusiones parciales

Después de realizar la modelación del negocio, se tiene un conocimiento general de cómo se desarrolla el negocio. Al realizar las actividades correspondientes se generaron los siguientes artefactos: Casos de Uso del Negocio, Actores del Negocio, Trabajadores del Negocio, Entidades del Negocio, Modelo de Casos de Uso del Negocio, Modelo de Objetos del Negocio, Reglas del Negocio y Realización de Casos de Uso del Negocio, posibilitando los mismos conocer cuáles son las actividades posibles a automatizar.

Aplicando las actividades y técnicas utilizadas en la Ingeniería de Requisitos, se logró captar las necesidades expresadas por el cliente, identificando de esta manera los requisitos funcionales y no funcionales permitiendo generar los artefactos: Especificación de Requisitos, Actores del Sistema, Casos de Uso del Sistema y Modelo de Casos de Uso del Sistema.

CAPÍTULO III: VALIDACIÓN

3.1 Introducción

En el presente capítulo se realiza la validación de los prototipos de interfaz con el cliente y se aplican técnicas de validación como son las Revisiones y Listas de Chequeo, se emplean métricas orientadas a la calidad de las especificaciones de los requisitos y casos de usos.

3.1.3 Revisiones

El equipo de calidad del proyecto SGF, realizó la lectura y corrección de toda la documentación generada en los flujos de trabajo Modelado del negocio y Requisitos, donde se señalaron errores de redacción (concordancia de ideas), falta de claridad en algunos requisitos identificados, casos de usos sin referencias y con errores en su estructura; según las pautas establecidas por el proyecto para la descripción de los mismos. Todos estos errores fueron recogidos en el documento de no conformidades (NC). Después de corregidas las NC detectadas, el equipo de calidad realizó una segunda revisión en las cuales se encontraron errores que no quedaron claros en la primera revisión, después de corregidas las mismas se efectuó una tercera revisión la cual concluyó con resultados satisfactorios (**véase Anexo 1**).

3.1.2 Listas de Chequeo

Otra técnica utilizada para la validación de los requisitos fue las Listas de Chequeo. La misma arrojó una serie de requisitos que no habían sido descritos correctamente y tenían cierto nivel de ambigüedad, también se pudo reestructurar algunos requisitos que no estaban lo suficientemente consistentes. Otro elemento a cambiar fue el nombre de algunos requisitos que no dejaban claro su función. De forma general se obtuvieron un total de 8 NC, de las cuales 3 eran significativas para el buen desarrollo de los requisitos. Con el apoyo de la Lista de Chequeo se le dio solución a todas las no conformidades existentes.

La Lista de Chequeo que se realizó se puede consultar en el **Anexo 2**.

3.1.3 Prototipos no funcionales

Los prototipos permiten validar los requisitos identificados que, sin tener la totalidad de la funcionalidad del sistema, permiten al usuario hacerse una idea de la estructura de la interfaz, lo que permite a los desarrolladores conseguir una importante retroalimentación en cuanto a si el sistema diseñado en función de los requisitos le permite al usuario hacer su trabajo de manera eficiente y efectiva. Un prototipo puede revelar errores u omisiones en los requisitos propuestos, favorece la comunicación entre clientes y desarrolladores, da una primera visión del producto.

Los prototipos no funcionales del módulo Ordinario se realizaron en la herramienta Axure RP Pro. 5.5, donde para cada caso de uso se diseñaron interfaces, integradas por un grupo de pantallas que reflejan el flujo que se realiza en las mismas, permitiendo verificar los requisitos que se cumplen en cada una de ellas. Estos prototipos pueden ser consultados en el documento: **Modelo de casos de uso del Sistema**.

3.2 Métricas

Métricas de la calidad de la especificación de los requisitos

En el presente trabajo se realizaron dos revisiones para lograr obtener requisitos fuertes y sin incoherencias, en las mismas se obtuvieron los siguientes resultados a partir de los datos:

$$n_r = 22 + 48 \quad n_r = 70$$

Revisión 1: En la primera revisión se encontraron 5 requisitos que necesitaban ser escritos en un lenguaje más comprensible tanto para el usuario como para los desarrolladores.

$$Q = 65/70 \quad Q = 0.93$$

Revisión 2: En esta revisión se solucionaron los problemas de ambigüedad y se reestructuraron los requisitos que fueron señalados.

Después de los cambios hechos se cuenta con iguales interpretaciones por parte de los revisores para los 70 requisitos.

$$Q = 70/70 \quad Q = 1$$

Después de haber aplicado esta métrica se demostró, gracias a que la mayoría de las interpretaciones de los revisores coincidían, que las especificaciones de los requisitos presentan un alto grado de claridad por lo que los requisitos responden a una única interpretación.

A continuación se representan gráficamente los resultados obtenidos anteriormente:

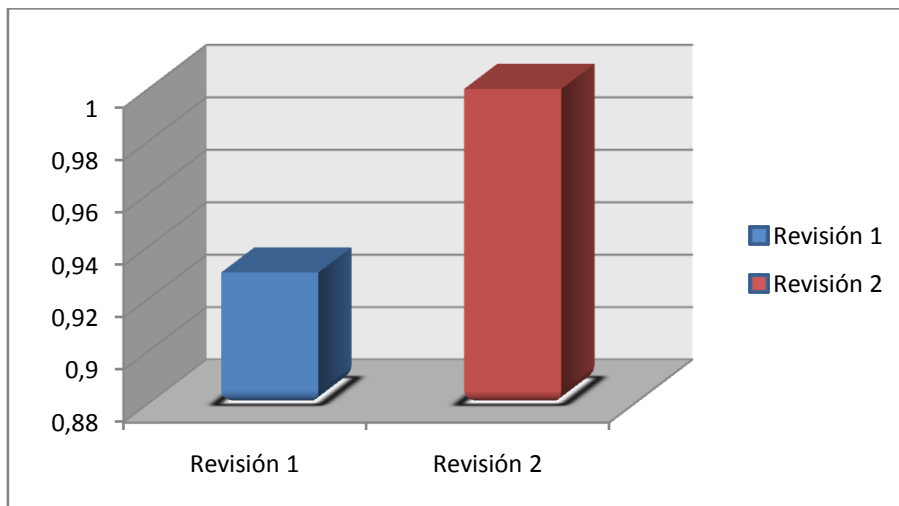


Gráfico 1 Calidad de la Especificación de Requisitos

Métricas aplicadas al diagrama de casos de uso del sistema

A continuación se exponen las métricas que fueron aplicadas al diagrama de caso de uso del sistema:

Primera Revisión:

| Atributo | Factor | Métrica asociada | Valor |
|------------|---|--|---|
| Compleitud | Factor 1. ¿Han sido definidos todos los roles relevantes de usuario encargados de generar/ modificar o consultar información? | Métrica 1 Número de roles relevantes omitidos. | Número de roles relevantes omitidos: 0 Se presenta un 100%. |
| | Factor 2. ¿Se presenta una descripción resumida de todos los casos de uso? | Métrica 2 Número de casos de uso que no tiene descripción resumida. | Número de casos de uso que no tiene descripción resumida: 0 Se presenta un 100%. |
| | Factor 3. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso? | Métrica 3 Número de requisitos omitidos por caso de uso. Métrica 4 Número de casos de uso que tienen requisitos omitidos. | Número de requisitos omitidos por caso de uso: 2 Se presenta un 66.6%. Número de casos de uso que tienen requisitos omitidos: 0 Se presenta un 100%. |
| | Factor 4. ¿Todos los casos de uso han sido clasificados de acuerdo a su relevancia en (crítico, | Métrica 5. Número casos de que no han sido clasificados. | Número casos de que no han sido clasificados: 0 Se presenta un 100%. |

| | | | |
|-------------------------------|--|--|--|
| | secundario, auxiliar, opcional)? | | |
| Se presenta un: 93.2% | | | |
| Consistencia | Factor 5. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario? | Métrica 6 Número de casos de uso que tienen un nombre incorrecto. | Número de casos de uso que tienen un nombre incorrecto: 2 Se presenta un 33.3%. |
| | Factor 6. ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos? | Métrica 7 Grado de adecuación de la descripción del flujo de eventos para un caso de uso. | La descripción se define en el lenguaje del usuario. Se define el responsable de cada acción: 0 Se presenta un 100%. |
| | Factor 7. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable? | Métrica 8 Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema. | Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema: 0 Se presenta un 100%. |
| | Factor 8. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados? | Métrica 9 Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos. | Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos: 0 Se presenta un 100%. |
| Se presenta un: 83.33% | | | |
| Correctitud | Factor 9. ¿Representa el caso de uso requisitos comprensibles por el usuario? | Métrica 10 Número de casos de uso en que los requisitos representados no son comprensibles por el usuario. | Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 0 Se presenta un 100%. |
| | Factor 10. ¿Las interacciones definidas describen la funcionalidad requerida del sistema? | Métrica 11 Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema. | Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema: 2 Se presenta un 33.3%. |
| | Factor 11. ¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado | Métrica 12 Grado en que se ajusta el diagrama | Grado en que se ajusta el diagrama del caso de uso a la metodología: 0 |

| | | | |
|-------------------------------|---|---|--|
| | en la metodología? | del caso de uso a la metodología | Se presenta un 100%. |
| | Factor 12. ¿Las interacciones definidas introducen mejoras al proceso actual? | Métrica 13 Número de casos de uso que deben ser modificados para mejorar el proceso actual. | Número de casos de uso que deben ser modificados para mejorar el proceso actual: 0 Se presenta un 100%. |
| Se presenta un: 83.33% | | | |
| Complejidad | Factor 13. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación? | Métrica 14 Número de elementos del diagrama que requieren reubicación. | Número de elementos del diagrama que requieren reubicación: 0 Se presenta un 100%. |
| Se presenta un: 100% | | | |

Segunda Revisión:

| Atributo | Factor | Métrica asociada | Valor |
|-----------------------------|--|--|--|
| Complejidad | Factor 1. ¿Han sido definidos todos los roles relevantes de usuario encargados de generar/ modificar o consultar información? | Métrica 1 Número de roles relevantes omitidos. | Número de roles relevantes omitidos: 0 Se presenta un 100%. |
| | Factor 2. ¿Se presenta una descripción resumida de todos los casos de uso? | Métrica 2 Número de casos de uso que no tiene descripción resumida. | Número de casos de uso que no tiene descripción resumida: 0 Se presenta un 100%. |
| | Factor 3. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso? | Métrica 3 Número de requisitos omitidos por caso de uso. Métrica 4 Número de casos de uso que tienen requisitos omitidos. | Número de requisitos omitidos por caso de uso: 0 Se presenta un 100%. Número de casos de uso que tienen requisitos omitidos: 0 Se presenta un 100%. |
| | Factor 4. ¿Todos los casos de uso han sido clasificados de acuerdo a su relevancia en (crítico, secundario, auxiliar, opcional)? | Métrica 5. Número de casos de que no han sido clasificados. | Número de casos de que no han sido clasificados: 0 Se presenta un 100%. |
| Se presenta un: 100% | | | |

| | | | |
|-----------------------------|--|--|--|
| Consistencia | Factor 5. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario? | Métrica 6 Número de casos de uso que tienen un nombre incorrecto. | Número de casos de uso que tienen un nombre incorrecto: 0 Se presenta un 100%. |
| | Factor 6. ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos? | Métrica 7 Grado de adecuación de la descripción del flujo de eventos para un caso de uso. | La descripción se define en el lenguaje del usuario. Se define el responsable de cada acción: 0 Se presenta un 100%. |
| | Factor 7. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable? | Métrica 8 Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema. | Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema: 0 Se presenta un 100%. |
| | Factor 8. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados? | Métrica 9 Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos. | Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos: 0 Se presenta un 100%. |
| Se presenta un: 100% | | | |
| Correctitud | Factor 9. ¿Representa el caso de uso requisitos comprensibles por el usuario? | Métrica 10 Número de casos de uso en que los requisitos representados no son comprensibles por el usuario. | Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 0 Se presenta un 100%. |
| | Factor 10. ¿Las interacciones definidas describen la funcionalidad requerida del sistema? | Métrica 11 Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema. | Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema: 0 Se presenta un 100%. |
| | Factor 11. ¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado en la metodología? | Métrica 12 Grado en que se ajusta el diagrama del caso de uso a la metodología | Grado en que se ajusta el diagrama del caso de uso a la metodología: 0 Se presenta un 100%. |

| | | | |
|-----------------------------|---|---|--|
| | Factor 12. ¿Las interacciones introducen mejoras al proceso actual? | Métrica 13 Número de casos de uso que deben ser modificados para mejorar el proceso actual. | Número de casos de uso que deben ser modificados para mejorar el proceso actual: 0 Se presenta un 100%. |
| Se presenta un: 100% | | | |
| Complejidad | Factor 13. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación? | Métrica 14 Número de elementos del diagrama que requieren reubicación. | Número de elementos del diagrama que requieren reubicación: 0 Se presenta un 100%. |
| Se presenta un: 100% | | | |

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

A continuación se representan gráficamente los resultados obtenidos anteriormente:

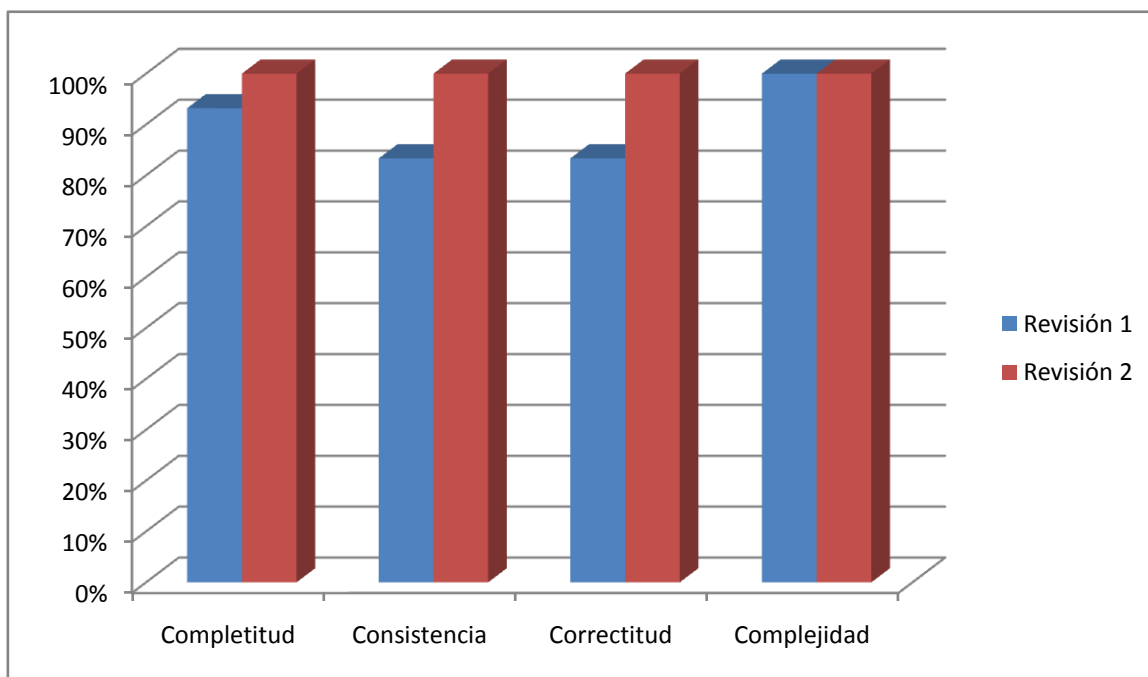


Gráfico 2 Resultados de la aplicación de las métricas para evaluar la calidad de los casos de uso del sistema.

3.3 Conclusiones parciales

Después de haber realizado un análisis de los requisitos y detallados a través de casos de uso, se validaron mediante las técnicas como las revisiones, listas de chequeo y prototipos no funcionales de interfaz de usuario y la aplicación de métricas. Estas técnicas arrojaron errores que fueron solucionados. En la aplicación de la métrica de calidad de los requisitos se obtuvo primeramente un valor de aceptación de 0.93, ya en la segunda revisión el valor fue 1. En la aplicación de la métrica para el caso de uso en la primera revisión el factor Completitud alcanzó el valor de 93.2%, mientras que los factores Correctitud y Consistencia alcanzaron el valor de 83.33% y el de Complejidad el 100%, después de obtenidos estos valores se realizó una segunda revisión donde todos los factores alcanzaron el 100% de aceptación. Después de validados los resultados se puede decir que cuentan con la calidad requerida para cumplir las exigencias planteadas por el cliente.

CONCLUSIONES GENERALES

Después de terminado este trabajo se obtuvieron las siguientes conclusiones:

- Se realizó un estudio del arte de los diferentes Sistema de Gestión Fiscal del mundo y de la Ingeniería de Requisitos, situación que permitió comprender su importancia.
- Se estudió diferentes metodologías, lenguajes de modelado, herramientas case y de modelado de prototipos de interfaz de usuario, permitiendo escoger la más acordes para el análisis del sistema.
- Aplicando las técnicas propuestas por la Ingeniería de Requisitos se logró conocer el negocio y los requisitos del cliente, mejorando la comunicación entre estos y los desarrolladores del módulo Ordinario.
- Con la construcción de los diferentes artefactos propios de los flujos de trabajo, se cumplió la meta y resolvió el problema propuesto.
- La puesta en prácticas de las métricas y otras técnicas de validación, permitieron medir el grado de corrección de los artefactos obtenidos, los cuales están listos a pasar al diseño y continuar el desarrollo del software.

RECOMENDACIONES

Tomando en cuenta los resultados obtenidos:

1. Se recomienda seguir trabajando con la metodología, herramienta y lenguaje seleccionado.
2. Se recomienda realizar la gestión de requisitos para tener un control sobre los riesgos que puedan afectar con el buen desempeño del software.
3. Se recomienda que se continúe con el diseño del módulo Ordinario, basado en el análisis realizado y a partir de la documentación generada.

REFERENCIAS BIBLIOGRÁFICAS

- 610.12-1990, I. S. (24 de 5 de 2004). "IEEE Standard Glossary of Software Engineering Terminology". Obtenido de http://standards.ieee.org/reading/ieee/std_public/description/se/610.12-1990_desc.html
- Arias Chaves, M. (Edición Digital. 2007). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. Revista InterSedes © Universidad de Costa Rica , 4-5.
- Beck, K. (1999). Extreme Programming Explained.
- CGAE. (Octubre de 2006). Obtenido de http://www.cgae.es/portalCGAE/printPortal.do?urlPagina=S001010005/es_ES.html
- Craig, L. (2003). Agile and Iterative Development: A Manager's Guide.
- Escalona, J. M., & Koch, N. (2002). Ingeniería de Requisitos en Aplicaciones para la Web - Un estudio comparativo.
- Escalona, Jose María; Koch, Nora. (2002). Ingeniería de Requisitos en Aplicaciones para la Web - Un estudio comparativo.
- FGR. (1 de Diciembre de 2008). Obtenido de Portal de la Fiscalía General de la República: http://www.cubagob.cu/des_soc/derecho.htm
- Fiscal.es. (2009). Obtenido de <http://www.fiscal.es>
- Gaceta Oficial de la República de Cuba. (s.f.). Obtenido de <http://www.gacetaoficial.cu>
- Gómez, J. (s.f.). Softonic. Obtenido de <http://axure-rp.softonic.com/>
- Infolex. (1998). Obtenido de Gestión Jurídica: <http://www.infolex.es/ie/infolexnet.aspx?menu=4&sub=1>
- Jacobson, I. (2000).
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). "El Proceso Unificado de Desarrollo de Software". Addison Wesley. Capítulos 7, 8 páginas 125-163, 187-202.
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). El Proceso Unificado de Desarrollo de Software.
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). Lenguaje Unificado de Modelado. Manual de Referencia.
- Lodoño. (julio de 2008). Revista EIA. Recuperado el enero de 2009, de [http://revista.eia.edu.co/articulos9/43-52%20\(articulo%203\).pdf](http://revista.eia.edu.co/articulos9/43-52%20(articulo%203).pdf)
- Microsoft Office Online. (s.f.). Recuperado el 2010, de <http://office.microsoft.com/es-es/visio/HA101656403082.aspx>
- Olsina, L. (1999). Metodología cualitativa para la evaluación y comparación de la calidad de sitios web.
- OMG. (2006). Business Process Modeling Notation Specification.
- Pérez, J. D. Notaciones y lenguajes de procesos. Una visión global.
- Pérez, M. (1999). Arquitectura para Ambientes CASE Integrados.

- Pressman, R. S. (2005). Ingeniería de software. Un enfoque práctico. Parte 1. Ciudad de la Habana: Félix Varela.
- Pressman, R. S. (2005). Ingeniería de Software. Un enfoque práctico. Parte II. Ciudad Habana: Félix Varela.
- Rational, S. C. (2003). Ayuda del Rational Unified Process.
- Softlex. (2006). Obtenido de <http://www.softlexplus.de>
- Sommerville, I. (2005). Ingeniería de Software. Séptima Edición.
- Valdés, M. P. (2007).

GLOSARIO DE TÉRMINOS

Requisitoriado: Se dice del despacho en que un juez requiere a otro para que ejecute un mandamiento del recurrente.

Sobreseimiento: Es un tipo de resolución judicial que dicta un juez o un tribunal, suspendiendo un proceso por falta de causas que justifiquen la acción de la justicia. Habitualmente es una institución del derecho procesal penal. En el sobreseimiento el juez, al ver la falta de pruebas o ciertos presupuestos, no entra a conocer el fondo del asunto o se abstiene de seguirlo haciendo, pudiendo terminar el proceso antes de dictar sentencia. Por ese motivo, dependiendo de la legislación, el sobreseimiento no provoca normalmente la situación de cosa juzgada y el proceso se podría reabrir más adelante. Normalmente, el sobreseimiento se dicta mediante un auto, que puede ser objeto de recurso.

Legislación: Conjunto de las leyes de un Estado y también conjunto de leyes relativo a una materia determinada. Estos conjuntos comprenden no solo las leyes propiamente dichas, sino también las normas consuetudinarias y las normas de carácter ejecutivo (reglamentos, etc.)

Ratificar: Aprobar o confirmar actos, palabras o escritos dándolos por valederos y ciertos.

Actuaciones: Todas las providencias, actos, declaraciones, emplazamientos, notificaciones y demás diligencias que constituyen los procedimientos administrativos a los procesos judiciales.

Recurso: En estricto sentido jurídico, aquel que se intenta por vía de alzada, ante un organismo de superior jerarquía para que anule, revoque o modifique actos y decisiones adoptadas por un organismo inferior.

Resolución: Es una decisión de carácter general o particular adaptadas y suscritas por los secretarios del Ejecutivo por disposición del Gobernador, por disposición específica de la ley.

Procura: Cuidado asiduo en los negocios.

Jurisprudencia: Ciencia del derecho, conjunto de las sentencias de los tribunales y doctrina que contienen o criterio sobre un problema jurídico establecido por una pluralidad de sentencias concordes

ANEXOS

Anexo 1 Acta de Liberación de Artefactos

Acta de Liberación de Artefactos, Grupo de Calidad Centro CEGEL de la Facultad 15 de la Universidad de las Ciencias Informáticas.

Martes, 22 de junio de 2010.

Luego de haber efectuado 3 iteraciones de revisiones a los artefactos: Especificación de Requisitos, Modelo de Negocio y Modelo de Sistema del módulo Ordinario del proyecto Sistema de Gestión Fiscal del Centro CEGEL de la Facultad 15 y haberse detectado un promedio de 25 No Conformidades, se puede afirmar que se han corregido los defectos encontrados, por lo que quedan liberados los artefactos.

A handwritten signature in blue ink, appearing to read 'Raúl', written over a horizontal line.

Firma del Asesor y Jefe del Grupo de Calidad Centro CEGEL

Ing. Raúl Velázquez Álvarez



Anexo 2 Lista de Chequeo

| Estructura del documento | | | | | |
|--|---|------|------|---------------------------------|-----------------------------------|
| Peso | Indicadores a Evaluar | Eval | (NP) | Cantidad de elementos afectados | Comentarios |
| Crítico | 1. ¿Está el documento acorde con la plantilla estándar del proyecto o del expediente de proyecto? | 0 | | | |
| | 2. ¿Contiene las secciones obligatorias definidas en el expediente? (Ver Expediente de Proyecto) | 0 | | | |
| Elementos definidos por la metodología | | | | | |
| Peso | Indicadores a Evaluar | Eval | (NP) | Cantidad de elementos afectados | Comentarios |
| Crítico | 1. ¿Están todos los requisitos escritos en un lenguaje entendible para el usuario? | 0 | | | |
| | 2. ¿Están los requisitos a un nivel bastante consistente? | 0 | | | |
| | 2.1 ¿Debería especificarse algún requisito con más detalle? | 1 | | RF_PP_OR_10, RF_PP_OR_7 | Deben describirse con más detalle |
| | 2.2 ¿Debería especificarse algún requisito con menos detalles? | 0 | | | |
| | 3. ¿Son los requisitos lo suficientemente claros para poderse enviar a un grupo independiente para su implementación y que lo entiendan? | 0 | | | |
| | 4. ¿Cada requisito establece lo que el sistema debe cumplir? | 0 | | | |
| | 5. ¿Los requisitos, incluyen aspectos relacionados con la funcionalidad, rendimiento, limitaciones de diseño, atributos (atributos de datos y estructura de datos)? | 0 | | | |

| | | | | | |
|--------------------------------|--|-------------|-------------|--|---|
| | 6. ¿Han sido abordadas e identificadas los valores de entradas y salidas? | 0 | | | |
| | 7. ¿Cada requisito ha sido identificado para indicar ya sea la importancia (proyecto y/o cliente) o la estabilidad de ese requisito? | 0 | | | |
| | 8. ¿Se puede verificar cada requisito? | 0 | | | |
| | 9. ¿Cada requisito es relevante al problema y a su solución? | 0 | | | |
| | 10. ¿Se han especificado todos los posibles cambios en los requisitos, incluyendo la probabilidad de cambio? | 1 | | | No están especificados los cambios posibles a los requisitos. |
| | 11. ¿No aparece un mismo requisito en más de un lugar del documento de especificación? | 0 | | | |
| | 12. ¿Hay requisitos que tienen más de una interpretación? | 0 | | | |
| | 13. ¿Hay un glosario en el cual los requisitos significativos y específicos están en términos definidos y claros? | 0 | | | |
| | 14. ¿Hay requisitos que produzcan inquietud? | 0 | | | |
| | 15. ¿Están incluidos todos los requisitos relacionados con el hardware? | 0 | | | |
| Semántica del documento | | | | | |
| Peso | Indicadores a Evaluar | Eval | (NP) | Cantidad de elementos afectados | Comentarios |
| Crítico | 1. ¿Está organizado correctamente? | 0 | | | |

| | | | | | |
|--|---|---|--|--|--|
| | 2. ¿Ha identificado errores ortográficos? | 0 | | | |
| | 3. ¿Se entiende claramente lo que se ha especificado en el documento? | 0 | | | |