

Universidad de las Ciencias Informáticas
Facultad 3

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

**Título: Análisis de los subprocesos asociados al
subsistema Administrativo del Proyecto Tribunales
Populares Cubanos.**

Autor: Roxana Pérez Arcia

Tutor: Ing. Aray Pérez Degue

Asesor Técnico: Chavely Larramendy Tellez

Habana, Cuba



Está mi alma apegada a tí, tu diestra me ha sostenido Señor.

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Roxana Pérez Arcia

(Autor)

Aray Pérez Degue (Tutor)

Chavely Tellez Larramendi (Asesor Técnico)

Agradecimientos

Le agradezco primeramente a mi Dios, quien me ha dado la vida y me ha ayudado a llegar hasta aquí, todo cuanto tengo y soy se lo debo a Él, en verdad yo puedo decir Ebenezer, hasta aquí me ha ayudado el Señor. Agradecerle a Dios por la oportunidad que me dio de venir a esta universidad que no solo formó en mí una ingeniera informática sino también una mujer apasionada por Jesús.

Agradecerles a mis padres por su amor incondicional por confiar siempre en mí y apoyarme en todas mis decisiones, los admiro mucho por criarme como lo han hecho y los amo tal como son, a mis padres lindos les doy muchas gracias. A mis dos abuelitas por ser cariñositas conmigo, a Illa por sus oraciones a Dios por mí en todos estos años de universidad y por su cariño sin igual. A mi familia en su totalidad por estar siempre pendientes de mis necesidades y dispuestos a darme cuanto necesité en la medida de sus posibilidades claro, a tía Nena por ayudarme y quererme como a una hija más y a mi primo Yunier por ser incondicional conmigo, a tía Celia por cargar conmigo aquí en La Habana. A mi hermana que aunque hace poco es que nos hemos venido a relacionar bien, ha sido de mucha bendición para mí, espero Dios permita que podamos permanecer comunicándonos y llevándonos así bien y aun mejor, cuando me valla para mi monte. A Lili que ya no me parece solamente hermana en la fe sino también de la familia, por su cariño y hospitalidad durante estos años en la universidad.

A mis hermanos queridos, cómo no agradecerles a ellos que tantas veces me han visto llorar y reír, a ellos les agradezco por ser una familia preciosa donde siempre encontré amor, cariño, respeto y palabra buena para crecer en la fe, presencia de Dios y ayuda en la necesidad. Gracias a todos mis hermanitos, no voy a mencionar nombres para que no se me pongan celositos los quiero mucho a todos, soy bienaventurada de haberles conocido a cada uno de ustedes, nunca, nunca los voy a olvidar.

A mis compañeros de aula y apartamento, también a ellos les quiero agradecer, porque aunque mi carácter y principios les eran chocantes en ocasiones, lograron aceptarme al menos y algunos hasta me cogieron cariño. De una forma extraña ellos también han sido instrumentos de Dios para mi vida, simplemente gracias. Espero haber sido de alguna forma extraña también de bendición a sus vidas.

A mi tutora Aray que viene desde tercer año tutorándome, por su amabilidad y comprensión, por ayudarme con la tesis siendo una buenísima tutora. A y a Chavely por su ayuda también con la tesis, gracias a ambas.

A mi tribunal por ayudarme a ir perfeccionando la tesis desde un inicio con sus recomendaciones e indicaciones para poder llegar hasta aquí.

Dedicatoria

Le dedico esta tesis a mi Señor Jesucristo por ser mi Salvador Eterno, al Espíritu Santo por ser mi guía, ayudador y consolador, a Mi Gran Dios le dedico esta tesis, como una pequeñísima corona que arrojé a sus sublimes pies.

A mis padres, por ser una bendición que Dios me ha dado, porque no hay para mí padres mejores en el mundo, ya que la mezcla del alegre y vivaz Héctor y la silenciosa y amable Norma dio lugar a un nuevo elemento X a Roxanita, jeje. A ustedes padres míos también les dedico esta tesis.

A mi familia bonita, amable y numerosa.

A mis hermanos de la UCI por ser una de las más grandes bendiciones que Dios me ha dado en la vida.

Resumen

La Universidad de las Ciencias Informáticas con el apoyo de los Tribunales Populares Cubanos desarrolla el proyecto TPC para la creación de una solución informática que permita la gestión automatizada de sus procesos. Uno de los procesos a automatizar es el procedimiento Administrativo que tiene como objetivo a través de la vía judicial darle solución a diferentes problemas que surgen por inconformidades ante resoluciones dictadas por los órganos administrativos y está compuesto por un proceso principal y cinco subprocesos asociados: ejecución, costas procesales, suspensión, solicitud de casación y archivo. Con el fin de comprender y realizar un correcto modelado de una solución informática para la informatización de los subprocesos asociados al procedimiento Administrativo en el presente trabajo se plasma el análisis realizado a los mismos.

Se identificaron, analizaron, especificaron y validaron los requisitos aplicando el proceso de Ingeniería de Requisitos. Se generaron los artefactos: Modelo del negocio, Especificación de requisitos de software, Modelo de sistema y Prototipo de interfaz no funcional. A partir de esta propuesta y con su seguimiento, se espera desarrollar una solución informática capaz de lograr el control de la información generada en los Tribunales Populares Cubanos, contribuyendo a la calidad de la tramitación de los procesos.

Índice de contenidos

Introducción	1
Capítulo 1: Fundamentación teórica.....	6
1.1 Introducción	6
1.2 Proceso Administrativo y subprocesos asociados.....	6
1.3 Informática Jurídica.....	7
1.4 Soluciones informáticas para la gestión de procesos judiciales	8
1.5 Ingeniería de requisitos	10
1.6 Metodologías de desarrollo de software	15
1.7 Lenguaje de modelado	20
1.8 Herramientas de modelado.....	24
1.9 Herramientas para modelado de prototipos no funcionales	27
1.10 Métricas de software.....	28
Conclusiones.....	30
Capítulo 2: Propuesta de solución.....	31
2.1 Introducción.....	31
2.2 Modelado del Negocio	31
2.3 Descripción de los subprocesos asociados al procedimiento Administrativo	31
2.4 Definición de Actores y Trabajadores del Negocio.....	32
2.5 Flujograma de subprocesos.....	33
2.6 Requisitos de Software.....	40
2.7 Definición de actores del Sistema.....	44
2.8 Diagrama de Caso de Uso del Sistema	45
2.9 Realización de los Casos de Uso del Sistema.....	49
Conclusiones.....	56
Capítulo 3: Análisis de los resultados.....	57
3.1 Validación de requisitos mediante prototipos no funcionales	57
3.2 Métricas de la calidad de la Especificación de requisitos.....	57
3.3 Modelo de métricas orientadas a objeto aplicadas al DCUS.....	59
Conclusiones generales.....	66

Recomendaciones	67
Bibliografía.....	68
Glosario	71

INTRODUCCIÓN

Una de las características del mundo actual es el desarrollo progresivo de las Tecnologías de la Información y las Comunicaciones (TIC). Las TIC pueden ser utilizadas ampliamente para beneficiar la forma de trabajo y de vida del hombre. Actualmente en una sociedad donde se requiere rapidez y efectividad en la realización del trabajo, el uso de las TIC constituye una necesidad. Por este motivo numerosos países buscan su inserción en todas las ramas sociales y económicas. Como resultado de esta revolución tecnológica han surgido una serie de ciencias enfocadas en estudiar y guiar el uso de las TIC en una rama específica de la sociedad, ejemplo de ello es la Informática Jurídica.

La Informática Jurídica estudia la aplicación de los ordenadores para ejercer el derecho de forma tal que permita la reducción de problemas jurídicos. La Informática Jurídica se divide en tres grandes ramas, la Informática Jurídica documental, la de apoyo a la decisión y la Informática Jurídica de gestión. Esta última facilita mediante la automatización, la labor de las oficinas relacionadas con el derecho a nivel público (los tribunales que administran justicia).

En Cuba se han trazado metas ambiciosas a la altura de países del primer mundo a fin de lograr la informatización de la sociedad. La Universidad de las Ciencias Informáticas (UCI) ha contribuido en gran medida a este proceso de informatización. Actualmente la UCI se encuentra organizada en centros de desarrollo con propósitos de producción específicos. Uno de estos centros de desarrollo es el Centro de Gobierno Electrónico (CEGEL). En este se agrupan varios proyectos donde se desarrollan servicios informáticos para las principales instituciones gubernamentales del país como parte de la evolución de la Informática Jurídica en Cuba.

El centro CEGEL con el apoyo del Ministerio de Justicia específicamente de los Tribunales Populares Cubanos desarrolla el proyecto Tribunales Populares Cubanos (TPC) para la creación de una solución informática que estandarice los procesos y los documentos jurídicos, que ejecute y supervise los procesos de manera automática conforme a las disposiciones legales que los regulan y que posibilite generar notificaciones y reportes electrónicos.

El proyecto TPC está conformado por siete subsistemas, siendo uno de ellos el subsistema Administrativo, cuyo objetivo fundamental es la informatización del procedimiento

Administrativo encargado de resolver los conflictos que se derivan por la inconformidad ante actos administrativos¹ dictados por los órganos administrativos² y que tienen prevista la reclamación en la vía judicial, este procedimiento está constituido por un proceso principal y cinco subprocesos asociados; ejecución, archivo, suspensión, costas procesales y solicitud de casación.

Los subprocesos asociados al procedimiento Administrativo se realizan en los tribunales provinciales de Cuba, donde se atiende de forma manual un enorme cúmulo de expedientes y disímiles trámites legales. Estos expedientes son guardados en estantes, lugar donde la búsqueda de documentos resulta engorrosa y se encuentran expuestos al deterioro y en ocasiones a la pérdida. Los jueces y abogados a la hora de realizar cada trámite deben tener en cuenta un término fijado en la Ley de Procedimiento, Civil, Laboral y Económico y al existir una forma de trabajo poco centralizada en ocasiones se violan alguno de estos importantes términos. Este hecho da lugar a una serie de errores como la duplicación de resoluciones y de asientos en los libros de la sala, errores a la hora de registrar una información proveniente de las parte, entre otros. Todo lo anterior provoca que no exista un control óptimo de la información y por ende un atraso a la hora de darle solución a los procesos.

La situación descrita evidencia la necesidad de crear un solución informática que permita la realización más eficiente de los subprocesos asociados al procedimiento Administrativo. Debido a que estos subprocesos son grandes y complejos, se hace necesario un previo análisis para poder identificar cuándo se realizan dichos subprocesos, cómo funcionan y quiénes los ejecutan. En fin identificar cuáles son las necesidades reales del cliente, para así traducirlas a un lenguaje entendible para los desarrolladores a fin de lograr un correcto desarrollo de los subprocesos del subsistema Administrativo del proyecto Tribunales Populares Cubanos.

Para dar solución a la situación problemática descrita anteriormente surge como problema de investigación: ¿Cómo transformar las necesidades del cliente a un lenguaje comprensible por los desarrolladores que facilite el desarrollo de los subprocesos del subsistema Administrativo?

¹ Actos administrativos: documento legal que se emite por parte de órgano administrativo.

² Órganos Administrativos: Algunos ejemplos son: ONAT, Vivienda, Pesca, Educación.

Para dar solución a este problema se define como objeto de estudio de la investigación al proceso de desarrollo del software y como campo de acción de la investigación la ingeniería de requisitos de software de los subprocesos asociados al procedimiento Administrativo.

El objetivo general que se persigue es: realizar la elicitación, análisis, especificación y validación de los requisitos de los subprocesos competentes al procedimiento Administrativo para transformar las necesidades del cliente a un lenguaje comprensible por los desarrolladores de modo que facilite su desarrollo.

Por lo anteriormente expresado se propone la siguiente idea a defender: Con el análisis de los subprocesos asociados al procedimiento Administrativo de los Tribunales Populares Cubanos se logrará el entendimiento entre clientes y desarrolladores, facilitando así su desarrollo.

Se tiene como objetivos específicos:

- Elaborar el marco teórico de la investigación.
- Identificar y especificar los requisitos funcionales de software.
- Validar los resultados obtenidos.

Para poder lograr los objetivos específicos propuestos se plantearon las siguientes tareas de investigación:

- ✓ Estudiar del estado del arte de la Ingeniería de Requisitos y de los sistemas de gestión de procesos judiciales de tribunales.
- ✓ Selección de las metodologías de desarrollo de software y las herramientas CASE candidatas que faciliten el modelado de procesos de negocio y los lenguajes de modelado que se utilizarán para lograr una comprensión de los subsistemas.
- ✓ Modelado de los subprocesos del procedimiento Administrativo en los Tribunales Provinciales.
- ✓ Identificación y especificación de los requisitos que deberá tener el software.
- ✓ Validación de los requisitos a través de prototipos no funcionales del sistema.
- ✓ Validación de los resultados obtenidos del módulo Administrativo para evaluar la calidad del producto.

Para dar cumplimiento a las tareas propuestas anteriormente se emplean métodos científicos, teóricos y empíricos de la investigación

Se emplearon los siguientes métodos teóricos:

- El método Analítico-Sintético, que permitió realizar un estudio teórico de la investigación y un análisis previo sobre el funcionamiento de los subprocesos asociados al procedimiento Administrativo y extraer los elementos más importantes relacionados con el objeto de estudio.
- El método Histórico-Lógico usado para analizar a nivel internacional y nacional el empleo de sistemas informáticos de gestión jurídicos, así como investigaciones realizadas anteriormente sobre el tema.
- El método de Modelación se empleó para la creación modelos que representen abstracciones con el objetivo de comprender el funcionamiento de los subprocesos asociados al procedimiento Administrativo que se lleva a cabo en los Tribunales Provinciales Cubanos.
- El método Inducción-Deducción, permitió arribar a la proposición de una vía de solución del problema planteado. Se utilizó para la formulación de la idea a defender.

Dentro de los métodos empíricos se usaron:

- La entrevista, realizada a jueces de la sala de lo Administrativo que se pusieron a disposición del equipo de analistas, esta permitió recoger toda la información referente al funcionamiento de los subprocesos asociados al procedimiento Administrativo.
- La observación se utilizó para lograr la comprensión de los subprocesos en cuestión, el prolongado análisis del funcionamiento de estos subprocesos en los propios Tribunales Provinciales Cubanos posibilita una mejor visión de los mismos.

Se propone estructurar el trabajo investigativo en tres capítulos y varios anexos como se muestra a continuación:

Capítulo1: Fundamentación Teórica: En este primer capítulo se plasma el resultado de la investigación realizada referente a los diferentes sistemas informáticos que gestionan procesos judiciales existentes, se definen los conceptos más importantes sobre la ingeniería de requisitos y se describen brevemente las técnicas de obtención de requisitos, se realiza

además un análisis crítico de las metodologías de desarrollo de software existentes, lenguajes de modelado y herramientas CASE para el desarrollo del software justificándose así el uso de las variantes seleccionadas. Y se muestra el resultado del estudio de algunas métricas para medir la calidad.

Capítulo 2: Solución Propuesta: En este capítulo se describe el negocio de los subsistemas asociados al procedimiento Administrativo que se lleva a cabo en los Tribunales Provinciales. A partir de aquí se realiza el Modelado del Negocio, obteniéndose los siguientes artefactos: Actores del Negocio, Trabajadores del Negocio y Flujograma de Procesos. Además se muestran los Requisitos Funcionales y los Requisitos no Funcionales identificados. Seguidamente se presentan los principales artefactos obtenidos en cuanto a sistema: Actores del Sistema, Diagrama de Casos de Uso y la descripción textual de cada Caso de Uso con el prototipo de interfaz.

Capítulo 3: Análisis de los Resultados: En este capítulo se aplican algunas métricas para garantizar la calidad de la especificación de los requisitos y del diagrama de casos de uso del sistema, así como los resultados obtenidos en la validación mediante prototipos y la validación realizada por el equipo de calidad de la Facultad 3

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se presenta un estudio sobre el procedimiento administrativo que se realiza en los Tribunales Provinciales Cubanos, así como los subprocesos asociados al mismo. Se muestra una breve reseña del concepto de Informática Jurídica y su evolución histórica. Se describen algunas soluciones informáticas que gestionan el proceso judicial en Cuba y el mundo, además se exponen los principales resultados obtenidos al estudiar el proceso de desarrollo de software, destacando la ingeniería de requisitos, sus actividades y las técnicas de obtención de requisitos. Se analizan las herramientas y metodologías actuales para la realización de ingeniería de requisitos de forma que se puedan justificar las empleadas.

1.2 Proceso Administrativo y subprocesos asociados

Los tribunales agrupan sus procesos por materia, siendo una de ellas la materia Administrativa, quien ejecuta tres procedimientos: Administrativo, Recurso de Casación y Revisión. El procedimiento Administrativo se atiende en la instancia Provincial y está compuesto por un proceso principal y 5 subprocesos. El presente trabajo se enfocará en los cinco subprocesos asociados al procedimiento Administrativo.

Se denomina procedimiento Administrativo, en término judicial, a la sucesión de trámites mediante los cuales una persona que se considere afectada por una decisión o acto de algún órgano de la administración pueda reclamar ante la Sala de lo Civil y Administrativo del Tribunal Provincial.

Se considera Administración a los Organismos de la Administración Central del Estado y a los Órganos Locales del Poder Popular, aunque existen otros como Direcciones Municipales de la Vivienda y la Oficina Cubana de la Propiedad Industrial. Las decisiones que adoptan las administraciones se plasman en actos judiciales que son notificados a los interesados y éstos cuentan con treinta días hábiles para establecer demanda ante el tribunal por hallarse inconformes con lo resuelto, porque consideran que con ella se lesiona un derecho, que en su opinión tienen legalmente preestablecido.

Los subprocesos asociados a este procedimiento son: Ejecución, Solicitud de casación, Archivo, Suspensión y Costas Procesales.

El subproceso de Ejecución se desarrolla cuando el proceso termina con una sentencia con lugar, con lugar en parte, por nulidad de oficio o con un auto de nulidad que implique que la administración dicte un nuevo acto y lo presente ante el tribunal en el término concedido. Cuando la administración no presenta este acto en el tribunal se realizan una serie de eventos jurídicos que terminan con la recepción del nuevo acto.

El subproceso de Solicitud de casación tiene lugar una vez dictada la sentencia cuando una de las partes inconforme con el resultado emitido presenta una solicitud para que su caso sea tramitado en el Tribunal Supremo Popular, lo que conlleva a una serie de trámites que finalizan cuando se eleva el expediente en cuestión al Tribunal Supremo Popular para que se tramite allí.

El subproceso de archivo es el que pone fin a un proceso en el cual, el expediente es cerrado y archivado, se devuelven todos los expedientes o documentos legales que contenga el expediente a la institución propietaria y la devolución o confiscación de una fianza pagada por cuestiones de suspensión si se solicitó en el proceso.

El subproceso de suspensión comienza cuando la parte actora solicita la suspensión del acto administrativo impugnado, esto implica un fallo por parte del juez ponente para decidir si admitirá o no la solicitud el pago de fianza y termina al notificar a la administración de lo dispuesto por el tribunal.

El subproceso de costas procesales comienza cuando la parte beneficiada con el resultado del proceso, solicita que se realice el cómputo del dinero que ha perdido por cuestiones del proceso, lo que genera una realización de este cómputo por parte de las secretarías y la notificación a la contraparte de su deber de pagarle a la parte afectada el dinero estimado.

1.3 Informática Jurídica

1.3.1 Concepto de Informática Jurídica

La Informática Jurídica es el resultado de la unión de la informática con el derecho, esta es la ciencia encargada de guiar la práctica del derecho a través de medios informáticos.

Es la interrelación de las materias informática y derecho que tiene como fin el análisis, la estructuración lógica y ordenada, la deducción e interpretación de la información jurídica a través de la utilización de la máquina computadora para su efectivo y eficaz tratamiento, administración, recuperación acceso y control y cuyos alcances están predeterminados al auxilio en la toma de decisiones jurídicas. (Riesta, 1995)

Constituye el tratamiento automatizado de las fuentes de conocimiento jurídico (sistemas de documentación legislativa, jurisprudencial y doctrinal), de las fuentes de producción jurídica y su organización (funcionamiento de organismos legislativos y judiciales) y de las decisiones judiciales. (Luño, 2008)

Según el jurista español Antonio Enrique Pérez Luño, la Informática Jurídica se divide en tres áreas:

- Informática Jurídica documental: automatización del conocimiento jurídico emanado de fuentes legislativas, jurisprudenciales y doctrinales;
- Informática Jurídica decisional: automatización de las fuentes de producción jurídica;
- Informática Jurídica de gestión: automatización de los procesos de administración judicial.

La Informática Jurídica como se plantea anteriormente constituye la práctica del derecho mediante el uso de ordenadores para lograr una efectiva y eficaz ejecución de los procesos judiciales.

1.4 Soluciones informáticas para la gestión de procesos judiciales

La Informática Jurídica tiene sus inicios en el año 1949 y desde entonces se han dado pasos lentos pero progresivos. Numerosos países ya han logrado insertar el uso de ordenadores en oficinas judiciales, como un medio de apoyo al almacenamiento y creación de documentos legales. Algunos países han llegado más lejos, creando sistemas informáticos que ayudan a la gestión de los procesos jurídicos, convirtiendo así al ordenador en una herramienta para ejercer el derecho.

1.4.1 Soluciones informáticas en el mundo

En este epígrafe se muestran algunas soluciones informáticas creadas en diferentes países para la gestión del proceso judicial en los mismos.

ATLANTE: Sistema Informático desarrollado en Canarias que comenzó a instalarse en todas las dependencias judiciales del Archipiélago en el año 2007. Este sistema registra todas las diligencias realizadas por las distintas instituciones, remite la información adecuada al siguiente paso, aunque los funcionarios esperan que lleguen los documentos en papel para procesarlos. Actualmente se despliega Atlante II, pero este ha ocasionado problemas en los juzgados, donde los funcionarios deben trabajar "a mano" debido a las continuas "caídas" de los servidores.

LEXNET: Es un sistema de gestión de notificaciones telemáticas usado en la Administración de Justicia española que permite la comunicación bidimensional entre las oficinas judiciales y los distintos operadores jurídicos. Actúa sobre la base de la tecnología de los certificados digitales, permite el uso de la firma digital, en los documentos electrónicos, la presentación de escritos y notificaciones. Es un sistema de intercambio de documentos judiciales en formato electrónico.

1.4.2 Soluciones informáticas en Cuba

SisProp: Sistema creado en Villa Clara propuesto para abarcar las instancias Supremo y Provincial en la materia de lo Penal. Se encarga de la tramitación de los procesos y apelaciones pero no capta datos de la fase judicial del proceso, de manera que no aporta ninguna estadística ni información.

SisEco: Sistema creado en La Habana en el año 2002 para el área de la estadística en el procedimiento Económico. En él se insertan los documentos radicados manualmente y las salvas diariamente se guardan en disquetes. La secretaria de estadística recoge el libro de radicación de escritos (LRE) e inserta los datos en la aplicación y cada cinco años borra la información, quedando solamente asentada en los libros.

1.4.3 Necesidad de una solución informática que gestione la tramitación judicial.

Una vez realizado el estudio de varios de los sistemas existentes para la gestión de procesos judiciales se ha llegado a la conclusión de que ninguno de estos satisface a las necesidades

presentadas por el organismo Tribunales Populares Cubanos. Se puede señalar que LEXNET es sólo un sistema para notificaciones, recepción de escritos digitales y traslado de copias. ATLANTE mantiene la tramitación de forma manual, sólo permite registrar diligencia y remite la información al siguiente paso que le corresponde. Es válido aclarar que estos sistemas no se guían por la Ley Judicial vigente en Cuba. Los sistemas realizados no se ajustan al proceso Administrativo debido a que fueron diseñados específicamente para la materia Penal SisProP y para la materia Económica SisEco.

Lo anteriormente planteado justifica la creación de un sistema que se rija por la Legislación vigente en Cuba que contenga la información general del proceso judicial y de las variantes posibles, que permita su tramitación digital de forma total, que notifique de forma clara a aquellos que interactuaran con el mismo el trámite que corresponde de acuerdo al estado en que se encuentre, genere automáticamente todas las resoluciones, oficios, citaciones y notificaciones propios de la tramitación de los procesos que se traten. Además que brinde la posibilidad de alertar el vencimiento de término de cada proceso y permita obtener reportes estadísticos en tiempo real.

Como parte de la solución informática, se realiza el módulo Administrativo, para desarrollar e integrar todas las funcionalidades que permitan gestionar el proceso Administrativo y sus subprocesos asociados. De manera que se sistematice, digitalice y centralice todas las acciones que se llevan a cabo en el Tribunal Provincial en esta materia.

1.5 Ingeniería de requisitos

Al comenzar un proceso de desarrollo de software se realiza la especificación del software, actividad en la que se define la funcionalidad y restricciones operacionales que debe cumplir el software. Para su realización es necesario utilizar la ingeniería de requisitos.

1.5.1 Requisito de software

Para comprender en que consiste la ingeniería de requisitos es necesario conocer el concepto de requisito de software. Muchos son los conceptos que se han emitido al respecto, para Sommerville: “Un requisito es simplemente una declaración abstracta de alto nivel de un

servicio que debe proporcionar el sistema o una restricción de este. En el otro extremo es una declaración detallada y formal de una función del sistema.” (Sommerville, 2005)

Clasificación de los requisitos de software

Para Straud Berros existen dos tipos de requisitos, los requisitos de usuario, que son descripciones ya sea en lenguaje natural o diagramas de lo que se espera que el sistema posea así como las restricciones que podrá presentar el mismo. También están los requisitos del sistema, los cuales establecen con más detalle las funciones, servicios y restricciones operativas que el sistema tendrá. Estos requisitos se conocen como requisitos de software, y se pueden dividir en dos partes: requisitos funcionales o requisitos no funcionales.

Según Summerville:

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y en algunos casos declarar explícitamente lo que el sistema no debe hacer. Los requisitos funcionales del sistema describen de forma detallada la función de este, sus entradas y salidas, excepciones, etcétera. Por su parte los requisitos no funcionales, son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema sino a las propiedades emergentes de este como son la fiabilidad, rendimiento, usabilidad, disponibilidad, el tiempo de respuesta, la capacidad de almacenamiento y otras propiedades. Los requisitos de dominio se derivan del dominio de aplicación del sistema más que de las necesidades específicas de los usuarios. Normalmente incluyen terminología especializada del dominio o referencias a conceptos del dominio. Pueden ser requisitos funcionales nuevos, restringir los existentes o establecer como se deben ejecutar cálculos particulares.

1.5.2 Ingeniería de requisitos

La ingeniería de requisitos es la ciencia encargada de identificar qué es lo que se debe construir, en verdad ocupa un papel importantísimo en un proceso de desarrollo de software. Constituye la base sobre la cual ha de edificarse el sistema o software a desarrollar, se puede afirmar que la eficacia del resultado depende en gran medida de la realización de una buena ingeniería de requisitos.

En la actualidad debido al alto coste de los errores que se generan por la incorrecta realización de las actividades de la ingeniería de requisitos ha crecido el interés por los requisitos de software. Esta actividad es considerada una disciplina clave en la Ingeniería de Software. Como consecuencia de esto ha surgido el término Ingeniería de Requisitos para de esta forma englobar los procesos de desarrollo y gestión de requisitos en el ciclo de vida del software y comprender las necesidades exactas de los usuarios y transformarlas al lenguaje de los desarrolladores para lograr un mejor entendimiento entre los clientes y todo el equipo de desarrollo. (Sommerville, 2005)

“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 de un sistema que satisfaga las necesidades del usuario. La ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán”. (Pressman, 2005)

1.5.3 Tareas para realizar ingeniería de requisitos

La ingeniería de requisitos comienza con la fase de inicio, tarea que define el ámbito y la naturaleza del problema que debe resolverse. Continúa con la obtención, que es una tarea que ayuda al cliente a definir sus necesidades; sigue con la elaboración, que es la fase donde se refinan y modifican los requisitos básicos. Cuando el cliente ha definido el problema se lleva a cabo la negociación, donde se define cuáles son las prioridades, cuales aspectos son esenciales y en qué momento se requieren. Por último el problema se especifica y después es validado y revisado para asegurar que la concepción que tiene el ingeniero informático coincide con la percepción del cliente. (Pressman, 2005)

Debido a la trascendencia que posee la ingeniería de requisitos para el desarrollo de un sistema, diferentes autores han propuesto una serie de tareas o actividades a realizar con el fin de ejecutar una correcta ingeniería de requisitos, de las cuales sobresalen: elicitación, análisis y negociación, especificación, validación y gestión de requisitos; encontrándose estas vinculadas estrechamente ya que las salidas de cada actividad constituye la entrada necesaria para la siguiente actividad, evidenciando que su ejecución es continua e iterativa.

Elicitación de requisitos: la etapa de elicitación de requisitos abarca la primera y quizás más importante fase dentro del desarrollo de un sistema informático. Uno de los retos más importantes de la elicitación de requisitos es garantizar que los requisitos del sistema sean consistentes con las necesidades de la organización. (Villanueva, 2000) Es el proceso durante el cual se identifica la información que determina las características deseadas y las restricciones que deberá satisfacer el software, que tendrán efectos satisfactorios para el usuario, en el ambiente donde se encuentra. Realizándose con el fin de conocer el dominio del problema y obtener una especificación preliminar detallada de las necesidades de los usuarios del software a desarrollar. (García Ávila, 2008)

Para el logro de resultados precisos durante la elicitación se utilizan técnicas o métodos como son: las entrevistas, arqueología de documentos, los prototipos y los escenarios.

- **Entrevistas:** las entrevistas sirven para obtener una comprensión general de lo que hacen los stakeholders, cómo podrían interactuar con el sistema y las dificultades a las que se enfrentan con los sistemas actuales. A la gente le gusta hablar sobre su trabajo y normalmente se alegran de verse implicados en entrevistas. (Sommerville, 2005)
- **Arqueología de Documentos:** permite determinar posibles requisitos sobre la base de inspeccionar la documentación utilizada y generada en la empresa. Sirve fundamentalmente como complemento de las demás técnicas y mediante la misma se obtiene información que de otra forma sería imposible, como son manuales de procedimientos, reglamentos, facturas, entre otras. (Davila, 2001)
- **Prototipos:** para validar los requisitos hallados durante la etapa de captura se crean prototipos, los cuales son simulaciones del posible producto, que luego son utilizadas por el usuario final. Provee a los analistas de una importante retroalimentación, ya que les permite conocer si el sistema diseñado sobre la base de los requisitos capturados, les permite a los usuarios realizar su trabajo de forma efectiva y eficiente. (Escalona, 2005)
- **Escenarios:** los escenarios pueden ser especialmente útiles para agregar detalle a un esbozo de la descripción de requisitos. Son descripciones de ejemplos de las sesiones de interacción. El escenario comienza con un esbozo de la interacción y, durante la

obtención, se agregan detalles para crear una descripción completa de esta interacción.(Sommerville, 2005)

Análisis y negociación de requisitos: una vez recopilados los requisitos, el producto obtenido configura la base del análisis de requisitos. 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. Es corriente en clientes y usuarios solicitar más de lo que puede realizarse, también es común en clientes proponer requisitos contradictorios. El ingeniero del sistema debe resolver estos conflictos a través de un proceso de negociación. Utilizando un procedimiento iterativo, se irán eliminando requisitos, se irán combinando y/o modificando para conseguir satisfacer los objetivos planteados. (Pressman, 2005)

Especificación de requisitos: una especificación puede ser un documento escrito, un modelo gráfico, un modelo matemático formal, una colección de escenarios de uso, un prototipo o una combinación de lo anteriormente citado. El resultado obtenido sirve como fundamento a la ingeniería de hardware, de software, de base de datos y la humana. (Pressman, 2005). En esta fase se documentan los requisitos acordados con el cliente, en un nivel apropiado de detalle. (Chaves, 2007). Durante la especificación se registran los requisitos de forma detallada en un documento formal, es decir constituye la documentación detallada de los requisitos, este documento es de gran utilidad para etapas posteriores en el desarrollo del software.

Validación de requisitos: la validación es la etapa final de la IR. Su objetivo es, ratificar los requisitos para asegurarse que representan una descripción, por lo menos, aceptable del sistema que se debe implementar. Se examinan las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedades, 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. (Pressman, 2005). Esta actividad es de gran importancia porque permite identificar los errores y detalles que no se hayan detectado en las actividades anteriores.

Gestión de requisitos: con esta actividad se pretende llevar un control sobre los cambios que pueden sufrir los requisitos debido a que se haya cambiado el problema que se estaba

resolviendo, o simplemente cambiaron las expectativas de los clientes. Para gestionar los requisitos se llevan a cabo “un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento”. (Pressman, 2005)

Para la aplicar una ingeniería de requisitos satisfactoria, acorde a las necesidades de los clientes y el equipo de desarrollo se decide realizar cuatro de las actividades de la IR: elicitación, análisis y negociación, especificación y validación de los requisitos. La etapa de gestión de requisitos no se recoge en el presente trabajo, debido a que los requisitos correspondientes a los subprocesos en cuestión están bien definidos y las expectativas del cliente están claramente especificadas por lo que es poco probable que ocurran cambios en los mismos, aunque si puede ocurrir un incremento del número de requisitos, lo cual se tendrá presente para próximas iteraciones del software. Para la realización de la elicitación se propone utilizar la Entrevista a los especialistas en la materia Administrativa y la Arqueología de documentos, aplicada a escritos y expedientes radicados en salas de lo Civil y Administrativo y que competen a la materia.

1.6 Metodologías de desarrollo de software

Si se quiere construir un software de alta calidad, desarrollado en el tiempo planificado, con los costes establecidos, es necesario enfocarse en trabajar de forma organizada, donde se controle y documente todo lo relacionado con el proyecto en cuestión y puedan eliminarse los riesgos que podrían presentarse durante el desarrollo del mismo, esto no podría lograrse sin el empleo de una metodología eficaz que se adapte a las características propias del software que se esté desarrollando.

Piattini define a la metodología de desarrollo de software como “un conjunto de procedimientos, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software”.

Las metodologías considerando su filosofía de desarrollo se dividen en dos grandes grupos las metodologías tradicionales y las metodologías ágiles.

1.6.1 Metodologías ágiles o ligeras

A principios de la década del 90, surgió un enfoque que fue bastante revolucionario para su momento ya que iba en contra de la creencia de que mediante procesos altamente definidos se iba a lograr obtener software en tiempo, costo y con la requerida calidad. El enfoque fue planteado por primera vez en 1991 y se dio a conocer en la comunidad de ingeniería de software con el mismo nombre que su libro, RAD o Desarrollo Rápido de Aplicaciones. RAD consistía en un entorno de desarrollo altamente productivo, en el que participaban grupos pequeños de programadores utilizando herramientas que generaban código en forma automática tomando como entradas sintaxis de alto nivel. (Pressman, 2002)

Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable. (Pressman, 2005).

Entre las metodologías ágiles más utilizadas tenemos a: EXtreme Programming y a SCRUM

➤ **EXtreme Programming (XP)**

Esta metodología fue desarrollada por Kent Beck. Está centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares que mantienen el código legible para los miembros del equipo, facilitando los cambios. Se basa en realimentación continua entre el cliente y el equipo de desarrollo. XP se define como especialmente adecuada para proyectos de corto plazo con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. (Penadés)

➤ **SCRUM**

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante

iteraciones, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Estas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (Penadés)

SCRUM no es una metodología “mágica”, sino un marco de posibles estrategias, respetando unas reglas prefijadas. Una plantilla a la que después se puede acoplar distintas técnicas de planificación, gestión, control, estimación, análisis, diseño, implementación y pruebas. (Sutherland, 1990)

1.6.2 Metodologías tradicionales

Las metodologías tradicionales son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. Estas metodologías han demostrado ser más efectivas y necesarias en proyectos de gran envergadura, que involucra a un amplio equipo de desarrollo. (Rodríguez, 2010)

Las metodologías tradicionales son muy usadas en proyectos para los que sea necesario realizar una planificación detallada de todo el trabajo, llevar un control del proceso, esta última función la asegura mediante una rigurosa definición de roles, artefactos, actividades, herramientas, notaciones para el modelado y documentación detallada. Entre las metodologías tradicionales existentes se pueden citar a MSF (Marco de Solución de Microsoft) y RUP (Proceso Unificado de Desarrollo).

Microsoft Solutions Framework (MSF)

MSF o Marco de Solución de Microsoft es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan el desarrollo y la planificación de proyectos que pueden adaptarse a cualquier proyecto de tecnología de información. Es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. MSF no es un marco estático, sin embargo evoluciona respondiendo a los cambios en la tecnología y en los requisitos de los proyectos, es una disciplina de desarrollo de soluciones enfocada en

los modelos únicos de equipos y procesos, usados para organizar grupos efectivos de trabajo y administración del ciclo de vida de un proyecto. (Sanchez, 2004)

Entre sus características principales se encuentran:

- Adaptable: es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- Escalable: puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- Flexible: es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología Agnóstica: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

Rational Unified Process (RUP)

RUP o Proceso Unificado de Desarrollo constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El Proceso Unificado es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema de software en construcción está formado por componentes de software (Ivar Jacobson, 2000). RUP brinda una guía para encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y restricciones. Utiliza una notación de Caso de Uso y escenarios para representar los requisitos. (Sanchez, 2004)

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales:

Proceso dirigido por CU: en RUP los casos de uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. (Kruchten, 2000)

Proceso centrado en la arquitectura: la arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo. En el caso de RUP se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. (Kruchten, 2000)

Proceso iterativo e incremental: RUP propone un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto. (Ivar Jacobson, 2000)

El Proceso unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Al final de cada uno de ellos se obtiene una versión final del producto, que no sólo satisface ciertos casos de uso, sino que está lista para ser entregada y puesta en producción. Cada ciclo se compone de varias fases y flujos de trabajo, y dentro de cada una, se pueden descomponer adicionalmente el trabajo en iteraciones, con sus incrementos resultantes (Ivar Jacobson, 2000).

RUP es una metodología de desarrollo de software marcada por una gran organización, que garantiza la evolución en el proceso de desarrollo y la creación de un producto que se ajuste a las necesidades de los usuarios. Establece la creación de un modelo visual conformado por una serie de artefactos modelados con el Lenguaje Unificado de Modelado (UML). Por todo lo anterior expresado se puede afirmar que RUP define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto.

1.6.3 Justificación de la metodología a utilizar

No existe una metodología mejor que otra, su uso está condicionado por la compatibilidad de las características particulares de dichas metodologías con las requeridas por el proyecto en cuestión o al software a desarrollarse. Al realizar una valoración de las características de las metodologías de desarrollo expuestas en el presente epígrafe se llega a la conclusión de que se utilizará RUP como metodología de desarrollo de software, debido a que es una

metodología robusta que se adapta muy bien a proyectos de larga duración, complejos y con un gran equipo de desarrollo como es el caso del proyecto TPC. Por las características del proyecto el cliente no siempre se encuentra presente y es de gran importancia la extensa documentación que ofrece RUP, lo que garantiza poder brindarle al cliente una visión clara de lo que se está realizando en cada etapa del desarrollo. No así la metodología XP que ha sido diseñada especialmente para proyectos de corto plazo, ni SCRUM que se usa mayormente en proyectos más bien pequeños y poco complejos debido al gran número de iteraciones que comprende y en el caso de MSF por ser un modelo de Microsoft exige que se utilicen sus herramientas, lo que impediría el uso de otras herramientas que posean mayores potencialidades para el proyecto.

1.7 Lenguaje de modelado

Un modelo es una representación parcial o simplificada de la realidad que recoge aquellos aspectos de relevancia y lenguaje podría definirse como el conjunto de símbolos útiles para comunicarnos. El lenguaje de modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar parte de un diseño de software orientado a objetos.

Las ramas más antiguas de la ingeniería han encontrado útil desde hace mucho tiempo representar los diseños mediante dibujos. En el presente trabajo se realiza una breve descripción del Método para la Modelación Funcional de Procesos (IDEF0), el Lenguaje Unificado de Modelado UML y la Notación de Modelado de Procesos BPMN.

Método para la Modelación Funcional de Procesos (IDEF0)

La traducción literal de las siglas IDEF es Definición de la integración para la modelización de las funciones. IDEF consiste en una serie de normas que definen la metodología para la representación de funciones modeladas. (Andrés, 2008)

IDEF0 constituye una técnica de modelación gráfica, especializada en la representación de las relaciones e interdependencias existentes entre los diferentes procesos. Es capaz de diferenciar entre tres tipos posibles de relación entre procesos:

- Relaciones que establecen las guías que debe tener en cuenta el proceso.
- Relaciones que aportan los recursos necesarios para llevar a cabo el proceso.

- Relaciones de encadenamiento lineal entre procesos (entrada – salida). (Andrés, 2008)

Un proceso en IDEF0 se representa mediante cinco elementos fundamentales: entrada (s), salida (s), control (es), sujeto (s) y la actividad. IDEF0 es un medio comprensivo que posibilita comunicar reglas y procesos de negocios. Se puede utilizar para modelar una variedad amplia de sistemas automatizados y no automatizados. Para los nuevos sistemas, se puede utilizar primero para definir los requisitos y para especificar las funciones, y después para diseñar una puesta en práctica que resuelva los requisitos y realice las funciones. Puede ser generado por una gran variedad de herramientas gráficas en computadores. (Knowledge Based System, 2006)

IDEF0 es aplicado comúnmente como medio para comunicar reglas y procesos de negocios, para la obtención de la visión estratégica de cualquier proceso de negocios y para facilitar el análisis para la identificación de áreas de mejora.

Unified Modeling Language (UML)

El lenguaje unificado de modelado es un lenguaje estándar de modelado para software. No define un proceso de desarrollo específico, tan solo se trata de una notación. Permite modelar sistemas de información, y su objetivo es lograr modelos que, además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los clientes o usuarios de aquello que se modela. (Ivar Jacobson, 2000) Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema. (Enciclopedia Online Wikipedia, 2008). UML es útil para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar. (Andrés, 2008)

Algunas de las propiedades como lenguaje de modelado son las que se exponen en (Ivar Jacobson, 2000):

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.

- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas. Incluye los siguientes diagramas: diagrama de casos de uso, diagrama de clases, diagrama de objetos, diagrama de secuencia, diagrama de colaboración, diagrama de estados, diagrama de actividades, diagrama de componentes, diagrama de despliegue. (Ivar Jacobson, 2000)

Bussines Process Modeling Notation (BPMN)

BPMN es una notación para el modelado de procesos de negocios que tiene como objetivo principal servir como soporte para la gestión por procesos, como una notación que pueda ser entendida fácilmente desde los analistas que crean los bocetos iniciales del proceso, los desarrolladores técnicos responsables de implementar la tecnología que ejecutará estos procesos, hasta las personas que los ejecutan y aquellas que llevarán a cabo el monitoreo y supervisión de los procesos. En otras palabras esta notación crea un enlace entre las etapas de diseño e implementación. Pero a pesar de ser intuitiva para todos los usuarios de negocio es capaz de representar semánticas de procesos complejos. (Andrés, 2008) Con BPMN un diagrama puede ser transformado en un código ejecutable automáticamente, sin la necesidad de programación. De esta forma, el analista de negocios puede definir, diseñar y generar una solución a sus procesos. (Rodríguez, 2010)

Las características principales de BPMN según Eric Durocher son:

- Visibilidad de los procesos de las empresas.
- Mayor flexibilidad y agilidad para adaptación al cambio.

- Brinda la posibilidad de integrar la información del negocio dispersa en diferentes sistemas y permite adquirir una ruta de mejoramiento y eficiencia continua al convertir actividades ineficientes en menores costos a través de uso de tecnología enfocada en procesos.

BPMN es un estándar internacional de modelado de procesos aceptado por la comunidad, es independiente de cualquier metodología de modelado de procesos. Crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos (Headquarters, 2008). Genera diagramas formados por un conjunto de elementos gráficos. Estos habilitan el fácil desarrollo de diagramas simples que serán familiares para la mayoría de analistas de negocio (diagrama de flujo). Las cuatro categorías básicas de elementos son: Objetos de flujo, Objetos de conexión, Swimlanes (Carriles de piscina) y Artefactos.

BPMN presenta como ventaja su gran expresividad a la hora de especificar procesos, mucho más expresivo que los diagramas de actividad de UML, es gráficamente más rico, con menos símbolos, lo que facilita su comprensión por parte de personas que no tienen dominio del lenguaje. (Business Process Managements Notation, 2007)

1.7.1 Justificación del lenguaje de modelado seleccionado

Al realizar una valoración detallada de los lenguajes para el modelado caracterizados anteriormente, se decidió utilizar un híbrido, a BPMN como lenguaje para la representación del negocio, ya que este es orientado a procesos y de esta forma es más entendible a los usuarios, quienes trabajan actualmente con un sistema de justicia estructurado por procesos y a UML para continuar con el ciclo de desarrollo de RUP debido a las grandes potencialidades que brinda, ejemplo de ello es que permite visualizar, especificar, construir y documentar un sistema de software, ventajas favorables para realizar un correcto modelado del sistema. Se desestimó el uso de IDEF0 que a pesar de permitir el modelado de procesos, es un lenguaje que visualiza fundamentalmente los elementos de entrada, salida y controladores de los procesos modelados con una simbología limitada y no muestra legiblemente el flujo o la relación entre los procesos, en cambio BPMN es un lenguaje que si muestra de forma clara la relación entre procesos del negocio.

1.8 Herramientas de modelado

Las herramientas CASE son diversas aplicaciones informáticas destinadas a aumentar los aspectos claves de todo el proceso de desarrollo de un software, desde el principio hasta el final. “CASE proporciona al ingeniero la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Las herramientas CASE ayudan a garantizar que la calidad se diseñe antes de llegar a construir el producto”. (Pressman, 2005)

Una herramienta no es más que un sistema de software que intenta proporcionar ayuda automatizada a las actividades a realizar durante el proceso de desarrollo de software que permitirá aumentar la velocidad de desarrollo de los sistemas. En el presente estudio se analizaron las herramientas: Rational Rose, Visual Paradigm y el Enterprise Architect.

Rational Rose

Rational Rose Enterprise es el producto más completo de la familia Rational Rose. Todos los productos Rational Rose incluyen soporte UML. Posee soporte para análisis de patrones y un control por separado de componentes modelo que permite una administración más granular y el uso de modelos. Brinda soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5 y genera código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables. (Innova, 2008)

Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

Algunas de las características propias de Rational son las nombradas por Rubén González Blanco y Sergio Pérez Tobalina:

- Permite Especificar, Analizar, Diseñar el sistema antes de Codificarlo
- Mantiene la consistencia de los modelos del sistema software
- Chequeo de la sintaxis UML
- Generación de documentación automáticamente

- Generación de código a partir de los modelos.
- Ingeniería Inversa (Tobalina)

Visual Paradigm

Visual Paradigm para UML 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. Ofrece un entorno de creación de diagramas para UML 2.1. Visual Paradigm permite realizar un diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad, posee capacidades de ingeniería directa e inversa. El modelo y código permanece sincronizado en todo el ciclo de desarrollo. Existen múltiples versiones disponibles para ser usadas según cada necesidad. Permite la integración con los principales IDEs³ de desarrollo y posee disponibilidad en múltiples plataformas. (Caballero, 2009)

Visual Paradigm es útil para realizar modelado UML, posee características gráficas muy cómodas que facilitan la realización de todos los diagramas de modelado que sigue el estándar de UML. Permite la integración con IDE's como son: NetBeans (de Sun), Eclipse (de IBM), JBuilder (de Borland). Permite ingeniería inversa para: JAVA, .NET, XML e Hibernate. Permite además exportar imágenes jpg, png y svg. (Altamirano, 2008)

Visual Paradigm posibilita la generación de código y la base de datos a partir de los diagramas UML realizados, aporta más características no menos interesantes para el desarrollador como es la realización de Ingeniería Inversa, generación automática de informes en formato PDF, Word o HTML; generación de máquinas de estados, integración con Visio y Rational Rose. (Pavón, 2008)

Enterprise Architect

El Enterprise Architect es un ambiente de modelado para trabajo en equipo, potenciado por UML 2.3, para la realización de un diseño gráfico que ayude a que el equipo de desarrollo construya y mantenga un robusto sistema. Enterprise Architect tiene incorporada la capacidad

³ Conocido por sus siglas en inglés Integrated Development Environment y traducido al español como Entorno de Desarrollo Integrado, que es un programa informático compuesto por un conjunto de herramientas de programación.

de manejar de forma clara los requisitos en un sistema o negocio, esto ayuda al arquitecto de empresa a desarrollar especificaciones con un alto nivel de análisis, diseño, implementación, prueba y mantenimiento usando UML, SysML⁴, BPMN y otras normas de modelado. Es una herramienta que abarca el ciclo de vida completo del desarrollo de software, con herramientas que pueden proveerle una estructura competitiva en modelado de negocio, diseño de software, ingeniería de sistemas, arquitectura de empresas, gestión de requisitos, pruebas y mucho más. (Sparx Systems, 2010)

Es una herramienta multiusuario⁵, con seguridad y administración de permisos incorporada. Soporta diferentes repositorios basados en DBMS (Sistemas Manejadores de Base de Datos), incluyendo Oracle, SQL Server, My SQL, PostgreSQL. Provee trazabilidad completa desde el análisis de requisitos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue (System, 2008)

1.8.1 Justificación de la herramienta de modelado seleccionada

Teniendo en cuenta las características antes expuestas de las herramientas analizadas se llega a la conclusión de que se utilizará Visual Paradigm como herramienta de modelado de software ya que posee alta capacidad de integración con el lenguaje orientado a procesos BPMN y UML que serán los utilizados durante el desarrollo del sistema. La aplicación a desarrollar será programada en PHP y Visual Paradigm se integra fácilmente con este lenguaje. Además el equipo de desarrollo cuenta con un mayor conocimiento de esta herramienta que de otras descritas como por ejemplo del Enterprise Architect que a pesar de ofrecer similares funcionalidades que RUP es desconocido por el equipo de desarrollo, posibilitando ahorro de tiempo en el desarrollo del software ofreciendo capacitación de los desarrolladores. Se decide no utilizar la herramienta Rational Rose debido a que no permite la integración con el lenguaje BPMN ni con el lenguaje de programación seleccionado.

⁴ Conocido por sus siglas en inglés Systems Modeling Language, que se traduce al español como Lenguaje de Modelado de Sistema.

⁵ Multiusuario: Propiedad que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente.

1.9 Herramientas para modelado de prototipos no funcionales

La creación de prototipos de programas informáticos es el proceso de crear un modelo incompleto del programa informático futuro, que se puede utilizar para dejar a los usuarios tener una primera idea del programa terminado o permitir que los clientes evalúen el programa. El cliente puede comparar si el software hecho se adecua a la especificación del software, según la cual se construye el programa informático.

Microsoft Office Visio 2007

Visio se utiliza para crear una gran variedad de dibujos, desde diagramas de red hasta calendarios y desde diseños de oficina hasta diagramas de flujo.

Mediante el uso de diagramas, Microsoft Office Visio 2007 facilita a los profesionales de informática y de los negocios la visualización, el análisis y la comunicación de la información, los sistemas y los procesos complejos. Microsoft Office Visio 2007 ofrece un entorno de dibujo de Microsoft dedicado y familiar que se completa con una amplia gama de plantillas, formas y herramientas sofisticadas diseñadas para facilitar la creación de una gran variedad de diagramas empresariales y técnicos. Microsoft Office Visio 2007 permite aumentar la productividad integrando diagramas con información de varias fuentes. Permite además visualizar y actuar sobre información compleja mostrando datos en diagramas, así como analizar datos y realizar de forma sencilla el seguimiento de tendencias, la identificación de problemas y la señalización de excepciones con diagramas dinámicos. Visio comunica información compleja con nuevas plantillas y formas y transmite información de manera eficiente con diagramas de aspecto profesional. (Visio, 2007)

Visio es una herramineta muy potente que permite representar de forma clara planos, maquetas, prototipos no funcionales y muchos otros diagramas. Los prototipos no funcionales creados mediante Visio son prototipos estáticos, que muestran los componentes que ha de tener una página web o formulario, pero Visio se ve limitado en cuanto a la forma de representar la secuencia o interrelación entre cada prototipo creado.

Axure RP Pro

Axure RP es una aplicación ideal para crear prototipos y especificaciones muy precisas para páginas web. Se trata de una herramienta especializada en la tarea, así que cuenta con todo lo necesario para crear prototipos de forma más eficiente. Permite componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad (Gómez, 2007). Axure RP es una herramienta fácil de usar dónde se producen de forma instantánea modelos funcionales. Esta herramienta posee un alto grado de personalización y una versión de administración que permite la creación de modelos interactivos con anotaciones y facilita la creación de un diseño efectivo. (Ladrin, 2010)

Axure RP Pro permite la rápida creación de prototipos no funcionales para aplicaciones web, debido en gran medida a que fue diseñada exclusivamente para esta función y está por tanto equipada con los elementos necesarios para esto. Es una de las herramientas más utilizadas sobre plataformas Windows en lo que a prototipado y gestión de requisitos se refiere.

1.9.1 Justificación de la herramienta para el modelado de prototipos no funcionales seleccionada.

Analizando las características de las herramientas de modelado estudiadas se decide utilizar la herramienta Axure RP Pro. Esta herramienta por ser diseñada explícitamente para el prototipado, posee todos los elementos necesarios para la creación de prototipos amigables que pueden ser exportados como un proyecto o como fotos de fácil uso para el equipo de desarrollo, también pueden generarse como páginas html dinámicas, funcionalidad que permite al usuario interactuar de forma directa con los prototipos, observando el flujo consecutivo de los eventos, permitiéndoles de esta manera tener una idea más clara de cómo será la aplicación en un futuro y de esta forma sentirse identificados y motivados con la solución a elaborar. Visio posee también los elementos necesarios para crear prototipos no funcionales amigables, pero se encuentra en desventaja frente a Axure RP Pro en cuanto a la representación de la interrelación entre cada prototipo, pues no muestra el flujo consecutivo de eventos.

1.10 Métricas de software

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo del software y los proyectos de mantenimiento. La medición persigue tres objetivos

fundamentales: ayudarnos a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que acontece en nuestros proyectos y poder mejorar nuestros procesos y nuestros productos. Las métricas del software es un término que se asigna a un amplio rango de actividades diversas, por ejemplo: medidas y modelos de estimación de coste y esfuerzo, modelos y medidas de productividad, aseguramiento y control de calidad, recogida de datos, medidas y modelos de calidad y modelos de fiabilidad. (Muñoz, 2008)

En el proceso de desarrollo de software son medibles fundamentalmente tres elementos: el proceso, el producto y los recursos. En el presente estudio se hace necesaria la medición de aquellos productos intermedios o artefactos que son generados en la fase inicial de desarrollo propuesta por RUP, para esto se empleará la Métrica de la calidad de Especificación de los requisitos, el Modelo de métricas orientadas a objeto aplicadas al DCUS y una Métrica para medir el nivel de aceptación del cliente.

Métricas de la calidad de la Especificación de requisitos

Es de gran importancia comenzar a medir la calidad desde las primeras etapas del ciclo de vida de desarrollo software para asegurar la calidad del producto final. La calidad de la especificación de requisitos generada en la etapa inicial del desarrollo puede ser medida.

Para realizar la validación de los requisitos existe toda una lista de características que sugieren el uso de una o más métricas como son: especificidad (ausencia de ambigüedad), corrección, compleción, comprensión, capacidad de verificación, consistencia externa e interna, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización. (Pressman, 2005).

Modelo de métricas orientadas a objeto aplicadas al DCUS.

Esta tecnología permite aumentar la productividad y mejorar la calidad del software, constituyendo las metodologías orientadas a objetos un marco idóneo para abordar la complejidad creciente en el desarrollo del software, gracias a sus modelos más realistas y consistentes, dando lugar a productos más flexibles y fáciles de mantener y reutilizar. Las razones del cambio desde el enfoque estructurado al enfoque orientado a objetos (OO), se refieren principalmente a la mejora de la productividad, de la calidad y del mantenimiento del software desarrollado que ofrece esta nueva tecnología. Otro aspecto del enfoque OO que lo

hace adecuado para garantizar la calidad de los productos si se aplica desde las primeras fases de los proyectos es su capacidad para modelar los requisitos de los usuarios de una forma muy intuitiva. (Hilera, 2003)

A raíz del estudio realizado, para medir la calidad de algunos de los productos intermedios que serán generados en la realización del subsistema Administrativo, es decir los artefactos generados en la fase inicial de desarrollo propuesta por RUP se empleará la Métrica de la Calidad de Especificación de los Requisitos y la Métrica Modelo de Métricas Orientadas a Objeto Aplicadas al DCUS.

Conclusiones

El análisis de algunas soluciones informáticas existentes que gestionan los procesos judiciales, demuestran la necesidad de crear el módulo Administrativo, debido a que estas no se ajustan a las leyes que rigen al procedimiento Administrativo, ni a las necesidades existentes en el Tribunal Provincial.

El estudio de las diferentes metodologías de desarrollo de software, lenguaje de modelado, herramientas CASE y herramientas para el modelado de prototipos no funcionales, permitió conocer cuáles son las que más se ajustan para el desarrollo del módulo Administrativo, en este caso las seleccionadas fueron: RUP, UML y BPMN y Axure RP Pro. Sentándose las bases para comenzar la realización del modelado del módulo Administrativo.

Se decide utilizar la Métrica de la calidad de la Especificación de requisitos y Modelo de Métricas orientadas a objeto aplicadas al DCUS para garantizar la calidad de los resultados que se obtendrán.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 Introducción

Teniendo como base la investigación realizada, en el presente capítulo se presenta una descripción de los subprocesos asociados al procedimiento Administrativo que se lleva a cabo en los Tribunales Provinciales mediante la identificación de las personas involucradas en el mismo y aquellas actividades que requieren ser informatizadas. Para esto se muestran los siguientes artefactos: actores y trabajadores del negocio, flujograma de procesos y reglas del negocio. Se realizará además la Especificación de requisitos funcionales y requisitos no funcionales y el Modelado del sistema para los subprocesos competentes al módulo Administrativo.

2.2 Modelado del Negocio

Una de las actividades de obligatoria ejecución en el proceso de desarrollo de software es el análisis y descripción del negocio, esta etapa permite entender cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito.

2.3 Descripción de los subprocesos asociados al procedimiento Administrativo

Durante la resolución de un proceso Administrativo ocurren una serie de eventos procesales que pueden o no tener lugar, porque dependen de la solicitud hecha por las partes involucradas en el proceso.

El subproceso de costas procesales depende de una solicitud presentada por la parte beneficiada luego de dictada sentencia en la que pida realizar una tasación de los costos del proceso para que la parte desfavorecida le pague los gastos en que incurrió durante el proceso esta puede ser admitida o no y de ser admitida generará una serie de actos judiciales y resoluciones asociadas, así como una disposición del pago de estas costas.

Los organismos pueden disponer actos administrativos que perjudiquen de forma apremiante a las partes demandantes, estas partes pueden presentar una solicitud de suspensión de este acto administrativo mientras se resuelva el caso, si el tribunal admite esta solicitud deberá

pagar una fianza que ha de ser devuelta una vez terminado el proceso, si este concluye a su favor y de lo contrario será confiscada.

Una vez dictado una auto que ponga fin al proceso o una sentencia las partes pueden presentar un escrito solicitando elevar el expediente en cuestión al Tribunal Supremo para que se resuelva allí, el juez ponente puede o no admitir esta solicitud y de hacerlo así el caso pasará a tramitarse en el Tribunal Supremo una vez emplazados los abogados por tablilla.

La ejecución de la sentencia o auto definitivo por su parte tendrá lugar solo en aquellos casos que la sentencia declare con lugar, con lugar en parte o con nulidad de oficio la demanda presentada o se termine el proceso mediante un auto de nulidad. Dichos supuestos implican que la administración dicte un nuevo acto diferente al impugnado y que presenten al tribunal en un término de treinta días una copia del mismo. En caso de que la administración presente el nuevo acto en término, la secretaria lo registraría, de no ser así se realiza un requisito a la administración y al jerárquico superior de la misma y en caso necesario la notificación de este hecho a la Fiscalía de la República.

El archivo ocurre siempre que se termina un proceso ya sea una vez dictada la sentencia o por un auto definitivo que ponga fin al proceso emitido por el juez ponente. Durante el archivo se desglosan los expedientes gubernativos presentados por la administración y otros expedientes o documentos pertenecientes a alguna institución que se hayan traído al tribunal como pruebas y que requieren ser devueltos. Además si se pagó alguna fianza por cuestión de suspensión es necesario realizar el desglose del slip bancario para la devolución de la fianza en caso de que la sentencia termine con lugar, con lugar en parte o por nulidad de oficio o enviar un oficio al banco para que se confisque la fianza.

2.4 Definición de Actores y Trabajadores del Negocio

Los actores son aquellas personas que se benefician directamente con el Negocio, para los que la realización de Casos de Uso tiene un resultado visible. Los trabajadores son las personas o sistemas automatizados o semi-automatizados que llevan a cabo las diversas tareas o actividades para dar cumplimiento al negocio.

Tabla 1: Descripción de Actores del Negocio.

Actores	Descripción
Demandante	Es la persona encargada de presentar legalmente ante el tribunal una demanda o escritos

Tabla 2: Descripción de Trabajadores del Negocio.

Trabajadores	Descripción
Abogado	Es la persona encargada de realizar todas las acciones legales pertinentes a las personas o entidades representadas por él.
Secretaria	Es la persona encargada de mantener la relación y comunicación entre el juez ponente, abogado y por ende entre las partes involucradas.
Juez Ponente	Es la persona con la máxima responsabilidad en un caso, encargada de guiar los procesos en el mismo, analizarlos y emitir un resultado, dándole fin al caso.

2.5 Flujograma de subprocesos

Todos los subprocesos fueron descritos y representados mediante BPMN utilizando la herramienta Visual Paradigm. A continuación se muestra la descripción correspondiente a los subprocesos Costas Procesales y Suspensión. Para obtener la información correspondiente al resto de los subprocesos ver el documento Modelo de negocio de los subprocesos administrativos, adjunto a la investigación.

Subproceso Costas procesales

Ficha de Proceso	
Proceso:	Costas procesales
Responsable:	Juez ponente
Ejecutor (es):	Secretaria, Juez ponente
Descripción:	El subproceso de costas procesales es aquel en el que una de las

partes (debe ser la parte demandada si es beneficiada con el resultado del proceso) solicita al tribunal que se realice la tasación de costas para que la otra parte realice el pago de las costas procesales. El juez ponente revisa el escrito presentado y decide si lo admite lo rechaza o lo manda a subsanar. Si lo admite dispone que la secretaria realice la tasación de las costas y una vez realizada la misma procede a darle traslado a las partes de la tasación realizada. Las partes pueden presentar un escrito de impugnación de la tasación y este siempre es admitido por el juez ponente quien señala una audiencia verbal para atender la cuestión y luego de celebrada esta, el juez dispone la rectificación o la admisión de la tasación y la parte actora procede a pagar las costas ante el tribunal.

Entradas:	Escrito de costas procesales Escrito de impugnación de la tasación de costas
Salidas:	Auto o providencia
Procesos asociados:	Administrativo
Reglas del Negocio:	RN. 51; RN.52; RN.53; RN.54

Clasificación:	Secundario
Flujo básico:	

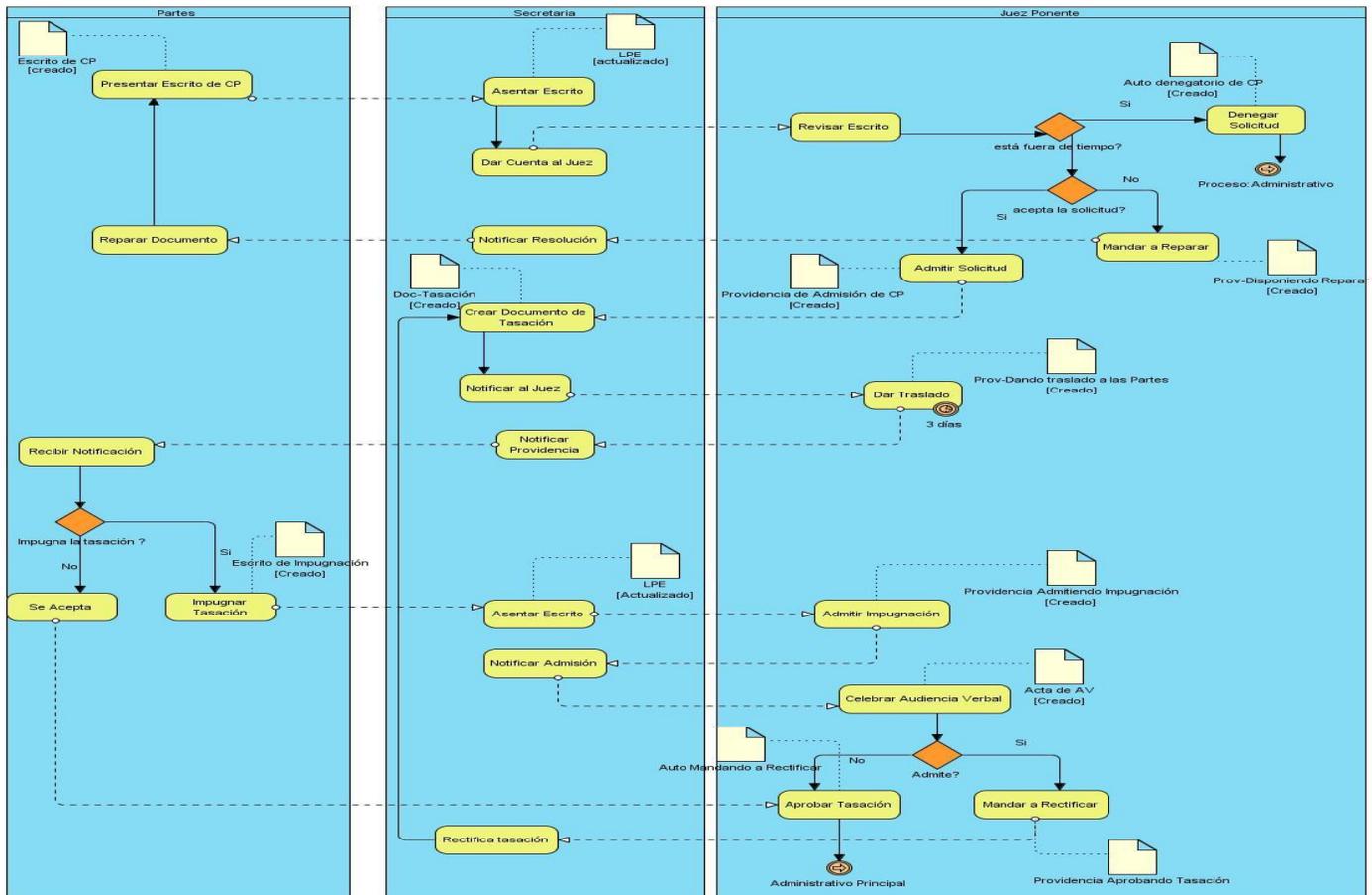
Actividades Principales

1. Presentar Escrito de costas procesales	La parte interesada presenta ante el tribunal un escrito solicitando el pago de las costas procesales, un Escrito de solicitud de costas procesales.
2. Asentar Escrito de costas procesales	La secretaria recibe el Escrito de costas procesales y lo asienta en el LPE, seguidamente se une el escrito al expediente.
3. Dar Cuenta	La secretaria da cuenta al juez ponente que se ha solicitado el pago de las costas procesales.
4. Revisar escrito	El juez ponente revisa el escrito y emite una resolución respecto a lo solicitado.
5. Denegar solicitud	Si el escrito es presentado fuera de tiempo, por ejemplo antes de que se dicte la resolución que pone fin al proceso, el juez deniega la solicitud mediante un auto y continua el proceso principal

normalmente, a menos que se presente recurso de súplica contra el mismo, que en dicho caso ocurriría el subproceso de súplica.

6. Mandar a reparar	Si el juez al revisar el escrito descubre que tiene algún problema en su redacción, como por ejemplo que no se encuentren bien desglosados los gastos incurridos emitiría una providencia disponiendo el reparo del escrito.
7. Notificar resolución	La secretaria notificaría a las partes la nueva resolución de la siguiente forma: <ol style="list-style-type: none"><li data-bbox="500 709 1411 947">1. La parte que asiste al tribunal en el día que sale la resolución, o al siguiente, se le notifica en este caso sobre la resolución emitida por el Juez Ponente en turno, para lo que la Secretaria de la mesa de notificación le aplica el cuño “Diligencia de Notificación” al dorso de la resolución, y registra la fecha.<li data-bbox="500 961 1411 1245">2. La parte que no asiste al tribunal en ninguno de los dos días indicados en el caso 1, será notificada por la Secretaria de la mesa de notificación a través del “Estado Diario” que publica en la “Tablilla de Anuncios” e igual le aplica el cuño “Diligencia de Notificación” al dorso de la resolución, y adjuntándolo al Expediente quedando este actualizado.
8. Reparar Escrito de costas procesales	La parte repararía los errores encontrados por el juez y presentaría nuevamente el escrito, seguidamente ocurriría el proceso de igual manera a cuando se presentara el escrito de solicitud de tasación de costas inicialmente.
9. Admitir solicitud	Si el juez ponente admite la solicitud emite una providencia de admisión de costas procesales y disponiendo que se dé traslado a las partes y se tase las costas por la secretaria.
10. Crear documento de tasación	La secretaria de la Sala confecciona un documento que contiene la tasación de las costas, calculadas por ella.
11. Notificar al juez	La secretaria notifica al juez que se ha creado el documento de tasación y se lo entrega.
12. Dar traslado	El juez revisa el documento y dicta una providencia dando traslado a

	las partes por tres días para que se informen de la tasación realizada por el tribunal.
13. Notificar providencia	La secretaria notificaría a las partes la nueva resolución por las vías antes explicadas.
14. Impugnar tasación	Si la parte no está de acuerdo esta providencia por la tasación realizada, impugna la tasación por medio de un Escrito de impugnación de costas procesales.
15. Asentar Escrito de impugnación de costas	La secretaria asienta el escrito presentado en el LPE.
16. Admitir impugnación	Se crea una providencia admitiendo la impugnación a la tasación de costas y convocando a las partes para una audiencia verbal.
17. Celebrar audiencia verbal	El día señalado se crea un acto de audiencia verbal en el que se exponen lo referente a las costas procesales y se redacta un Acta de audiencia verbal de impugnación de tasación de costas elaborada por la secretaria.
18. Mandar a rectificar	Si se resuelve en dicho acto judicial que la parte tiene razón, se emite una providencia mandando a rectificar la tasación a la secretaria.
19. Rectifica tasación	La secretaria rectifica la tasación y elabora un nuevo documento de tasación y le flujo de acciones consecutivas a este acto es similar a partir de la actividad 10.
20. Aprobar tasación	Si se resuelve que la tasación realizada por la secretaria es correcta se emite un Auto aprobando la tasación de costas creada por la secretaria y termina el proceso. Esto también procede cuando la parte acepta la tasación hecha por la secretaria la primera vez que fue notificada.



Subproceso de Suspensión

Ficha de Proceso

Proceso:	Suspensión
Responsable:	Juez ponente
Ejecutor (es):	Secretaria, Juez ponente
Descripción:	La suspensión consiste en la petición escrita por la parte demandante de la suspensión del acto administrativo impugnado en el caso, es decir que se suspenda la ejecución de lo dictado por la administración hasta que se termine el caso, para esto la parte debe pagar una fianza al banco en solo si el juez ponente admite la solicitud. El juez dispone que se envíe un oficio al banco informándole del pago de la fianza. La parte procede a pagar la fianza y presenta en el tribunal el slip o comprobante bancario que le devuelve el banco una vez

pagada la fianza para que se una al expediente e inmediatamente el juez ponente dispone que se envíe oficio a la administración informándole que debe suspender el acto.

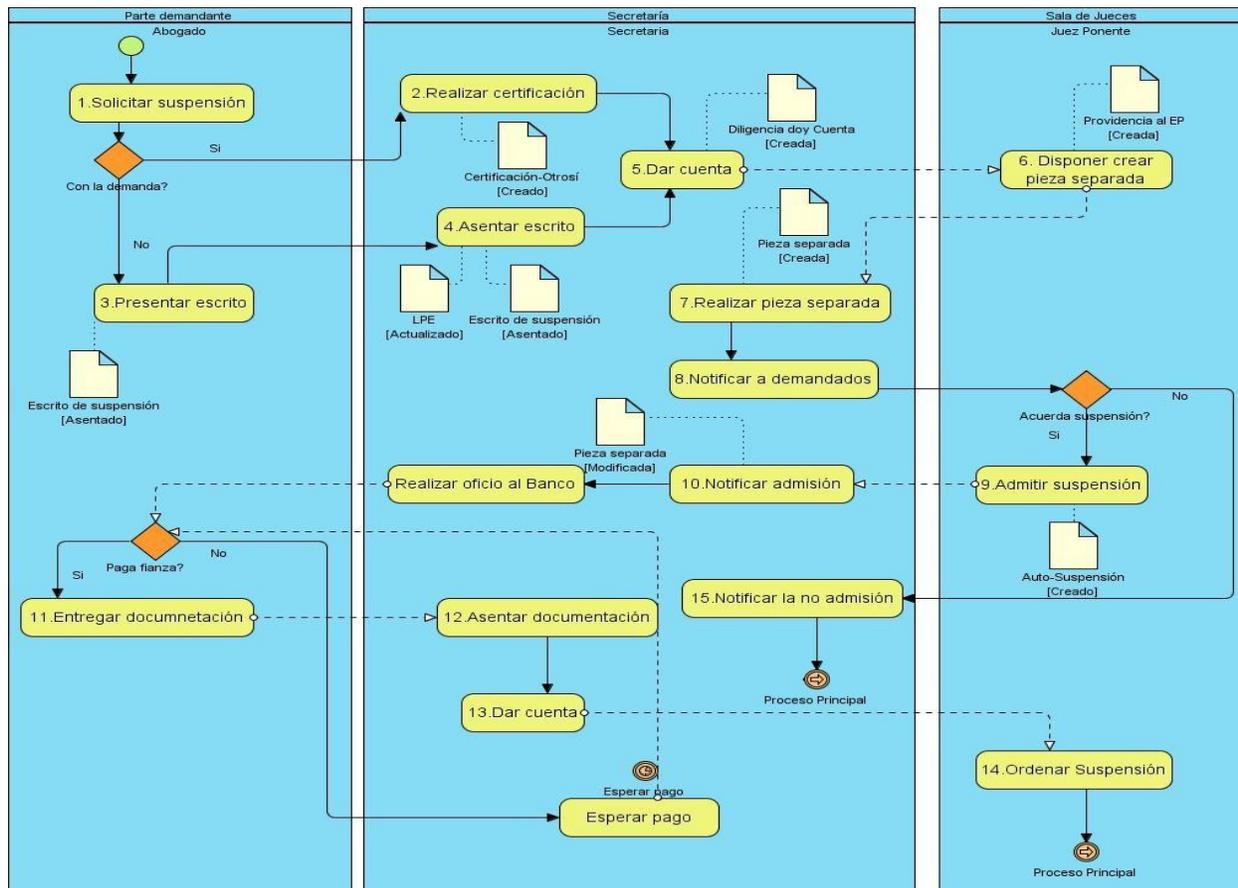
Entradas:	Escrito de demanda Escrito de solicitud de suspensión
Salidas:	Providencia disponiendo la suspensión del acto administrativo impugnado. Oficio a la administración para la suspensión del acto administrativo impugnado.
Procesos asociados:	Administrativo, Archivo
Reglas del Negocio:	RN.1; RN.2; RN.3; RN.14
Clasificación:	Secundario
Flujo básico:	

Actividades Principales

1.Solicitar suspensión	Es cuando la parte actora solicita al tribunal la suspensión, esta puede pedirse al inicio del proceso en el escrito de demanda presentado en cuyo caso la secretaria realizaría una certificación o en cualquier otro momento en un documento independiente que sería asentado por la secretaria.
2.Realizar certificación	Es cuando la parte actora pide la suspensión en la demanda, la secretaria realiza una certificación del "Otro sí" donde se realizó la petición.
3.Presentar escrito	Es cuando la parte actora pide la suspensión después de haber presentado la demanda, esto lo hace presentando un escrito en el tribunal.
4.Asentar escrito	La secretaria asienta en el LPE el escrito de suspensión presentado
6.Dictar providencia	El juez ponente dicta providencia admitiendo el escrito y mandando a realizar la pieza separada.
7.Realizar pieza separada	La secretaria realiza un expediente que es unido a cuerda floja al expediente principal del proceso y pone dentro la petición de

suspensión.

8. Notificar a demandados	La secretaria les informará a los demandados de la petición presentada por la parte actora y se oirá, por el plazo común de tres días, al representante de la Administración y a los coadyuvantes que hubieren comparecido, luego el Juez ponente decidirá la aceptación o no de la suspensión.
9. Admitir suspensión	El juez decide, mediante auto, que es firmado por todos los jueces que intervienen en el proceso, admitir la suspensión del acto objeto del proceso, fijando en dicho auto el monto de la fianza a pagar por la parte actora.
10. Notificar admisión	Es cuando la secretaria asienta el auto en el LNA y actualiza la Pieza separada. Le notifica a la parte actora la admisión de la suspensión y del monto de la fianza a pagar.
11. Realizar Oficio al banco	La secretaria elabora y envía un oficio al banco disponiendo el pago de la fianza por parte de los demandados.
12. Entregar documentación.	Si la parte paga la fianza, luego de pagada en la agencia bancaria la fianza fijada por el tribunal, presenta el Slip bancario lo cual prueba que dicho pago se efectuó. Seguidamente la secretaria une este slip al expediente. Si no se presenta este slip el proceso administrativo transcurre normalmente sin suspender el acto impugnado.
13. Dar cuenta	La secretaria crea una diligencia doy cuenta que será enviada al juez ponente informándole que ha sido pagada la fianza.
14. Ordenar suspensión	El Juez ponente, luego de recibir la notificación del pago de la fianza, ordenará el inmediato cumplimiento de la suspensión a la autoridad administrativa que corresponda mediante un auto.
15. Notificar la no admisión	Si el juez decide no admitir la solicitud de suspensión lo hace mediante un auto. Luego de esto la secretaria le notifica a la parte actora la decisión y si la parte no está de acuerdo con la decisión puede suplicarla si no el proceso administrativo continúa normalmente.



2.6 Requisitos de Software

Los requisitos de software deben ser especificados por escrito, como todo contrato o acuerdo entre dos partes, deben ser posibles de probar o verificar debido a que si un requisito no se puede comprobar no hay forma de saber si se cumplió con él o no, deben estar descritos como una característica del sistema a entregar, esto es, lo que el sistema debe hacer y no cómo debe hacerlo. Es importante destacar que deben estar redactados de la manera más abstracta y concisa posible para evitar malas interpretaciones. (UCI, 2007-2008)

Para identificar los requisitos funcionales que los clientes deseaban que tuviera el sistema, específicamente para los subprocesos asociados al procedimiento Administrativo se emplearon diferentes técnicas. En las primeras reuniones con los clientes se llevó a cabo la Tormenta de Ideas para obtener una vista general de estos subprocesos y se identificaron las necesidades fundamentales que el sistema debía satisfacer, estas primeras ideas se fueron concretando

mediante las Entrevistas en las cuales se detallaron cada una de las funcionalidades que el sistema debía cumplir, sirvieron como técnica de apoyo la Observación y Arqueología de Documentos.

2.6.1 Requisitos

Durante el desarrollo de las tareas de elicitación, análisis y especificación de requisitos se identificaron cuarenta y ocho requisitos funcionales. Los treinta y cuatro requisitos no funcionales especificados fueron identificados fundamentalmente por el equipo de arquitectura del proyecto. Para consultar información detallada de los requisitos obtenidos ver documento Modelo de Especificación de requisitos de software y el Documento de requisitos no funcionales del software, adjuntos a la investigación. A continuación se muestra el listado de los requisitos funcionales más significativos así como algunos requisitos no funcionales identificados agrupados por categorías.

Principales requisitos funcionales

Subproceso Costas Procesales

RF.2 Presentar escrito de solicitud de costas procesales

RF.3 Crear providencia admitiendo solicitud de costas procesales.

RF.6 Crear documento de tasación de las costas.

RF.8 Presentar escrito de impugnación de costas.

RF.11 Crear auto resolviendo costas procesales.

Subproceso de Suspensión

RF.14 Presentar escrito de solicitud suspensión del acto administrativo.

RF.16 Crear providencia admitiendo escrito y dando traslado de la solicitud de suspensión del acto administrativo.

RF.18 Crear providencia admitiendo la suspensión del acto administrativo.

RF.19 Crear providencia teniendo por pagada la fianza de la suspensión y disponiendo librar oficio a la administración para la suspensión del acto administrativo.

Subproceso de Solicitud de Casación

RF.24 Presentar escrito solicitando recurso de casación.

RF.26 Crear Providencia de admisión de recurso de casación sin de evacuar reparo o luego de evacuar reparo.

RF.28 Crear providencia disponiendo emplazamiento del recurso de casación de los abogados por tablilla.

RF.32 Crear providencia ordenando devolución de los Expedientes gubernativos luego de nombrarse resuelto el recurso de casación en el TSP.

Subproceso de Ejecución

RF.33 Disponer enviar oficio requiriendo a la administración demandada por inejecución de sentencia o auto definitivo.

RF.35 Disponer enviar oficio al superior jerárquico de la administración demandada por inejecución de sentencia o auto definitivo.

RF.37 Disponer dar cuenta a la fiscalía por medio de un oficio sobre la inejecución de sentencia o auto definitivo por la administración demandada.

Subproceso de Archivo

RF.40 Crear providencia disponiendo la devolución de los expedientes gubernativos y expedientes como medio de prueba traído al proceso.

RF.42 Crear providencia disponiendo librar oficio al banco para confiscar fianza por suspensión.

RF.43 Crear providencia disponiendo librar oficio al banco para devolución de la fianza por suspensión.

Algunos requisitos no funcionales

- Facilidad de uso o Usabilidad

RNF.01. 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.02. Existirán servidores locales con capacidad necesaria para el procesamiento de las solicitudes del conjunto de aplicaciones de las diferentes oficinas.

➤ Fiabilidad

RNF.07. El sistema estará disponible 24 horas al día, 7 días a la semana.

RNF.08. Disponibilidad de los casos asignados desde cualquier parte del país.

➤ Eficiencia

RNF.13. Tiempo de respuesta promedio de las peticiones que se realizan al servidor no deberá ser mayor de 3 segundos.

RNF.14. El número de clientes o transacciones que el sistema puede alojar es de 2000.

➤ Seguridad

RNF.15. Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

RNF.16. El sistema debe mantener en todo momento la seguridad de la información asegurando la autenticidad de la misma.

➤ Soporte

RNF.21. Soporte para grandes volúmenes de datos y velocidad de procesamiento.

RNF.22. Tiempo de respuesta rápido en accesos concurrentes.

➤ Restricciones de diseño

RNF.23. El lenguaje de programación es PHP.V. 5.2 y JavaScript.

RNF.24. El framework de desarrollo es Sauxe 1.0.

➤ Interfaz

RNF.30. El sistema tiene que ofrecer una interfaz amigable, fácil de operar.

RNF.31. El sistema tiene que mantener la línea de diseño establecida para la institución que mantiene la uniformidad y representatividad de la misma.

➤ Legales y de Derecho de Autor

RNF.33. El Sistema de Gestión para los Tribunales Populares Cubanos tiene que garantizar el cumplimiento de lo dispuesto en las normas jurídicas, fundamentalmente las que se emitieron con rango de Ley, permitiendo adecuar el sistema de forma fácil, a cambios en dichas normas. Debe igualmente hacer referencia al Registro y depósito legal de la aplicación.

➤ Estándares Aplicables

RNF.34. El estándar utilizado en el proyecto para la especificación de requisitos es el IEEE 830. Los estándares de codificación utilizados en el proyecto están basados en normas seguidas por el código Symphony, específicamente el estándar “UpperCamelCase” para nombre de clases y variables.

2.7 Definición de actores del Sistema

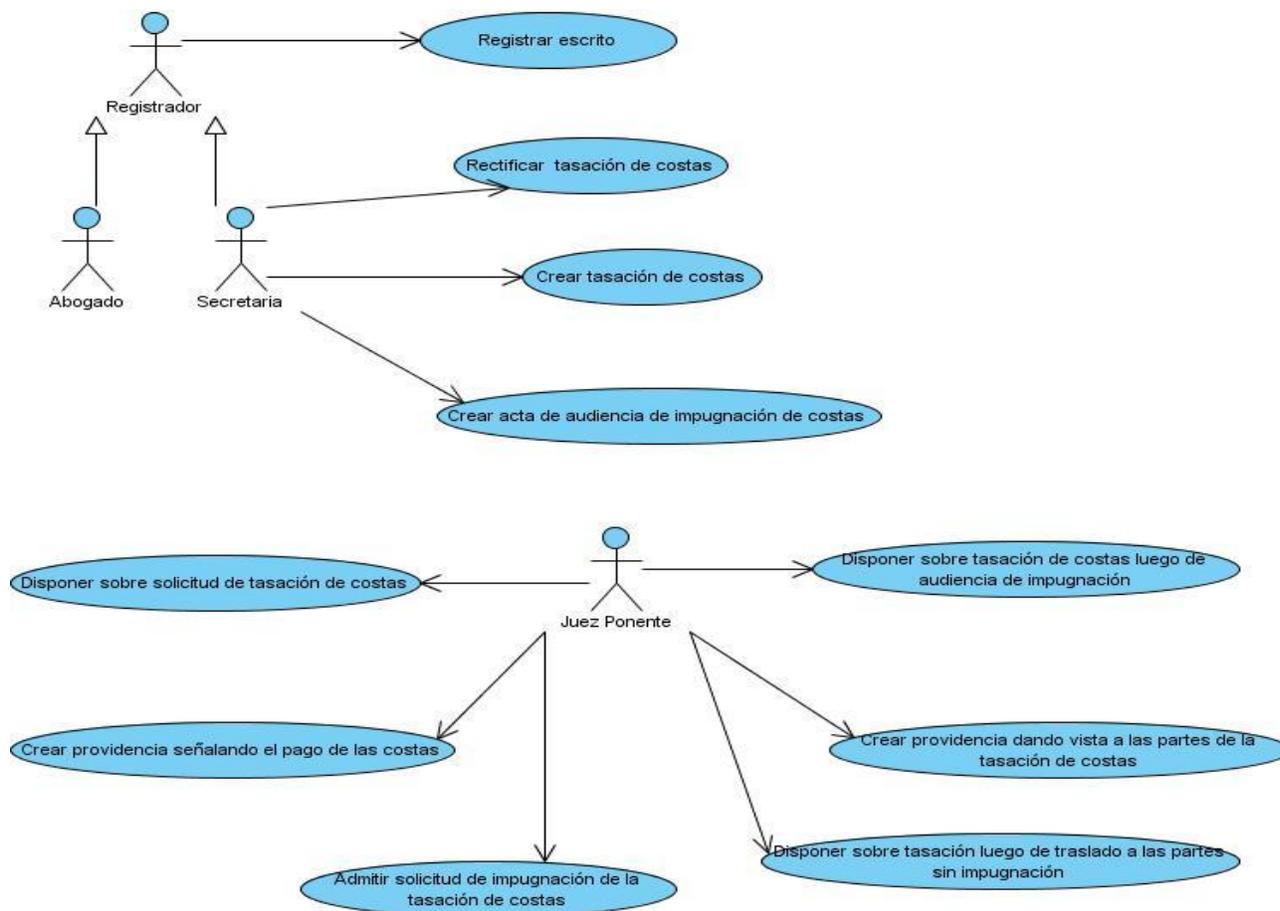
Tabla 3: Descripción de actores del Sistema.

Actores	Descripción
Abogado	Realizará las siguientes acciones: <ul style="list-style-type: none">• Notificarse de las resoluciones dictadas por el juez ponente
Secretaria	Podrá realizar las siguientes acciones en el sistema: <ul style="list-style-type: none">• Crear oficios• Crear actas• Realizar y rectificar la tasación de costas procesales.• Crear Citaciones y emplazamientos.

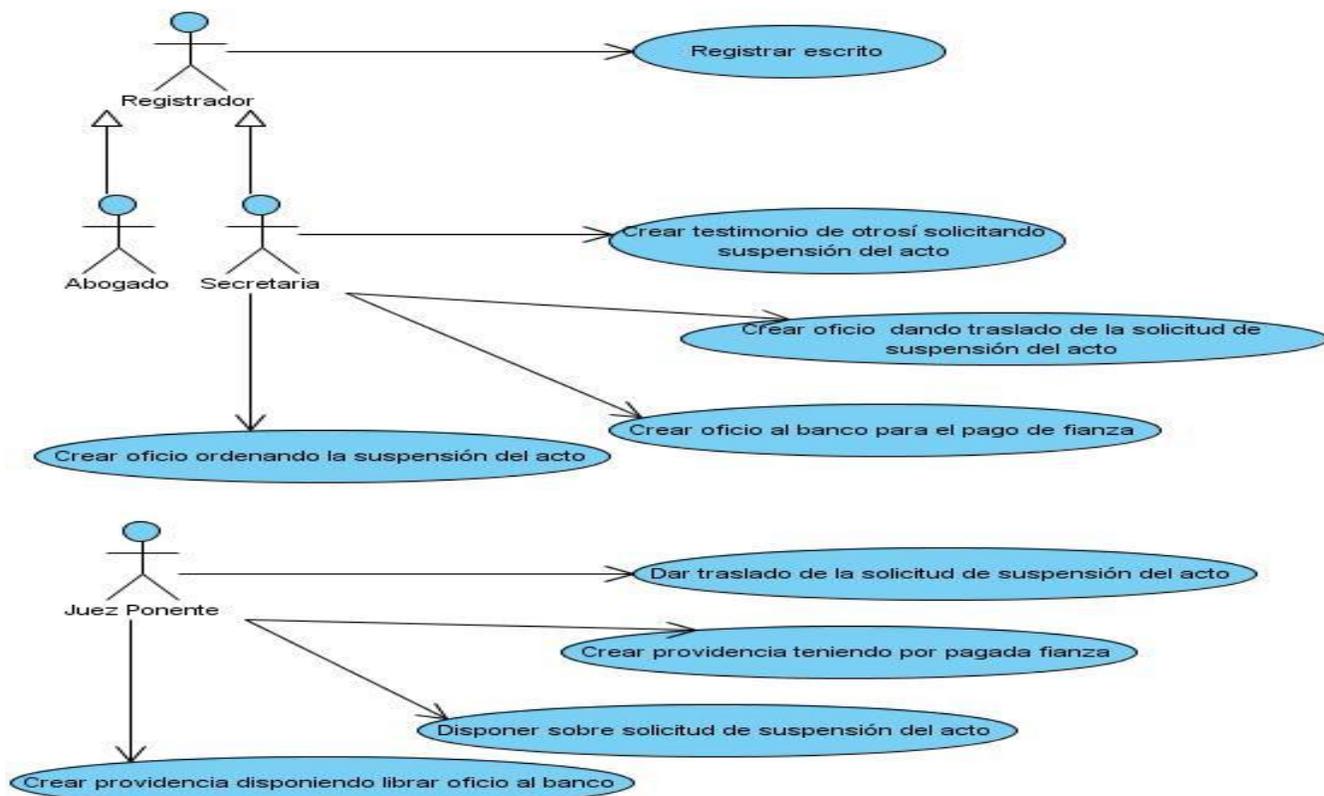
Juez ponente	<p>Es la persona encargada de revisar todo escrito presentado en el tribunal. Realiza las siguientes acciones:</p> <ul style="list-style-type: none">• Crear las providencias admitiendo las solicitudes.• Crear providencias disponiendo librar oficios.• Crear providencias disponiendo el reparo de algún escrito.• Crear autos de rechazo de plano de algún escrito presentado.• Crear auto de inadmisión de alguna solicitud presentada.
Registrador	<p>Es una generalización de los actores Secretaria y Abogado. Podrá realizar las siguientes acciones en el sistema:</p> <ul style="list-style-type: none">• Registrar datos de un escrito (Escrito de suspensión, Escrito de solicitud de casación y Escrito de costas procesales).

2.8 Diagrama de Caso de Uso del Sistema

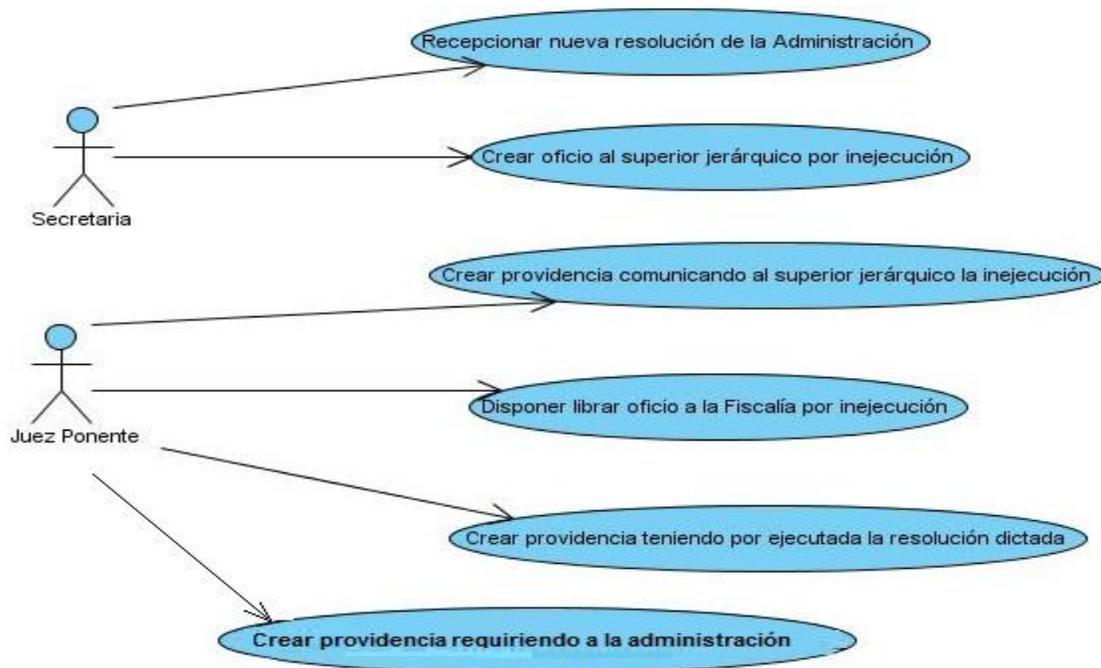
Los cincuenta y uno requisitos funcionales obtenidos fueron agrupados en treinta y cuatro casos de uso, inicializados por cuatro actores fundamentales. Para la realización del Diagrama de Casos de Uso del Sistema se tuvieron en cuenta los patrones de casos de uso existentes, aplicándose en este el patrón Actores múltiples.



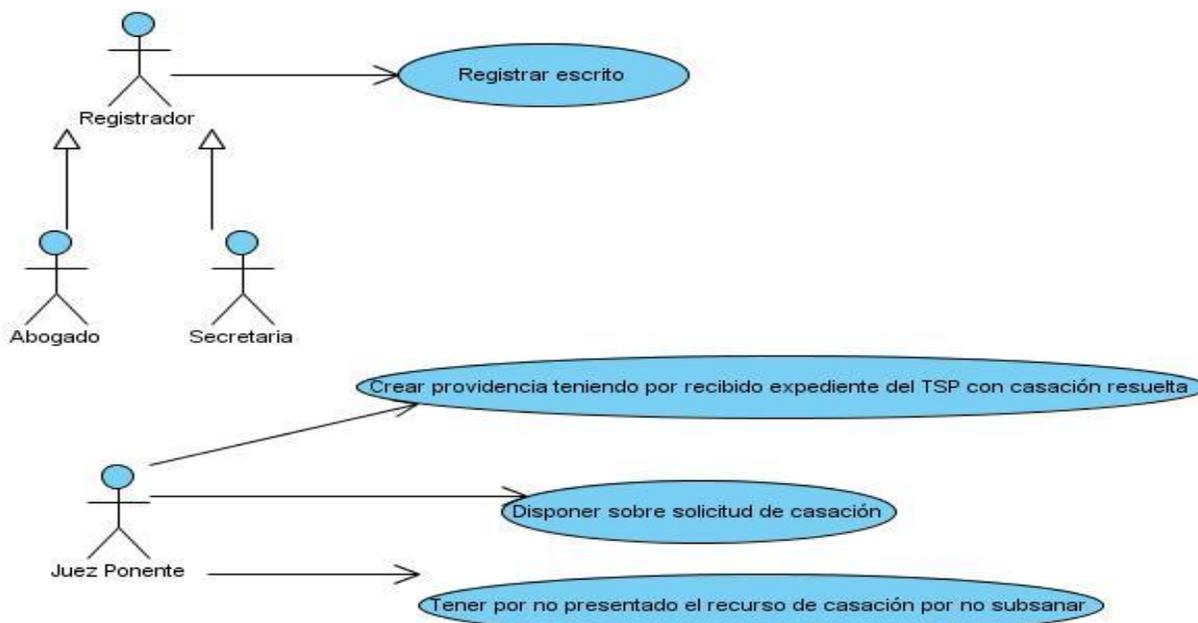
1. Diagrama de casos de uso del subproceso Costas procesales



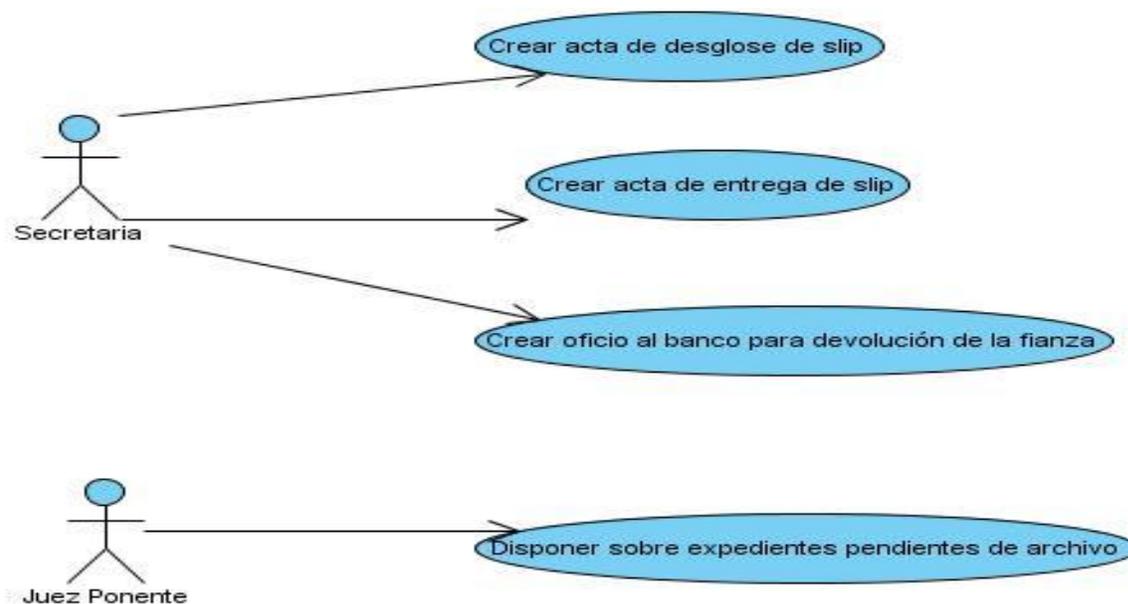
2. Diagrama de casos de uso del subproceso Suspensión



3. Diagrama de casos de uso del subproceso Ejecución



4. Diagrama de casos de uso del subproceso Solicitud de casación



5. Diagrama de casos de uso del subproceso de Archivo

2.9 Realización de los Casos de Uso del Sistema

Todos los casos de uso identificados fueron descritos y graficados mediante prototipos no funcionales. A continuación se muestra la descripción de uno de los Casos de Uso del Sistema con los prototipos no funcionales correspondientes. Las restantes descripciones se pueden encontrar en el documento Modelo de Sistema, que se adjunta a la investigación.

Descripción de caso de uso: Disponer sobre solicitud de suspensión del acto administrativo.

Caso de Uso:	Disponer sobre solicitud de suspensión del acto administrativo.	
Actores:	Juez ponente.	
Resumen:	El caso de uso inicia cuando el actor necesita disponer sobre solicitud de suspensión del acto administrativo. Consiste en que el actor puede admitir o no la solicitud de suspensión del acto administrativo, el sistema genera la resolución correspondiente, terminando así el caso de uso.	
Precondiciones:	<ul style="list-style-type: none"> • Que haya vencido el término en que se le dio traslado a la administración (3 días) habiéndose recibido o no una respuesta por parte de la administración. • El expediente se muestra en el trámite: Disponer sobre solicitud de Suspensión del acto administrativo. 	
Referencias	RF. 19, RF.20 y RF.21	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	

<p>1-El caso de uso comienza cuando selecciona el expediente con el que necesita trabajar.</p>	<p>2-Muestra una interfaz con los datos principales del expediente seleccionado así como la posibilidad de ver los siguientes escritos a través de un vínculo:</p> <p>Escrito presentado.</p> <ul style="list-style-type: none">• Escrito solicitando suspensión del acto administrativo(sólo si se presentó) o testimonio del Otro sí solicitando suspensión del acto administrativo(sólo si se pidió en la demanda)• Escrito de la administración contra la suspensión(sólo si se presentó) y las opciones:• Admitir la solicitud de suspensión del acto administrativo.• Rechazar la solicitud de suspensión del acto administrativo. <p>Además de:</p> <ul style="list-style-type: none">• Pasar a definitivo• Vista previa• Cancelar
--	--

3-Selecciona la opción Rechazar la solicitud de suspensión del acto.

Si selecciona la opción Admitir la solicitud de suspensión del acto administrativo, ver sección 1.

4-Selecciona la Pasar a definitivo.

Si selecciona la opción Vista previa invoca caso de uso Vista previa.

5- Valida que los datos introducidos son correctos y que los campos obligatorios no están vacíos y/o que haya sido elegida una disposición.

6-Crea la resolución guardando definitivamente los datos.

7- Notifica a las partes de la resolución, terminando así el caso de uso.

Prototipo de Interfaz

The screenshot displays the 'Tribunales Populares Cubanos' web application. The interface includes a navigation menu with options like 'Materias', 'Buscar', 'Reportes', 'Calendario', and 'Administración y Gobierno'. The current page is titled 'Inicio > Administrativo > Disponer sobre solicitud de suspensión del acto administrativo'. It shows a case summary with details such as 'Nro de Expediente: 20100714125', 'Tipo de Proceso: Administrativo', and 'Abogado(s): Anabel Gómez Albear'. A section titled 'Escrito presentado.' contains three links: 'Escrito solicitando suspensión del acto administrativo', 'Otro sí solicitando suspensión del acto administrativo', and 'Escrito de la administración contra la suspensión'. Below this, there are two radio button options: 'Admitir la solicitud de suspensión del acto administrativo' (marked with a yellow '1') and 'Rechazar la solicitud de suspensión del acto administrativo'. At the bottom, there are three buttons: 'Vista Previa' (marked with a yellow '2'), 'Pasar a Definitivo' (marked with a yellow '3'), and 'Cancelar'. The footer of the application reads 'COPYRIGHT © 2010, Tribunales Populares Cubanos'.

Tribunales Populares Cubanos

Materias | Buscar | Reportes | Calendario | Administración y Gobierno

Inicio > Administrativo > Disponer sobre solicitud de suspensión del acto administrativo

Bienvenido "Juez Ponente" 14 de Julio de 2010

- Expedientes en curso
 - Pedntes de Admisión
 - Pedntes de Resulta
 - Pedntes de Gubernativos
 - Solicitan Completamiento
 - Pedntes de Completamiento
 - Pedntes de Contestación
 - Pedntes de EPP
 - Pedntes de Admisión de P
 - En Prueba
 - Pedntes de EPV
 - Conclusos para sentencia
 - Pednte de Firmeza
 - Firmes
 - Solicitan Tasación de Cost.
 - Solicitan Suspensión
 - Solicitan desistimiento
 - Solicitan delegación en TA
 - Aclarar sent o auto de oficio
 - Disponer sobre conclusos
 - Citar comparecencia art.42

Nro de Expediente: 20100714125
Tipo de Proceso: Administrativo
Abogado(s): Anabel Gómez Albear

Escrito presentado.

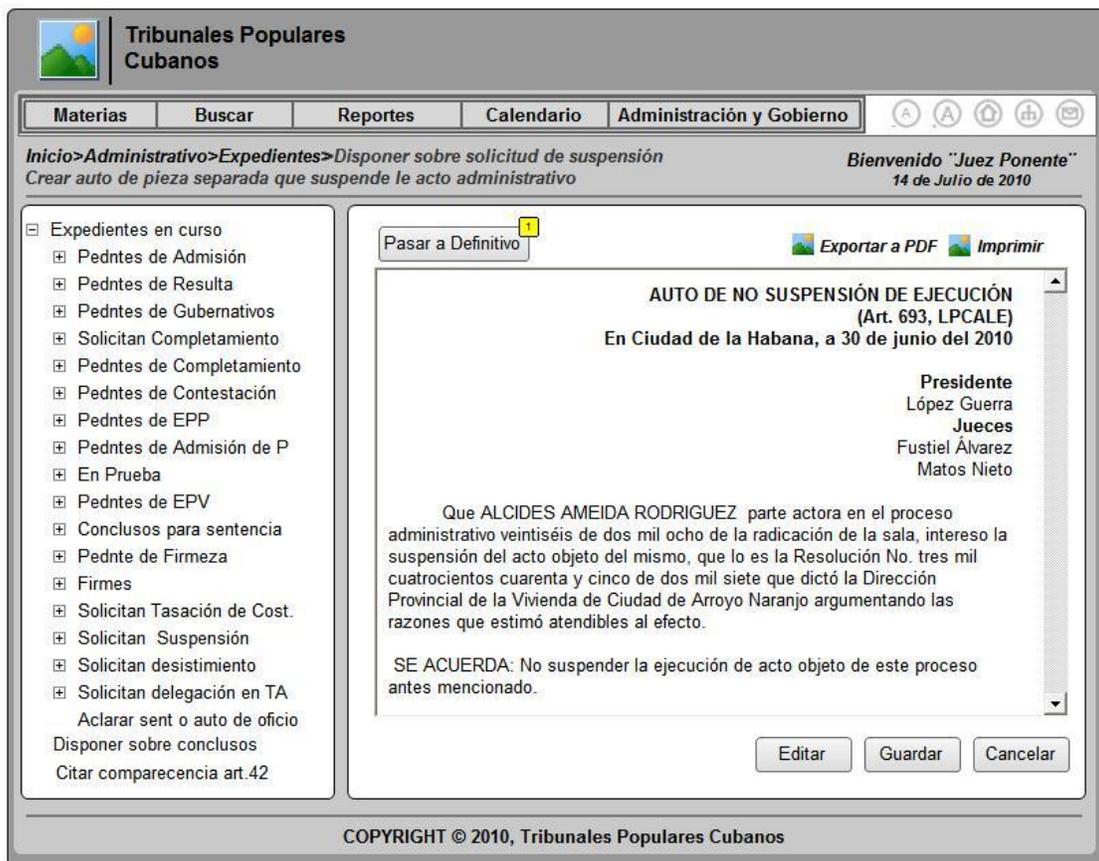
- [Escrito solicitando suspensión del acto administrativo](#)
- [Solicitud de suspensión con la demanda](#)
- [Escrito de la administración contra la suspensión](#)

Admitir la solicitud de suspensión del acto administrativo 1

Rechazar la solicitud de suspensión del acto administrativo

Vista Previa 2 | Pasará a Definitivo 3 | Cancelar

COPYRIGHT © 2010, Tribunales Populares Cubanos



Sección 1 “Admitir la solicitud de suspensión del acto administrativo”

Acción del Actor	Respuesta del Sistema
	<p>1- Muestra una interfaz donde se pueden llenar las siguientes opciones:</p> <ul style="list-style-type: none"> • Auto admitiendo la suspensión de la ejecución <ul style="list-style-type: none"> ○ Cantidad a pagar por fianza (obligatorio) <p>Y las opciones:</p> <ul style="list-style-type: none"> • Vista previa • Pasar a definitivo • Cancelar

2-Introduce los datos necesarios, continúa el paso 4 del flujo normal de eventos.

Flujos Alternos

Flujo Alterno 1 “Insertar datos incorrectos y/o campos vacíos”

Acción del Actor

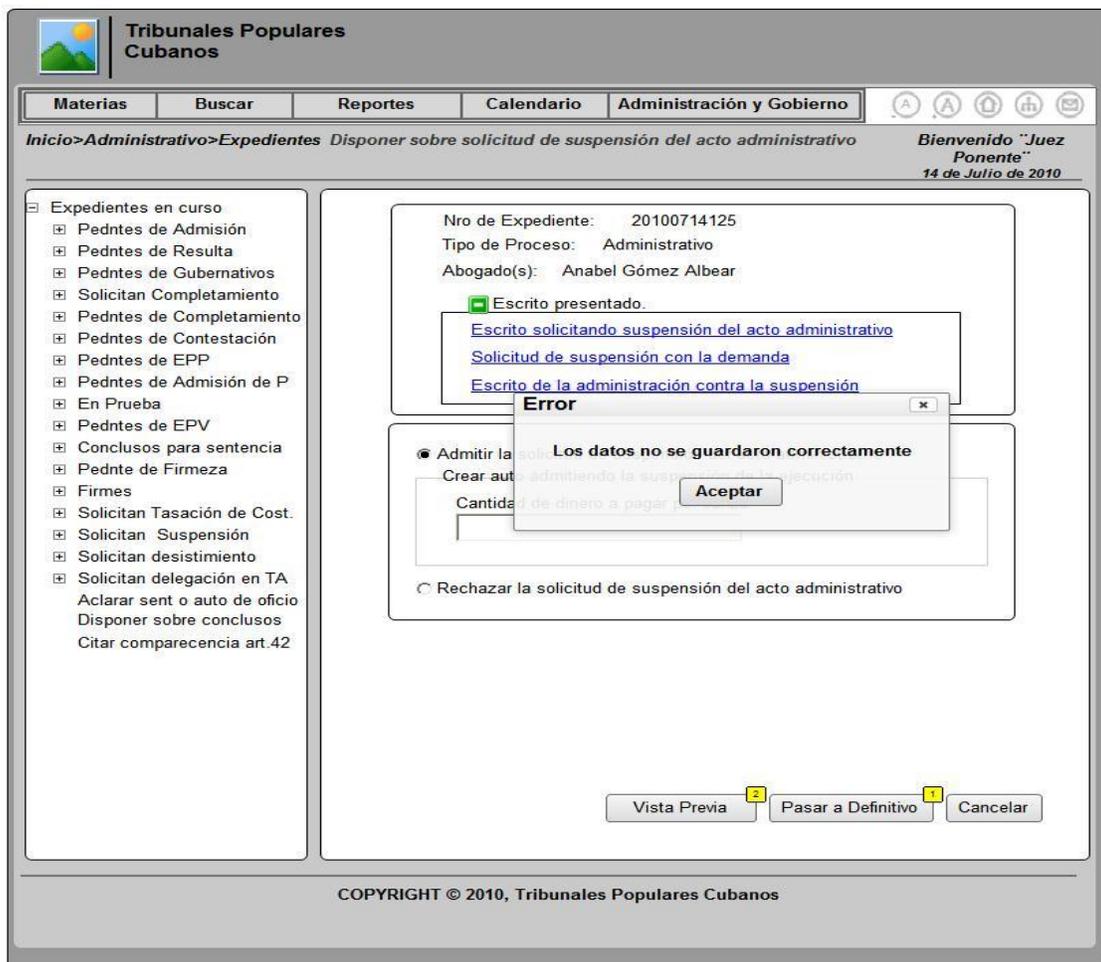
Respuesta del Sistema

5.1- Señala el campo que se dejó vacío o con error y muestra un mensaje indicando que existen campos vacíos y/o con error.

Flujo Alterno 2 “Cancelar”

Acción del Actor	Respuesta del Sistema
	4.1- El sistema cierra la interfaz y termina el caso de uso.

Prototipo de Interfaz



Poscondiciones

- El expediente pasa al nuevo estado que le corresponde.
- Se crea el auto de disponiendo la suspensión de la ejecución del acto administrativo o el Auto no suspensión de ejecución en dependencia

de la acción desarrollada por el Juez ponente.

Conclusiones

El Modelo del Negocio realizado permitió entender la estructura y la dinámica de los subprocesos asociados al procedimiento Administrativo que se realizan en Tribunales Populares Cubanos, en los que se va a implantar la solución informática a crear. Constituyendo una entrada fundamental para el siguiente flujo de trabajo propuesto por RUP: Levantamiento de Requisitos.

Los actores y trabajadores del negocio identificados sirven de base para la definición posterior de los usuarios que interactuarán con el sistema.

Mediante la realización del Levantamiento de Requisitos y con la estrategia de obtención de requisitos puesta en práctica, se consiguió identificar las funcionalidades que el módulo Administrativo debe brindar y las restricciones sobre las que va a operar, siendo fieles a las necesidades del cliente y estas funcionalidades identificadas se plasman en el modelado del sistema, donde se describen los casos de uso del sistema identificados y a desarrollar.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

En este capítulo se aplica la validación mediante prototipos no funcionales y métricas para medir la calidad de la especificación de los requisitos identificados. Se aplica también el Modelo de métricas orientadas a objeto aplicadas al DCUS para medir la funcionalidad del diagrama de casos de uso del sistema realizado, todo esto permitirá medir de cierta forma la calidad del análisis de los subprocesos correspondientes al módulo Administrativo de la solución informática desarrollada en el proyecto TPC.

3.1 Validación de requisitos mediante prototipos no funcionales

El método validación mediante prototipos no funcionales consiste en construir una maqueta del futuro sistema software a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud. Los prototipos a utilizar para la validación en el presente trabajo son conocidos como storyboards o guiones gráficos que permiten mostrar interfaces y la secuencia de acciones, o escenarios, que se deben realizar con el programa. Para que la evaluación del prototipo sea lo más efectiva posible, deben seleccionarse adecuadamente los usuarios que participarán en la evaluación.

Para la presente validación se crearon prototipos del tipo storyboards haciendo uso de la poderosa herramienta Axure RP.Pro, donde se generaron páginas html dinámicas que fueron presentadas primeramente a varios especialistas de la materia Administrativa de forma aislada y luego a un conjunto de ellos en un Taller realizado en el Tribunal Supremo Popular, alcanzándose resultados satisfactorios y un alto grado de aceptación por parte de los especialistas. Manifestándose estos satisfechos con los requisitos identificados en la presente etapa de análisis de los subprocesos asociados al procedimiento Administrativo. Todo esto se ve reflejado en una carta de aceptación emitida por el cliente.

3.2 Métricas de la calidad de la Especificación de requisitos

Los requisitos identificados en los subprocesos correspondientes al módulo Administrativo fueron comprobados para determinar su especificidad (ausencia de ambigüedad) a través de la métrica para la calidad de especificación de los requisitos de software, basada en la consistencia de la interpretación de los revisores para cada uno de los requisitos. Para ello el

documento de especificación de requisitos fue revisado por 5 personas, un juez especialista en la materia administrativa, un desarrollador y tres analistas con experiencia en el rol.

Para llevar a cabo este proceso primeramente se determina el total de requisitos mediante la suma de los requisitos no funcionales y los requisitos funcionales identificados, por lo que se define que: la denominación **NR** es el número de requisitos identificados en la investigación, **NF** es el número de requisitos funcionales y **NNF** es el número de requisitos no funcionales.

$$NR = NF + NNF$$

Luego se procede a medir la especificidad que no es más que el resultado de la división entre el total de requisitos con igual interpretación por parte de los revisores y la cantidad total de requisitos, para ello se define la siguiente fórmula:

$$Q1 = Nu1 / NR$$

Donde **Nu1** es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas y **Q1** es la consistencia de la interpretación de los revisores.

El valor óptimo de **Q1** es 1, significa la ausencia de ambigüedad en los requisitos. Por tanto mientras más cercano esté el valor de **Q1** a 1 mayor será la consistencia de la especificación de los requisitos.

Un resumen de los resultados obtenidos se muestra a continuación:

Atributo de calidad	Tipo de requisito	Interpretaciones	
		Iguales	Desiguales
Especificidad	Funcionales	45	3
	No funcionales	34	0
	TOTAL	79	3

$$NF = 48 \quad NNF = 23 \quad NR = NF + NNF = 48 + 34 = 82$$

$$Q1 = Nu1 / NR = 79 / 82 = 0.963$$

Se obtuvo **Q1** con un valor de 0.963 éste resultado demuestra que el grado de ambigüedad en la especificación de los requisitos de software del módulo Administrativo fue muy bajo y por consiguiente hubo calidad en la especificación.

3.3 Modelo de métricas orientadas a objeto aplicadas al DCUS.

Para medir la funcionalidad del DCUS se le aplicará al mismo un modelo de métricas, donde se tendrán en cuenta cuatro atributos: completitud, consistencia, correctitud y complejidad, quienes por su parte son medidos por varios factores. Cada uno de estos factores tendrá asociada una o más métricas, que establecen una medida cuantitativa del grado en que los factores indiquen una mala calidad.

- ✓ 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.

A continuación se muestran los resultados obtenidos al evaluar los factores de la calidad correspondientes al documento de descripción de casos de uso del sistema y el diagrama de casos de uso del sistema.

Tabla 4: Factores por atributo.

Factores de Completitud	Métricas asociadas	Evaluación para los subprocesos del subsistema Administrativo
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 Umbral: < 10% Acciones sugerida: Revisar el alcance del sistema e Involucrar tipos de usuarios	Total de roles relevantes: 4 Número de roles relevantes omitidos: 0 Representa: 0%

Análisis de los subprocesos asociados al subsistema Administrativo del Proyecto Tribunales Populares Cubanos.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS

	representativos de cada una de las áreas funcionales	
Factor 2. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?	<p>Métrica 2: Número de requisitos omitidos por caso de uso</p> <p>Umbral < 10%</p>	<p>Total de requisitos: 49</p> <p>Número de requisitos omitidos por caso de uso: 0</p> <p>Representa: 0%</p>
	<p>Métrica 3: número de casos de uso que tienen requisitos omitidos</p> <p>Umbral < 10%</p> <p>Acción sugerida: revisar la lista de requisitos para determinar cuáles serán apoyados por cada caso de uso</p>	<p>Total de casos de uso: 29</p> <p>Número de casos de uso que tienen requisitos omitidos : 0</p> <p>Representa: 0%</p>
Factor 3. ¿Se describen las condiciones de excepción relevantes que debe contemplar cada flujo de eventos?	<p>Métrica 4: número de casos de uso que no describen condiciones de excepción relevante.</p> <p>Umbral < 20%</p> <p>Acción sugerida: revisar las excepciones presentadas en el flujo de eventos que producen un mensaje de error al usuario.</p>	<p>Total de casos de Uso: 29</p> <p>Número de casos de uso que no describen condiciones de excepción relevantes: 1</p> <p>Representa: 3,45%</p>

Factores de Consistencia	Métricas asociadas	Evaluación para los subprocesos del subsistema Administrativo
Factor 4. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario?	Métrica 5: número de casos de uso que tienen un nombre incorrecto Umbral < 20% Acción sugerida: modifique el nombre del caso de uso de tal manera que signifique una acción desde el punto de vista del usuario	Total de casos de Uso:29 Número de casos de uso que tienen un nombre incorrecto: 0 Representa: 0%
Factor 5. ¿Representa el caso de uso una interacción observable por un actor?	Métrica 6: número de casos de uso que no representan una interacción observable por un actor Umbral< 5% Acción sugerida: elimine el caso de uso e incorpore su funcionalidad como una responsabilidad del sistema dentro de otro caso de uso	Total de casos de Uso: 29 Número de casos de uso que no representan una interacción observable por un actor: 0 Representa: 0%
Factor 6. ¿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	Métrica 7: número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema	Total de casos de Uso: 29 Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema: 0

identificable?	Umbral: < 10% Acción sugerida: complete la definición del caso de uso incluyendo la acción fuera del sistema que da inicio al caso de uso o la condición interna que el sistema tiene controlar para dar inicio al caso de uso	Representa: 0%
Factor 7. Si en el caso de uso interviene más de un actor, ¿existe claridad en cuál de ellos es el actor iniciador?	Métrica 8: número de casos de uso con más de un actor, que no describe cuál es el actor iniciador Umbral: < 20% Acción sugerida: revise los puntos de inicio del caso de uso y asigne el actor que inicia la acción	Total de casos de Uso: 29 Número de casos de uso con más de un actor, que no describe cuál es el actor iniciador: 0 Representa: 0%
Factores de Correctitud	Métricas asociadas	Evaluación para los subprocesos del subsistema Administrativo
Factor 8. ¿Representa el caso de uso requisitos comprensibles por el usuario?	Métrica 9: grado en que los requisitos representados por el caso de uso son comprensibles por el usuario	Total de requisitos: 49 Cantidad de requisitos que no son comprensibles por el usuario: 0 Representa: 0%

	<p>Métrica 10: número de casos de uso en que los requisitos representados no son comprensibles por el usuario</p> <p>Umbral: < 5%</p> <p>Acción sugerida: discuta con el usuario la interacción que describe el caso de uso y ajuste dicha descripción de manera que sea comprensible por el usuario.</p>	<p>Total de casos de Uso: 29</p> <p>Número de casos de uso en que los requisitos representados no son comprensibles por el usuario: 0</p> <p>Representa: 0%</p>
Factores de Complejidad	Métricas asociadas	Evaluación para los subprocesos del subsistema Administrativo
<p>Factor 9. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación?</p>	<p>Métrica 11: número de elementos del diagrama que requieren reubicación</p> <p>Umbral: < 30%</p> <p>Acción sugerida: modifique la ubicación de los elementos del diagrama de manera que los elementos relacionados se encuentren lo más cercano posible</p>	<p>Total de casos de Uso: 29</p> <p>Número de elementos del diagrama que requieren reubicación: 1</p> <p>Representa: 3.45%</p>

A continuación se grafican los resultados de las métricas por atributo:

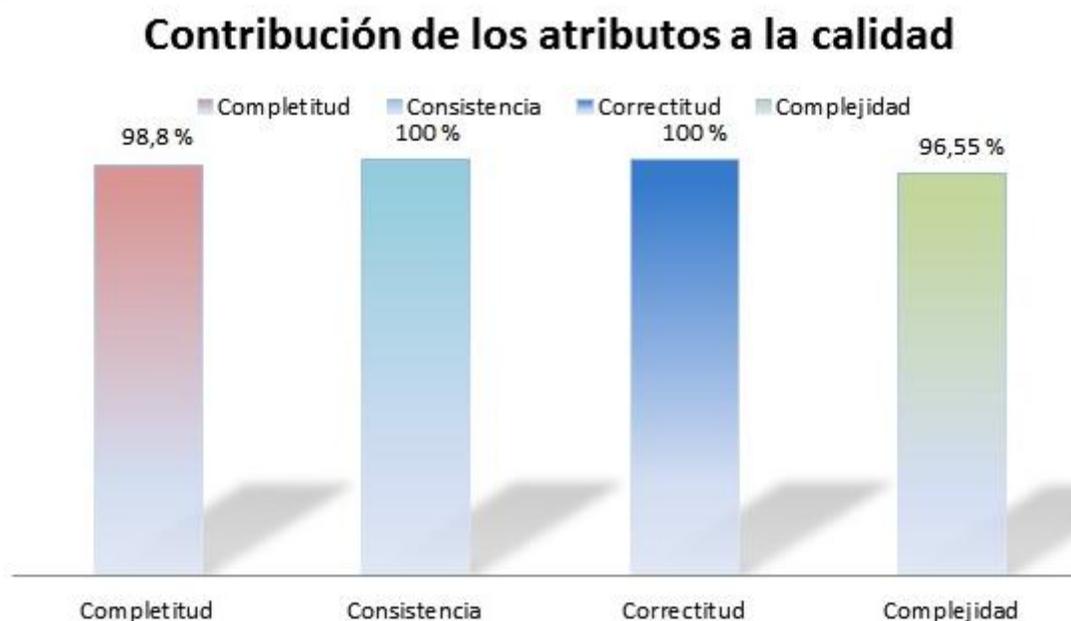


Figura 10: Grado de funcionalidad del diagrama de casos de uso del sistema

Luego de la aplicación del Modelo de métricas orientada a objeto al DCUS, se demostró que el mismo posee la calidad requerida para el futuro diseño del sistema debido a que: cumple con todos los requisitos identificados a través de los casos de uso, los cuales presentan una descripción detallada con todas las acciones del flujo de eventos. En las descripciones de los casos de uso complejos están bien definidas las acciones que corresponden al flujo básico de eventos, a los flujos alternos y a los flujos subordinados. Para lograr una mejor comprensión del DCUS, todos los casos de uso se nombraron con una expresión verbal que describe una funcionalidad relevante para el usuario, representando así una interacción observable para un actor del sistema. Los elementos dentro del diagrama están ubicados lo que facilita su interpretación.

En la evaluación realizada, el DCUS alcanzó una calificación de 98.84% de funcionalidad, la contribución de cada atributo a la calidad total fue: Completitud 98.85%, Consistencia 100%, Correctitud 100% y Complejidad 96.55%.

Conclusiones

Se puede concluir que al aplicar las métricas para la calidad de la especificación de requisitos de software se detectó que el grado de ambigüedad en la especificación de los requisitos es muy bajo. La aplicación de las métricas para la funcionalidad del DCUS por su parte demostró que se representaron de forma funcional los casos de uso del sistema y se puede afirmar que el cliente se encuentra satisfecho con el análisis de los subprocesos asociados al procedimiento Administrativo realizado, debido a los resultados obtenidos en la validación mediante prototipos no funcionales.

CONCLUSIONES GENERALES

Una vez finalizado el desarrollo del presente trabajo se puede afirmar que el estudio realizado sobre metodologías de desarrollo de software, lenguajes de modelado y herramientas CASE, permitió justificar la selección de las utilizadas en el presente trabajo, para el análisis y modelación de los subprocesos asociados al procedimiento Administrativo.

Mediante el modelado del negocio se documentó la representación de forma detallada de los subprocesos asociados al procedimiento Administrativo, asegurando una comprensión común del modelamiento de los mismos por parte de los clientes, usuarios finales y desarrolladores.

Los Requisitos identificados fueron descrito y se agruparon en casos de uso representados en diagramas de casos de uso del sistema, los cuales fueron descritos y prototipados alcanzando un alto nivel de comprensión por parte de los desarrolladores.

La validación mediante los Prototipos no funcionales y la validación de la Especificación de Requisitos y del DCUS mediante métricas demostraron que los artefactos generados presentan la calidad requerida para dar continuidad al ciclo de desarrollo de los subprocesos asociados al procedimiento Administrativo desarrollados en el subsistema Administrativo del proyecto TPC.

RECOMENDACIONES

Terminado el análisis de los subprocesos asociados al procedimiento Administrativo se recomienda continuar con la realización de los flujos de trabajo propuestos por RUP, generando los artefactos correspondientes y realizando las actividades implícitas en esta guía de desarrollo de software a fin de crear una solución informática que sea capaz de resolver los problemas reales para lo cual fue ideada.

BIBLIOGRAFÍA

Acuña, Kareny Brito. 2009. 2009.

Altamirano, Alfonso Valdez. 2008. *Comparativo IDE's.* 2008.

Andrés, Rolando Rodríguez. 2008. <http://www.gestiopolis.com>. <http://www.gestiopolis.com>. [En línea] 2008. [Citado el: 17 de febrero de 2011.] <http://www.gestiopolis.com/administracion-estrategia/lenguajes-notaciones-y-herramientas-en-analisis-de-procesos.htm>.

Argentina, Sparx Systems. 2010. SPARX SYSTEMS. *SPARX SYSTEMS.* [En línea] 2010. [Citado el: 23 de febrero de 2011.] <http://www.sparxsystems.com.ar/>.

Aurora Vizcaíno, Felix Óscar García, Ismael Caballero. INGENIERÍA DEL SOFTWARE I.

Brooks, Frederick P. 1987. *Essence and Accidents of Software Engineering.* Estados Unidos : s.n., 1987.

Business Process Managements Notation. **Durocher, Eric. 2007.** 2007.

Caballero, Aurora Vizcaíno y Felix Óscar García y Ismael. 2009. *INGENIERÍA DEL SOFTWARE I.* 2009.

Estavillo, Juan José Ríos. 1997. *DERECHO E INFORMÁTICA EN MÉXICO. INFORMÁTICA JURÍDICA Y DERECHO DE LA INFORMÁTICA.* Mexico : s.n., 1997.

FUNDIBEQ. 2008. [En línea] 2008. www.fundibeq.org.

Gómez, Julián. 2007. softonic. [En línea] 2007. [Citado el: 23 de febrero de 2011.] <http://axure-rp.softonic.com/>.

Headquarters, Corporate. 2008. www.bizagi.com. www.bizagi.com. [En línea] 2008. [Citado el: 21 de febrero de 2011.] www.bizagi.com/docs/BPMNbyExampleSPA.pdf.

Innova, Grupo de Soluciones. 2008. www.rational.com.ar. www.rational.com.ar. [En línea] 2008. [Citado el: 23 de febrero de 2011.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.

Introduction to BPMN. **White, Stephen A. 2004.** Canada : s.n., 2004.

—. **Wite, Stephen. 2004.** Canada : s.n., 2004.

Ivar Jacobson, Grady Booch, James Rumbaugh. 2000. *El Proceso Unificado de Desarrollo de Software .* s.l. : Addison Wesley, 2000.

- Jiménez, Beatriz Bernárdez. 2003.** *Una Aproximación Empírica a la Verificación de Especificaciones de Requisitos para sistemas de información.* Sevilla : s.n., 2003.
- 2006.** Knowledge Based System. *Knowledge Based System.* [En línea] 2006.
<http://www.idef.com/IDEFO.html>.
- Kruchten, Philip. 2000.** *The Rational Unified Process: An Introduction.* Canadá : s.n., 2000.
- Ladrin, Lucas. 2010.** *tipete.com.* [En línea] 2010.
<http://www.tipete.com/userpost/descargas-gratis/axure-rp-pro-v5602158>.
- Luño, Antonio Enrique Pérez. 2008.** *Legislación Informática.* 2008.
- Muñoz, Coral Calero. 2008.** *MÉTRICAS DEL SOFTWARE:.* Castilla : s.n., 2008.
- Pavón, Eduardo León. 2008.** *Tutorial Visual Paradigm.* La Mancha : s.n., 2008.
- Penadés, Patricio Letelier y Maria Carmen.** *Metodologías ágiles para el desarrollo de software, sXtreme Programming(XP) .* Valencia : s.n.
- Pereiro, Jorge. 2008.** *portalcalidad.* [En línea] 2008. [Citado el: 3 de marzo de 2011.]
http://www.portalcalidad.com/articulos/71-la_satisfaccion_del_cliente_iso_9001.
- Pérez, Matías Sales y Marcelo. 2008.** *El Diagrama de Pareto en la Toma de Decisiones.* 2008.
- Pressman, R. 2005.** *Ingeniería del Software. Un enfoque práctico.* La Habana : Félix Varela, 2005. pág. 601.
- Pressman, Roger S. 2001.** *Ingeniería del Software. Un enfoque práctico.* 2001.
- Revista Intersedes.* **Chaves, Michael Arias. 2007.** Costa Rica : s.n., 2007.
- Riesta, Enma. 1995.** *Informática Jurídica aplicada a la enseñanza de derecho.* Mexico : s.n., 1995.
- Rodríguez., Yaily Morales. 2010.** *Análisis y diseño del subsistema administrativo de los tribunales cubanos.* Ciudad Habana : s.n., 2010.
- Rumbaugh, James. 2000.** *El Proceso Unificado del Desarrollo de software.* 2000.
- Sanchez, María A Mendoza. 2004.** *Metodologías De Desarrollo De Software.* 2004.
- Sommerville. 2005.** *Ingeniería de Software.* s.l. : Editorial Pearson, 2005.
- Souñé, Emilio. 1986.** *Introducción a la Informática Jurídica y el derecho de la informática.* España : s.n., 1986.

Sparx Systems, Group. 2010. Sparx Systems. *Sparx Systems*. [En línea] 2010. [Citado el: 23 de febrero de 2011.] <http://www.sparxsystems.com.ar/>.

Sutherland, Ken Schwaber y Jeff. 1990. *Scrum, una metodología ágil I*. 1990.

Tobalina, Rubén González Blanco Sergio Pérez. *LESE-2 Introducción a Rational Rose*. Barcelona : s.n.

UCI. 2007-2008. *eva.uci.cu. eva.uci.cu*. [En línea] 2007-2008. [Citado el: 1 de marzo de 2011.] <http://eva.uci.cu/mod/resource/view.php?id=8865..>

Villanueva, Isabel. *Elicitación de requisitos en sistemas de gestión orientados a procesos1*. España : s.n.

Visio, Grupo de desarrollo de Microsoft Office. 2007. *Microsoft Office Visio 2007*. 2007.

Wieggers, Karl. 2003. *Software Requirement, Second Edition*. Washington : s.n., 2003.

GLOSARIO

Auto: Resolución judicial fundada durante el proceso que no resuelve el asunto en lo principal, que decide cuestiones secundarias, incidentales, de ejecución o aquellas para las que la ley previene esta forma de resolución.

Acto administrativo: Decisión emitida por algún órgano de la administración del gobierno respecto a algún asunto legal de su competencia que se registra de forma escrita y se da a conocer a los afectados para su ejecución.

Asientos: Una sección correspondiente al libro de presentación de escrito, donde se registran los datos fundamentales de un escrito presentado en la sala o de una sentencia emitida.

Asentar: Proceso de registrar en el libro de presentación de escrito los datos fundamentales a un escrito presentado en la sala.

Casación: Acción de casar o anular. Recurso extraordinario, destinado a la anulación de sentencias de los tribunales inferiores por defectos de forma, infracción de ley o doctrina legal.

Costas Procesales: Gastos en los que incurre la parte por eventos relacionados con el proceso judicial en el que se encuentra involucrado. Por ejemplo los gastos en notificaciones, sellos y pago de asistencia letrada entre otros.

Demanda: Petición o escrito que inicia un proceso ejercitando una o varias acciones ante el tribunal o juzgado que se considere competente.

Demandado: Contra quien va dirigida una demanda. En el orden penal será más conocido como acusado.

Demandante: Quien entabla o inicia una acción judicial en solicitud de un derecho. También se conoce como actor o litigante.

Expediente: Es un legajo que contiene toda la tramitación judicial en un caso concreto que se enumera consecutivamente por año.

Impugnar: Es una acción mediante la cual se pretende corregir actos y resoluciones judiciales, cuando a juicio del recurrente son deficientes, erróneas o ilegales.

Juez Ponente: Magistrado que en los tribunales colegiados, y por turno, examina las actuaciones, dirige las pruebas y presenta un proyecto de sentencia que somete a la aprobación de los restantes magistrados que forman el tribunal.

Libro de presentación de escritos (LPE): Es un libro registro en el que se deja constancia oficial de la presentación de escritos y documentos acompañados.

Libro de Radicación (LR): Es el libro en el que se registra por su número consecutivo los expedientes para su tramitación.

Notificación: Acto en que, con las formalidades legales, se comunica a los interesados una resolución de carácter judicial o administrativo. Medio legal por el cual se da a conocer a las partes o a un tercero el contenido de una resolución judicial.

Parte: Persona que litiga, compareciendo por sí misma o por medio de otras que la representan. Persona interesada en un juicio, compareciendo por sí mismo o por medio de otras que la representan real o presuntivamente.

Plazo: Espacio de tiempo establecido para realizar un determinado acto.

Procedimiento: Conjunto de formalidades o trámites que constituyen los actos jurídicos.

Proveer: Dictar una resolución judicial.

Providencia: Resolución judicial que decide cuestiones de trámite y peticiones secundarias o accidentales. Tienen por objeto la ordenación material del proceso.

Sentencia: Es la resolución final de un expediente, mediante esta el tribunal expone el fallo. Siempre deben estar numeradas ya que siempre ponen fin al proceso.

Suspensión: Es el proceso mediante el cual una administración suspende la ejecución del acto judicial que ha sido impugnado.

Tablilla: Es el medio de publicidad que normalmente produce el efecto de la notificación y se concretan otras diligencias.

Término: Espacio de tiempo establecido para realizar un determinado acto.