

**Universidad de las Ciencias Informáticas**

**Facultad 3**



**Título:**

**Personalización del subproceso Desistimiento para el Sistema de Informatización para la Gestión de los Tribunales Populares Cubanos**

**Trabajo de Diploma para optar por el título de  
Ingeniero Informático**

**Autor(es):** Yudiel Honnier Robert García  
Alejandro Hernández Herrería

**Tutor(es):** Ing. Yoslenys Roque Hernández  
Ing. Reinier Fernández Coello

**Junio, 2018**

***“Si decides que vas hacer solo las cosas que sabes que van a funcionar, dejarás un montón de oportunidades de lado”***

***Jeff Bezos***



*DECLARACIÓN DE AUTORÍA*

Declaramos ser autores de la presente tesis que tiene por título: Personalización del subproceso Desistimiento para el Sistema de Informatización para la Gestión de los Tribunales Populares Cubanos, y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firman la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Autores:**

\_\_\_\_\_  
Yudiel Honnier Robert García

\_\_\_\_\_  
Alejandro Hernández Herrería

**Tutores:**

\_\_\_\_\_  
Ing. Yoslenys Roque Hernández

\_\_\_\_\_  
Ing. Reinier Fernández Coello

## AGRADECIMIENTOS

*A mi padre Raúl, gracias PIPO por apoyarme sin cesar en todo momento y por ser exigente conmigo con el único objetivo de convertirme en un profesional. Por siempre guiarme, por todos tus consejos y por ser más que un padre un amigo.*

*A mi madre Arelis por su sacrificio, por sus consejos y regaños, por todo su amor y cariño.*

*A mi hermanita Arlettys que a pesar de que estamos lejos uno del otro siempre hemos permanecidos unidos. Te quiero mucho Tita.*

*A toda mi familia que siempre me apoyó incondicionalmente, en especial a mi abuela Gricelda, que nunca ha dejado de preocuparse por mí pese a todas sus enfermedades.*

*A mis abuelos Jesús, Mario y Lidia que me miran desde el cielo y sé que hoy están orgullosos de que me convierta en ingeniero.*

*A mis tutores por toda la ayuda brindada y las horas dedicadas a esta tesis, sé que se esforzaron y dieron todo de sí para poder brindarnos sus conocimientos. Gracias Reinier por la paciencia que tuviste con nosotros, sé que estuvimos al punto de volverte loco.*

*A todas las personas del proyecto, principalmente a Yoannys por brindarnos de su tiempo cada vez que teníamos cualquier duda.*

*A todos mis compañeros de estudio durante estos 5 años, gracias por permitirme tener el privilegio de haberlos conocido. A las chikis, al Noe, a Robinson y a Ismael por los días de domingo y chinchirinchin. A mi compañero de cuarto Francis por soportar todas mis pesadeces e inventos. Al Javi por ser un gran amigo y por tener más corazón que tamaño. Al Placeres por los momentos de Freestyle y de descarga. A una persona en especial que hizo inolvidable esta etapa de mi vida en la UCI. Al cerebritito Barby por saber ser un buen amigo y compañero. A Omarito y Yaniel por ser siempre tan carismático.*

*A mi compañero de tesis Alejandro por demostrar que con sacrificio y entrega podemos realizar lo que nos proponíamos en la vida.*

*A* piquete de los Lan Party (El chino, Reicer, Alain, Guille, Supernatural, Unleashed, El migue, Fidel, El Primo, El Punky, Roy, Eduardo y al resto que no menciono por los malos que son. Juntos pasamos noches de chucho inolvidables.

*A* todos los educadores que han contribuido a mi formación como profesional en esta institución.

*A* todo el tribunal por las sugerencias, inconformidades y recomendaciones que me hicieron, permitiendo la creación de un mejor trabajo de diploma.

Sé que me quedan muchos sin mencionar, pero, en fin: Gracias a todas aquellas personas que de una forma u otra formaron parte de mi vida universitaria, gracias a ustedes por su apoyo y por sus saludos, gracias.

*Yudiel*

*A* mis padres que me han apoyado siempre, sin ellos no hubiera llegado a este momento.

*A* Alexis, mi amigo y hermano que me acompañó durante estos cinco años.

*A* mis compañeros de apartamento Alberto y Nelson, doy gracias por haberlos conocidos.

*A* mi compañero de tesis, por ayudarme en la realización de este trabajo y ser mi amigo.

*A* Bati, Alea y el Trimi, mis amigos y compañeros de ocio.

*A* los profesores que me impartieron clases estos 5 años.

*Alejandro*

*DEDICATORIA*

*Quiero dedicar esta tesis a todas las personas que han estado conmigo en las buenas y en las malas, en especial a mis padres por haberme educado y criado con esfuerzo y sacrificio.*

*Yudiel*

*Dedico el presente trabajo a mis padres, por brindarme su amor y estar ahí para mí todos estos años.*

*Alejandro*

*RESUMEN*

En el año 2015 fue desplegada la primera fase del Sistema de Informatización para la Gestión de los Tribunales Populares Cubanos, con el objetivo de informatizar todos los procesos que se llevan a cabo en los tribunales cubanos. El despliegue se realizó con muy buena aceptación por parte del cliente de forma tal que, a partir de ese momento, la tramitación actual de los módulos definidos en esta fase comenzó a realizarse mediante dicho sistema. El subproceso de Desistimiento solamente se tuvo en cuenta para las instancias municipales. De ahí que los expedientes se almacenen una parte digital y la otra física, lo que trae consigo gasto de recursos y tiempo a la hora de imprimir los documentos digitales para conformar el expediente en formato duro. Esto provoca su deterioro debido a la humedad, insectos y roedores. Por tal motivo la presente investigación tiene como objetivo desarrollar la personalización de dicho subproceso para las instancias provinciales. Para el desarrollo de esta propuesta de solución se tuvieron en cuenta las herramientas y tecnologías utilizadas en esta fase. Por otro lado, se obtuvo el modelado de sistema, de implementación y, por último, se validó dicha propuesta a través de las pruebas de software, corroborando que la personalización se había llevado a cabo de acorde con las especificaciones del cliente y que el sistema contaba con la calidad para ser utilizado por los especialistas funcionales.

**PALABRAS CLAVE:** expediente, personalización, subproceso de Desistimiento, Tribunales Populares Cubanos.

## *ÍNDICE DE CONTENIDOS*

INTRODUCCIÓN .....	11
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	16
1.1    Conceptos asociados al dominio del problema.....	16
1.2    Sistema de Informatización para la Gestión de los TPC.....	17
1.3    Subproceso Desistimiento.....	18
1.4    Descripción de la metodología de desarrollo de software.....	18
1.5    Herramienta CASE.....	19
1.5.1    Lenguaje de modelado UML.....	20
1.6    Marco de trabajo .....	21
1.6.1    Symfony .....	21
1.7    Bootstrap v 2.1 .....	21
1.8    Lenguajes de programación .....	22
1.9    Motor de plantillas Twig v 1.9.....	24
1.10    Sistema gestor de bases de datos .....	25
1.11    Herramienta de mapeo objeto relacional .....	25
1.12    Servidor web.....	26
1.13    Entorno de desarrollo integrado.....	26
1.14    Arquitectura de software .....	27
1.14.1    Patrón Modelo-Vista-Controlador (MVC) .....	27
1.15    Patrones de diseño.....	28
Conclusiones del capítulo.....	29
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA .....	30
2.1    Descripción del proceso de negocio.....	30
2.2    Modelo conceptual .....	32
2.3    Requisitos .....	33
2.3.1    Requisitos funcionales.....	33
2.3.2    Requisitos no funcionales del sistema .....	42
2.4    Arquitectura de software .....	43
2.4.1    Vista.....	43
2.4.2    Controlador .....	44
2.4.3    Modelo .....	44
2.5    Modelo de diseño.....	45
2.5.1    Patrones GRASP .....	45



2.5.2	Patrones GoF .....	47
2.5.3	Diagrama de clases del diseño .....	48
2.5.4	Diagrama de secuencia .....	49
2.5.5	Diagrama de despliegue .....	50
2.5.6	Modelo de datos .....	51
2.5.7	Estándares de codificación .....	52
	Conclusiones del capítulo .....	55
CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN PROPUESTA .....		56
3.1	Validación del diseño .....	56
3.1.1	Métrica Tamaño operacional de las clases (TOC) .....	56
3.1.2	Métrica Relación entre clases (RC) .....	58
3.2	Pruebas de software .....	60
3.2.1	Pruebas unitarias .....	61
3.2.2	Pruebas funcionales .....	64
3.2.2	Pruebas de rendimiento .....	65
3.3	Validación de las variables de la investigación .....	66
	Conclusiones del capítulo .....	69
CONCLUSIONES GENERALES .....		70
RECOMENDACIONES .....		71
REFERENCIAS BIBLIOGRÁFICAS .....		72
ANEXOS .....		¡ERROR! MARCADOR NO DEFINIDO.

*ÍNDICE DE ILUSTRACIONES*

Ilustración 1: Escenario No. 3 para encapsular los requisitos de software.....	19
Ilustración 2: Arquitectura del sistema .....	28
Ilustración 3: Descripción del proceso de negocio para el subproceso Desistimiento .....	31
Ilustración 4: Modelo conceptual del subproceso Desistimiento .....	32
Ilustración 5: Método indexAction() .....	46
Ilustración 6: Patrón Decorador. Clase plantilla_Ejecutivo.html.twig .....	48
Ilustración 7: Patrón Inyección de dependencias. Método getGestor() .....	48
Ilustración 8: Diagrama de clase del diseño del RF “Registrar escrito de Desistimiento” .....	49
Ilustración 9: Diagrama de secuencia del RF “Registrar escrito de Desistimiento” .....	50
Ilustración 10: Diagrama de despliegue del SITPC.....	51
Ilustración 11: Modelo de datos para el subproceso Desistimiento.....	52
Ilustración 12: Estándar de codificación. Cabecera de archivo .....	53
Ilustración 13: Estándar de codificación. Método capturarDatos.....	54
Ilustración 14: Estándar de codificación. Denominación de los servicios.....	54
Ilustración 15: Estándar de codificación. Denominación de las rutas.....	55
Ilustración 16: Resultado de la métrica TOC .....	57
Ilustración 17: Resultado de la métrica RC.....	60
Ilustración 18: Método pasarDefinitivo() utilizado para aplicarle la prueba de caja blanca .....	62
Ilustración 19: Grafo de flujo a partir del método pasarDefinitivo().....	62
Ilustración 20: Total de NC detectadas por iteración .....	65
Ilustración 21: Total de NC por tipo de clasificación .....	65
Ilustración 22: Tiempo de respuesta al registrar escrito de Desistimiento.....	66

*ÍNDICE DE TABLAS*

Tabla 1: Especificación de Requisitos funcionales del subproceso Desistimiento .....	33
Tabla 2: Categoría por atributos y criterio de evaluación. Métrica TOC.....	56
Tabla 3: Resultados obtenidos luego de aplicada la métrica TOC.....	57
Tabla 4: Categoría por atributos y criterio de evaluación. Métrica RC. ....	58
Tabla 5: Resultados obtenidos luego de aplicada la métrica RC .....	59
Tabla 6: Caso de prueba de la ruta independiente 1 .....	63
Tabla 7: Validación de las variables de investigación .....	67

*INTRODUCCIÓN*

Desde el año 2007 el Centro de Gobierno Electrónico (CEGEL) de la Universidad de Ciencias Informáticas (UCI) y los Tribunales Populares Cubanos (TPC) trabajan en conjunto para lograr la estandarización de procesos y documentos jurídicos, además de la ejecución y supervisión de manera automática de las actividades judiciales y disposiciones legales que lo regulan. Con este fin se creó el Sistema de Informatización para la Gestión de los Tribunales Populares Cubanos (SITPC). Como consecuencia de la gran extensión y complejidad de la solución propuesta se decidió dividirla en tres fases. La fase 1 está compuesta por los procedimientos: Ejecutivo de la materia de lo Económico, Disciplina y derecho laboral de la materia de lo Laboral, Administrativo de la materia de lo Administrativo y Ordinario de las materias de lo Penal y Civil. La solución informática es un sistema integrado donde existe para cada materia un proceso general sobre el cual inciden otros subprocesos como es el caso del Desistimiento.

El Desistimiento es la terminación anormal de un proceso por el que el actor manifiesta su voluntad de abandonar su pretensión, pero sin renunciar al derecho en que la basaba, es decir, que tiene la posibilidad de poder plantear la misma Litis<sup>1</sup> posteriormente (Enciclopedia-Jurídica, 2014). En otras palabras, es uno de los modos de extinción de la demanda, pues permite poner fin a la solicitud realizada por parte del demandante.

En el año 2015 fue desplegada la primera fase del proyecto con muy buena aceptación por parte de cliente, a partir de este momento la tramitación actual de los módulos definidos en esta fase se realiza mediante el SITPC. En las instancias provinciales los procedimientos automatizados son el Ordinario de las materias Penal y Civil, así como el Ejecutivo de la materia Económica; por su parte en las instancias municipales el Administrativo de la materia Administrativo y el de Derecho y disciplina de lo Laboral. El subproceso Desistimiento se encuentra automatizado en las instancias municipales en el módulo Común a las materias de la Ley del Procedimiento Civil, Administrativo, Laboral y Económico (LPCALE), pero solo se ajustó a los procedimientos que actualmente se trabajan en dicho sistema. Por lo que, en las instancias provinciales, si alguna de las partes que representan al demandante o la parte en sí, desiste de continuar con el proceso que se encuentra sobre un expediente que se viene tramitando en el sistema, a partir de ese entonces, se continúa de la forma tradicional, por no contar con ese subproceso ajustado a las necesidades actuales en las instancias provinciales del SITPC. De ahí que los expedientes se almacenen una parte digital y la otra física, lo que trae consigo gasto de

---

<sup>1</sup> Litis es un vocablo latino que en idioma español se traduce como litigio, significando disputa o controversia judicial; diferencia de intereses entre dos partes, llamadas litigantes, sometidas a decisión de un Juez.

recursos y tiempo a la hora de imprimir los documentos digitales para conformar el expediente en formato duro. Esto provoca su deterioro debido a la humedad, insectos y roedores.

Durante la ejecución de los procesos se generan documentos como escritos, providencias y autos, en los cuales pudieran cometerse errores de radicación ya que el sistema los numera automáticamente y no lleva la constancia de los que se numeraron de forma manual. El Desistimiento carece de retroalimentación estadística rápida, debido a que los datos son almacenados en estantes, provocando demoras en las búsquedas y obtención de la información. Si se archiva el expediente luego del desistimiento, en el sistema, este continúa en ejecución, lo que hará poco fiables los reportes estadísticos y continúan apareciendo en pantalla expedientes que ya llegaron a fin de proceso al haber desistido sobre ellos.

El acceso simultáneo por el personal autorizado a los expedientes de cada uno de los procesos legales que se realizan en los tribunales se dificulta, puesto que se realiza a través de la Secretaria Judicial, conllevando a demoras en su respuesta por el esfuerzo humano requerido, devenido en la organización actual de los documentos, unos en formato duro y otros en formato digital.

Los problemas antes mencionados influyen negativamente en la rapidez del proceso y en ocasiones el tiempo de tramitación es elevado, según la encuesta realizada a los especialistas funcionales de los TPC, con más de 5 años de experiencia atendiendo casos respecto a la tramitación del Desistimiento de alguna de las partes en un expediente determinado (ver Anexo 1). Esta encuesta se realizó en el 2017 a una muestra de 62 especialistas judiciales de todo el país, mientras recibían las capacitaciones para trabajar con el SITPC. El objetivo de la misma fue determinar el tiempo promedio que demoran habitualmente dichos especialistas en realizar las actividades comprendidas en la tramitación del subproceso en cuestión. A partir de esta encuesta se obtuvo que un 70% de estos especialistas plantean que se demoran entre 20 y 30 minutos para la búsqueda de los expedientes en los tribunales. Por otro lado, a pesar de que un 33% emplean menos de 10 minutos para capturar los datos al registrar los escritos en el subproceso Desistimiento, un 60% manifiestan que demoran de 10 a 15 minutos para esta actividad. La realización de las notificaciones es otra de las actividades que se ven afectadas en la actualidad, ya que un 60% manifiestan que tardan de 1 a 3 días para realizar esta notificación a las partes. De igual forma el tiempo empleado en la generación de los documentos es elevado, ya que un 77% manifiestan que tardan más de 30 minutos para conformar estos documentos en correspondencia con lo establecido por el TSP. La consulta de los expedientes de forma simultánea no es posible, por lo que, si un expediente se encuentra entregado a una de las partes, en caso de que la otra parte solicite el mismo expediente, en el 60% de los casos, esta parte debe esperar entre 1 y 3 días, según los resultados de la encuesta. Por último, según lo que plantea

la ley, existe un término para realizar cada trámite en los expedientes, en caso de que estos términos se venzan, la Secretaría judicial debe estar al tanto de esto, hacer los cálculos pertinentes e informarle al Juez ponente que se ha vencido el término. Esta notificación es otra de las actividades que se ve afectada debido al cúmulo de información que se maneja, ya que un 50% manifiestan que demoran entre 1 y 3 días para notificarle a dicho juez sobre el vencimiento de un plazo determinado en el subproceso.

Dada esta situación, los TPC se han trazado entre sus objetivos estratégicos, la adaptación del subproceso Desistimiento de las instancias municipales para las instancias provinciales en el SITPC.

A razón de la problemática existente surge el siguiente **problema de la investigación**: ¿Cómo contribuir a la agilización de la ejecución del subproceso Desistimiento en el SITPC para las instancias provinciales de los tribunales cubanos?

Tomando en cuenta el problema antes propuesto se define como **objeto de estudio**: Proceso de desarrollo de software en la Informática Jurídica de Gestión.

De esta forma se determina como **objetivo general**: Desarrollar la personalización del subproceso Desistimiento en el SITPC de forma tal que contribuya a la agilización de la ejecución del mismo en las instancias provinciales de los tribunales cubanos.

Para ello se identifica como **campo de acción**: Personalización del subproceso de Desistimiento en el SITPC para las instancias provinciales de los tribunales cubanos.

Por lo tanto, se plantea la siguiente **idea a defender**: Si se desarrolla la personalización del subproceso Desistimiento en el SITPC, se contribuirá a la agilización de la ejecución del mismo en las instancias provinciales de los tribunales cubanos.

Se desglosan del objetivo principal los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos en los que se sustenta el desarrollo de software.
- Obtener el modelo de diseño como aproximación a la implementación.
- Realizar la implementación para obtener la personalización del subproceso de Desistimiento.
- Validar los resultados obtenidos mediante la realización de pruebas de software, así como las variables de la investigación.

Para dar cumplimiento a los objetivos propuestos se definen las siguientes **tareas de la investigación**:

1. Definición de los principales conceptos relacionado con el objeto de estudio y campo de acción.
2. Descripción del proceso de desarrollo de software y su metodología.
3. Caracterización de las tecnologías y las herramientas a utilizar para el desarrollo de la solución.
4. Descripción del negocio del subproceso de Desistimiento.
5. Descripción de la arquitectura de software base para la implementación del sistema.
6. Caracterización y selección de los patrones de diseño más factibles para esta propuesta de solución.
7. Realización del diagrama de clases de diseño, de secuencia y de despliegue, así como el modelo de datos.
8. Implementación de la personalización del subproceso de Desistimiento para las instancias provinciales en el SITPC.
9. Ejecución de las pruebas de software para evaluar la calidad de la propuesta de solución.
10. Solución de no conformidades al finalizar cada iteración de las pruebas.
11. Validación de las variables de la investigación.

Para dar solución a las tareas antes propuestas se definen los métodos científicos, clasificados en teóricos y empíricos. De los cuales se emplearon:

**Métodos teóricos:**

**Histórico-lógico:** empleado en el estudio de la bibliografía relacionada con la informática jurídica, además de adquirir conocimientos respecto a cómo se realizan las actividades jurídicas en los TPC.

**Analítico-sintético:** esta es la unión de dos métodos, el analítico, el cual permitió la división del negocio en general en pequeñas partes para facilitar su estudio, y el sintético que basándose en los elementos analizados con el método anterior ayudó a realizar un resumen de los elementos más importantes del negocio.

**Modelación:** se utilizó para la realización de los diagramas necesarios en el proceso de desarrollo de software, haciendo una representación abstracta de la solución, facilitando así el desarrollo de la misma.

**Métodos empíricos:**

**Entrevista:** permitió el intercambio verbal con el cliente para obtener la mayor cantidad de información posible, entender todo el proceso de negocio, así como comprender la estructura y funcionamiento de la organización, así como las deficiencias existentes que permitieron definir el problema a resolver y establecer el objeto de estudio.

**Encuesta:** permitió mediante un cuestionario pre-elaborado, obtener información sobre el tiempo real que demoran los especialistas funcionales jurídicos en realizar la tramitación de los expedientes que se encuentren enmarcados en un subproceso de Desistimiento en los tribunales cubanos.

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

**Capítulo 1. Fundamentación teórica:** Se hace un estudio del SITPC, de igual forma se abordan los conceptos de interés para comprender mejor el problema que se plantea, se realiza un análisis de la metodología de desarrollo de software empleada, se profundiza en el marco de trabajo y arquitectura orientada por el equipo de arquitectura para el proceso de construcción del sistema, así como los lenguajes, tecnologías y herramientas a utilizar.

**Capítulo 2. Descripción de la solución propuesta:** Se presenta la descripción de la solución propuesta a través de la identificación de los requisitos funcionales y no funcionales del sistema, obteniéndose la base de la arquitectura del sistema. A partir de los requisitos identificados se modelarán los Diagramas de clases del diseño y de secuencia, los cuales permiten obtener una visión más clara de la implementación del sistema.

**Capítulo 3. Evaluación de la solución propuesta:** se evalúa el grado de calidad y fiabilidad de los resultados obtenidos en el desarrollo de este trabajo luego de haber personalizado el subproceso de Desistimiento. Esta evaluación se lleva a cabo a partir de la validación del diseño a través de las métricas: Tamaño Operacional de las Clases y Relación entre Clases. Por último, se aplican pruebas de software para verificar y revelar la calidad del producto software antes de ser entregado al cliente.



*CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA*

En el presente capítulo se hace un estudio del Sistema de Informatización de los Tribunales Populares Cubanos (SITPC), se abordan los conceptos de interés para comprender mejor el problema que se plantea, se realiza un análisis de la metodología de desarrollo de software empleada, así como del lenguaje de modelado y de la herramienta CASE<sup>2</sup> escogida para entender y manejar los artefactos durante todo el desarrollo del software. También se exponen los patrones de diseño, así como el marco de trabajo y arquitectura orientada a componentes que se emplea en el proceso de construcción del sistema. Además, se profundiza en las características de las herramientas de implementación que fueron definidas en el SITPC y que permiten una adecuada construcción del software.

### **1.1 Conceptos asociados al dominio del problema**

Con el objetivo de lograr un entendimiento del contenido a tratar en la investigación, se requiere el conocimiento de los conceptos enunciados a continuación:

**Agilización:** Según lo que plantea el Diccionario de la Real Academia Española, se refiere a la acción y efecto de agilizar, o sea, facilitar o acelerar la ejecución de un proceso.

Los autores de la presente investigación, basándose en esta definición adoptan la agilización, en el caso del subproceso de Desistimiento, a la acción de facilitar la tramitación de los expedientes que se enmarcan en dicho subproceso, de forma tal que disminuya el tiempo de ejecución y que contribuya al cumplimiento de los términos en dicho subproceso.

**Informática Jurídica:** Es la interrelación entre 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 (A. Peña, 2015).

**Informática Jurídica de Gestión:** La informática jurídica de gestión está encaminada a organizar y controlar la información jurídica de documentos, expedientes, libros, etc., mediante la aplicación de programas de administración que permitan crear identificadores y descriptores para la clasificación de dicha información. Este tipo de informática es conocida como de administración y/o control, es

---

<sup>2</sup> Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador

utilizada en tribunales, estudios jurídicos, notarías, entre otras; se utiliza sobre todo para llevar el seguimiento de trámites y procesos con el objeto de mantener actualizada la información y llevar un buen control de la misma (A. Peña, 2015).

**Personalizar:** Se define la personalización como sinónimo de *customizar*, palabra que no está concebida en el Diccionario de la Real Academia Española, sin embargo, tiene un uso bastante frecuente en la lengua española. Se trata de una adaptación del término inglés *customize*, que refiere a modificar algo de acuerdo a las preferencias personales (Fundeu.es, 2013).

Los autores de la presente investigación, a raíz de lo mencionado anteriormente, definen la personalización de productos informáticos como la acción de adaptar un producto según las nuevas funcionalidades o servicios a raíz de las nuevas peticiones de mercado. Para el caso de la presente investigación, la personalización se basa en la adaptación del subproceso de Desistimiento de acuerdo a las necesidades del usuario o cliente al cual va dirigido la propuesta de solución. Las acciones que se tendrán en cuenta para la personalización son:

- La estandarización del subproceso, de forma tal que se comporte de la misma forma, o sea, que solamente se pueda llevar de forma digital al igual que el resto de los procesos.
- La homogenización de los documentos de forma tal que se genere una plantilla común a los módulos con la diferencia del lugar donde se crea, con el fin de disminuir en tiempo y utilización de recursos.
- La integración con todos los módulos de la LPACALE, de forma tal que permita la tramitación completa para todas las materias, sin errores ni caminos nulos.

### 1.2 Sistema de Informatización para la Gestión de los TPC

El Sistema de Informatización para la Gestión de los TPC (SITPC) surge a partir de la necesidad del Tribunal Supremo Popular (TSP) de mejorar y agilizar la tramitación de sus procesos, así como las interacciones con los ciudadanos, destacando las implicaciones positivas que tiene para toda la sociedad cubana. El sistema cuenta, en una primera entrega, con 5 subsistemas y un total de 5 módulos que informatizan los procesos jurídicos que se desarrollan actualmente en los Tribunales Populares Cubanos. El desarrollo del sistema integra tecnologías basadas en software libre y productos elaborados en la Red de Centros de Desarrollo de la Universidad de las Ciencias Informáticas. El software se desplegó en los 169 Tribunales Municipales Populares (TMP) y los 16 Tribunales Populares Provinciales (TPP) del país, con buena aceptación por parte de los especialistas funcionales de los Tribunales Populares Cubanos. Este cuenta con 3000 usuarios que

gestionan los procesos judiciales asignados a cada territorio. De esta forma se asegura la rapidez y eficiencia en la gestión judicial y se les garantiza a los ciudadanos transparencia, celeridad e imparcialidad en sus procesos judiciales, teniendo en cuenta las disposiciones legales que regula la Constitución de la República de Cuba (González Ochoa, y otros, 2018).

### **1.3 Subproceso Desistimiento**

El Desistimiento está regido por el artículo 652 de la Ley del procedimiento Civil, Administrativo, Laboral y Económico (LPCALE), y es el proceso mediante el cual el demandante expresa su deseo de no continuar con el proceso comenzado y que hará innecesaria la sentencia. Para ello es necesario que se haga constar mediante una declaración formulada por el demandante.

En correspondencia a lo que plantea dicha ley, del escrito en que se formule se dará traslado a las demás partes por término de cinco días a fin de que muestren su conformidad o no con él, y en su vista, el Tribunal resolverá lo que corresponda sobre la continuación o no del proceso. Aunque todas las partes hayan mostrado su conformidad con el Desistimiento, el Tribunal, no obstante, previo traslado al Fiscal por tres días, podrá disponer que el proceso continúe hasta su terminación con arreglo a derecho siempre que el Desistimiento sea contrario al interés social o a los derechos de terceros protegidos por la ley. En este caso el Fiscal asumirá la representación de dichos intereses o derechos. La aprobación del Desistimiento indica ponerle fin al proceso, y se realiza mediante un auto definitivo. Sin embargo, el Desistimiento por sí solo producirá todos sus efectos si se formula antes de la contestación a la demanda o de haber transcurrido el término para contestarla, sin que se hubiere efectuado, teniendo el demandante cinco días hábiles para subsanar los errores encontrados en la declaración formulada para la petición del Desistimiento.

### **1.4 Descripción de la metodología de desarrollo de software**

#### **Proceso Unificado Ágil (AUP) variación UCI**

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable, se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, para ello nos apoyaremos en el Modelo CMMI-DEV v1.3. El

cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad (Rodríguez Sánchez, 2015).

Con la adaptación de AUP que se propone para la actividad productiva de la UCI:

- Se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3.
- Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos.
- Se redujo a 1 la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a 11 (Rodríguez Sanchez, 2015).

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas Internas, Pruebas de Liberación y Pruebas de Aceptación (Rodríguez Sánchez , 2015).

De igual forma se definen por parte de la metodología 4 escenarios para la disciplina de Requisitos. Para el caso del presente trabajo se selecciona el escenario 3. El mismo aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez y su continuidad. Se debe tener presente que este escenario es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados. En general plantea que los proyectos que modelen el negocio con DPN<sup>3</sup> solo pueden modelar el sistema con DRP<sup>4</sup> (Rodríguez Sánchez , 2015).



**Ilustración 1:** Escenario No. 3 para encapsular los requisitos de software

### 1.5 Herramienta CASE

Las herramientas de Ingeniería de Software Asistida por Ordenador (CASE por sus siglas en inglés: Computer Aided Software Engineering) son diversas aplicaciones informáticas destinadas a aumentar

---

<sup>3</sup> Descripción de Proceso del Negocio

<sup>4</sup> Descripción de Requisitos por Proceso

la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan al proceso de desarrollo del software en tareas como realizar un diseño del proyecto, cálculo de costos, implementación de parte del código con el diseño dado, compilación automática, documentación o detección de errores entre otras (Menéndez-Barzanallana Asensi, 2016).

### **Visual Paradigm**

Esta herramienta ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (Visual Paradigm, 2017).

Esta herramienta CASE utiliza como el lenguaje Unificado de Modelado v2.0 (UML). A continuación, se plantean sus principales características:

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Fácil de instalar y actualizar.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Diagramas de flujo de datos.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo (Visual Paradigm, 2017).

A partir de los elementos antes expuestos se decide por parte del equipo de arquitectura del proyecto utilizar Visual Paradigm en su versión 8.0, teniendo en cuenta que esta herramienta propicia un conjunto de funcionalidades para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Además, se tuvo en cuenta que la UCI posee una licencia académica para su uso.

#### **1.5.1 Lenguaje de modelado UML**

UML (Lenguaje Unificado de Modelado o Unified Modeling Language) es un lenguaje de modelado de sistemas de software respaldado por el OMG (Object Management Group) y aprobado por éste como

un estándar. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Ofrece un estándar para describir modelos del sistema, incluyendo aspectos conceptuales tales como procesos de negocios, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Cuenta con una sintaxis y una semántica con reglas sobre cómo agrupar los elementos del lenguaje y el significado de esta agrupación (lenguaje). Es visual, mediante su sintaxis se modelan distintos aspectos de situaciones reales, que permiten una mejor interpretación y entendimiento (modelado). Además, unifica varias técnicas de modelado en una sola (unificado) (Larman, 2016).

## **1.6 Marco de trabajo**

Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar (Recaman Chaux, y otros, 2014).

### **1.6.1 Symfony**

Symfony es un marco de trabajo que simplifica el desarrollo de una aplicación web mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Está basado en el clásico patrón de diseño web conocido como arquitectura MVC (Modelo-Vista-Controlador).

Para el desarrollo de la propuesta de solución se definió por parte del equipo de arquitectura la utilización de Symfony 2, debido a que es fácil de instalar y configurar en cualquier plataforma. De igual forma las aplicaciones desarrolladas con Symfony son compatibles con la mayoría de las plataformas, bibliotecas e infraestructuras que existen y se adaptan a entornos de negocio en cambio permanente, requiriendo menos esfuerzo para su mantenimiento.

En los epígrafes que se muestran a continuación, se describen las tecnologías que utiliza dicho marco de trabajo, así como las herramientas definidas por parte del equipo de arquitectura del proyecto SITPC, y por ende, utilizadas para el desarrollo de la propuesta de solución (Pérez Avila, 2012).

### **1.7 Bootstrap v 2.1**

Es un marco de trabajo de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de

diseño basado en HTML<sup>5</sup> y CSS<sup>6</sup>, así como, extensiones de JavaScript opcionales. Los desarrolladores eligen en un formulario los componentes y ajustes deseados, y de ser necesario, los valores de varias opciones a sus necesidades. El paquete consecuentemente generado ya incluye la hoja de estilo pre-compilada (Getbootstrap.com, 2018).

### **1.8 Lenguajes de programación**

Un lenguaje de programación es un lenguaje formal creado para describir el conjunto de acciones que un equipo debe ejecutar. El lenguaje de programación está compuesto de una serie de reglas sintácticas y semánticas que permiten expresar instrucciones que posteriormente serán interpretadas por el equipo (SoftDoit, 2018).

A raíz de que la presente investigación se enmarca sobre una arquitectura definida por el equipo de arquitectura del SITPC, a continuación, se describen cada uno de los lenguajes utilizados por el marco de trabajo definido y, por consiguiente, en el desarrollo de la propuesta de solución.

#### **HTML**

HTML (por sus siglas en inglés de HyperText Markup Language) es el lenguaje básico de la Web para crear documentos y aplicaciones para que los desarrolladores lo utilicen en cualquier lugar (W3C, 2018).

HTML en su versión 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Mejora el elemento <canvas>, capaz de renderizar elementos 3D en los navegadores, por ejemplo: Firefox, Chrome y Opera (W3C, 2018). En el caso de Symfony 2, trae definido HTML en su versión 5.

#### **JavaScript**

JavaScript es un lenguaje de programación ligero, interpretado por la mayoría de los navegadores y que les proporciona a las páginas web, efectos y funciones complementarias a las consideradas como estándar HTML. Este tipo de lenguaje de programación, con frecuencia son empleados en los sitios web, para realizar acciones en el lado del cliente, estando centrado en el código fuente de la página web. Además, ayuda a mejorar la gestión cliente/servidor; entre sus funciones básicas se encuentran: abrir y cerrar ventanas; cambios eficaces en una página (en lo que respecta a su contenido y aspecto; desarrollo de cadenas de texto; procedimientos aritméticos. Dado que su misión es extender el HTML, JavaScript es un lenguaje que contempla ciertas restricciones, que, de manera

---

<sup>5</sup> por sus siglas en inglés de HyperText Markup Language

<sup>6</sup> Hojas de estilo en cascada (CSS por sus siglas en inglés)

indirecta, terminan por brindarle seguridad al usuario (JavaScript.com, 2018). En el caso de Symfony 2, trae definido JavaScript en su versión 1.8.

### jQuery

jQuery es una librería en JavaScript de fuente abierta (código abierto), que simplifica mucho la creación de páginas web, sobre todo en interfaces web que se comportan de modo muy similar a las aplicaciones de escritorio. Esta nos propone una metodología de programación orientado a objetos en un flujo controlado por eventos, donde encontramos características tales como:

- **Alta compatibilidad entre navegadores:** El uso de jQuery permite desarrollar un código que no tenga problemas con una implementación particular de un navegador, por ejemplo, en el trabajo con AJAX<sup>7</sup>. Es decir, una mayor compatibilidad entre navegadores.
- **Iteración implícita:** Los selectores que se usan en jQuery para encontrar elementos en el DOM, poseen iteración implícita, esto significa que evita tener que programar bucles de código para buscar todos elementos en el DOM<sup>8</sup> que cumplen el criterio dado. Por ejemplo, seleccionar elementos “<div>”<sup>9</sup> del HTML con una clase CSS de nombre PROMO se programa en una sola línea de código.
- **Encadenamiento de métodos:** Permite en una sola línea de código hacer mucho más que usando solamente las características propias del JavaScript. Esto es muy útil cuando se recorre el DOM de un documento HTML, buscando un elemento determinado. Es decir, con jQuery se hace más con menos líneas de código (Jquery.com, 2018).

En el caso de Symfony 2, trae definido jQuery en su versión 1.8.

### PHP

El lenguaje de programación PHP (siglas del inglés Hypertext Pre-processor) fue desarrollado puntualmente para diseñar páginas web dinámicas programando scripts del lado del servidor. El lenguaje PHP siempre va incrustado dentro del HTML y generalmente se le relaciona con el uso de servidores Linux (Programación.net, 2018).

### Características principales:

---

<sup>7</sup> AJAX, acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas.

<sup>8</sup> DOM acrónimo de Document Object Model (Modelo de Objetos del Documento) es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML

<sup>9</sup> Etiqueta HTML que sirve para crear secciones o agrupar contenidos en documentos HTML.



- **Velocidad:** PHP no solo es rápido al ser ejecutado, sino que no genera retrasos en la máquina, por esto no requiere grandes recursos del sistema. PHP se integra muy bien junto a otras aplicaciones, especialmente bajo sistemas operativos de software libre.
- **Estabilidad:** PHP utiliza su propio sistema de administración de recursos y posee un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- **Seguridad:** PHP maneja distintos niveles de seguridad, estos pueden ser configurados desde el archivo “.ini”.
- **Simplicidad:** Usuarios con experiencia en los lenguajes de programación C y C++ podrán utilizar PHP rápidamente. Además, PHP dispone de una amplia gama de librerías, y permite la posibilidad de agregarle extensiones. Esto le permite su aplicación en múltiples áreas, tales como encriptado, gráficos, XML y otras (Programación.net, 2018).

En el caso de Symfony 2, trae definido PHP en su versión 5.3.

### 1.9 Motor de plantillas Twig v 1.9

Es un motor de plantillas para el lenguaje de programación PHP. Su sintaxis se origina a partir de Jinja y plantillas de Django. El producto de código abierto se distribuye bajo licencia BSD<sup>10</sup> y desarrollado por Fabien Potencier (Pacheco, 2014).

#### Entre sus características se encuentran:

- **Rápido:** compila las plantillas a código PHP sencillo optimizado. La sobrecarga en comparación con el ordinario de código PHP se ha reducido considerablemente.
- **Seguro:** tiene un modo de recinto para evaluar código de la plantilla no es de confianza. Esto permite Twig para ser utilizado como un lenguaje de plantillas para aplicaciones donde el usuario puede modificar el diseño de la plantilla.
- **Flexible:** es impulsado por un léxico flexible y analizador. Esto permite al desarrollador definir sus propias etiquetas y filtros personalizados (Pacheco, 2014).

En el caso del marco de trabajo seleccionado, trae definido para la elaboración de las vistas el motor de plantillas Twig en su versión 1.9.

---

<sup>10</sup> Licencia de software otorgada para los sistemas Berkeley Software Distribution, tipo de sistema operativo Unix-like, o sea, licencia de software libre.

### **1.10 Sistema gestor de bases de datos**

Un Sistema gestor de base de datos es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarios para el almacenamiento y búsqueda de la información del modo más eficiente posible (Iruela, 2016).

#### **PostgreSQL**

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Sus principales características son:

- Alta concurrencia: mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés).
- Amplia variedad de tipos nativos: provee nativamente varios soportes.
- Ahorros considerables de costos de operación.
- Estabilidad y confiabilidad (Iruela, 2016).

En el caso del marco de trabajo seleccionado, trae definido como sistema gestor de base de datos PostgreSQL en su versión 9.2.

### **1.11 Herramienta de mapeo objeto relacional**

El mapeo objeto relacional es una técnica de programación para convertir datos del sistema de tipos utilizado en un lenguaje de programación orientado a objetos al utilizado en una base de datos relacional. En la práctica esto crea una base de datos virtual orientada a objetos sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos, esencialmente la herencia y el polimorfismo (Mapeo Objeto Relacional, 2014).

#### **Doctrine v 2.0**

Brinda persistencia a los objetos PHP de manera transparente, no enfoca su trabajo en la inserción o búsqueda de filas y/o columnas en la base de datos; sino que trata de hacer persistir y buscar objetos hacia y desde la base de datos. Además, dicho lenguaje permite a los desarrolladores escribir poderosas consultas de una manera sencilla y flexible, puesto que es Doctrine internamente quien se encarga de las conversiones para gestionar la información de la base de datos. Las entidades de Doctrine v2.0 son clases escritas en lenguaje PHP puro (Doctrine-project.org, 2018).

Symfony 2, trae definido como mapeo de objeto relacional ORM Doctrine en su versión 2.

### **1.12 Servidor web**

Un servidor web es un programa que utiliza HTTP (Hypertext Transfer Protocol) para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP de sus computadoras. Las computadoras y los dispositivos dedicados también pueden denominarse servidores Web (Pavón Mestras, 2014).

#### **Apache**

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP. Entre sus principales características están:

- Multiplataforma.
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor (Ibrugor, 2014).

El equipo de arquitectura del proyecto SITPC define Apache como software para el servidor de aplicaciones en su versión 2.2, por su flexibilidad, rapidez, por ser libre de licencia, multiplataforma, así como la facilidad de interpretar varios lenguajes como PHP.

### **1.13 Entorno de desarrollo integrado**

Un entorno de desarrollo integrado (IDE por sus siglas en inglés), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (Martin Olivera, y otros, 2016).

#### **Netbeans**

Netbeans es un IDE creado en Java que permite el desarrollo de aplicaciones de escritorio en Java, aplicaciones Web, HTML5, JavaScript y CSS. Brinda una serie de herramientas para los desarrolladores de C, C++ y PHP. Netbeans es una herramienta libre, de código abierto, multiplataforma y posee una gran comunidad de desarrolladores y usuarios en el mundo. Entre las principales características que posee este IDE se encuentran: (NetBeans.org, 2018)

- El mejor soporte para las últimas tecnologías Java.
- La edición rápida e inteligente del código fuente.
- Administración fácil y eficiente de los proyectos.
- Desarrollo rápido de interfaces de usuario.

- Posee una gran comunidad que provee plugins para el IDE. (NetBeans.org, 2018)

El equipo de arquitectura del proyecto SITPC define Netbeans como IDE en su versión 8.1, porque es gratuito, multiplataforma e interpreta varios lenguajes como PHP.

### 1.14 Arquitectura de software

La arquitectura de un sistema es la estructura del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente y las relaciones entre ellos (Pressman, 2010).

Representa la base para todo el funcionamiento de la aplicación; es el pilar principal del producto que se quiere construir. En su forma más simple, es la estructura u organización de los componentes del programa (módulos), la manera en que estos componentes interactúan y la estructura de datos que utilizan. La solución informática que se pretende obtener como resultado de la investigación no está aislada, por lo que debe seguir la arquitectura definida por el equipo de arquitectura. La misma es una arquitectura en capas basada en el patrón arquitectónico Modelo-Vista-Controlador (MVC) que es uno de los más utilizados en las aplicaciones web y también viene integrado al marco de trabajo Symfony 2 que fue el seleccionado para la implementación de la aplicación. La estructura que define el marco de trabajo Symfony 2 para la separación de estas capas deja bien delimitado dónde quedan las clases del modelo, las de la vista y las del controlador.

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

**Modelo:** Es un objeto que representa cierta información del dominio. Constituye un objeto no visual y contiene todos los datos y comportamiento del mismo.

**Vista:** La vista representa la muestra del modelo mediante la interfaz visual.

**Controlador:** Se encarga de cambiar cualquier información existente en el modelo. Toma las entradas del usuario, manipula el modelo y actualiza la vista apropiadamente.

A continuación, se muestra la representación de la arquitectura definida para SITPC descrita anteriormente:

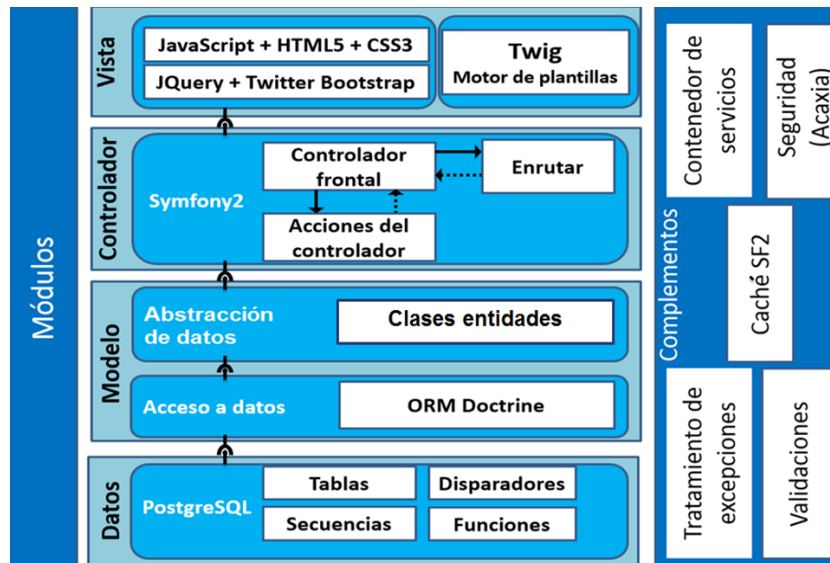


Ilustración 2: Arquitectura del sistema

Uno de los aspectos más importantes que proporciona la división de estas capas es la separación clara de dónde tiene que ir cada tipo de lógica, facilitando el mantenimiento y la escalabilidad del sistema en desarrollo, ya que brinda la oportunidad de que se puedan realizar cambios en la vista sin afectar el modelo y simplifica el trabajo de los desarrolladores que estén trabajando en ambas capas (Fowler, 2002).

### 1.15 Patrones de diseño

Cuando se desarrolla una aplicación software es frecuente encontrarse en la situación de tener que volver a resolver problemas similares a otros que ya hemos solucionado anteriormente, y debemos volver a hacerlo partiendo de cero una y otra y otra vez (incluso dentro del mismo proyecto). Debido a ello y basándose en la programación orientada a objetos surgieron los patrones de diseño, donde cada uno de ellos define la solución para resolver un determinado problema, facilitando además la reutilización del código fuente (Larman, 2016).

- **Patrones GRASP:** son una ayuda en el aprendizaje que permiten al desarrollador entender lo esencial del diseño de objetos y a aplicar el razonamiento de una forma metódica, racional y explicable. Este enfoque se entiende en el uso de los principios del diseño basado en patrones para la asignación de responsabilidades a las clases y objetos de una aplicación (Larman, 2016).
- **Patrones GoF:** En (Gamma *et al.*, 1997) se propone un total de 23 patrones que pueden ser utilizados para el correcto diseño de un software, estos patrones son reconocidos como los

patrones GoF (Gang of Four). A continuación, se presentan las formas de clasificar estos patrones según su propósito:

- Creacionales: Conciernen al proceso de creación de objetos.
- Estructurales: Tratan con la composición de las clases o los objetos.
- Comportamiento: Caracterizan la vía en que las clases u objetos interactúan y distribuyen las responsabilidades (Larman, 2016).

### **Conclusiones del capítulo**

De manera general en este capítulo se abordaron conceptos que facilitaron la comprensión del trabajo y características de las herramientas, arquitectura del sistema, metodología de desarrollo de software y características fundamentales de los lenguajes de programación tanto del lado del cliente como del lado del servidor para el desarrollo de aplicaciones web. Este estudio resultó de gran importancia para una mejor comprensión del objetivo del actual trabajo y permitió sentar las bases para el inicio de la personalización del subproceso Desistimiento del SITPC.

*CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

En el presente capítulo se presenta la descripción de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales del sistema, obteniéndose la base de la arquitectura del sistema. A partir de los requisitos identificados se modelarán los Diagramas de clases del diseño y de secuencia, los cuales permiten obtener una visión más clara de la implementación del sistema.

**2.1 Descripción del proceso de negocio**

El subproceso desistimiento tiene comienzo cuando la parte demandante presenta un escrito donde manifiesta su deseo de terminar la causa iniciada. La secretaria judicial del TPP registra el escrito con los documentos adjuntos necesarios que se presentan. El juez se hará cargo de tomar la decisión si el proceso continúa o no; este recibe el escrito de desistimiento y lo revisa, si no cumple con las formalidades requeridas ordena subsanarlo de lo contrario crea una providencia de admisión y traslado.

Si la demanda fue contestada la secretaria notifica la admisión del desistimiento a las partes y el demandado tiene 5 días para contestar el desistimiento. El juez ponente recibe el escrito de contestación del desistimiento y lo revisa, si no cumple con las formalidades requeridas dicta una providencia de subsanación de la contestación; en caso contrario el juez dicta un auto de No ha lugar a Admitir (ANHLA) rechazando la contestación o lo admite, para lo que crea una resolución. En dicha resolución el juez propone dar traslado al fiscal o resolver el desistimiento.

En el caso que el juez decida darle traslado al fiscal, la secretaria elabora un oficio que se envía al fiscal para que se persone en el término de 3 días. Luego que se persone el fiscal, el juez dispone sobre este: si el escrito de personería del fiscal no cumple con las formalidades requeridas el juez dicta una resolución de subsanación de personería; en caso de que el escrito esté correcto el juez dicta una resolución de admisión o rechazo y la secretaria es quien notifica dicha resolución.

El juez dicta la providencia de subsanación dando un término de 2 a 5 días y la secretaria notifica a la parte la providencia de subsanación. Si pasa el término y no se subsana el escrito el juez dicta una resolución sobre el vencimiento del término que se le concedió a las partes para subsanar los errores en el escrito y la secretaria notifica a la parte la resolución dictada. Finalmente se resuelve la causa llegando así al fin del proceso con la admisión o rechazo del desistimiento. En la ilustración que se presenta a continuación, se describen los pasos del negocio enunciados anteriormente:

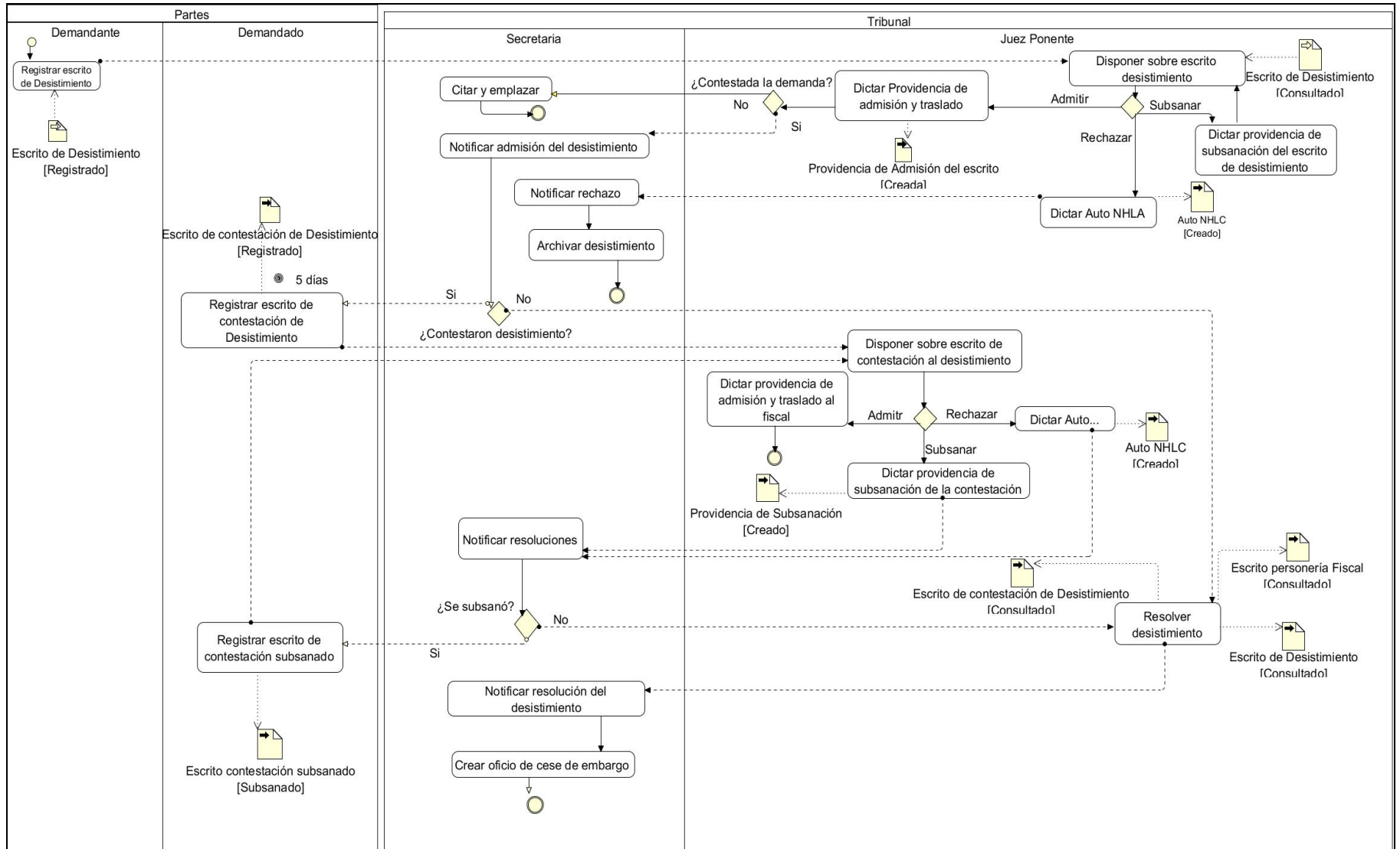
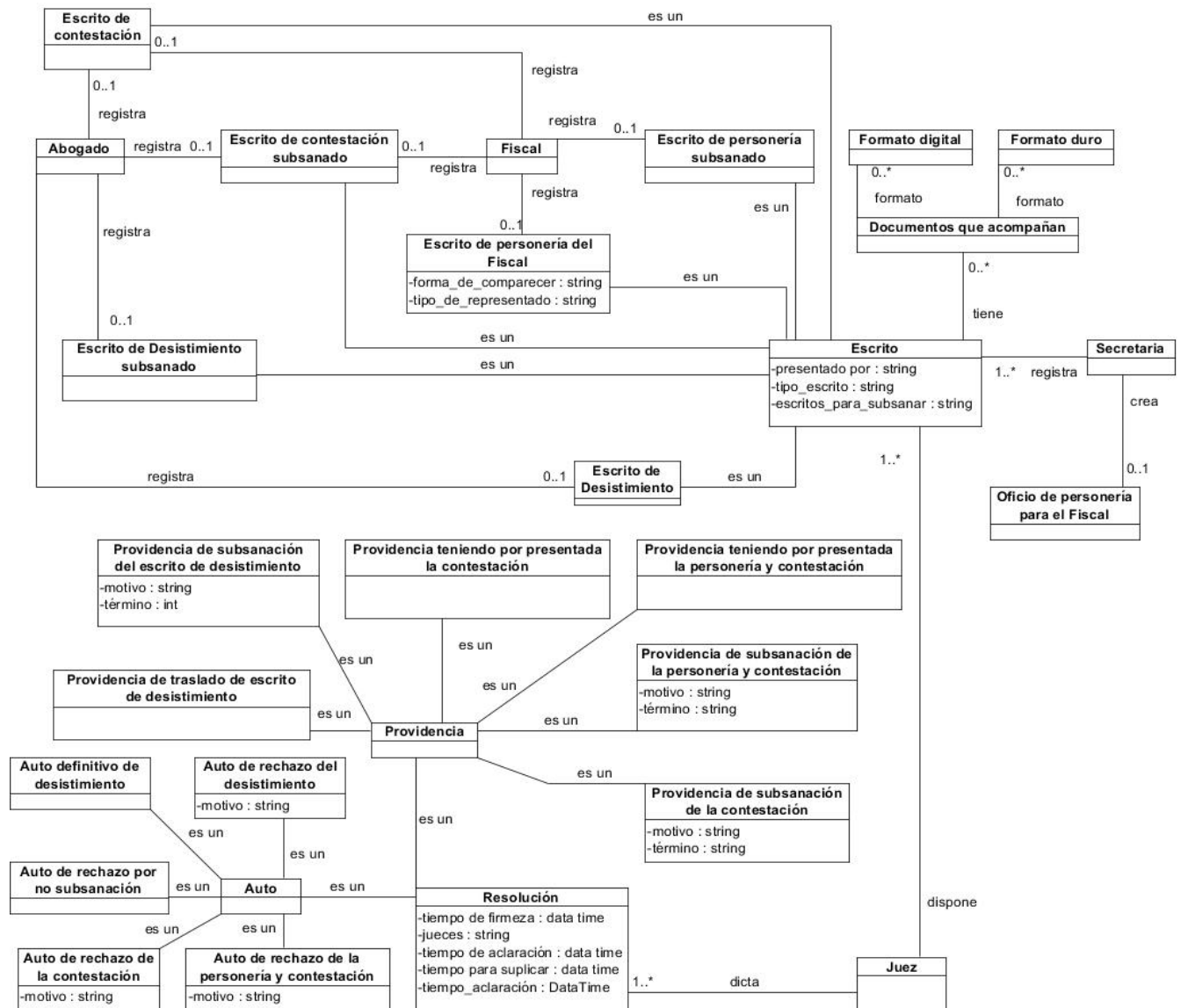


Ilustración 3: Descripción del proceso de negocio para el subproceso Desistimiento



## 2.2 Modelo conceptual

El modelo conceptual es la descripción de cómo se relacionan los conceptos en un problema. Sirve para representar un problema de manera gráfica a través de diagramas entidad relación, diccionarios/glosarios y diagramas de clases. Además, es importante para abstraer un problema e identificar como interactúa el sistema en el cual se desenvuelve la solución. Con el modelo conceptual se modela un problema identificando su funcionamiento (Valenzuela Buitrago, y otros, 2014). Con el objetivo de relacionar los conceptos del negocio en el que se enmarca la presente investigación, en la Ilustración 4 se muestra el Modelo conceptual propuesto para el subproceso de Desistimiento.



**Ilustración 4:** Modelo conceptual del subproceso Desistimiento

Una vez concebido el Modelado del Negocio, con el objetivo de comprender el negocio que se desea informatizar, se da paso a la disciplina de Requisitos planteada por la metodología de desarrollo de software que guía la presente investigación.

## **2.3 Requisitos**

### **2.3.1 Requisitos funcionales**

Los requisitos funcionales (RF) de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio, cumplimiento, seguridad u otra índole (Pressman, 2010).

El equipo de desarrollo identificó 111 requisitos funcionales (RF), en la Tabla 1 se muestra la especificación de estos requisitos:

**Tabla 1:** Especificación de Requisitos funcionales del subproceso Desistimiento

<b>No.</b>	<b>Funcionalidad</b>	<b>Descripción</b>
RF 1.	Registrar escrito de desistimiento.	El sistema permitirá registrar el escrito de desistimiento en caso de que la parte desee desistir del proceso.
RF 2.	Seleccionar la persona que presenta el escrito	El sistema permitirá seleccionar al abogado, representante legal, o la parte personada en el proceso que registra el escrito.
RF 3.	Seleccionar el tipo de escrito	El sistema permitirá seleccionar el tipo de escrito que será registrado.
RF 4.	Adicionar documentos en formato digital	El sistema permitirá adicionar documentos en formato digital adjuntos a los escritos que se presentan.
RF 5.	Eliminar documentos en formato digital	El sistema permitirá eliminar documentos en formato digital que estén adjuntos a los escritos que se presentan.
RF 6.	Adicionar documentos en formato duro	El sistema permitirá adicionar documentos en formato duro adjuntos a los escritos que se presentan.

*CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

---

RF 7.	Eliminar documentos en formato duro	El sistema permitirá eliminar documentos en formato duro que estén adjuntos a los escritos que se presentan.
RF 8.	Visualizar el documento del escrito de desistimiento	El sistema debe visualizar el documento: escrito de desistimiento para su edición.
RF 9.	Editar el documento escrito de desistimiento	El sistema permitirá editar el documento: escrito de desistimiento.
RF 10.	Guardar el documento escrito de desistimiento	El sistema permitirá guardar el documento: escrito de desistimiento.
RF 11.	Visualizar el documento escrito de desistimiento en formato .pdf	El sistema permitirá visualizar el documento: escrito de desistimiento en formato .pdf.
RF 12.	Registrar escrito de contestación al desistimiento.	El sistema permitirá registrar el escrito de contestación al desistimiento en caso de que la parte desee contestar al subproceso de desistimiento presentado en el expediente.
RF 13.	Visualizar el documento del escrito de contestación al desistimiento	El sistema permitirá visualizar el documento escrito de contestación al desistimiento para su edición.
RF 14.	Editar el documento escrito de contestación al desistimiento	El sistema permitirá editar el documento escrito de contestación al desistimiento.
RF 15.	Guardar el documento escrito de contestación al desistimiento	El sistema permitirá guardar el documento escrito de contestación al desistimiento.
RF 16.	Visualizar el documento escrito de contestación al desistimiento en formato .pdf	El sistema permitirá visualizar el documento escrito de contestación al desistimiento en formato .pdf.
RF 17.	Registrar escrito de personería.	El sistema permitirá registrar el escrito de personería en caso de que un abogado o representante fiscal desee personarse en el expediente.
RF 18.	Visualizar el documento del escrito de personería de abogado	El sistema permitirá visualizar el documento escrito de personería de abogado para su edición.

*CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

---

RF 19.	Editar el documento escrito de personería de abogado	El sistema permitirá editar el documento escrito de personería de abogado.
RF 20.	Guardar el documento escrito de personería de abogado	El sistema permitirá guardar el documento escrito de personería de abogado.
RF 21.	Visualizar el documento escrito de personería de abogado en formato .pdf	El sistema permitirá visualizar el documento escrito de personería de abogado en formato .pdf.
RF 22.	Guardar el documento escrito de personería de abogado como documento definitivo	El sistema permitirá guardar el documento escrito de personería de abogado como documento definitivo.
RF 23.	Registrar escrito de subsanación.	El sistema permitirá registrar el escrito de subsanación en caso de que el juez disponga la Subsanación sobre la presentación del escrito.
RF 24.	Visualizar el documento del escrito a subsanar	El sistema permitirá visualizar el documento del escrito a subsanar.
RF 25.	Guardar escrito de subsanación	El sistema permitirá guardar definitivamente los datos del escrito que se registra.
RF 26.	Adicionar el motivo del desistimiento al escrito de desistimiento	El sistema permitirá registrar el motivo por el que se desiste al escrito de desistimiento.
RF 27.	Seleccionar el tipo de escrito a subsanar en el escrito de subsanación	El sistema permitirá seleccionar el tipo de escrito que será subsanado en el escrito de subsanación.
RF 28.	Mostrar trámite para disponer sobre el escrito de desistimiento	El sistema permitirá mostrar el trámite para disponer sobre el escrito de desistimiento en la pantalla del juez ponente.
RF 29.	Mostrar los documentos necesarios para disponer sobre el escrito de desistimiento	El sistema debe mostrar el documento escrito de desistimiento y los documentos adjuntos al escrito en formato duro o formato digital como documentos necesarios, que ayuden en la disposición.
RF 30.	Seleccionar la opción de dar traslado al fiscal	El sistema debe permite seleccionar la opción de dar traslado al fiscal en caso de que se necesite

*CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

		la intervención del mismo en el asunto.
RF 31.	Visualizar la providencia de admisión del escrito de desistimiento	El sistema permitirá visualizar la providencia de admisión del escrito de desistimiento para su edición.
RF 32.	Editar la providencia de admisión del escrito de desistimiento	El sistema permitirá editar la providencia de admisión del escrito de desistimiento.
RF 33.	Guardar la providencia de admisión del escrito de desistimiento	El sistema permitirá guardar la providencia de admisión del escrito de desistimiento.
RF 34.	Visualizar la providencia de admisión del escrito de desistimiento en formato .pdf	El sistema permitirá visualizar la providencia de admisión del escrito de desistimiento en formato .pdf.
RF 35.	Guardar la providencia de admisión del escrito de desistimiento como documento definitivo	El sistema permitirá guardar la providencia de admisión del escrito de desistimiento como documento definitivo.
RF 36.	Adicionar el término para subsanar el escrito de desistimiento	El sistema permitirá adicionar el término para subsanar el escrito de desistimiento.
RF 37.	Visualizar la providencia de subsanación del escrito de desistimiento	El sistema permitirá visualizar la providencia de subsanación del escrito de desistimiento para su edición.
RF 38.	Editar la providencia de subsanación del escrito de desistimiento	El sistema permitirá editar la providencia de subsanación del escrito de desistimiento.
RF 39.	Guardar la providencia de subsanación del escrito de desistimiento	El sistema permitirá guardar la providencia de subsanación del escrito de desistimiento.
RF 40.	Visualizar la providencia de subsanación del escrito de desistimiento en formato .pdf	El sistema permitirá visualizar la providencia de subsanación del escrito de desistimiento en formato .pdf.
RF 41.	Visualizar el auto de rechazo del escrito desistimiento para su edición	El sistema permitirá visualizar el auto de rechazo del desistimiento para su edición.
RF 42.	Editar el auto de rechazo del escrito de desistimiento	El sistema permitirá editar el auto de rechazo del escrito de desistimiento.
RF 43.	Guardar el auto de rechazo del escrito	El sistema permitirá guardar el auto de rechazo

*CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

	de desistimiento	del escrito de desistimiento.
RF 44.	Visualizar auto de rechazo del escrito de desistimiento en formato .pdf	El sistema permitirá visualizar el auto de rechazo del escrito de desistimiento en formato .pdf.
RF 45.	Guardar el auto de rechazo del escrito de desistimiento como documento definitivo	El sistema permitirá guardar el auto de rechazo del escrito de desistimiento como documento definitivo.
RF 46.	Mostrar trámite para disponer sobre el escrito de contestación al desistimiento	El sistema permitirá mostrar el trámite para disponer sobre el escrito de contestación al desistimiento en la pantalla del juez ponente.
RF 47.	Mostrar los documentos necesarios para disponer sobre el escrito de contestación al desistimiento	El sistema debe mostrar el documento escrito de contestación al desistimiento y los documentos adjuntos al escrito en formato duro o formato digital como documentos necesarios, que ayuden en la disposición.
RF 48.	Visualizar la providencia de admisión del escrito de contestación al desistimiento	El sistema permitirá visualizar la providencia de admisión del escrito de contestación al desistimiento para su edición.
RF 49.	Editar la providencia de admisión del escrito de contestación al desistimiento	El sistema permitirá editar la providencia de admisión del escrito de contestación al desistimiento.
RF 50.	Guardar la providencia de admisión del escrito de contestación al desistimiento	El sistema permitirá guardar la providencia de admisión del escrito de contestación al desistimiento.
RF 51.	Visualizar la providencia de admisión del escrito de contestación al desistimiento en formato .pdf	El sistema permitirá visualizar la providencia de admisión del escrito de contestación al desistimiento en formato .pdf.
RF 52.	Adicionar el término para subsanar el escrito de contestación al desistimiento	El sistema permitirá adicionar el término para subsanar el escrito de contestación al desistimiento.
RF 53.	Visualizar la providencia de subsanación del escrito de	El sistema permitirá visualizar la providencia de subsanación del escrito de contestación al

*CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

	contestación al desistimiento	desistimiento para su edición.
RF 54.	Editar la providencia de subsanación del escrito de contestación al desistimiento	El sistema permitirá editar la providencia de subsanación del escrito de contestación al desistimiento.
RF 55.	Guardar la providencia de subsanación del escrito de contestación al desistimiento	El sistema permitirá guardar la providencia de subsanación del escrito de contestación al desistimiento.
RF 56.	Visualizar la providencia de subsanación del escrito de contestación al desistimiento en formato .pdf	El sistema permitirá visualizar la providencia de subsanación del escrito de contestación al desistimiento en formato .pdf.
RF 57.	Editar el auto de rechazo de contestación al desistimiento	El sistema permitirá editar el documento de rechazo de contestación al desistimiento.
RF 58.	Guardar el auto de rechazo de contestación al desistimiento	El sistema permitirá guardar el documento de rechazo de contestación al desistimiento.
RF 59.	Visualizar el auto de rechazo de contestación al desistimiento en formato .pdf	El sistema permitirá visualizar el documento de rechazo de contestación al desistimiento en formato .pdf.
RF 60.	Mostrar trámite para disponer sobre el escrito de personería	El sistema permitirá mostrar el trámite para disponer sobre el escrito de personería en la pantalla del juez ponente.
RF 61.	Mostrar los documentos necesarios para disponer sobre el escrito de personería de abogado	El sistema debe mostrar el documento escrito de personería de abogado y los documentos adjuntos al escrito en formato duro o formato digital como documentos necesarios, que ayuden en la disposición.
RF 62.	Visualizar la providencia de admisión del escrito de personería	El sistema permitirá visualizar la providencia de admisión del escrito de personería para su edición.
RF 63.	Editar la providencia de admisión del escrito de personería	El sistema permitirá editar la providencia de admisión del escrito de personería.
RF 64.	Guardar la providencia de admisión	El sistema permitirá guardar la providencia de

*CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

	del escrito de personería	admisión del escrito de personería.
RF 65.	Visualizar la providencia de admisión del escrito de personería en formato .pdf	El sistema permitirá visualizar la providencia de admisión del escrito de personería en formato .pdf.
RF 66.	Adicionar el término para subsanar el escrito de personería	El sistema permitirá adicionar el término para subsanar el escrito de personería.
RF 67.	Visualizar la providencia de subsanación del escrito de personería	El sistema permitirá visualizar la providencia de subsanación del escrito de personería para su edición.
RF 68.	Editar la providencia de subsanación del escrito de personería	El sistema permitirá editar la providencia de subsanación del escrito de personería.
RF 69.	Guardar la providencia de subsanación del escrito de personería	El sistema permitirá guardar la providencia de subsanación del escrito de personería.
RF 70.	Visualizar la providencia de subsanación del escrito de personería en formato .pdf	El sistema permitirá visualizar la providencia de subsanación del escrito de personería en formato .pdf.
RF 71.	Editar el documento de rechazo de la personería	El sistema permitirá editar el documento de rechazo de la personería.
RF 72.	Guardar el documento de rechazo de la personería	El sistema permitirá guardar el documento de rechazo de la personería.
RF 73.	Visualizar el documento de rechazo de la personería en formato .pdf	El sistema permitirá visualizar el documento de rechazo de la personería en formato .pdf.
RF 74.	Buscar expedientes con resoluciones pendientes a notificar	El sistema permitirá buscar aquellos expedientes que tienen resoluciones pendientes a notificar.
RF 75.	Seleccionar persona a notificar	El sistema permitirá adicionar la persona que será notificada.
RF 76.	Listar resoluciones pendientes de notificación	El sistema permitirá listar las resoluciones que estén pendientes de notificación.
RF 77.	Visualizar providencia de admisión y traslado del escrito de desistimiento para notificar	El sistema permitirá visualizar la providencia de admisión y traslado del escrito de desistimiento para notificar.



*CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

---

RF 78.	Visualizar providencia de subsanación del escrito de desistimiento para notificar	El sistema permitirá visualizar la providencia de subsanación del escrito de desistimiento para notificar.
RF 79.	Visualizar auto de rechazo del escrito de desistimiento para notificar	El sistema permitirá visualizar el auto de rechazo del escrito de desistimiento para notificar.
RF 80.	Visualizar providencia de admisión y traslado del escrito de contestación al desistimiento para notificar	El sistema permitirá visualizar la providencia de admisión y traslado del escrito de contestación al desistimiento para notificar.
RF 81.	Visualizar providencia de subsanación del escrito de contestación al desistimiento para notificar	El sistema permitirá visualizar la providencia de subsanación del escrito de contestación al desistimiento para notificar.
RF 82.	Guardar fecha de notificación de la providencia de subsanación del escrito de contestación al desistimiento	El sistema permitirá guardar la fecha de notificación de la providencia de subsanación del escrito de contestación al desistimiento.
RF 83.	Visualizar auto de rechazo del escrito de contestación al desistimiento para notificar	El sistema permitirá visualizar el auto de rechazo del escrito de contestación al desistimiento para notificar.
RF 84.	Visualizar providencia de admisión del escrito de personería para notificar	El sistema permitirá visualizar la providencia de admisión del escrito de personería para notificar.
RF 85.	Visualizar providencia de subsanación del escrito de personería para notificar	El sistema permitirá visualizar la providencia de subsanación del escrito de personería para notificar.
RF 86.	Visualizar auto de rechazo del escrito de personería para notificar	El sistema permitirá visualizar el auto de rechazo del escrito de personería para notificar.
RF 87.	Visualizar auto de admisión del desistimiento para notificar	El sistema permitirá visualizar el auto de admisión del desistimiento para notificar.
RF 88.	Guardar fecha de notificación del auto de admisión del desistimiento	El sistema permitirá guardar la fecha de notificación del auto de admisión del desistimiento.
RF 89.	Visualizar auto de rechazo del desistimiento para notificar	El sistema permitirá visualizar el auto de rechazo del desistimiento para notificar.

## *CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA*

---

RF 90.	Mostrar trámite para declarar no haber lugar a admitir	El sistema permitirá mostrar el trámite para declarar no haber lugar a admitir en la pantalla del juez ponente.
RF 91.	Visualizar auto para declarar no haber lugar a admitir	El sistema permitirá visualizar el auto para declarar no haber lugar a admitir para su edición.
RF 92.	Editar auto para declarar no haber lugar a admitir	El sistema permitirá editar el auto para declarar no haber lugar a admitir.
RF 93.	Guardar auto para declarar no haber lugar a admitir	El sistema permitirá guardar el auto para declarar no haber lugar a admitir.
RF 94.	Visualizar el auto para declarar no haber lugar a admitir en formato .pdf	El sistema permitirá visualizar el documento del auto para declarar no haber lugar a admitir en formato .pdf.
RF 95.	Mostrar trámite para declarar no haber lugar a continuar	El sistema permitirá mostrar el trámite para declarar no haber lugar a continuar en la pantalla del juez ponente.
RF 96.	Visualizar auto para declarar no haber lugar a continuar	El sistema permitirá visualizar el auto para declarar no haber lugar a continuar para su edición.
RF 97.	Editar auto para declarar no haber lugar a continuar	El sistema permitirá editar el auto para declarar no haber lugar a continuar.
RF 98.	Guardar auto para declarar no haber lugar a continuar	El sistema permitirá guardar el auto para declarar no haber lugar a continuar.
RF 99.	Visualizar el auto para declarar no haber lugar a continuar en formato .pdf	El sistema permitirá visualizar el documento del auto para declarar no haber lugar a continuar en formato .pdf.
RF 100.	Mostrar trámite para archivar el desistimiento	El sistema permitirá mostrar el trámite para archivar el desistimiento en la pantalla de la secretaria.
RF 101.	Visualizar providencia de archivo	El sistema permitirá visualizar la providencia de archivo para su edición.
RF 102.	Editar providencia de archivo	El sistema permitirá editar la providencia de archivo.

RF 103.	Guardar providencia de archivo	El sistema permitirá guardar la providencia de archivo.
RF 104.	Visualizar providencia de archivo en formato .pdf	El sistema permitirá visualizar la providencia de archivo en formato .pdf.
RF 105.	Mostrar trámite para resolver el desistimiento	El sistema permitirá mostrar el trámite para resolver el desistimiento en la pantalla del juez ponente.
RF 106.	Editar el auto de admisión del desistimiento	El sistema permitirá editar el auto de admisión del desistimiento.
RF 107.	Guardar el auto de admisión del desistimiento	El sistema permitirá guardar el auto de admisión del desistimiento.
RF 108.	Visualizar auto de admisión del desistimiento en formato .pdf	El sistema permitirá visualizar el auto de admisión del desistimiento en formato .pdf.
RF 109.	Editar el auto de rechazo del desistimiento	El sistema permitirá editar el auto de rechazo del desistimiento.
RF 110.	Guardar el auto de rechazo del desistimiento	El sistema permitirá guardar el auto de rechazo del desistimiento.
RF 111.	Visualizar auto de rechazo del desistimiento en formato .pdf	El sistema permitirá visualizar el auto de rechazo del desistimiento en formato .pdf.

**Fuente:** Elaboración propia

### **2.3.2 Requisitos no funcionales del sistema**

Los requisitos no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Suelen presentar dificultades en su definición dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir (Pressman, 2010).

El equipo de arquitectura del proyecto SITPC definió los requisitos no funcionales del sistema íntegro en el documento *“CEGEL Levantamiento de Información SITPC Especificación de requisitos de software.docx”*. Seguidamente se presenta un resumen de los mismos: (Roque Hernández, y otros, 2014)

**Usabilidad:**

- Se deben visualizar todos los mensajes en idioma español.

- La tipografía debe ser uniforme, de un tamaño adecuado. Debe tener un contraste adecuado que resalte los textos.

**Confiabilidad:**

- Debe estar disponible durante el horario laboral, efectuándose en períodos de tiempos definidos el proceso de actualización de la información del servidor local de base de datos con el servidor de la instancia superior.
- El sistema tendrá un tiempo de inactividad de 10 minutos y una vez que quede inactivo el usuario deberá autenticarse nuevamente.
- El sistema contará con una solución que será capaz de registrar trazas de error, de rendimiento y de acceso, con el objetivo de poder auditar el recorrido histórico de los recursos y los usuarios del sistema.
- Se debe configurar un sistema local de recuperación ante fallos del servidor que garantice una copia de la aplicación, así como del estado diario de la base de datos.

**Seguridad:**

- Se definirá una jerarquía de usuarios en correspondencia con las acciones que puedan realizar, estos podrán ser activados y desactivados por el administrador a través del sistema.
- Las políticas de acceso serán definidas por el administrador del sistema, los permisos serán limitados basados en los roles definidos y el usuario no podrá gestionarlos. Dichos permisos se harán corresponder con los niveles de acceso a la información que se define como clasificada.
- Se establecerán mecanismos de autenticación personalizada para todos los usuarios que harán uso de la aplicación mediante dirección o rango de direcciones IP autorizadas a hacer uso del sistema.

**2.4 Arquitectura de software**

La arquitectura para el diseño es una arquitectura en capas basada en el patrón arquitectónico Modelo-Vista-Controlador (MVC). A continuación, se describen cada una de las capas en la propuesta de solución.

**2.4.1 Vista**

En esta capa se encuentra todo lo que se refiere a la visualización de la información, el diseño, colores, estilos y la estructura visual de las páginas. En el desarrollo de la solución propuesta, y en SITPC en general, queda evidenciada esta capa a través del uso de los archivos JavaScript, HTML y

CSS, que en conjunto con la librería de JavaScript JQuery, el marco de trabajo Bootstrap y el motor de plantillas Twig, son los encargados de crear las páginas de la aplicación. Según la estructura de carpetas que trae definida Symfony, las vistas de la aplicación se encontraran en la carpeta “view” dentro de cada módulo.

### **2.4.2 Controlador**

Esta capa es la responsable de procesar y mandar a mostrar los datos obtenidos en la capa de acceso a datos, es decir, trabaja de intermediaria entre la capa de la vista y la capa de acceso a datos. En la solución informática propuesta el uso de esta capa se observa a través de las clases controladoras que son las encargadas de realizar el procesamiento de los datos y lógica de negocio asociada a las peticiones realizadas desde la capa de la vista. Según la estructura de carpetas que trae definida Symfony, las clases controladoras de la aplicación se encontraran en la carpeta “Controller” dentro de cada módulo.

### **2.4.3 Modelo**

En la arquitectura definida para el desarrollo del SITPC esta capa está dividida en dos subcapas; la de acceso a datos y la de abstracción de datos.

#### **Capa de acceso a datos**

Esta capa se encarga del manejo de la lógica de negocio, además sirve como intermediaria entre la capa controladora y el sistema gestor de base de datos. Las clases que componen esta capa son las gestoras y las clases repositorios, ubicadas en las carpetas Negocio y Repository respectivamente. Las clases gestoras reciben la información obtenida de la vista por medio de las controladoras y posteriormente se encargan de tratar dicha información en conjunto con las clases repositorios. Las clases repositorios contienen un conjunto de consultas para acceder y realizar acciones sobre los datos de tablas específicas de la base de datos. En esta capa también se encuentra el ORM Doctrine que posibilita la separación de la aplicación respecto al sistema gestor de base de datos mediante su lenguaje propio de consultas DQL.

#### **Capa de abstracción de datos**

Esta capa contiene las clases entidades, las cuales son una representación de las tablas de la base de datos mapeadas por el ORM Doctrine, de ahí que constituyen una abstracción de los datos reales almacenados en la base de datos. Las clases entidades se encuentran en la carpeta Entity en la carpeta ComunBundle.

## 2.5 Modelo de diseño

Esta etapa de desarrollo del software representa el enlace entre los requisitos y la implementación del sistema. El objetivo de esta etapa es crear prototipos o modelos que describan cómo deben interactuar las clases, para el posterior desarrollo de la aplicación.

### 2.5.1 Patrones GRASP

**Patrón Controlador:** El objetivo de este patrón es asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas, o sea, delega la responsabilidad en otras clases con las que mantiene un modelo de alta cohesión. Este patrón se evidencia en la solución a través de las clases controladoras que se encuentran en la carpeta Controller perteneciente a cada Bundle de la solución, y en la estructura de Symfony2 se evidencia en el uso del controlador frontal que se encuentra en la carpeta web de cada proyecto creado con dicho marco de trabajo. Para una mayor comprensión consultar la ilustración de la arquitectura en el capítulo 1 (Ver Ilustración 2).

**Patrón Experto:** Este patrón define que la clase experta en información es la que debe llevar la responsabilidad, ya que esta última es la que contiene toda la información para cumplir con la responsabilidad. Este patrón se observa en la solución al crear las vistas para mostrárselas al usuario; así como en las clases Controladoras que son las expertas en información y las encargadas de crear los formularios y las vistas que se le mostrarán al usuario. En la Ilustración 5 se señala un ejemplo al declarar el método *indexAction* de la clase controladora *RegistrarEscritoDesistimientoController*, cuando se crea el formulario y la vista con dicho formulario.

**Patrón Creador:** En la implementación del SITPC este patrón se evidencia en las clases Gestoras en el momento de crear nuevas instancias de las clases Entidades. Otro ejemplo son las clases Controladoras, que son las responsables de la creación de los formularios que se mostrarán en las vistas (Ver Ilustración 5).

```
public function indexAction($idTramite)
{
    $opcion = 1;
    $termino = 5;
    $documento = $this->getGestor()->getDocumentoTramite($idTramite);
    $tramite = $this->getGestor()->getTramiteXId($idTramite);
    $idExpediente = $tramite->getTramiteExpediente()->getExpediente()->getId();

    // cuando se ha creado anteriormente el documento
    if (isset($documento) && $documento != null) {...}

    $formulario = $this->createForm(new MotivoRechazoSubsanacionType());

    // Crear la vista
    return $this->render('EjecutivoBundle:Paginas\DisponerSobreEscritoDesistimiento:DisponerSobreEscritoDesistimiento.html.twig', array(
        'formulario' => $formulario->createView(),
        'idTramite' => $idTramite,
        'idExpediente' => $idExpediente,
        'opcion' => $opcion,

        'termino' => $termino,
        'materia' => $this->recuperar( llave: 'materiaActual'),
        'expediente' => $tramite->getTramiteExpediente()->getExpediente(),
        'seleccion' => 'Seleccion'
    ));
}
```

Ilustración 5: Método indexAction()

**Patrón Bajo acoplamiento:** Este patrón plantea la baja dependencia que debe existir entre las clases, además está estrechamente relacionado con los patrones Experto o Alta Cohesión. Symfony favorece ampliamente el bajo acoplamiento de las clases en el sistema, esto ocurre porque a cada clase se le asignan solamente las responsabilidades necesarias de manera que no dependan en gran medida de otras. Este patrón se observa en el uso del patrón arquitectónico MVC del SITPC ya que las clases de la Vistas no tienen relación con las clases gestoras y las entidades, que se refieren al Modelo, y estas últimas no presentan ninguna relación con las clases controladoras que representan el Controlador. Esto favorece la escalabilidad del sistema, la reutilización y el mantenimiento futuro de la aplicación. Para una mejor visión consultar la ilustración de la arquitectura en el capítulo 1 (Ver Ilustración 2).

**Patrón Alta cohesión:** Este patrón tiene como propósito asignar responsabilidades de manera que la cohesión siga siendo alta. El marco de trabajo Symfony favorece la alta cohesión asignando responsabilidades a las clases de tal manera que estas se encuentren estrechamente relacionadas entre sí y no lleguen a realizar un trabajo excesivo. En la implementación del SITPC este patrón se pone de manifiesto en la interrelación que existe entre las clases controladora, las clases gestora y las funcionalidades del sistema, donde cada una de estas presenta una clase controladora y una gestora, la primera encargada de manejar la lógica de presentación y el flujo de los datos provenientes de la vista y la segunda encargada de manejar la lógica del negocio de cada

funcionalidad. Para una mejor visión consultar la ilustración de la arquitectura en el capítulo 1 (Ver Ilustración 2).

### 2.5.2 Patrones GoF

**Patrón Método de fabricación (Factory Method):** Define una interfaz para crear un objeto, pero deja que sean las subclasses quienes decidan qué clase instanciar; su objetivo es devolver una instancia de múltiples tipos de objetos, que normalmente provienen de una misma clase padre y sólo se diferencian entre ellos por algún aspecto de comportamiento. En la implementación del SITPC este patrón es utilizado cuando se realiza una petición al contenedor de servicios, en la que se le pasa por parámetros el servicio que tiene definido el repositorio que se desea instanciar; internamente el contenedor de servicios determina cuál es la entidad que corresponde a dicho repositorio y realiza una llamada a la clase EntityManager.php, pasándole por parámetro la entidad calculada. Dicha clase es la encargada de implementar el patrón Factory Method, esto se realiza específicamente en el método getRepository(entityName). En la ilustración que se muestra a continuación se señala un ejemplo al declarar el método *obtenerExpedienteXId* de la clase gestora *RegistrarEscritoDesistimientoGtr*, cuando se accede al repositorio de la clase *Expediente* en la base de datos.

```
public function obtenerExpedienteXId($idexpediente)
{
    return $this->getEm()->getRepository('ComunBundle:AG\Expediente')->find($idexpediente);
}
```

**Ilustración 5:** Patrón Factory Method. Clase RegistrarEscritoDesistimientoGtr

**Patrón Decorador:** El principal objetivo de este patrón es añadir responsabilidades a objetos concretos de manera dinámica y transparente sin afectar a otros objetos. Este patrón brinda más flexibilidad que la herencia estática y evita que las clases más altas en la jerarquía estén demasiado cargadas de funcionalidad y sean complejas. En la implementación del SITPC este patrón se evidencia en el uso de una plantilla global para las vistas que decorará las demás páginas de la aplicación. En la ilustración que se muestra a continuación se señala un ejemplo al decorar la plantilla base del módulo Ejecutivo perteneciente a la materia Económico, la cual es decorada a partir de la plantilla base definida para todo el sistema con el nombre *plantilla\_SIT.html.twig*.



```
{% extends "ComunBundle::plantilla_SIT.html.twig" %}
{%block contenido%}
<style>
#wizard_btn_group {margin-left: 70%;}
#wizard_btn_group *{margin-right: 5px;}
</style>
<div id="div-contenido-wizard">
  {%block contenido_wizard%}
  !!! Contenido que va dentro del wizard
  {%endblock contenido_wizard%}
</div>
<div id="wizard_btn_group" class="btn-group">
  <a id="btn-wizard-ant" class="btn" style="...">Anterior</a>
  <a id="btn-wizard-sig" class="btn">Siguiete</a>
</div>
{%endblock contenido%}
```

Ilustración 6: Patrón Decorador. Clase plantilla\_Ejecutivo.html.twig

**Patrón Inyección de dependencias:** Este patrón es utilizado en los contenedores de servicios (o contenedores de inyección de dependencias), los cuales son objetos PHP que gestionan la creación de instancias de servicios, es decir, objetos. Esto se evidencia en la relación que se establece entre los gestores y los controladores. Cada controlador tiene asociado un gestor para realizar las acciones de acceso a datos. Estos gestores son publicados como servicios en el archivo “services.yml” y cuando una clase controladora necesita hacer uso de su gestor; en lugar de crear una instancia del mismo, simplemente llama a un método denominado getGestor() el cual accede al contenedor de servicios, que es el encargado de hacer la instancia del gestor solicitado, en tiempo de ejecución y de destruirla posteriormente cuando ya no la necesite (Ver Ilustración 7).

```
private function getGestor()
{
    if (!$this->container->has('economico.ejec.gtr.RegistrarEscritoDesistimiento')) {
        throw new \LogicException('Este servicio no esta registrado en la aplicacion');
    }
    return $this->container->get('economico.ejec.gtr.RegistrarEscritoDesistimiento');
}
```

Ilustración 7: Patrón Inyección de dependencias. Método getGestor()

Una vez abordados los elementos de la arquitectura del SITPC, a continuación, se muestran el diseño del diagrama de clases, de secuencia y de paquetes, mediante el uso del lenguaje UML v2.0.

### 2.5.3 Diagrama de clases del diseño

El Diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y las interfaces que participan en el sistema con sus relaciones estructurales y de herencia. Los diagramas de este tipo contienen las definiciones de las entidades del software en vez de conceptos del mundo real y son utilizados durante el proceso de análisis y diseño de los sistemas,

donde se crea una vista lógica de la información que se maneja en el sistema, los componentes que se encargarán del funcionamiento y la relación entre uno y otro (Larman, 2016).

Teniendo en cuenta que el proceso comienza a partir de la presentación de forma verbal o escrita del escrito de Desistimiento por una de las partes en el tribunal, a continuación se muestra el diagrama de clases del diseño para el requisito funcional “Registrar escrito de Desistimiento”:

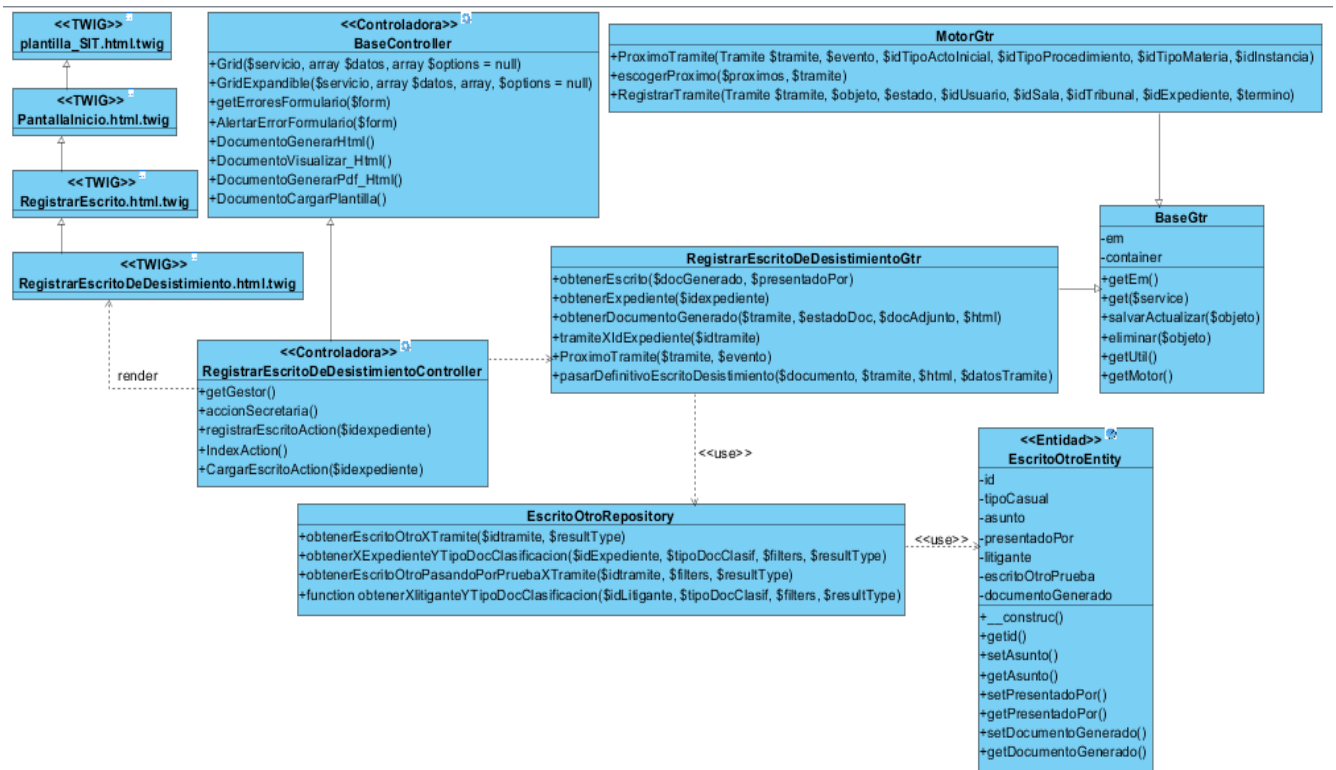


Ilustración 8: Diagrama de clase del diseño del RF “Registrar escrito de Desistimiento”

### 2.5.4 Diagrama de secuencia

El diagrama de secuencia muestra la secuencia de mensajes entre objetos durante un escenario concreto (paso de mensajes).

- En la parte superior aparecen los objetos que intervienen.
- La dimensión temporal se indica verticalmente (el tiempo transcurre hacia abajo).
- Las líneas verticales indican el período de vida de cada objeto.
- El paso de mensajes se indica con flechas horizontales u oblicuas (cando existe demora entre el envío y la atención del mensaje).

- La realización de una acción se indica con rectángulos sobre las líneas de actividad del objeto que realiza la acción (Network, 2018).

A continuación, se muestra del diagrama de secuencia para el requisito funcional tomado como base en la presente investigación:

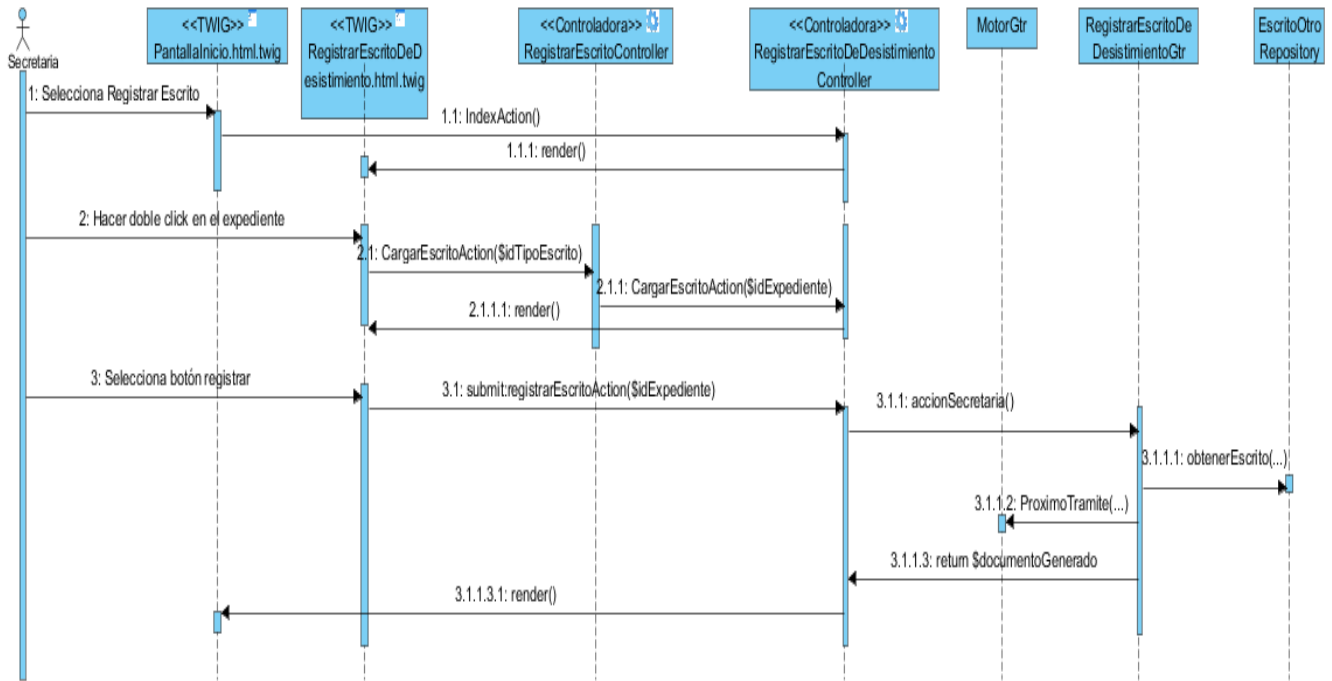


Ilustración 9: Diagrama de secuencia del RF “Registrar escrito de Desistimiento”

### 2.5.5 Diagrama de despliegue

Este tipo de diagrama se realiza con el objetivo de mostrar las relaciones físicas de los distintos nodos que componen un sistema y la distribución de los componentes sobre dichos nodos. Además, se modela la arquitectura en tiempo de ejecución de un sistema (Larman, 2016).

Mediante un análisis íntegro por parte del equipo de arquitectura se definió como quedaría estructurado el SITPC en los TMP, así como su comunicación con las instancias superiores (TPP o TSP), a continuación, se muestra el resultado:

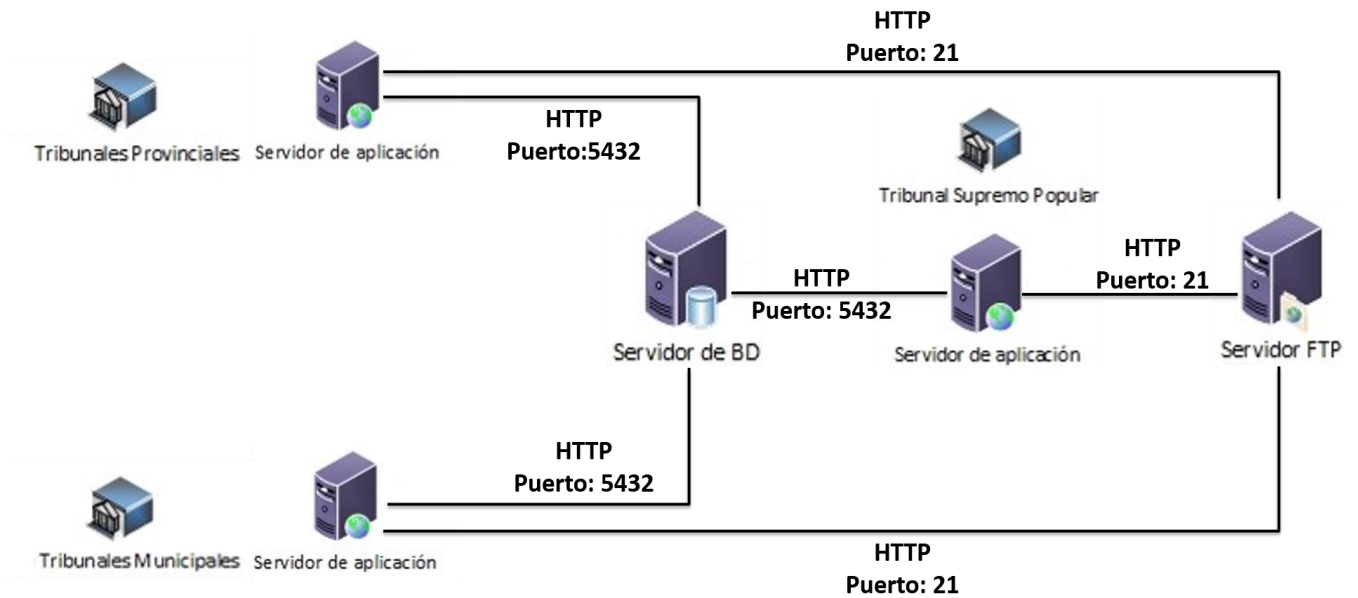


Ilustración 10: Diagrama de despliegue del SITPC

### 2.5.6 Modelo de datos

El modelo de datos se utiliza para describir la estructura lógica y física de la información persistente gestionada por el sistema. Se utiliza para definir la correlación entre las clases de diseño persistentes y las estructuras de datos persistentes, y para definir las estructuras de datos persistentes (Lucidchart, 2018).

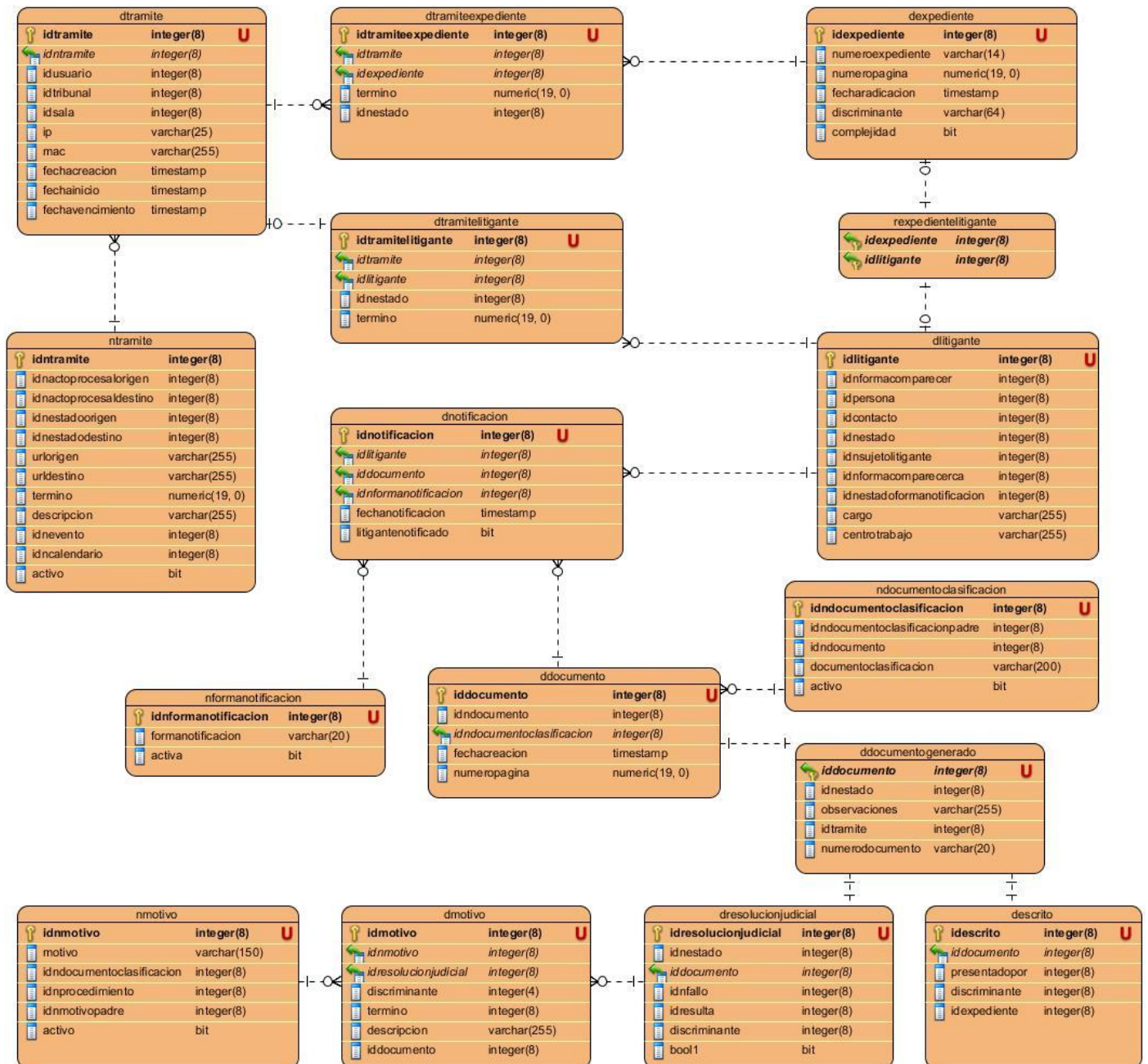


Ilustración 11: Modelo de datos para el subproceso Desistimiento

### 2.5.7 Estándares de codificación

Los estándares de codificación son normas que deben seguirse durante todo el ciclo de implementación del sistema (Ohmyroot, 2017).

Los mismos fueron seleccionados por el equipo de arquitectura y diseño del proyecto, a continuación, se describen dichos estándares:

### (1) Cabecera del archivo

Todos los archivos.php deben iniciar con una cabecera específica que muestre información del autor del código y fecha de los últimos cambios realizados, entre otros aspectos. Decidir agregar o no más datos depende de cada módulo.

```
/**
 * Description of CrearANHLAController
 * @FechaCreacion 13 de Marzo 2018
 * @FechaUltimaModificacion 14 de Marzo 2018
 * @author Yudiel
 */
```

**Ilustración 12:** Estándar de codificación. Cabecera de archivo

### (2) Indentación <sup>11</sup>

Es importante que el contenido siempre se indente con tabulaciones nunca utilizando espacios en blanco, para mejorar la legibilidad del código fuente por parte de los programadores y delimitar la estructura del programa permitiendo establecer bloques de código (Ver Ilustración 13).

### (3) Comentarios

Todo método debe ser comentado con el objetivo de describir la funcionalidad que realiza, sin necesidad de que el programador ajeno al código tenga que estudiarlo a profundidad para su comprensión. Además, debe ofrecer información referente a los valores y parámetros que debe devolver cada método comentado (Ver Ilustración 13).

### (4) Denominación de archivos

Las variables y funciones deben comenzar con letra minúscula, en el caso de las clases y archivos con letra mayúscula. En caso que sean palabras compuestas, estas no deben exceder de cuatro cadenas de letras, siguiendo además la notación Camello (Ver Ilustración 13).

---

<sup>11</sup> Espacio en blanco al inicio de línea o sangría antes del código.

```
// Permite capturar los datos a partir de los parametros $idTramite, $idExpediente, $opcion, $termino, $darTraslado
private function capturarDatos($idTramite, $idExpediente, $opcion, $termino = null, $darTraslado )
{
    $escritoContestacionDemanda = $this->getGestor()->buscarEscritoXClasificacion($idExpediente, ETipoEscrito::Contestacion_Demanda);
    $contestacion = 0;
    switch ($opcion) {
        case 1:
            if($darTraslado != null){
                return $this->getGestor()->buscarDatosProvidenciaTraslado($idExpediente);
            }
            else{
                if (isset($escritoContestacionDemanda) || $escritoContestacionDemanda != null) {
                    $contestacion=1;
                }
                return $this->getGestor()->buscarDatosAutoDefinitivoAdmision($idTramite, $contestacion, $idExpediente);
            }
            break;
        case 2:
            return $this->getGestor()->buscarDatosDocumentoSubsanacion($termino, $idExpediente);
        case 3:
            return $this->getGestor()->buscarDatosDocumentoRechazo($idExpediente);
        default:
            break;
    }
}
```

Ilustración 13: Estándar de codificación. Método capturarDatos

### (5) Denominación de las clases

Las clases gestoras deben llamarse igual que las clases controladoras, solo se diferenciarán por la terminación de cada nombre que identifica el tipo de clase al que se hace referencia, ejemplo: *RegistrarEscritoDesistimientoGtr*, se refiere a la clase gestora del requisito funcional “Registrar escrito de Desistimiento”.

### (6) Denominación de los servicios

Es importante que los parámetros estén en minúscula utilizando como separador el carácter punto (.); igual para los servicios, pero sin el sufijo “.class”. La estructura que se propone para los parámetros es: materia.procedimiento.gestor|tableModel.categoria.class.

```
economico.ejec.gtr.RegistrarEscritoDesistimiento:
class: SIT\ECONOMICO\EjecutivoBundle\Negocio\Gestor\RegistrarEscritoDesistimientoGtr
parent: arquitectura.basegtr
```

Ilustración 14: Estándar de codificación. Denominación de los servicios

### (7) Denominación de las rutas

Las rutas se deben escribir en minúscula utilizando como separador el carácter guión bajo (\_) de la siguiente forma: materia\_procedimiento\_controlador\_accion. El patrón de la ruta debe tener relación con la acción que se llama y ser lo más entendible posible, en caso de que el nombre esté compuesto por más de una palabra serán separadas por el signo de menos (-) y no debe ponerse la materia ni el procedimiento en el mismo.

```
##### Inicio Registrar escrito de desistimiento (alejandro) #####  
economico_ejec_escritoDesistimiento_index:  
  pattern: /desistimiento-ejec/registrar-escrito/index  
  defaults: {_controller: EjecutivoBundle:RegistrarEscritoDesistimiento:index}
```

**Ilustración 15:** Estándar de codificación. Denominación de las rutas

**Otras consideraciones:**

- Las constantes deben de escribirse siempre en mayúsculas y tanto estas como las variables globales, deben de tener como prefijo el nombre de la clase a la que pertenecen.
- Las llaves de apertura irán al final de la sentencia que delimitan, y las llaves de cierre alineadas con el inicio de la sentencia en una nueva línea.
- Los nombres deben ser descriptivos y concisos.
- No usar ni grandes frases ni pequeñas abreviaciones para las variables, de manera que con solo conocer su nombre se sepa la función que realizan. Esto aplica para los nombres de variables, funciones, argumentos de funciones y clases.

**Conclusiones del capítulo**

La descripción del proceso de negocio permitió lograr un mayor entendimiento del negocio a informatizar y sentar las bases para la disciplina de Requisitos. Por su parte, la especificación de requisitos ayudó a obtener una visión de las funcionalidades al personalizar el subproceso de Desistimiento. Por último, la descripción de la arquitectura del sistema y el modelado del sistema, facilitó establecer una guía para lograr la correcta personalización del módulo Desistimiento en las instancias provinciales, teniendo siempre en cuenta los patrones de diseño definidos, así como los estándares de implementación.



*CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN PROPUESTA*

En este capítulo se evalúa el grado de calidad y fiabilidad de los resultados obtenidos en el desarrollo de este trabajo luego de haber personalizado el subproceso de Desistimiento. Esta evaluación se lleva a cabo a partir de la validación del diseño a través de las métricas: Tamaño Operacional de las Clases y Relación entre Clases. Por último, se aplican pruebas de software para verificar y revelar la calidad del producto software antes de ser entregado al cliente.

**3.1 Validación del diseño**

Para validar el diseño realizado se aplicaron un conjunto de métricas. Los resultados obtenidos por cada una de las métricas utilizadas se muestran a continuación.

**3.1.1 Métrica Tamaño Operacional de las Clases (TOC)**

Las métricas basadas en el Tamaño Operacional de las Clases se basan en contar la cantidad de atributos y la cantidad de operaciones que tiene una clase individual y el promedio que presenta el sistema en su totalidad. Se evalúa a partir de los siguientes atributos de calidad:

- Responsabilidad
- Complejidad de implementación
- Reutilización

Una vez analizado el indicador tamaño de clase, si el valor resultante tiende al crecimiento, es probable que la clase posea un alto grado de Responsabilidad; en consecuencia, el nivel de Reutilización sería mínimo y la Implementación altamente compleja. Para medir los atributos de calidad se definieron los umbrales que se muestran en la Tabla 2. En la misma se emplean la abreviatura Prom (promedio).

**Tabla 2:** Categoría por atributos y criterio de evaluación. Métrica TOC.

Atributo de calidad	Categoría	Criterio
Responsabilidad	Baja	TOC < =Promedio (Prom).
	Media	TOC Entre Prom. y 2* Prom.
	Alta	TOC >2*Prom.
Complejidad de implementación	Baja	TOC < =Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	TOC >2*Prom.
Reutilización	Baja	TOC >2*Prom.

*CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN PROPUESTA*

	Media	TOC Entre Prom. y 2* Prom.
	Alta	TOC < =Prom.

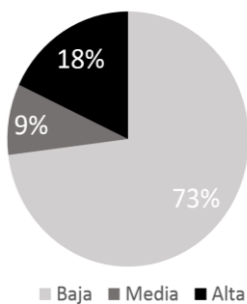
La Tabla 3 muestra las medidas de los parámetros de calidad obtenidas luego de aplicar la métrica TOC al diseño de clases del requisito que se toma como ejemplo en la presente investigación. En la misma se utilizaron las abreviaturas **Rp** (Responsabilidad), **CI** (Complejidad de implementación) y **R** (Reutilización).

**Tabla 3:** Resultados obtenidos luego de aplicada la métrica TOC

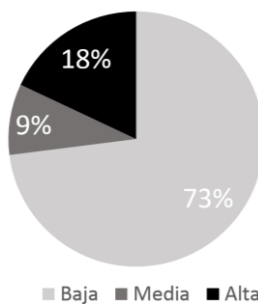
No.	Clase	Cantidad de Procedimientos	Rp	C	R
1	RegistrarEscritoDesistimientoController	8	Baja	Baja	Alta
2	RegistrarEscritoDesistimientoGtr	12	Media	Media	Media
3	BaseController	23	Alta	Alta	Baja
4	EscritoOtroRepository	36	Alta	Alta	Baja
5	EscritoOtroEntity	5	Baja	Baja	Alta
6	MotorGtr	6	Baja	Baja	Alta
7	BaseGtr	8	Baja	Baja	Alta
8	Plantilla_SIT.html.twig	4	Baja	Baja	Alta
9	PantallaInicio.html.twig	4	Baja	Baja	Alta
10	RegistrarEscrito.html.twig	2	Baja	Baja	Alta
11	RegistrarEscritoDesistimiento.html.twig	3	Baja	Baja	Alta

Al aplicar la métrica TOC a estas 11 clases y después de haber estudiado los resultados se obtuvieron para cada atributo de calidad los siguientes resultados:

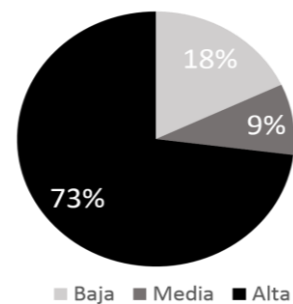
Complejidad de implementación



Responsabilidad



Reutilización



**Ilustración 16:** Resultado de la métrica TOC

Después de evaluar los resultados obtenidos de cada uno de los atributos de calidad que mide esta métrica, se obtuvieron los siguientes resultados:

- El 73% de las clases poseen una baja responsabilidad y complejidad de pruebas.
- El 73% de las clases poseen una alta reutilización.

Estos resultados demuestran la calidad del diseño de la solución propuesta, ya que, al obtener bajos índices de responsabilidad y complejidad, unidos a un alto grado de reutilización de las clases se facilita en gran medida la implementación de las mismas.

### **3.1.2 Métrica Relación entre clases (RC)**

El resultado de la aplicación de la métrica RC está dado por el número de relaciones de uso que se establecen entre una clase y las demás clases existentes. Se evalúa a partir de los siguientes atributos de calidad:

- Acoplamiento: Un aumento del RC implica un aumento del acoplamiento de la clase.
- Complejidad de Mantenimiento: Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
- Reutilización: Un aumento del RC implica una disminución en el grado de reutilización de la clase.
- Cantidad de Pruebas: Un aumento del RC implica un aumento de la cantidad de pruebas necesarias para probar una clase (Pressman, 2010).

Para obtener un nivel óptimo de relación entre clases, el valor obtenido al aplicar dicha métrica debe ser directamente proporcional al acoplamiento y a la complejidad de mantenimiento; además debe ser inversamente proporcional al nivel de reutilización del código. Para aplicar la métrica RC es necesario categorizar cada una de las clases según la cantidad de relaciones que esta contenga.

En la Tabla 4 se muestran las categorías para clasificar cada uno de los atributos de calidad anteriormente mencionados, así como el criterio de evaluación. En la misma se emplean la abreviatura Prom (promedio).

**Tabla 4:** Categoría por atributos y criterio de evaluación. Métrica RC.

<b>Atributo</b>	<b>Categoría</b>	<b>Criterio</b>
Acoplamiento	Ninguno	RC = 0
	Bajo	RC = 1
	Medio	RC = 2
	Alto	RC > 2

Complejidad de Mantenimiento	Baja	$RC \leq Prom$
	Media	$Prom \leq RC \leq 2 * Prom$
	Alto	$RC > 2 * Prom$
Reutilización	Baja	$RC > 2 * Prom$
	Media	$Promedio < RC \leq 2 * Promedio$
	Alto	$RC \leq Promedio$
Cantidad de pruebas	Baja	$RC \leq Promedio$
	Media	$Promedio \leq RC < 2 * Promedio$
	Alto	$RC \geq 2 * Promedio$

La Tabla 5 muestra las medidas de los parámetros de calidad obtenidas luego de aplicar la métrica RC al diseño de clases del requisito que se toma como ejemplo en la presente investigación. En la misma se utilizaron las abreviaturas **Ac** (Acoplamiento), **CM** (Complejidad de mantenimiento) y **R** (Reutilización) y **CP** (Cantidad de prueba).

**Tabla 5:** Resultados obtenidos luego de aplicada la métrica RC

No.	Clase	Cantidad de Relaciones de Uso	Ac	CM	R	CP
1	RegistrarEscritoDesistimientoController	3	Alto	Alto	Media	Alto
2	RegistrarEscritoDesistimientoGtr	3	Alto	Alto	Media	Alto
3	BaseController	1	Bajo	Bajo	Alta	Bajo
4	EscritoOtroRepository	1	Bajo	Bajo	Alta	Bajo
5	EscritoOtroEntity	1	Bajo	Bajo	Alta	Bajo
6	MotorGtr	1	Bajo	Bajo	Alta	Bajo
7	BaseGtr	2	Medio	Medio	Media	Medio
8	Plantilla_SIT.html.twig	1	Bajo	Bajo	Alta	Bajo
9	PantallaInicio.html.twig	2	Medio	Medio	Media	Medio
10	RegistrarEscrito.html.twig	2	Medio	Medio	Media	Medio
11	RegistrarEscritoDesistimiento.html.twig	2	Medio	Medio	Media	Medio

Después de haber aplicada la métrica, se tomaron todos los resultados obtenidos individualmente y se agruparon para ser analizados de manera general, promediando los valores obtenidos por categoría en cada atributo. En la Ilustración 17 se muestran estos resultados.

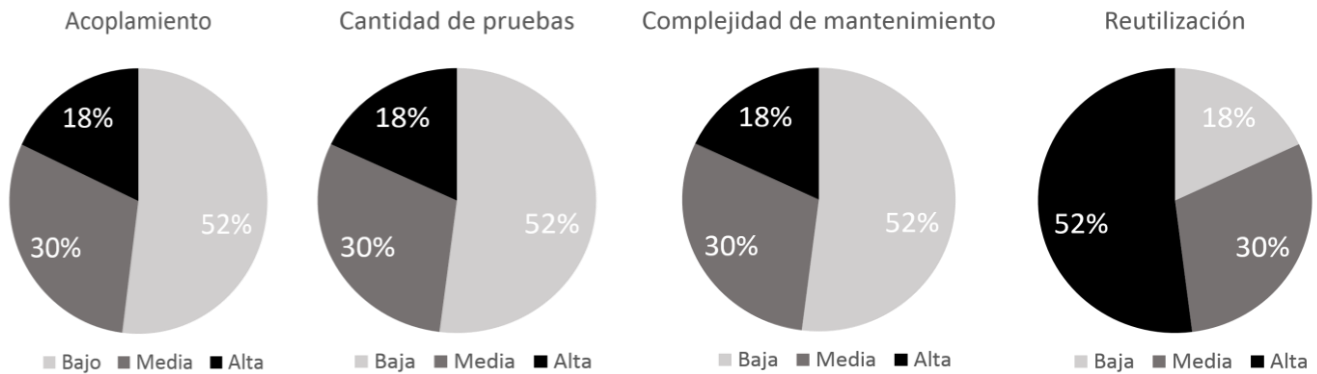


Ilustración 17: Resultado de la métrica RC

Después de evaluar los resultados obtenidos de cada uno de los atributos de calidad que mide esta métrica, se obtuvieron los siguientes resultados:

- El 52% de las clases tienen un bajo acoplamiento, baja complejidad de mantenimiento y un porcentaje bajo de cantidad de pruebas.
- El 52% de las clases poseen una alta reutilización.

Como se puede observar en la figura el acoplamiento posee un nivel bajo por lo que existe poca dependencia entre las clases trayendo como consecuencia una alta probabilidad de reutilización. También se puede apreciar que existe un bajo nivel de complejidad de mantenimiento por lo que a la hora de optimizar métodos y demás operaciones no es necesario realizar una gran cantidad de pruebas.

### 3.2 Pruebas de software

Procesos que permite verificar y revelar la calidad de un producto software antes de ser entregado al cliente. Básicamente, es una fase en el desarrollo de software que consiste en probar las aplicaciones construidas. Es importante mencionar que una campaña de pruebas de software extensa no garantiza con plena seguridad que el sistema estará limpio de errores y que siempre funcionará según las especificaciones.

### 3.2.1 Pruebas unitarias

Las pruebas unitarias se realizan sobre las funcionalidades internas de un módulo y se encargan de comprobar los caminos lógicos, ciclos (bucles) y condiciones que debe cumplir el programa (Pressman, 2010).

Para aplicar las pruebas unitarias, los autores de la presente investigación definen utilizar el método de Caja blanca.

Con el empleo de este método es posible desarrollar casos de prueba que garanticen la ejecución, al menos una vez, de los caminos independientes<sup>12</sup> (Pressman, 2010). Para aplicar este método se define la técnica de ruta básica.

#### Técnica de ruta básica

La técnica de ruta básica es empleada en el método de Caja blanca, la misma tiene como objetivo comprobar que cada camino se ejecute independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Esta técnica de prueba debe ser utilizada para evaluar la efectividad de los métodos asociados a una clase, con el objetivo de asegurar que cada camino independiente sea ejecutado por lo menos una vez en el sistema (Pressman, 2010).

Pressman propone como estrategia para aplicar la ruta básica, realizar un análisis de la complejidad ciclométrica de cada procedimiento que componen las clases del sistema; una vez concluido este paso se selecciona el método con valor de contener errores, además de que ofrece una medida del número de pruebas que deben diseñarse para validar la correcta implementación de una determinada función. Esta métrica se calcula sobre un grafo y se puede realizar mediante tres formas distintas:

1.  $V(G) = R$
2.  $V(G) = E - N + 2$
3.  $V(G) = P + 1$

Conociendo que:

- **G**: Grafo de flujo (grafo)
- **R**: El número de regiones contribuye a estimar el valor de la complejidad ciclométrica.
- **E**: Número de aristas
- **V(G)**: Complejidad ciclométrica
- **N**: Número de nodos del grafo
- **P**: Número de nodos predicados<sup>13</sup> incluidos en el grafo.

---

<sup>12</sup> Camino que recorre por lo menos una nueva arista en el grafo de flujo.

Una vez calculada la complejidad ciclomática, el valor obtenido representa el límite superior de pruebas que deberán aplicarse. (Pressman, 2010)

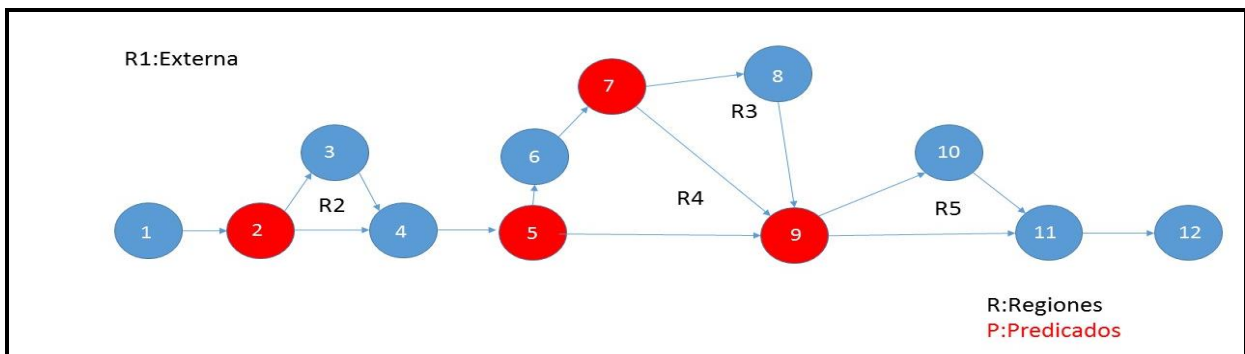
En la Ilustración 18 se muestra el método pasarDefinitivo() del Requisito Disponer sobre escrito de Desistimiento. Se selecciona el mismo porque es el método principal de este requisito, ya que es la acción donde se pasa a definitivo la disposición del juez generando el documento, continuando así el proceso.

```

public function pasarDefinitivo($datos, $idUserario, $pdf, $documento) {
    $tramite = $this->getTramiteId($datos['idTramite']);
    $expediente = $this->getExpedienteId($datos['idExpediente']);
    $tramiteExp = $tramite->getTramiteExpediente();
    $estado = EEstado::TramiteExp_Esperando_Dias;
    if ($documento->getTipoDocumentoClasificacion()->getId() == ETipoResolucionAuto::Rechazo_Desistimiento
        || $documento->getTipoDocumentoClasificacion()->getId() == ETipoResolucionAuto::Definitivo_Desistimiento) {
        $estado = EEstado::TramiteExp_Cerrado;
    }
    $tramite_nuevo = $this->insertarTramite($datos['idTramite'], $datos['evento'], $idUserario, $expediente, $estado, $expediente->getId(),
        $datos['termino']);
    $this->salvarActualizar($tramite_nuevo);
    $tramiteExp->setTipoEstado($this->getTipoEstado( idTipoEstado: EEstado::TramiteExp_Cerrado));
    $this->salvarActualizar($tramiteExp);
    $this->pasarDefinitivoDocumento($documento, $pdf, $tramite_nuevo);
    if ($documento->getTipoDocumentoClasificacion()->getId() == ETipoResolucionProvidencia::Traslado_Escrito_Desistimiento) {
        $fiscal = $this->getEm()->getRepository('ComunBundle:AG\Litigante')
            ->obtenerXFormaComparecerYExpediente(EFormaComparecer::Fiscal, $expediente->getId());
        if (!isset($fiscal) && $fiscal != null) {
            $tramite_nuevo_fiscal = $this->insertarTramite($datos['idTramite'],
                evento: EEvento::Admision_Crear_Oficio_Personeria, $idUserario, $expediente,
                tipoEstado: EEstado::TramiteExp_Resuelto, $expediente->getId());
            $this->getEm()->persist($tramite_nuevo_fiscal);
        }
    }
    if ($documento->getTipoDocumentoClasificacion()->getId() == ETipoResolucionAuto::Definitivo_Desistimiento) {
        $this->cerrarExpediente($tramite_nuevo);
    }
    $this->get('comunes.notificacion.gtr')->notificar($expediente, $documento);
    $this->getEm()->flush();
    return $documento;
}
    
```

**Ilustración 18:** Método pasarDefinitivo() utilizado para aplicarle la prueba de caja blanca

A continuación, se realiza el grafo de flujo para el método previamente descrito:



**Ilustración 19:** Grafo de flujo a partir del método pasarDefinitivo()

<sup>13</sup> Está caracterizado porque dos o más aristas emergen de él.

A dicho grafo de flujo se le aplicó la métrica de complejidad ciclomática, calculada por las tres vías conocidas, obteniéndose los siguientes resultados:

- $V(G) = R = 5$ .
- $V(G) = 15 - 12 + 2 = 5$ .
- $V(G) = 4 + 1 = 5$ .

Después de calcular la complejidad del grafo, se pudo comprobar que los resultados obtenidos son iguales a 5; por tanto, se deben realizar 5 casos de pruebas, uno por cada ruta independiente. Las rutas independientes resultantes fueron:

- 1) 1-2-3-4-5-9-11-12
- 2) 1-2-4-5-9-10-11-12
- 3) 1-2-3-4-5-6-7-8-9-10-11-12
- 4) 1-2-4-5-9-11-12
- 5) 1-2-3-4-5-9-10-11-12

**Tabla 6:** Caso de prueba de la ruta independiente 1

<b>Descripción</b>	El escrito sobre el que se dispone se pasa a definitivo.
<b>Condición de ejecución</b>	El juez selecciona la opción de Rechazo al disponer sobre el escrito de Desistimiento.
<b>Entradas</b>	Se le asocia el escrito, el identificador del usuario, el escrito en formato pdf y los documentos adjuntos.
<b>Resultados esperados</b>	El escrito se pasa a definitivo y se notifica dicho evento.

El resto de los casos de pruebas obtenidos a raíz del cálculo de la complejidad ciclomática se pueden consultar en los Anexos 2, 3, 4 y 5 correspondiente a las rutas básicas 2, 3, 4 y 5.

### **Resultados al aplicar la técnica de ruta básica**

Esta técnica se aplicó a los métodos de las clases controladoras; dichas clases fueron seleccionadas debido a que engloban las funcionalidades del sistema. Para comprobar la fiabilidad del código fuente se realizaron dos iteraciones completas en busca de errores de codificación, como condición de parada se tuvo en cuenta el criterio de estimación anteriormente expuesto. Se realizó una primera iteración donde se obtuvo como resultado 0 cantidad de No conformidades, por lo que se puede concluir que luego de realizar la prueba de caja blanca, donde se diseñaron y ejecutaron los casos de prueba correspondientes, se logró asegurar el cumplimiento del proceso de mejora del código.



### **3.2.2 Pruebas funcionales**

Las pruebas funcionales son pruebas diseñadas tomando como referencia las especificaciones funcionales de un componente o sistema (lo que vamos a testear, el software o una parte de él). Se realizan para comprobar si el software cumple las funciones esperadas. (Pressman, 2010)

Para llevar a cabo estas pruebas se tuvo en cuenta el método de Caja negra que a continuación se describe.

#### **Método de Caja negra**

El método de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La misma no es una alternativa a las técnicas de método de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca. Estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación (Pressman, 2010).

Para llevar a cabo el método de Caja negra se utiliza la técnica de Partición equivalente que a continuación se describe.

#### **Técnica de prueba: Partición equivalente**

Esta técnica divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. El método se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada (Pressman, 2010).

Para aplicar esta técnica, se deben primeramente realizar el Diseños de casos prueba (DCP) con el objetivo de obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software.

Un DCP es, en ingeniería del software, un conjunto de condiciones o variables bajo las cuales un analista determinará si una aplicación o una característica de éstos es parcial o completamente satisfactoria (Pressman, 2010).

Como primero paso en el DCP, se muestra en el Anexo 6 la descripción de las variables para el caso del requisito “Registrar escrito de Desistimiento”. Luego de obtener la descripción de cada una de estas variables, se procede a realizar cada uno de los escenarios a probar en el sistema. Para el caso del requisito en cuestión, en el Anexo 7 se muestran estos escenarios.

Para aplicar el DCP a la solución, se efectuaron un total de 3 iteraciones para poder alcanzar resultados satisfactorios atendiendo al correcto comportamiento del sistema ante diferentes situaciones. En la Ilustración 20 se muestra una gráfica con el número total de No conformidades (NC) por iteración y en la Ilustración 21, un resumen según su clasificación por cada iteración.

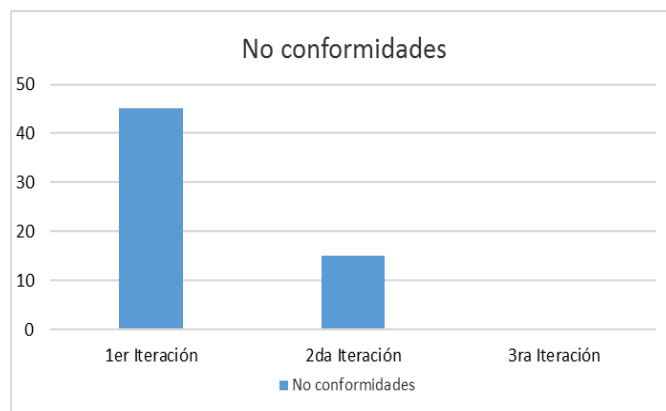


Ilustración 20: Total de NC detectadas por iteración

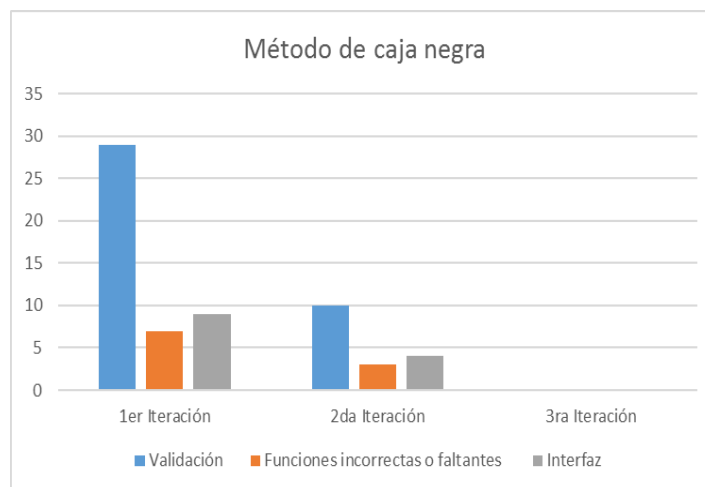


Ilustración 21: Total de NC por tipo de clasificación

### 3.2.2 Pruebas de rendimiento

Las pruebas de rendimiento se diseñan para poner a prueba el rendimiento del software en tiempo de corrida, dentro del contexto de un sistema integrado. Estas ocurren a lo largo de todos los pasos del

proceso de prueba. Estas requieren instrumentación de hardware y de software, es decir, con frecuencia es necesario medir la utilización de los recursos, por ejemplo: ciclos del procesador. La instrumentación externa puede monitorear intervalos de ejecución y eventos de registro conforme ocurren, y los muestreos del estado de la máquina de manera regular (Pressman, 2010).

Para evaluar el tiempo de respuesta del sistema durante las condiciones de carga esperada, y así corroborar la agilización en el subproceso de Desistimiento, los autores de la presente investigación definen el empleo de la herramienta automatizada JMeter que a continuación se describe.

### **JMeter**

JMeter es un software de código abierto diseñado en Java que permite realizar pruebas de rendimiento y de funcionalidad sobre aplicaciones tipo cliente/servidor escritas en cualquier lenguaje. Se puede utilizar para simular una carga pesada en un servidor, la red o un objeto para poner a prueba su resistencia o para analizar el rendimiento global en diferentes tipos de carga. Puede usarlo para hacer un análisis gráfico de rendimiento o para probar el comportamiento de diferentes elementos con un gran volumen de carga y concurrencia (Apache-JMeter.tm, 2018).

A continuación, se muestra el resultado al ejecutar las pruebas de Rendimiento al registrar un escrito de Desistimiento para evaluar el tiempo de respuesta utilizando esta herramienta, el resto de los resultados se muestran en los anexos 8, 9, 10 y 11.

Label	# Muestras	Media	Mín	Máx	Std. Dev.	% Error	Rendimiento	Kb/sec	Avg. Bytes
/servicios/comunes/buscar-expediente-comun	5	106	69	166	43,36	0,00%	5,2/sec	45,02	8897,2
/servicios/comunes/resultado-busqueda-expedien...	5	114	54	182	52,82	0,00%	4,7/sec	41,62	9145,0
/servicios/comunes/obtener-resultado-busqueda-...	5	131	50	193	56,34	0,00%	4,3/sec	39,27	9304,4
/servicios/comunes/escrito/registrar/1264	5	117	79	145	22,58	0,00%	4,1/sec	35,42	8937,4
/servicios/comunes/dpaPorPadre	10	134	77	170	27,85	0,00%	7,0/sec	61,15	8985,0
/servicios/comunes/buscar-fiscal-cargar-fiscal	5	122	76	206	48,14	0,00%	3,7/sec	32,95	9090,8
/servicios/comunes/escrito/ presentadoPor/1264	5	125	82	172	31,12	0,00%	3,7/sec	32,84	9113,4
/servicios/comunes/escrito/tipoEscrito/126119	5	106	68	179	43,82	0,00%	3,8/sec	34,44	9325,2
/servicios/comunes/escrito/cargarEscrito/1264/1015	5	88	56	133	28,47	0,00%	3,8/sec	34,36	9163,2
/servicios/comunes/escrito/rutaEnvioSecretaria/12...	5	87	42	150	43,19	0,00%	4,0/sec	36,22	9242,0
/economico/ejecutivo/desistimiento-ejec/registrar-...	5	81	43	157	40,29	0,00%	4,4/sec	39,97	9323,6
TOTAL	60	112	42	206	44,01	0,00%	28,8/sec	256,83	9126,0

**Ilustración 22:** Tiempo de respuesta al registrar escrito de Desistimiento

### **3.3 Validación de las variables de la investigación**

Teniendo en cuenta la idea a defender que se plantea en la presente investigación, para analizar la relación causa efecto entre la variable independiente “la personalización del subproceso Desistimiento en el SITPC” y la variable dependiente “agilización de la ejecución del mismo en las instancias provinciales de los tribunales cubanos”, los autores de la presente investigación definen un conjunto de criterios de medida que permiten validar cómo a través de la propuesta de solución se

logra la relación entre ambas variables. La obtención de estos criterios se realiza a partir de las principales deficiencias identificadas en la situación problemática y de los resultados de la encuesta realizada a los especialistas funcionales judiciales de los TPC.

**Criterios de medida definidos:**

- ✓ Búsqueda de los expedientes.
- ✓ Captura de los datos al registrar los escritos del subproceso de Desistimiento.
- ✓ Notificación de las tramitaciones que se realizan.
- ✓ Generación de los documentos.
- ✓ La consulta de los expedientes de forma simultánea
- ✓ Las notificaciones acerca de los vencimientos de los términos

A continuación, en la Tabla 7, se evalúan cada uno de los criterios de medida definidos. La evaluación se realiza a través de un antes (SITPC sin la personalización del proceso de Desistimiento) y un después (SITPC con la personalización del proceso de Desistimiento), con el propósito de verificar cómo mediante la solución propuesta se contribuye a la agilización de la ejecución del mismo en las instancias provinciales de los tribunales cubanos. Los datos tenidos en cuenta en la comparación fueron tomados a partir de la realización de las pruebas de Rendimiento, comparando el tiempo de respuestas del sistema con el tiempo promedio que se demoran los especialistas para cada una de las actividades que responden a los criterios de medida.

**Tabla 7:** Validación de las variables de investigación

<b>Atributo</b>	<b>Antes</b>	<b>Después</b>
Búsqueda de los expedientes	Se dificulta la búsqueda del expediente ya que los documentos que se generan al crearse un subproceso de Desistimiento se archivarían en los estantes de los tribunales, dificultando la búsqueda y afectando la agilización del subproceso en cuestión. A raíz de la encuesta el 70% de estos especialistas plantean que se demoran entre 20 y 30 minutos para	La búsqueda de los expedientes se realiza de forma segura y rápida a través del sistema, permitiendo la completitud en el manejo de la información estadística al tener todos los documentos de los expedientes en formato digital. Esta actividad utilizando el SITPC se comprobó que se disminuyó el tiempo a 39 segundos

*CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN PROPUESTA*

	realizar esta actividad.	aproximadamente.
Captura de los datos al registrar los escritos del subproceso de Desistimiento	Según los resultados de la encuesta, se determinó que al registrar los escritos en el subproceso de forma manual se consume un tiempo elevado ya que, el 60% de estos especialistas plantean que se demoran entre 10 y 15 minutos para realizar esta actividad.	Se disminuye considerablemente el uso del papel al llevar el expediente solo de forma manual. Solo demora alrededor de 4 minutos en realizar un trámite del subproceso en el sistema. Esta actividad utilizando el SITPC se comprobó que se disminuyó el tiempo a 29 segundos aproximadamente.
Notificación de las tramitaciones (autos, escritos y providencias)	A raíz de la encuesta el 60% de estos especialistas plantean que se tardan entre 1 a 3 días para realizar esta actividad.	Esta actividad utilizando el SITPC se comprobó que se disminuyó el tiempo a 35 segundos aproximadamente.
Generación de los documentos	Los documentos que se generan en el subproceso están expuesto a tachaduras, borrones, debido al tratamiento manual. Además, no todos contienen el mismo formato según el tipo de clasificación del documento. El 77% de los especialistas encuestados plantean que para generar los documentos tardan más de 30 minutos.	El sistema genera todos los documentos en formato pdf y además los usuarios manipulan los documentos de manera digital por lo que se eliminan las tachaduras y los borrones. Esta actividad utilizando el SITPC se comprobó que se disminuyó el tiempo a 35 segundos aproximadamente.
Consulta de los documentos en el expediente	La consulta de los expedientes de forma simultánea no es posible, por lo que, si un expediente se encuentra entregado a una de las partes, en caso de que la otra parte	El almacenamiento del expediente, una vez de haber iniciado un subproceso de Desistimiento, es en formato digital, esto trae beneficios al

	solicite el mismo expediente, en el 60% de los casos, esta debe esperar entre 1 y 3 días.	personal autorizado a acceder a los mismos ya que solo sería iniciar sesión en el sistema y buscar los documentos del expediente. Esta actividad utilizando el SITPC se comprobó que se disminuyó el tiempo a 31 segundos aproximadamente.
Notificación acerca de los vencimientos de los términos.	A raíz de la encuesta el 50% de estos especialistas plantean que se tardan entre 1 a 3 días para realizar esta actividad.	Esta actividad utilizando el SITPC se comprobó que automáticamente que se vence el término, se le muestra en la pantalla del usuario el trámite para disponer sobre el vencimiento del término.

**Fuente:** Elaboración propia

La comparación realizada en la tabla anterior, a través de los criterios de medida antes definidos, demuestra cómo, a través de la personalización obtenida como resultado de la presente investigación, se contribuye a la agilización del subproceso de Desistimiento en los TPC.

### **Conclusiones del capítulo**

En el presente capítulo se evaluó el cumplimiento de las funcionalidades del sistema de acuerdo al análisis de las métricas, validándose el diseño y verificando que la implementación se haya hecho correctamente. En general, los resultados obtenidos fueron favorables, desde el punto de vista funcional todos los requisitos realizan las funcionalidades requeridas y responden a las necesidades del cliente. Además, se comprobó el grado mínimo de complejidad en el diseño de las clases y un bajo nivel de acoplamiento entre estas, logrando así un alto por ciento de reutilización.

*CONCLUSIONES GENERALES*

- La elaboración del marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos en los que se sustenta la investigación, facilitó la adquisición de los conocimientos necesarios para la solución propuesta.
- La construcción del modelo del diseño, proporcionó generar los elementos fundamentales para facilitar la implementación de la solución.
- Teniendo como base el modelo de diseño, se llevó a cabo la personalización del subproceso Desistimiento en el SITPC, de forma tal que contribuye a la agilización de la ejecución del mismo en las instancias provinciales de los tribunales cubanos.
- La validación de la solución informática mediante la aplicación de pruebas de software, permitió avalar con buenos resultados la calidad del diseño y la implementación de la aplicación.

*RECOMENDACIONES*

Una vez cumplidos los objetivos de la investigación y teniendo en cuenta las experiencias obtenidas en la misma, se recomienda:

- Realizar la integración del subproceso de Desistimiento con el resto del SITPC.
- Ejecutar pruebas de Aceptación con el interés de validar en la práctica el subproceso de Desistimiento integrado al resto del SITPC del cual forman parte indisoluble la propuesta de solución.



REFERENCIAS BIBLIOGRÁFICAS

- **Peña, Carlos. 2015.** Informática Jurídica y Derecho Informático. [En línea] 2015. <http://www.palermo.edu/ingenieria/downloads/pdfwebc&T8/8CyT05.pdf>.
- **Alonso Velazquez, José Luis . 2014.** *Lenguaje de programación: Introducción C/C++ (IDE)*. s.l. : Universidad de Guanajuato, 2014.
- **Apache-JMeter.tm. 2018.** Apache Software Foundation. [En línea] 2018. [Citado el: 18 de Junio de 2018.] <http://jmeter.apache.org/>.
- **Doctrine-project.org. 2018.** Doctrine project. What is Doctrine? [En línea] 2018. [Citado el: 14 de Marzo de 2018.] <http://docs.doctrine-project.org/en/latest/tutorials/getting-started.html>.
- **Enciclopedia-Jurídica. 2014.** Enciclopedia Jurídica. [En línea] 2014. <http://www.encyclopedia-juridica.biz14.com/d/desistimiento/desistimiento.htm>.
- **Fundeu.es. 2013.** Fundeu.es. [En línea] 10 de Julio de 2013. <https://www.fundeu.es/recomendacion/adaptar-o-personalizar-mejor-que-customizar/>.
- **Getbootstrap.com. 2018.** Getbootstrap.com. [En línea] 2018. <https://getbootstrap.com/>.
- **González Ochoa, Darián y Dominguez, Yosviel. 2018.** *SITPC, una herramienta informática cubana para la administración de la justicia*. La Habana : s.n., 2018.
- **Iruela, Juan. 2016.** [En línea] Revista digital INESEMA, 2016. <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>.
- **JavaScript.com. 2018.** JavaScript.com. [En línea] 2018. <https://www.javascript.com/>.
- **Jquery.com. 2018.** Jquery.com. [En línea] 2018. <https://jquery.com/>.
- **Larman, Craig. 2016.** *UML y patrones*. s.l. : 3ra Edición, 2016.
- **Lucidchart. 2018.** Lucidchart. [En línea] 2018. <https://www.lucidchart.com/pages/es>.
- *Mapeo Objeto Relacional. Yanes Enriquez, Osmel y Garcia del Busto, Hansel. 2014.* 3, s.l. : Revista Telem@tica, 2014, Vol. I. ISSN-1729-3804.
- **Martin Olivera, Yonnys Pablo y Zamora Sanchez, Geysler. 2016.** *ENTORNO DE DESARROLLO INTEGRADO LIBRE Y MULTIPLATAFORMA PARA DESARROLLAR SOFTWARE EDUCATIVO EN FORMATO MULTIMEDIA*. La Habana : s.n., 2016.
- **Menéndez-Barzanallana Asensi, Rafael. 2016.** Ingeniería del software. [En línea] Universidad de Murcia, 2016. [http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE\\_principales.html](http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html).
- **NetBeans.org. 2018.** NetBeans.org. [En línea] 2018. [Citado el: 14 de Marzo de 2018.] <https://netbeans.org/kb/docs/ide/git.html>.

- **Network, Developer. 2018.** DeveloperNetwork. [En línea] 2018. [Citado el: 14 de Marzo de 2018.] <https://msdn.microsoft.com/es-es/library/dd409377.aspx>.
- **Ohmyroot. 2017.** Ohmyroot. [En línea] 17 de Enero de 2017. <https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/>.
- **Pacheco, Nacho. 2014.** GitHub. [En línea] 2014. <http://gitnacho.github.io/Twig/templates.html>.
- **Pavón Mestras, Juan. 2014.** *Servidores Web – Apache*. Madrid : Facultad de Informática. Departamento Ingeniería del Software. Universidad Complutense Madrid, 2014.
- **Pérez Avila, Rodolfo. 2012.** *Documento de Arquitectura de Software. Proyecto SITPC*. La Habana : Universidad de las Ciencias Informáticas, 2012.
- **Pressman, Roger S. 2010.** *Ingeniería de Software.Un enfoque práctico*. Séptima. s.l. : MC Graw Hill, 2010.
- **Programación.net. 2018.** Programación.net. [En línea] 2018. [https://programacion.net/articulo/funciones\\_y\\_caracteristicas\\_de\\_php\\_que\\_todo\\_programador\\_deberia\\_de\\_conocer\\_509](https://programacion.net/articulo/funciones_y_caracteristicas_de_php_que_todo_programador_deberia_de_conocer_509).
- **Recaman Chaux, Hernando y Guerrero Alarcón, Carlos Andrés. 2014.** *Marco de trabajo para aplicaciones web de código abierto en instituciones universitarias*. Colombia : Ediciones, 2014. ISSN: 2382-4239.
- **Rodríguez Sanchez, Tamara. 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : Universidad de las Ciencias Informáticas, 2015.
- **Roque Hernández, Yoslenys y Brito, Mauris. 2014.** *Especificación de requisitos no funcionales del sistema. Proyecto de Informatización de los Tribunales Populares Cubanos*. La Habana : s.n., 2014.
- **SoftDoit, S.L. 2018.** SoftDoit. S.L. [En línea] 2018. <https://www.softwaredoit.es/definicion/definicion-lenguaje-de-programacion.html>.
- **Sommerville, Ian. 2011.** *Ingeniería del Software*. Capítulo 6. : Novena edición, 2011.
- **Symfony.com. 2018.** Symfony.com. [En línea] 2018. <https://symfony.com/>.
- **Valenzuela Buitrago, Jorge y Ferrer Infante, Carlos Alberto. 2014.** *GUÍA DE ELABORACIÓN DE MODELOS CONCEPTUALES*. Departamento Nacional de Planeación. Colombia : s.n., 2014.
- **Visual Paradigm, International Ltd. 2017.** *Visual Paradigm*. 2017.
  - **W3C. 2018.** W3C. [En línea] Mayo de 2018. <https://www.w3.org/html/>.

