

**Universidad de las Ciencias Informáticas**

**Facultad 3**



*Desarrollo del subsistema Reportes del Proyecto de  
Informatización de los Tribunales Populares Cubanos  
Fase I.*

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

**Autores:**

Alayna Pantoja Trincado

Yoslenys Roque Hernández

**Tutores:**

Ing. Martín Villalón Cruzata

Ing. Chavelys Téllez Larramendi

**La Habana, 20 de junio de 2012**

**“Año 54 de la Revolución”**

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

---

Alayna Pantoja Trincado

---

Yoslenys Roque Hernández

---

Ing. Martín Villalón Cruzata

---

Ing. Chavelys Téllez Larramendi

## AGRADECIMIENTOS

*Agradecemos a todas las personas que de alguna forma han colaborado con la realización del presente trabajo de diploma, a nuestros tutores Chavelys Téllez y Martín Villalón por guiarnos, y en especial a Baby por ayudarnos tanto.*

### *Alayna:*

*Agradezco a las dos personas más importantes para mí, a mami y a papi, porque sin ellos nada hubiera sido posible, gracias por darme todo en la vida.*

*Al resto de mi familia por apoyarme y siempre creer que puedo lograr lo que me proponga, en especial a mi abuelita Gladys por enseñarme simplemente a crecer y a tener fe.*

*A mis amigos Mónica, Bárbara, Analina, Liamne, Félix, Lino, Pedro, Papito, Leo, y a Daimarys, Alberto y Anabel que ya no están en la universidad pero que espero volver a ver algún día, a todos los quiero muchísimo, sin ellos estos cinco años no hubieran tenido sentido.*

*A mis demás amistades y compañeros de aula con los que he compartido tantos momentos buenos y malos.*

*A todos los profesores que me han impartido clases en esta universidad, porque de cada uno de ellos he aprendido algo en especial y a mis compañeros del proyecto Tribunales por apoyarme en la realización de este trabajo.*

*Gracias especialmente a Yoslenys, porque no me puedo imaginar mi tesis con otra persona que no fuera él, porque entre los dos, con preocupación y momentos de estrés pero con mucho esfuerzo hemos logrado realizar nuestro sueño.*

*Yoslenys:*

*Agradezco especialmente mi abuela Eneida por ser la mejor madre que he podido tener, porque ha sido la luz que guía mi camino y sin su apoyo incondicional no hubiera podido llegar hasta aquí, porque siempre me ha enseñado que para lograr grandes cosas hay que dedicar mucho tiempo, esfuerzo y sacrificio y que al final todo se puede.*

*A mi abuelo Armando que me ha malcriado y consentido más que a sus propios hijos y que ha sacrificado todo lo que tiene para que no me falte nunca nada.*

*A mi mamá Gladis por estar presente en cada paso que doy alumbrándome con su cariño y su amor maternal.*

*A mis hermanos Alexis y Lianelis, porque aunque no lo demuestre muy a menudo, yo los quiero tanto como ellos a mí.*

*A toda mi familia por apoyar mis sueños. Incluyendo a mi padrastro Tony que siempre ha estado ahí para mí.*

*Gracias a Ania, Yiliam y Doa, por haber sido los mejores y los más insoportables amigos. Gracias a Ricnelys, Irmel, Yandys, Lisandra, Juan Carlos, Yasmín, Félix, Tatiana, Rancés, Daileén, Robinson y a todos con los que he convivido, los cuales espero que a pesar de que tomemos rumbos diferentes nunca nos olvidemos.*

*Gracias a Dayanis, Anita, Suyen y a todas esas personas que me apoyan desde lejos, a los viejos y nuevos amigos, a mi gente de la FEU y especialmente a mis compañeros del 3505 y de mi grupo CDI.*

*Un agradecimiento muy especial a mi chuchi Alayna, mi compañera de tesis, por haberme soportado en mis momentos de estrés, por haber puesto empeño y sacrificio en cada página de este trabajo, porque de todos no me pudo haber tocado una mejor compañera para esta contienda que ella.*

*¡Y a todo el que cuando lea esto sienta que falta, es que son muchos y se olvida!*

## DEDICATORIA

*Alayna:*

*A mis padres por guiarme y estar siempre  
ahí en todo momento, porque son lo mejor  
que tengo en la vida.*

*A mi familia por apoyarme y creer en mí.*

*Yoslenys:*

*A mis abuelos por todo el amor y el apoyo  
que siempre me han brindado.*

*A mi mamá y hermanos por confiar en mí.*

*A mi familia por el respeto y la admiración  
que siempre me han demostrado.*

## RESUMEN

El Ministerio de Justicia de la República de Cuba es una entidad gubernamental que en conjunto con la Universidad de las Ciencias Informáticas desarrollan un software para los Tribunales Populares Cubanos que permitirá la informatización de procesos y documentos jurídicos, además de la ejecución y supervisión de manera automática de las actividades judiciales y disposiciones legales que lo regulan.

El presente trabajo de diploma hace referencia al módulo de reportes, perteneciente al sistema de informatización de los Tribunales Populares Cubanos, que tiene como objetivo la gestión y publicación de la información contenida en forma de reportes estadísticos y no estadísticos que se requieran en los demás módulos y en sus respectivos procesos, de forma tal que facilite la estandarización, integración y a su vez se utilice como herramienta de control de la información en el proceso de toma de decisiones.

Para el desarrollo de la investigación se empleó como metodología el Proceso Unificado de Desarrollo (RUP). Los artefactos se generaron usando el Lenguaje Unificado de Modelado (UML) y auxiliados por el Visual Paradigm 5.0 como herramienta de Ingeniería de Software Asistida por Computadora (CASE). De igual forma se utilizó la herramienta de prototipado Axure RP Pro 5.5 en la realización de prototipos no funcionales y el Generador de Reportes Dinámico 1.7.0 para la creación y diseño de los reportes. Además para garantizar la calidad de los artefactos generados se realizaron las validaciones correspondientes y se aplicaron pruebas funcionales, que arrojaron resultados positivos.

**Palabras claves:** Estandarización, informatización, judiciales, procesos, reportes, software.

## ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....</b>	<b>6</b>
<b>Introducción .....</b>	<b>6</b>
1.1 Conceptos fundamentales asociados al dominio del problema .....	6
1.2 Metodología de desarrollo de software RUP .....	7
1.3 Lenguaje de modelado UML .....	10
1.4 Ingeniería de Requisitos .....	10
1.4.1 Requisito de software .....	10
1.4.1.1 Requisitos funcionales .....	11
1.4.1.2 Requisitos no funcionales .....	11
1.4.2 Actividades de la Ingeniería de Requisitos .....	11
1.5 Modelo de Dominio .....	14
1.6 Herramienta CASE .....	15
1.6.1 Visual Paradigm 5.0 .....	15
1.7 Patrones de Casos de Uso .....	16
1.7.1 Patrón Inclusión Concreta: .....	16
1.7.2 Patrón Múltiples Actores - Roles comunes: .....	16
1.8 Herramienta para modelado de prototipos no funcionales .....	17
1.8.1 Axure RP Pro 5.5 .....	17
1.9 Aplicación Web .....	18
1.10 Herramienta para generación de reportes .....	18
1.10.1 Generador Dinámico de Reportes 1.7.0 .....	19
1.11 Servidor web Apache2 .....	23
1.12 Herramienta para Gestor de Base de Datos .....	24
1.12.1 PostgreSQL 8.4 .....	25
1.12.2 PL/Pgsql .....	25
<b>Conclusiones del capítulo .....</b>	<b>26</b>
<b>CAPÍTULO 2: PROPUESTA DE SOLUCIÓN .....</b>	<b>27</b>
<b>Introducción .....</b>	<b>27</b>
2.1 Descripción del sistema .....	27
2.2 Modelo de dominio .....	27
2.2.1 Conceptos asociados al modelo de dominio .....	28
2.2.2 Descripción del modelo de dominio .....	29
2.2.3 Diagrama de objetos del modelo de dominio .....	30
2.3 Especificación de requisitos funcionales .....	30
2.4 Descripción de actores del sistema .....	32
2.5 Diagrama de casos de uso del sistema .....	34
2.6 Descripción de casos de uso del sistema .....	35
2.7 Especificación de requisitos no funcionales .....	39
2.7.1 Apariencia o Interfaz Externa .....	39
2.7.2 Usabilidad .....	40
2.7.3 Fiabilidad .....	40

---

2.7.4 Requisitos de Software .....	40
2.7.5 Seguridad .....	40
2.8 Modelo de diseño .....	41
2.8.1 Diagramas de clases del diseño .....	41
2.9 Diagrama de componentes .....	43
2.9.1 Descripción del diagrama de componentes.....	44
2.10 Diagrama de despliegue .....	45
2.11 Desarrollo de los casos de uso .....	46
2.11.1 Código fuente .....	47
2.12 Interfaces del sistema .....	48
<b>Conclusiones del capítulo .....</b>	<b>50</b>
<b>CAPÍTULO 3: VALIDACIÓN Y PRUEBAS .....</b>	<b>51</b>
<b>Introducción .....</b>	<b>51</b>
3.1 Iteraciones.....	51
3.2 Validación de requisitos .....	51
3.2.1 Prototipos funcionales .....	51
3.2.2 Matriz de trazabilidad .....	53
3.3 Pruebas.....	53
3.3.1 Diseño de casos de prueba .....	53
3.3.2 Pruebas de caja negra .....	54
3.3.2.1 Técnica de partición de equivalencia .....	54
3.3.3 Casos de pruebas diseñados.....	55
3.4 Registro de errores y dificultades detectadas.....	59
3.5 Análisis general de resultados.....	60
3.6 Validación de las variables de la investigación.....	62
<b>Conclusiones del capítulo .....</b>	<b>63</b>
<b>CONCLUSIONES GENERALES .....</b>	<b>65</b>
<b>RECOMENDACIONES.....</b>	<b>66</b>
<b>BIBLIOGRAFÍA .....</b>	<b>67</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>70</b>
<b>ANEXOS .....</b>	<b>72</b>



## ÍNDICE DE FIGURAS

Fig.1. Presencia de los CU en todo el proceso de desarrollo de software.....	8
Fig.2. Línea del tiempo de la arquitectura del ciclo de vida del software .....	9
Fig.3. Iteraciones del proceso de desarrollo de software.....	9
Fig.4. Página principal del Generador Dinámico de Reportes 1.7.0 .....	19
Fig.5. Área de trabajo del Visor de reportes.....	20
Fig.6. Área de Trabajo del Diseñador de Modelos.....	20
Fig.7. Área del Diseñador de Reportes, seleccionando reporte. ....	21
Fig.8. Área Diseñador de Reportes. ....	22
Fig.9. Área Diseñador de Consultas .....	23
Fig.10. Área Diseñador de Consultas más la vista SQL.....	23
Fig.11. Modelo de dominio.....	30
Fig.12. Modelo de casos de uso del sistema del módulo económico.....	34
Fig.13. Presencia del patrón múltiples actores roles comunes .....	35
Fig.14. Presencia del patrón de inclusión concreta .....	35
Fig.15. Diagrama de clases diseñador de reportes.....	42
Fig.16. Diagrama de clases Visor de reportes.....	43
Fig.17. Modelo de componentes para el CU Visualizar Reporte Diligencias Previas del Proceso Ejecutivo .....	44
Fig.18. Modelo de despliegue.....	46
Fig.19. Diseño del Reporte Diligencias Previas del Proceso Ejecutivo.....	47
Fig.20. Código fuente del reporte Diligencias Previas del Proceso Ejecutivo .....	48
Fig.21. Interfaz principal del subsistema Reportes (Recuperaciones).....	49
Fig.22. Interfaz de acceso al Visor de reporte.....	49
Fig.23. Gráfico de NC encontradas por iteraciones. ....	61
Fig.24. Gráfico tipo de NC significativas encontradas por iteraciones. ....	61
Fig.25. Gráfico de resultados de pruebas de caja negra por módulo.....	62

## **ÍNDICE DE TABLAS**

Tabla 1: Descripción de los actores del sistema .....	33
Tabla 2: Descripción de Casos de Uso del Sistema .....	35
Tabla 3: Secciones a probar en el caso de uso.....	56
Tabla 4: Descripción de variables.....	57
Tabla 5: SC1: Reporte diligencias previas del proceso ejecutivo .....	58
Tabla 6: No conformidades de la aplicación .....	59
Tabla 7: Variables de la investigación .....	62

## **INTRODUCCIÓN**

Las Tecnologías de la Información y las Comunicaciones (TIC), se han convertido en una poderosa fuente que alimenta el conocimiento del mundo entero, sobre todo en la actualidad, debido al progresivo desarrollo con que evolucionan y forman parte de la sociedad. Las TIC han marcado transformaciones en la economía, la cultura y la sociedad en general, destacándose además en la rama del Derecho, surgiendo de esta manera la disciplina Informática Jurídica, la cual no es más que la aplicación de la tecnología a la ciencia del derecho para reducir los problemas jurídicos.

Cuba por su parte no permanece al margen de los avances tecnológicos en esta rama. Para facilitar el nuevo desempeño que lleva a cabo nuestro país en el terreno de la informática existe la Universidad de las Ciencias Informáticas (UCI), la cual fue creada con el objetivo de ayudar en la formación de futuros ingenieros informáticos. La UCI desde su surgimiento tiene entre sus principales lineamientos la vinculación del estudio con el trabajo, por lo que la producción de software se ha convertido en su eje central. En la universidad el desarrollo de software está organizado por centros productivos que tienen fines específicos, CEGEL (Centro de Gobierno Electrónico) surge con el objetivo de informatizar procesos en las instituciones gubernamentales del país y con ello la posibilidad de agilizar los canales de información, ahorrar tiempo y recursos en las acciones que se realizan en estas organizaciones.

En este centro, el cual está estrechamente relacionado con el Ministerio de Justicia se desarrolla una solución informática para los Tribunales Populares Cubanos (TPC) que permitirá 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.

De manera general la solución informática propuesta cuenta con ocho módulos, Económico, Laboral, Administrativo, Penal, Civil, Administración y Gobierno, Común y Reportes, este último se encarga de gestionar y publicar la información contenida en forma de reportes estadísticos y no estadísticos que se requieran en los demás módulos y en sus respectivos procesos. Como consecuencia de la gran extensión y complejidad de la solución propuesta se decidió dividirla en tres fases. Reportes incluye específicamente la fase 1, la cual está compuesta solamente por los procedimientos diligencias previas de la sala de lo Económico, disciplina y derecho laboral de la sala de lo Laboral, administrativo de la sala de lo Administrativo y ordinario en su instancia municipal de las salas de lo Penal y Civil (TPC, 2011).

La obtención de información íntegra a través de estos reportes en los tribunales del país se realiza de forma semiautomática, pues se cuenta con pequeñas aplicaciones que han sido realizadas por los informáticos del área pero no abarcan todo el contenido estadístico necesario en los procesos que lo requieren, sólo existen para algunas materias y determinados procesos, que además no generan ni publican los reportes por lo que los trabajadores de la estadística (secretario auxiliar, secretario judicial y presidente de sala), realizan todo este proceso de consulta y revisión de forma manual, haciendo uso de estas soluciones de manera aislada, con desorganización y lentitud.

Debido a que dichas aplicaciones no cuentan con las funcionalidades que se necesitan se pierde la información que no es controlada durante la recogida de los datos, haciendo compleja la búsqueda, supervisión y el almacenamiento de los mismos, pues el cúmulo de información aumenta considerablemente con el paso del tiempo. De no existir estas dificultades los procesos se realizarían con mayor agilidad permitiendo que sus trabajadores eliminen cualquier posibilidad de equivocación.

A razón de la problemática existente surge el siguiente **problema de la investigación**: Los mecanismos utilizados en los Tribunales Populares Cubanos para obtener los reportes no garantizan la estandarización, integridad y control de la información de la estadística judicial.

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

De esta forma se determina como **objetivo general**: Desarrollar los reportes del Proyecto de Informatización de los Tribunales Populares Cubanos que permita garantizar la estandarización, integridad y control de la información de la estadística judicial.

Para ello se identifica como **campo de acción**: Los reportes de la estadística judicial de los procedimientos de la fase 1 del SIT.

Por lo tanto se plantea la siguiente **idea a defender**: Si se desarrollan los reportes de los procesos definidos en la fase I, se puede lograr una estandarización, integridad y control de la información de la estadística judicial.

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

- ❖ Definir el marco teórico-referencial de la investigación.

- ❖ Realizar el modelo de sistema.
- ❖ Desarrollar el modelo de implementación.
- ❖ Validar los resultados obtenidos.

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

- ✓ Estudio del estado del arte de la Ingeniería de Requisitos de los Tribunales Populares Cubanos en su fase I.
- ✓ Estudio e identificación de los reportes estadísticos y no estadísticos que se requieren en el módulo Reportes de los tribunales provinciales y municipales cubanos en su fase I.
- ✓ Análisis de la metodología de desarrollo de software y la herramienta CASE seleccionada para el modelado y diseño de los procesos del sistema.
- ✓ Análisis del lenguaje de modelado que se utilizará para lograr una comprensión de los subsistemas.
- ✓ Análisis de la herramienta de prototipado seleccionada previamente.
- ✓ Realización del modelo de dominio.
- ✓ Realización de diagramas de caso de uso del sistema.
- ✓ Realización de prototipos no funcionales.
- ✓ Realización de diagrama de componentes.
- ✓ Análisis de la herramienta para el desarrollo de los CU definidos.
- ✓ Implementación de los componentes.
- ✓ Diseño de los casos de prueba.
- ✓ Validación de la implementación mediante las pruebas de caja negra.

Existen métodos para dar solución a las tareas antes propuestas, estos son los métodos teóricos, empíricos y científicos. De los cuales se emplearon:

#### **Métodos Teóricos:**

**Analítico-Sintético**, este método permitió realizar un estudio teórico de la investigación permitiendo el análisis de documentos y la extracción de elementos importantes acerca del proceso de generación de reportes, se utilizó además para la precisión de las características del diseño propuesto, así como para el arribo de las conclusiones de la investigación.

**Histórico-Lógico**, este método fue usado para analizar las características de las herramientas utilizadas y las tendencias evolutivas actuales de su uso en los proyectos productivos de la universidad, así como la definición de los antecedentes de las mismas y la vinculación con tecnologías confinadas para implementar los reportes.

**Hipotético-Deductivo**, este método ha sido utilizado para la observación de los procesos que se siguen en el marco de la investigación, lo que permitió crear, partiendo de dichos procesos, las deducciones o proposiciones más elementales de la idea a defender, verificándose además la veracidad de los enunciados deducidos.

**Modelación**, se llevó a cabo una modelación del objeto lo que permite predecir la respuesta del proceso a variaciones de algunos de sus parámetros, sin tener que ejecutar el proceso en la realidad.

#### **Métodos Empíricos:**

**Entrevista**, para lograr definir las necesidades de los clientes y el entendimiento del problema fue necesaria la recopilación y registro de información útil referente a la estadística judicial para entender el proceso de la gestión de los reportes, realizar el levantamiento de requisitos y determinar las principales características de la propuesta de solución.

**Observación**, para lograr entender el funcionamiento real de los procesos fue necesario observar las actividades realizadas para determinar cómo ocurre realmente la gestión de los reportes.

**Medición**, para medir los atributos de calidad que posee el software, además para obtener información numérica sobre el grado del esfuerzo realizado en el momento de efectuar las pruebas de calidad.

El contenido de este trabajo se encuentra estructurado en tres capítulos, los que se definen de la siguiente manera:

**Capítulo 1: Fundamentación Teórica.** En este capítulo se abordan los conceptos fundamentales que están relacionados con la estadística judicial. Se hace referencia a la metodología por la cual se guiará todo el proceso de desarrollo del software, así como el

lenguaje utilizado por la misma. Además se presenta la fundamentación de las herramientas utilizadas para el diseño del sistema y las propuestas para su implementación y desarrollo.

**Capítulo 2: Propuesta de Solución.** En este capítulo se realiza un análisis de las características del sistema a desarrollar a partir de la problemática planteada, se describen las principales funcionalidades que brindará el módulo de Reportes y se identifican los trabajadores que intervienen. También se obtienen los artefactos generados a partir de la aplicación de la metodología de desarrollo utilizada.

**Capítulo 3: Validación y Prueba.** En este capítulo se realizan las validaciones y pruebas correspondientes a la solución propuesta en el capítulo anterior para determinar la calidad de la misma.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### Introducción

En este primer capítulo se abordan algunos 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 seleccionado y de la herramienta CASE escogida para entender y manejar los artefactos durante todo el desarrollo del software.

Un tema de suma importancia en este capítulo es la ingeniería de requisitos, la cual es imprescindible para identificar lo que se quiere construir y de igual forma se analizan los patrones de casos de uso utilizados en el modelado de los diagramas. Además se profundiza en las características del generador dinámico de reportes definido en la arquitectura del sistema para la informatización de los Tribunales Populares Cubanos y en las herramientas de implementación que permiten una adecuada construcción del software, justificando en cada caso, el uso de las seleccionadas.

### 1.1 Conceptos fundamentales asociados al dominio del problema

**Reporte:** Es utilizado cuando se quiere informar o dar noticia acerca de una determinada cuestión. Incluye elementos persuasivos, que pueden ser recomendaciones o sugerencias y también algunas conclusiones a través de las cuales se pueden tomar acciones en el futuro. Constituye un documento estructurado y resumido que contiene datos relevantes y es generado por un sistema (Populares, 2010).

El diseño de un reporte consta de varios pasos los cuales conforman una actividad secuencial que concluye en la mayoría de los casos con el resultado esperado: primeramente se procede al análisis del prototipo que no es más que el modelo que se necesita para realizar el posterior diseño del reporte, seguidamente se realizan los procedimientos, se obtienen a través de consultas los datos que son necesarios para conformar el informe, además de las expresiones, parámetros, imágenes, cuadros de texto, tablas y cualquier otro elemento de diseño que se pueda incluir en el mismo. Por último se verifica el estado del reporte en la vista previa y se chequea la veracidad de los datos.

Un reporte está compuesto por el encabezado del documento, el encabezado de página, el cuerpo, un pie de página y el pie de documento. Puede incluir imágenes, cuadros de texto y



líneas en los encabezados y pies de página. El encabezado de página contiene los campos del reporte que son estáticos y el cuerpo contiene los datos que son el resultado de los procedimientos aplicados. El reporte puede contener cualquier tipo de elemento en el cuerpo, por ejemplo tablas, matrices, listas, gráficos y medidores (Colectivo de Profesores de DATEC, 2010).

**Reporte Estadístico:** Es un tipo de reporte con características especiales, por lo que constituye un documento estructurado y resumido que contiene datos relevantes organizados en tablas, pero esta vez los datos van a ser solamente estadísticos, o sea que la información que brinda sobre una determinada cuestión es meramente numérica, dígase cantidades, por cientos, promedios o sumas totales.

**Reporte no Estadístico:** Es el documento compuesto por datos que ofrecen una información determinada en forma de listado, aunque organizados en tablas igual a los datos estadísticos, estos se diferencian por constituir fechas, nombres o cualquier otro dato, y que a su vez estén relacionados entre sí.

**Estadística Judicial:** Trabajo o labor realizada por los secretarios judiciales y auxiliares periódicamente, operación que consiste en registrar los datos estadísticos que se presentan y la información correspondiente a los procesos que se inicien en los tribunales hasta su fecha de culminación, para su posterior control y supervisión mediante reportes que brinden de manera organizada los datos requeridos por parte de los jueces y presidentes de sala (Populares, 2010).

## 1.2 Metodología de desarrollo de software RUP

Una metodología de desarrollo de software, no es más que un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la construcción de un software, donde el éxito del producto depende en gran parte de la metodología escogida por el equipo, ya sea tradicional o ágil, que son los dos tipos existentes (J. Solís, y otros).

Entre las principales metodologías tradicionales, o formales, como también son conocidas, están RUP y MSF, que centran su atención en llevar una documentación exhaustiva de todo el proyecto, además se enfocan en cumplir con un plan de proyecto, todo esto definido en la fase inicial del desarrollo del software.

RUP es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, es el más empleado dentro de las metodologías tradicionales, tiene un ciclo de vida definido por cuatro fases (Inicio, Elaboración, Construcción y Transición), que concluyen cada una con un hito, los cuales generan una serie de artefactos permitiendo el desarrollo del proyecto, estas fases están constituidas a su vez por flujos de trabajo o disciplinas. RUP utiliza el lenguaje UML como lenguaje de notación y posee tres características esenciales:

1. **Dirigido por Casos de Uso:** son los rectores del proceso de desarrollo, integran todo el trabajo, pues inician el proceso de desarrollo, guían el diseño, implementación y prueba de los requisitos del sistema.

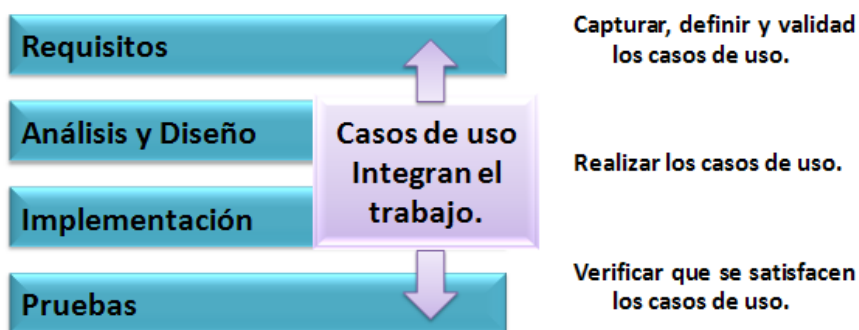


Fig.1. Presencia de los CU en todo el proceso de desarrollo de software

2. **Centrado en la Arquitectura:** es la organización o estructura de las partes más relevantes del sistema, permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar su desarrollo.

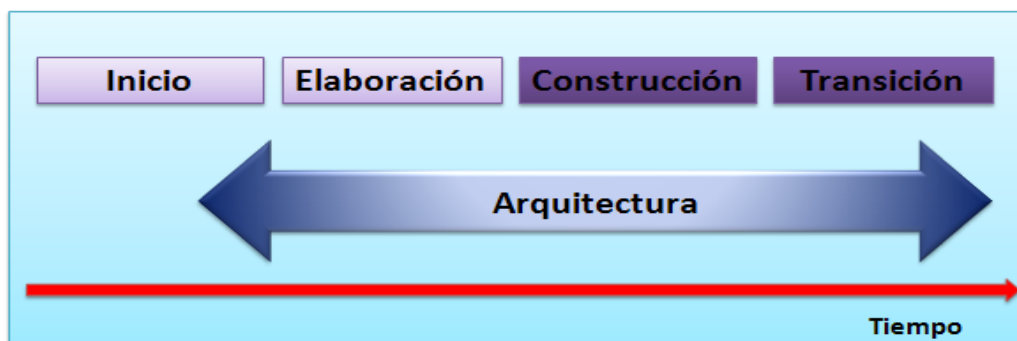


Fig.2. Línea del tiempo de la arquitectura del ciclo de vida del software

3. **Iterativo e incremental:** el trabajo se divide en partes más pequeñas o mini proyectos, lo que permite que el equilibrio entre casos de uso y arquitectura se vaya logrando durante cada uno, durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto (Kruchten, 2004).

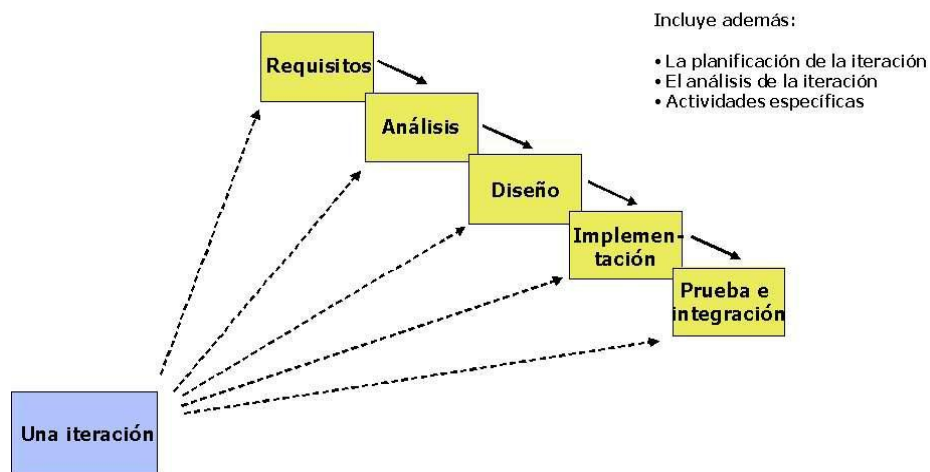


Fig.3. Iteraciones del proceso de desarrollo de software

La metodología RUP es la seleccionada para guiar el proceso de desarrollo del Sistema de Informatización de Tribunales por el Plan de Desarrollo de la Ingeniería de Requisitos del SIT porque sigue los pasos intuitivos a la hora de desarrollar el software y ofrece un seguimiento detallado en cada una de las fases, así como una evaluación continua en cada una de ellas, lo cual permite cambios en los objetivos. Define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto.

Además de las ventajas y características antes vistas, RUP garantiza la evolución en el proceso de desarrollo y la creación de un producto que se ajuste a las necesidades de los usuarios. Es una metodología robusta que se adapta muy bien a proyectos de larga duración, complejos y con un gran equipo de desarrollo, como es el caso del Proyecto de Informatización de los TPC.

### 1.3 Lenguaje de modelado UML

Un modelo es una representación parcial o simplificada de la realidad que recoge aspectos preeminentes, un lenguaje puede definirse como el conjunto de símbolos útiles para comunicarnos, por ende se puede arribar que el lenguaje de modelado de objetos está formado por un conjunto estandarizado de símbolos y de maneras de disponerlos para modelar parte de un diseño de software orientado a objetos (Larman, 1999).

El Lenguaje Unificado de Modelado es un lenguaje de modelado que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software, no define un proceso de desarrollo específico, tan solo se trata de una notación. Se utiliza para detallar los artefactos en el sistema y definir un sistema de software. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar (UML, 2012).

El análisis de este lenguaje se realiza debido a que UML es la notación utilizada por la herramienta CASE que se emplea para la creación del SIT, además es el lenguaje estándar de modelado para software que emplea la metodología RUP, la cual rige el proceso de desarrollo de software del proyecto Tribunales Populares Cubanos.

### 1.4 Ingeniería de Requisitos

La ingeniería de requisitos constituye la base sobre la cual ha de edificarse el sistema o software a desarrollar, la eficacia del resultado depende en gran medida de la realización de una buena ingeniería de requisitos. Mediante la misma podremos saber las restricciones que debe cumplir el software así como sus funcionalidades (Young, 2004).

#### 1.4.1 Requisito de software

Es primordial conocer el concepto de requisito para entender en qué consiste la ingeniería de requisitos. Aunque han sido muchas las definiciones de este término, una de las más abarcadoras es la planteada por Sommerville: “Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este. En el otro extremo es una declaración detallada y formal de una función del sistema.”

### 1.4.1.1 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, o sea como éste debe reaccionar a determinadas entradas e incluso declarar explícitamente lo que el sistema no debe hacer. Los requerimientos funcionales del sistema describen de forma detallada la función de este, sus entradas y salidas, excepciones, etcétera.

Estos requisitos deben ser especificados por escrito, como todo contrato o acuerdo entre dos partes, deben ser posibles de probar de lo contrario no hay forma de saber si se cumplió con él o no, deben estar descritos como una característica del sistema a entregar, esto es, lo que el sistema debe hacer y no cómo debe hacerlo. Es primordial redactarlos de la manera más abstracta y concisa posible para evitar malas interpretaciones (Sommerville, 2005).

### 1.4.1.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, y que harán del mismo un sistema confiable y seguro, forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, ya que aunque se conozca que el sistema cumple con la toda la funcionalidad requerida, propiedades no funcionales como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación (Sommerville, 2005).

## 1.4.2 Actividades de la Ingeniería de Requisitos

Existen cinco actividades básicas de la Ingeniería de Requisitos que son necesarias para que el proceso de desarrollo de software se ejecute de forma adecuada. Estas actividades son las siguientes:

**Elicitación de requisitos:** esta etapa abarca la primera fase dentro del proceso de desarrollo de software, con ella se intenta garantizar que los requisitos del sistema sean consistentes para que cumplan con las necesidades de la organización. La elicitación es el proceso donde se identifican las características que se desean y las restricciones que debe satisfacer el software. Para que los resultados sean puntuales se emplean varias técnicas de elicitación como son: la

observación, análisis de tareas, entrevistas, arqueología de documentos y los prototipos no funcionales.

- Entrevistas: las entrevistas sirven para obtener una comprensión general de lo que hacen los stakeholders (partes interesadas), cómo podrían interactuar con el sistema y las dificultades a las que se enfrentan con los sistemas actuales (Sommerville, 2005). Existen diferentes tipos de entrevistas recomendadas, se mencionan a continuación las propuestas por el propio Sommerville:
  - Entrevistas cerradas, donde los clientes responden a un conjunto predefinido de preguntas.
  - Entrevistas abiertas, donde no hay un programa predefinido. El equipo de la ingeniería de requisitos examina una serie de cuestiones con los clientes del sistema y, por tanto, desarrolla una mejor comprensión de sus necesidades.
- Arqueología de documentos: Permite determinar posibles requisitos sobre la base de inspeccionar la documentación utilizada y generada en la empresa. Sirve fundamentalmente como complemento de las demás técnicas y mediante la misma se obtiene información que de otra forma sería imposible, como son manuales de procedimientos, reglamentos, facturas, entre otras.
- Observación y análisis de tareas: Consiste en el análisis del trabajo diario y la anotación de estas tareas.
- Prototipos no funcionales: Para elicitar los requisitos durante la etapa de captura se crean prototipos, los cuales son simulaciones del posible producto, que luego son utilizadas por el usuario final. Provee a los analistas de una importante retroalimentación, ya que les permite conocer si el sistema diseñado sobre la base de los requisitos capturados, les permite a los usuarios realizar su trabajo de forma efectiva y eficiente (Escalona, 2005).

**Análisis y negociación de requisitos:** luego de resumidos los requisitos, se agrupan por categorías y se organizan en subconjuntos, además se estudia cada requisito en relación con el resto, se examinan en su consistencia, completitud y ambigüedad. Pueden surgir requisitos contradictorios por parte de los usuarios, los cuales el ingeniero del sistema debe ser capaz de resolver a través de un proceso de negociación. Luego de manera iterativa se irán eliminando requisitos, combinando y/o modificando para conseguir satisfacer los objetivos planteados.

**Especificación de requisitos:** documento escrito, modelo gráfico, modelo matemático formal, colección de escenarios de uso, prototipo o una combinación de todos ellos. El resultado que se obtiene sirve de apoyo a la ingeniería de hardware, de software, de base de datos y la humana. Durante la especificación se registran los requisitos de forma detallada en un documento formal, constituye la documentación detallada de los requisitos, este documento es de gran utilidad para etapas posteriores en el desarrollo del software.

**Validación de requisitos:** etapa final de la Ingeniería de Requisitos cuyo objetivo es confirmar los requisitos para asegurarse que representan una descripción admisible del sistema que se debe implementar. Se examinan las especificaciones para cerciorar que todos los requisitos del sistema han sido establecidos sin ambigüedades, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto.

**Gestión de Requisitos:** mediante esta tarea se procura llevar un control sobre los cambios que pueden sufrir los requisitos en caso que el problema que se estaba resolviendo haya sufrido cambios. Para gestionar los requisitos se llevan a cabo “un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento” (Pressman, 2005).

Durante todo el proceso de desarrollo se pretende realizar una IR completa, contando con el apoyo del Plan de Desarrollo de la Ingeniería de Requisitos del SIT, siguiendo las actividades que plantea Pressman para un adecuado y total desarrollo del software, y empleando la guía que ofrece la metodología RUP seleccionada. Para comenzar se debe realizar la elicitación de los requisitos mediante la técnica de la entrevista que establece Sommerville, pues aunque gran parte de la información que se maneja en los informes estadísticos ha sido analizada por separado en las materias que requieren de la visualización de estos reportes, se hace necesario identificar el reporte que se necesita visualizar de cada una de ellas. Posterior a esta etapa se pretende entender al detalle cada campo que contenga la tabla de salida del reporte en cuestión, pues estos constituyen en sí los requisitos que debe tener el sistema, y los mismos deben ser consistentes, completos y nunca ambiguos. Seguidamente se realizarán las demás etapas de la IR, en las cuales se especificarán los requisitos capturados, para continuar con la

validación y la gestión de los mismos, manteniéndose esta última actividad durante todo el proceso de desarrollo que se llevará a cabo para lograr una mayor calidad del sistema.

## 1.5 Modelo de Dominio

Considerando que RUP es la metodología escogida para regir el proceso de desarrollo del SIT y lo planteado en la ingeniería de requisitos previamente, es imprescindible analizar el comportamiento de Reportes en la primera disciplina de RUP, modelado empresarial o de negocio.

Teniendo en cuenta que el proceso de negocio de los reportes no está bien definido ya que no se ven claramente las fronteras del mismo, las personas que lo inician o, incluso, quién o quiénes van a ser los beneficiados, se determinó realizar un modelo de dominio pues no se cuenta con los elementos necesarios para realizar el modelado del negocio.

Un modelo de dominio captura los tipos más importantes de objetos del sistema, por lo que su objetivo es contribuir a la comprensión del contexto del mismo y de los requisitos del sistema que se desprenden de este contexto. Los objetos del dominio representan eventos que suceden en el entorno en que trabaja el sistema. Muchos de los objetos del dominio o clases, para emplear una terminología más precisa, pueden obtenerse de una especificación de requisitos o mediante entrevistas con los expertos del dominio. Las clases del dominio aparecen en tres formas típicas:

- ✚ Objetos del negocio que representan entidades que se manipulan en el mismo.
- ✚ Objetos del mundo real y conceptos de los que el sistema debe hacer un seguimiento.
- ✚ Sucesos que ocurrirán o han ocurrido.

El modelado del dominio se realiza habitualmente en reuniones organizadas por los analistas del mismo, en las cuales comprenden y describen los conceptos más importantes del sistema. En algunos casos el modelo de dominio es muy corto y no se hace necesario desarrollar un modelo de objetos, se puede realizar solamente un glosario de términos. El modelo de dominio se describe mediante diagramas de UML.

Las clases del modelo de dominio y el glosario de términos se utilizan en el desarrollo de los modelos de casos de uso y de análisis, permiten describir los casos de uso y diseñar la interfaz de usuario (Rumbaugh, y otros, 2000).



## 1.6 Herramienta CASE

Una herramienta CASE es un sistema de software que intenta proporcionar ayuda automatizada a las actividades que se realizan a través de todo el proceso de desarrollo del software, lo que contribuye a que el desarrollo del mismo se haga de manera más rápida y fácil (Sommerville, 2005).

“CASE proporciona al ingeniero la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Las herramientas CASE ayudan a garantizar que la calidad se diseñe antes de llegar a construir el producto” (Pressman, 2005).

Algunos de los beneficios que pueden proveer las herramientas CASE en todas las etapas del proceso de desarrollo de software, son:

- ❖ Hacer el trabajo de diseño de software más fácil y agradable.
- ❖ Verificar el uso de todos los elementos en el sistema diseñado.
- ❖ Ayudar en la documentación del sistema.
- ❖ Ayudar en la creación de relaciones en las bases de datos.
- ❖ Generar estructuras de código.
- ❖ Reducción del costo de producción de software (Kendall & Kendall, 1997).

### 1.6.1 Visual Paradigm 5.0

Herramienta CASE multiplataforma, diseñada para desarrollar software que utilicen programación orientada a objetos. Visual Paradigm 5.0 permite dibujar todos los tipos de diagramas necesarios para el modelado. Esta herramienta genera toda la documentación de lo que se hace, cumpliendo con estándares establecidos (Paradigm, 2012).

Para el modelado en el SIT se utiliza Visual Paradigm 5.0, el cual fue definido por el Plan de Desarrollo de la Ingeniería de Requisitos del SIT como herramienta de modelado de software ya que posee alta capacidad de integración con el lenguaje orientado a procesos BPMN y UML, de los anteriores sólo será utilizado UML durante el modelado del sistema para Reportes. Como Visual Paradigm 5.0 es una herramienta UML profesional, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Es una herramienta fácil de instalar y actualizar y una de

las características más importantes es que cuenta con una licencia uci para un mejor manejo por parte de los usuarios.

## **1.7 Patrones de Casos de Uso**

Un patrón “describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que se puede utilizar un millón de veces sin tener que hacer dos veces lo mismo” (Christopher, 1979).

Los patrones utilizados para el modelamiento y diseño de casos de uso reflejan los requisitos reales del sistema, haciendo más fácil el trabajo y mucho más simple su mantenimiento.

### **1.7.1 Patrón Inclusión Concreta:**

Se incluye una relación del caso de uso base al caso de uso de inclusión. El último puede ser instalado en sí mismo. El caso de uso base puede ser concreto o abstracto.

### **1.7.2 Patrón Múltiples Actores - Roles comunes:**

Puede suceder que los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso (Rumbaugh, y otros, 2000).

El patrón múltiples actores roles comunes será empleado en el modelado de diagramas de casos de uso necesarios para el correcto desarrollo de la solución informática, porque permite representar el comportamiento de los usuarios y las relaciones que se establecen entre ellos, mientras que el patrón de inclusión concreta será utilizado para definir la dependencia de diferentes casos de uso de otros más generales.

## 1.8 Herramienta para modelado de prototipos no funcionales

### 1.8.1 Axure RP Pro 5.5

Aplicación para crear prototipos y especificaciones muy precisas para páginas web que cuenta con la mayor cantidad de opciones que se puedan necesitar para el prototipado de la forma más eficiente.

Axure RP Pro 5.5 le permite al usuario componer la página web visualmente, añadiendo, quitando y modificando los elementos con suma facilidad.

Principales características de esta herramienta:

- Realiza el diseño del prototipo de manera eficiente con las características que requiere específicamente la futura aplicación.
- Ofrece la generación inmediata de prototipos interactivos.
- Permite el trabajo más fácil en un ambiente sencillo, con prototipos interactivos, anotaciones y especificaciones.

Existen librerías muy atractivas para Axure, además de las que se pueden crear para satisfacer las necesidades del prototipado, también ofrece una activa comunidad y tutoriales para aprender.

La especialización de Axure RP Pro 5.5 es en las anotaciones. En este punto, permite especificar el estado de cada elemento (Propuesto, Aceptado, Incorporado), el beneficio esperado (Crítico, Importante, Útil), el riesgo, la estabilidad, a quién va dirigido y a quién se le asignará la tarea (Axure, 2012).

La arquitectura del proyecto decide utilizar Axure RP Pro 5.5 como herramienta para modelado de prototipos no funcionales del SIT, pues posee todos los elementos necesarios para la creación de prototipos amigables que pueden ser exportados como un proyecto o como imágenes, de manera que facilitan el uso para el equipo de desarrollo. Igualmente, pueden generarse como páginas HTML dinámicas, funcionalidad que permite al usuario interactuar de forma directa con los prototipos, observando el flujo consecutivo de los eventos. Esta característica posibilita al usuario obtener una idea más clara de cómo será la aplicación en un futuro y de esta forma sentirse identificado y motivado con la solución a elaborar.

## 1.9 Aplicación Web

Es una solución informática en forma de página web, que tiene una base de datos asociada y que utiliza un navegador web que permite visualizar su interfaz a través de internet o una red interna (intranet) logrando el vínculo interactivo entre el usuario y la aplicación.

### **Ventajas:**

- Posibilita el acceso a varios usuarios simultáneamente al sistema.
- No requiere infraestructura de distribución por lo que tiene un precio menor que las aplicaciones de escritorio.
- Los datos están en línea, no es necesario encontrarse precisamente en la computadora en la que se instaló el software para tener acceso al mismo.
- Constituye un medio de promoción rápido y eficiente.
- Cuenta con una mayor inmediatez de acceso, estas no necesitan ser descargadas, instaladas, ni configuradas.
- Los datos y el procesamiento están centralizados en el servidor.
- Compatible con todos los sistemas operativos (Mora, 2002).

## 1.10 Herramienta para generación de reportes

El Generador de Reportes Dinámico es una aplicación web que tiene como objetivo la generación de informes de forma rápida e interactiva. Con el uso del mismo se pueden estandarizar procesos para confeccionar reportes independientemente del Sistema Gestor de Base de Datos que se utilice, ya sea MySQL, Oracle o PostgreSQL, lo que lo hace compatible con la arquitectura establecida para la solución informática de los Tribunales Populares Cubanos donde se define como gestor de Base de datos a PostgreSQL en su versión 8.4.

El Generador permite a los usuarios agilizar la toma de decisiones y generar reportes en varios formatos y con gran variedad de opciones en su diseño. El sistema está compuesto por varias aplicaciones entre las que se encuentran el Visor de reportes, Diseñador de modelos, Diseñador de reportes, Diseñador de consulta y Administrador de reporte (Colectivo de Profesores de DATEC, 2010).

### 1.10.1 Generador Dinámico de Reportes 1.7.0

Aplicación web compuesta por varias aplicaciones que están estrechamente relacionadas entre sí, permitiendo al usuario realizar primero el diseño y después la relación de los datos que se van a mostrar en el informe final. Esta herramienta presenta un módulo para la autenticación de usuarios que posibilita la seguridad y el control en el trabajo a realizar (Rodríguez Durán, 2012).

Se debe seleccionar el dominio de entidades al cual se quiere acceder y si lo desea hacer con el usuario que se autenticó anteriormente, sino se debe marcar la opción usuario diferente e introducir la contraseña a utilizar. Una vez autenticado, el usuario accederá a la página inicial de la aplicación como se muestra en la **Fig.4** donde podrá seleccionar el área en la cual trabajará.



**Fig.4. Página principal del Generador Dinámico de Reportes 1.7.0**

El Área de Trabajo del Visor de Reportes está estructurada, como muestra la **Fig.5**, por dos componentes fundamentales:

**Reportes existentes:** donde se muestran los reportes generados con anterioridad.

**Visor:** donde se visualiza el reporte seleccionado.

En esta área existen varias opciones que permiten al usuario agilizar procesos como por ejemplo la búsqueda de reportes y el filtrado por campos. Se cuenta además con las funcionalidades que permiten exportar el modelo al formato que se requiera e imprimir dichos informes.

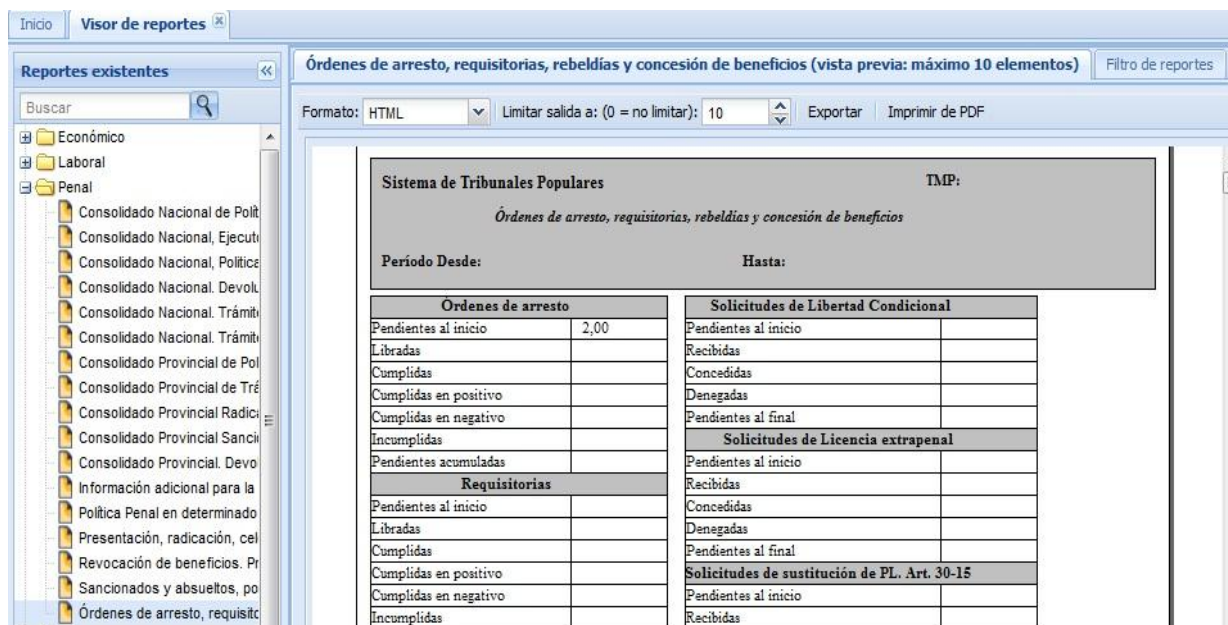


Fig.5. Área de trabajo del Visor de reportes

El paso previo al diseño de un modelo es poseer un origen de datos al cual conectarse para extraer los metadatos de las bases de datos clientes. El Área de Trabajo del Diseñador de Modelos está estructurada, como muestra la Fig.6, por dos componentes fundamentales: **Seleccionar un origen de datos** y **Modelos existentes**.

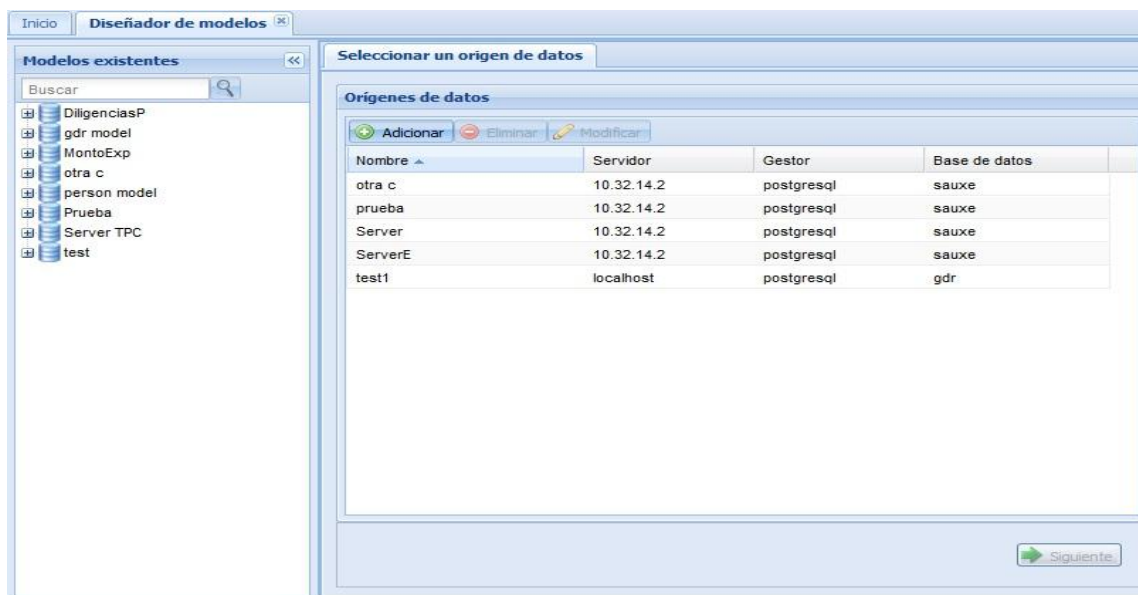


Fig.6. Área de Trabajo del Diseñador de Modelos

Uno de los pasos más importantes en la confección de los reportes es el diseño y para esto se establece el área Diseñador de Reportes la cual está estructurada, como muestra en las **Fig.7 y 8**, por tres componentes fundamentales:

**Paleta de componentes:** Donde se muestran todos los componentes y las propiedades que necesiten ser utilizadas.

**Área de diseño:** Es donde se encuentra cada uno de los espacios que tiene la estructura de un documento y soporta la utilización de los componentes en cada uno de los espacios.

**Inspector:** Aparece en forma de árbol la estructura de reporte o documento a realizar. En este componente se selecciona además la fuente de datos con que se va a trabajar en el diseño del reporte.

Con el Diseñador de reportes se puede obtener el diseño que se establece utilizando las propiedades que se brindan en la paleta de componentes, se pueden aplicar a estos modelos el formato de las plantillas que ya estén definidas, el origen de los datos queda establecido en el informe lo que posibilita la vista previa del mismo que sirve de guía al diseñador para verificar el resultado del trabajo.



**Fig.7. Área del Diseñador de Reportes, seleccionando reporte.**

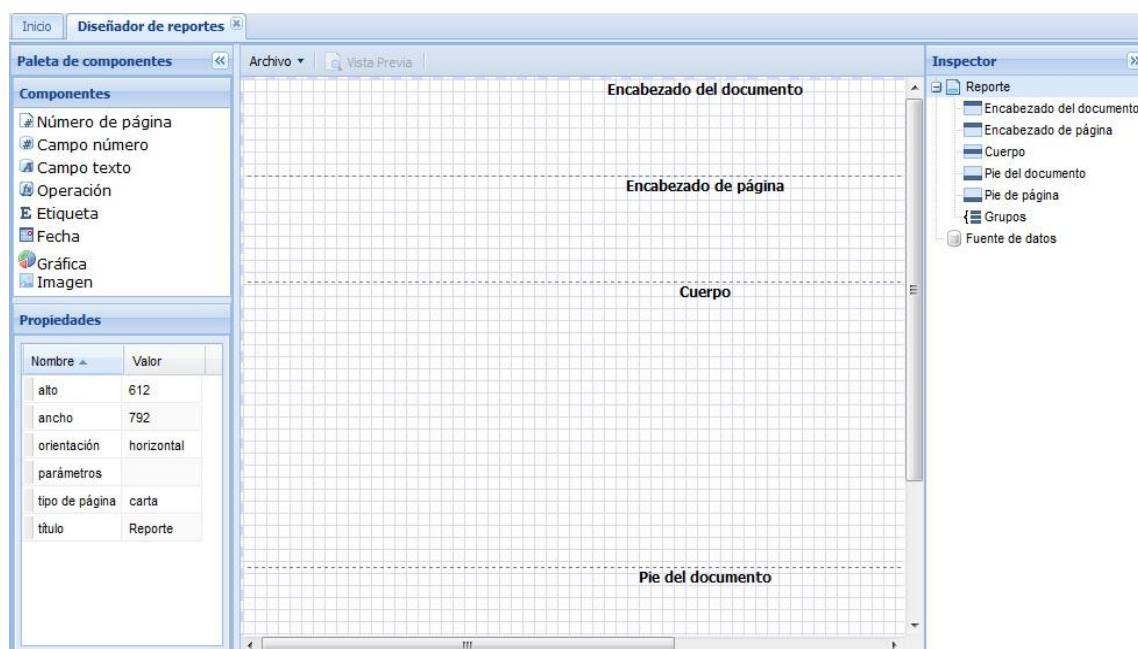


Fig.8. Área Diseñador de Reportes.

Posterior al diseño es necesaria la obtención de los datos a través de las diferentes consultas y procedimientos que se definan. El área Diseñador de Consultas está estructurada por dos componentes fundamentales como se muestra en la Fig.9:

**Área de desarrollo:** Área sobre la cual se diseñará la consulta, la cual consta de dos pestañas: Diseño y Editor de Consultas.

1. Diseño: Se establecen las consultas visualmente.
2. Editor de Consultas: Se verifica la sintaxis de la consulta realizada.

**Explorador de consultas:** muestra los modelos existentes y el árbol de construcción de la consulta que se diseñe.

El Diseñador de consultas permite realizar funciones, relacionar tablas y establecer condiciones de manera gráfica, contando además con la Vista SQL, Fig.10, que muestra el código SQL de la consulta en la cual se esté trabajando (Colectivo de Profesores de DATEC, 2010).



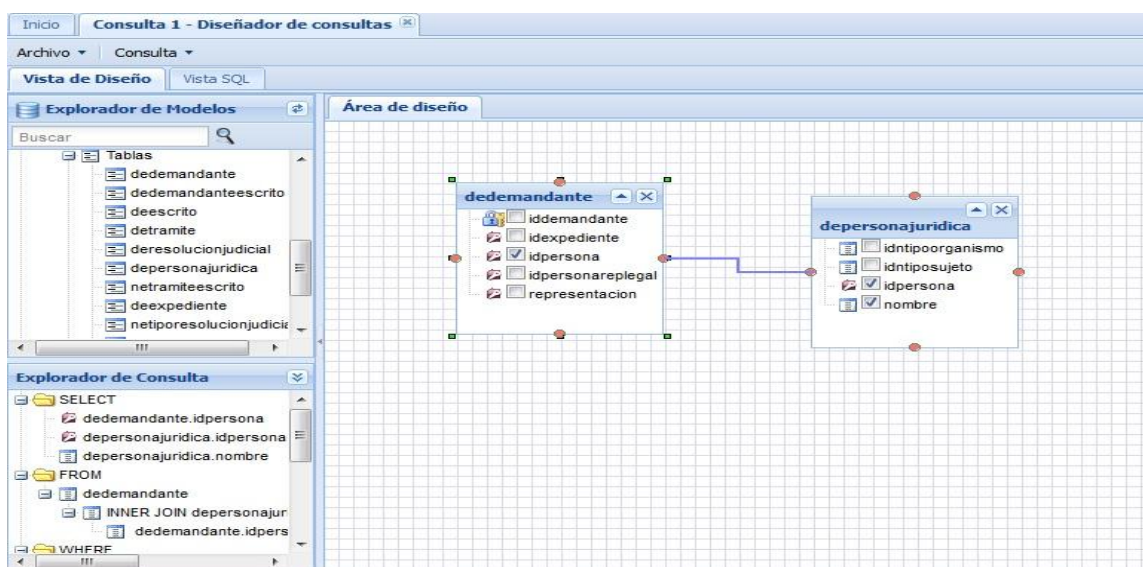


Fig.9. Área Diseñador de Consultas

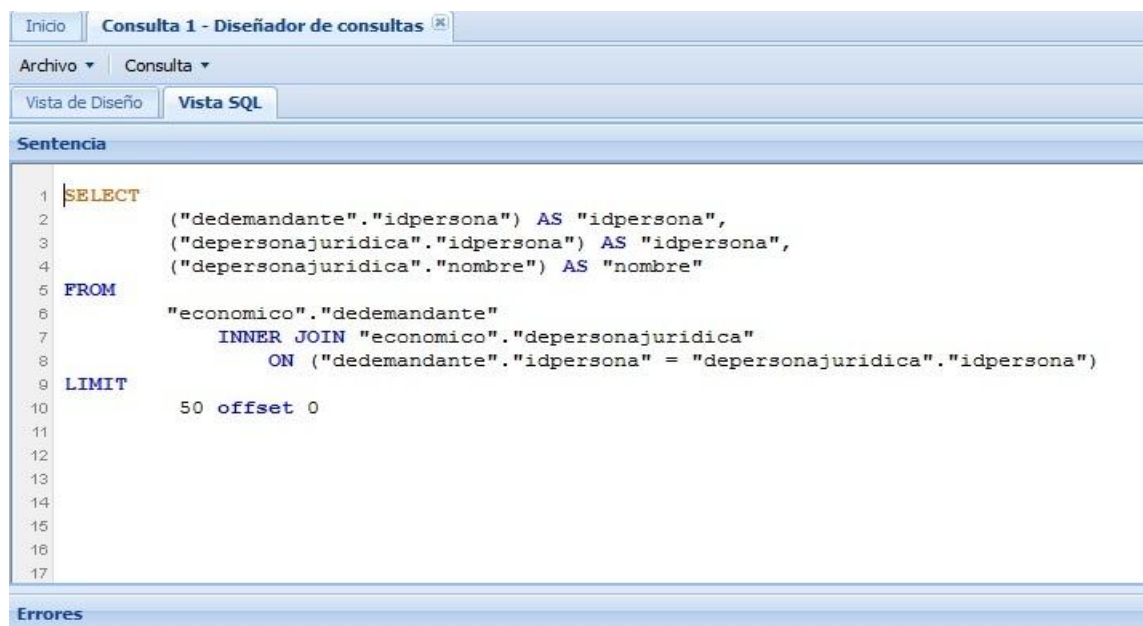


Fig.10. Área Diseñador de Consultas más la vista SQL

## 1.11 Servidor web Apache2

Apache es un servidor web muy utilizado en el mundo, la versión considerada estable del nuevo apache 2.0, incorpora muchas mejoras y novedades sobre la versión anterior. Es un

software de código abierto que funciona sobre cualquier plataforma. Entre sus ventajas se pueden mencionar las siguientes:

- ❖ Posee licencia libre, por lo cual se pueden realizar múltiples cambios para mejorar su rendimiento sin tener que pagar un centavo.
- ❖ Servidor altamente configurable de diseño modular: Se pueden escribir módulos para realizar determinadas funciones.
- ❖ Provee interfaz a todas las bases de datos.
- ❖ Es un servidor muy potente, que posee una gran seguridad y robustez (Reyes Lomeli, 2007).

## 1.12 Herramienta para Gestor de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

Analizando el concepto anterior se puede deducir que un Sistema Gestor o Manejador de Bases de Datos (SGBD) abarca al conjunto de programas que permiten a los usuarios crear y mantener una base de datos (BD), por lo tanto, el SGBD es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Estos sistemas están diseñados para gestionar grandes volúmenes de información. Algunas de las acciones que realizan son:

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.

Entre los gestores de Bases de datos más potentes se encuentran Oracle, DB2, PostgreSQL, MySQL, MS SQL Server, etc. (SQL Server 2008, 2011).

### 1.12.1 PostgreSQL 8.4

PostgreSQL 8.4 es un sistema de gestión de bases de dato objeto-relacional, diseñado para administrar grandes volúmenes de información. Incluye características propias de la orientación a objetos, como la herencia, tipos de datos, funciones, restricciones disparadoras, reglas e integridad transaccional (PostgreSQL-es, 2012).

PostgreSQL 8.4 es un poderoso sistema de código abierto que se encuentra entre los más avanzados del mundo, por estas y muchas otras características se usa de una forma sorprendente. En la Universidad de las Ciencias Informáticas (UCI) el uso de PostgreSQL 8.4 es muy extendido. Existen más de 16 proyectos productivos que trabajan con este gestor de bases de datos entre los cuales se puede mencionar el Sistema de Informatización para Los Tribunales Populares Cubanos.

#### Ventajas:

- Los lenguajes más populares del medio: PHP, C, C++, Perl, Python entre otros, le brindan soporte.
- Puede ser utilizado en los principales sistemas operativos: Linux, Unix, Mac OS, Beos y Windows.
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos espaciales y minería de datos.
- Incorpora funciones de diversa índole: manejo de fechas, reglas y soporte para el Protocolo de Capa de Conexión Segura (SSL, por sus siglas en inglés).
- Soporta el uso de índices, reglas y vistas (PostgreSQL-es, 2012).

### 1.12.2 PL/Pgsql

PL/Pgsql es un lenguaje estructurado en bloques. Como mínimo tendremos un bloque principal en nuestro procedimiento almacenado y dentro de este podremos tener subbloques. Un bloque se define de la siguiente manera (Todo entre los corchetes [] es opcional):

```
[<<etiqueta>>]
[DECLARE
  declaraciones de variables]
BEGIN
  codigo
END [etiqueta];
```

### **Ventajas de usar PL/Pgsql:**

SQL es el lenguaje estándar para realizar consultas a un servidor de base de datos. Cada sentencia SQL se ejecuta de manera individual por el servidor, lo cual implica que las aplicaciones cliente deben enviar cada consulta al servidor, esperar a que la procese, recibir los resultados, procesar los datos y después enviar la siguiente sentencia.

Al usar PL/Pgsql es posible realizar cálculos, manejo de cadenas y consultas dentro del servidor de la base de datos, combinando el poder de un lenguaje procedimental y la facilidad de uso de SQL, minimizando el tiempo de conexión entre el cliente y el servidor.

Desde PL/Pgsql se pueden realizar cálculos complejos y crear nuevos tipos de datos de usuario. Como un verdadero lenguaje de programación, dispone de estructuras de control repetitivas y condicionales, además de la posibilidad de creación de funciones que pueden ser llamadas en sentencias SQL normales o ejecutadas en eventos de tipo disparador (trigger).

Una de las principales ventajas de ejecutar programación en el servidor de base de datos es que las consultas y el resultado no tienen que ser transportadas entre el cliente y el servidor, ya que los datos residen en el propio servidor. Además, el gestor de base de datos puede planificar optimizaciones en la ejecución de la búsqueda y actualización de datos (Piloto, y otros, 2010).

### **Conclusiones del capítulo**

En el presente capítulo se han abordado los principales conceptos asociados al dominio del problema que posibilitan el mejor entendimiento de los procesos que se siguen en la metodología de software definida para el SIT. Se ha realizado el análisis del lenguaje de modelado empleado y de la herramienta CASE seleccionada para la aplicación durante todo el desarrollo del software. La Ingeniería de Requisitos definida y que será aplicada, permitirá la identificación de qué es lo que se quiere construir, además de organizar el trabajo mediante patrones de casos de uso. Se ha profundizado también en las características del Generador Dinámico de Reportes, software definido en la arquitectura del SIT que permite convertir en reportes estadísticos y no estadísticos toda la información recopilada en los procesos anteriores.

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

### **Introducción**

En el presente capítulo se realiza un análisis de las características del sistema a desarrollar a partir de la problemática planteada, se describe brevemente la solución propuesta, sus requisitos funcionales y no funcionales, así como los trabajadores que intervienen en ella. Se toman los conceptos, entidades y actores que se encuentran en el modelo de dominio para a partir de los mismos identificar y describir los diagramas de casos de uso y los prototipos de interfaz elaborados durante los flujos previos. También se obtienen los artefactos generados a partir de la aplicación de la metodología de desarrollo utilizada y una descripción de las entidades que tienen persistencia en la base de datos.

### **2.1 Descripción del sistema**

Por las características de los TPC, en los cuales se almacena un gran volumen de datos asociados a cada una de las materias que lo componen y además de brindar servicios a un gran número de usuarios, se hace necesario contar con un módulo de Reportes para llevar el control de las estadísticas. Cualquier trabajador que cuente con los permisos necesarios para realizar los reportes que brinda el módulo, podrá tener acceso a los mismos después de autenticarse en el sistema con su usuario y contraseña. Con el nuevo sistema se brinda la posibilidad de generar reportes de forma diaria, mensual y anual. El módulo ofrece la facilidad de filtrar rangos de fecha en dependencia del reporte que el usuario desee obtener. Además el sistema permitirá mostrar estos reportes en el mismo navegador y exportarlo en formato PDF o Excel para su posterior impresión.

Con la realización del presente trabajo el sistema a desarrollar deberá lograr un medio que facilite el cumplimiento de los objetivos estratégicos de los TPC en las diferentes Instancias, así como una mejor conservación de los documentos y un mejor acceso a la información.

### **2.2 Modelo de dominio**

Con la realización del modelo de dominio se pretende comprender los conceptos con que trabajan los usuarios y con los que la aplicación trabajará. Para ello fue necesaria la participación de los expertos del negocio, arribando a la necesidad de realizar las siguientes tareas:

- 1) Identificar las clases conceptuales.
- 2) Graficarlas en un diagrama de clases del dominio.
- 3) Añadir relaciones y atributos (Rumbaugh, y otros, 2000).

### 2.2.1 Conceptos asociados al modelo de dominio

**Sistema de tribunales:** Conjunto de todos los Tribunales Populares Cubanos que está conformado por instancias y estas a su vez por procedimientos, es el órgano principal que ejerce justicia.

**Instancia:** Forma de organización o estructura en que los Tribunales ejercen su jurisdicción en correspondencia con la división político-administrativa del país.

**TPP:** Tribunal Provincial Popular, instancia del sistema de tribunales que se encarga de resolver los conflictos dentro de los límites de su competencia.

**TMP:** Tribunal Municipal Popular, instancia del sistema de tribunales que se encarga de resolver los conflictos dentro de los límites de su competencia.

**Materia:** Disciplina que abarca cierta cantidad de procedimientos según su competencia.

**Económico:** Materia que se aplica a la solución de los litigios que se susciten entre entidades económicas ya sean personas naturales o jurídicas.

**Administrativo:** Materia con motivo de las decisiones que adoptan los órganos de la administración del estado.

**Civil:** Materia encargada de resolver los conflictos entre las personas naturales o jurídicas con motivo del patrimonio, el estado civil de las personas y las relaciones paterno-filiales entre otras.

**Penal:** Materia que se aplica a la solución de los conflictos que puedan surgir entre el estado y los individuos que infringen, violan las normas penales o por la posible comisión de delito.

**Laboral:** Materia que se aplica a la solución de los litigios que se susciten entre los trabajadores, o entre estos y las administraciones de las entidades laborales, abarca los procedimientos Disciplina y Derecho Laboral.

**DP:** Diligencia Previas, conjunto de actos procesales que se necesitan para resolver conflictos de la materia Económica.

**Administrativo:** Conjunto de actos procesales que se necesitan para resolver los conflictos de la materia Administrativa.

**Ordinario civil:** Procedimiento modelo en materia civil, conjunto de actos procesales que se necesitan para resolver los conflictos de esta materia.

**Ordinario penal:** Conjunto de actos procesales que se necesitan para resolver los conflictos que regulan el procedimiento básico modelo en materia penal con motivo de la comisión de delitos sancionables.

**Disciplina laboral:** Conjunto de actos procesales que se necesitan para resolver los conflictos de esta materia.

**Derecho laboral:** Conjunto de actos procesales que se necesitan para resolver los conflictos de esta materia.

**Usuario revisor:** Persona encargada de consultar los reportes en las diferentes Instancias.

**Presidente de sala/sección:** Usuario revisor que se encarga de visualizar sus reportes y los pertenecientes a los jueces y secretarías, será de sala o sección según la materia en la que se encuentre.

**Juez ponente:** Usuario revisor que se encarga de visualizar determinados reportes dentro de su competencia.

**Secretaria:** Usuario revisor que se encarga de visualizar determinados reportes dentro de su competencia.

**Reportes:** Subsistema conformado por los reportes estadísticos y no estadísticos contenidos en los procedimientos de las diferentes materias.

## 2.2.2 Descripción del modelo de dominio

El Sistema de Tribunales está compuesto por diferentes Instancias, entre las que se encuentra el Tribunal Supremo Popular, los Tribunales Provinciales Populares y los Tribunales Municipales Populares. En el alcance definido para el desarrollo de la fase 1 del presente trabajo de diploma solamente se tienen en cuenta las instancias municipales y provinciales. En estas áreas los procedimientos son llevados a cabo por el presidente de sala o sección en dependencia de la instancia, el juez ponente y la secretaria, comportándose los mismos como usuarios revisores dependiendo de los permisos asociados a la competencia de cada uno. Las instancias provinciales tienen diferentes materias entre las cuales se encuentran la Económica con el procedimiento de Diligencias Previas y la Administrativa con el procedimiento

Administrativo. En el caso de las instancias municipales las materias que la componen son la Civil y la Penal con sus procedimientos Ordinarios y la materia Laboral con sus procedimientos de Disciplina y Derecho Laboral. Todos estos procesos utilizan el módulo de Reportes para la gestión de los diferentes informes estadísticos y no estadísticos que se requieran en las materias.

### 2.2.3 Diagrama de objetos del modelo de dominio

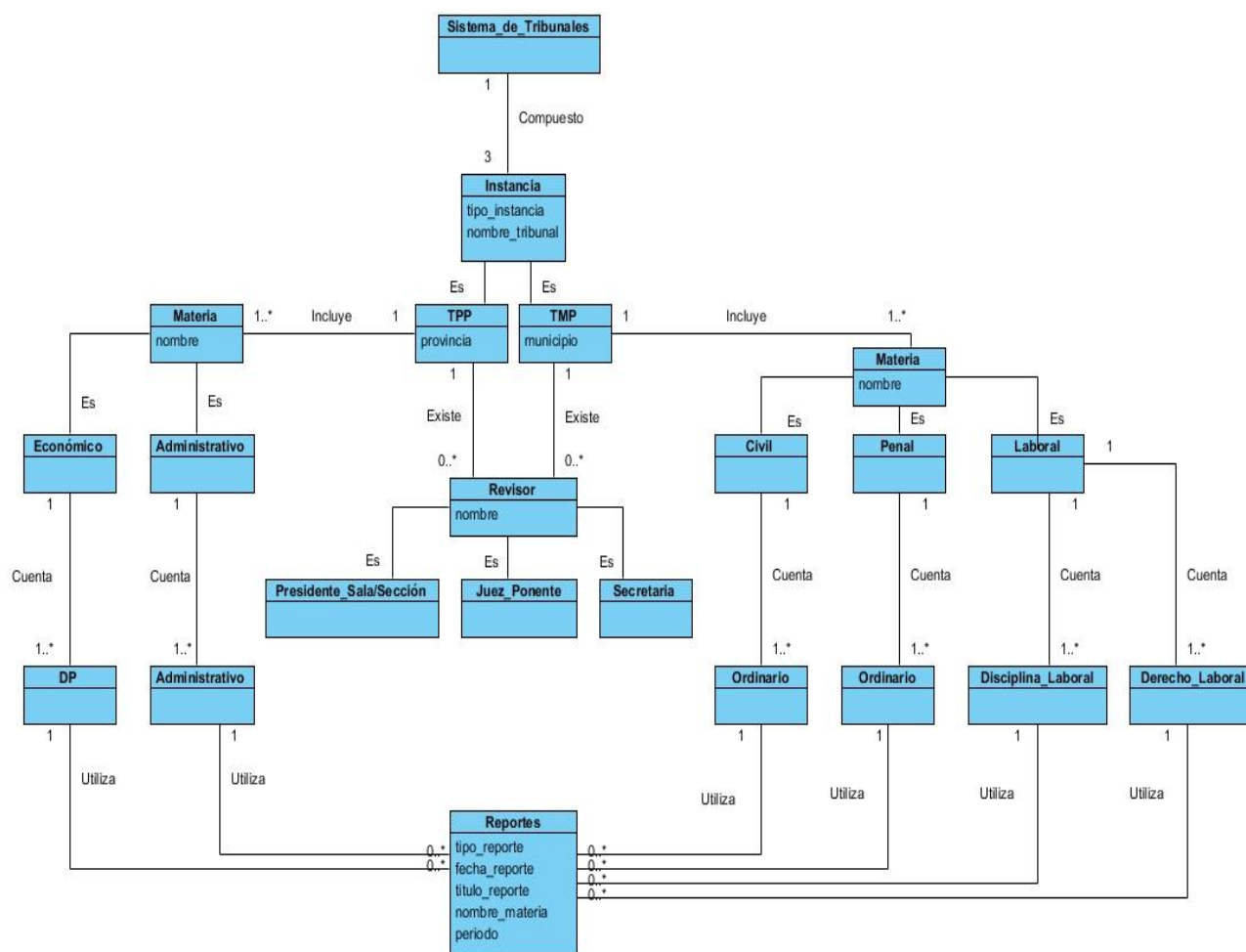


Fig.11. Modelo de dominio

## 2.3 Especificación de requisitos funcionales

Para lograr la correcta obtención de los requisitos funcionales del sistema se realizó la elicitación de los mismos a través de entrevistas abiertas realizadas al cliente, en las que se



estableció un intercambio de opiniones y preguntas (Ver Anexo 1), para obtener un mejor entendimiento de las necesidades del cliente. Luego de concluida esta actividad se generó el documento de especificación de requisitos de software, artefacto que se encuentra en el Expediente de Proyecto de TPC, y en el cual están contenidos los ochenta y cuatro requisitos identificados durante los encuentros con el cliente (TPC, 2011).

A continuación se muestran las especificaciones de tres de los requisitos que debe cumplir el subsistema Reportes en la materia económica para su procedimiento Diligencias Previas, para los que se definió el CU Visualizar Reporte Diligencias Previas del Proceso Ejecutivo. Todos los requisitos del subsistema cuentan con el mismo formato que el caso de uso en cuestión.

#### **RF.44 Visualizar Reporte Diligencias Previas del Proceso Ejecutivo**

El sistema permitirá mostrar por sujeto demandante los siguientes datos en forma de tabla:

- Pendientes al Inicio (numérico, automático)
- Radicadas
  - Total (numérico, automático)
  - Confesión de Deuda (numérico, automático)
  - Reconocimiento de Firma (numérico, automático)
- Total a Resolver (numérico, automático)
- Resolución
  - Inadmisibles (numérico, automático)
  - Desistidas (numérico, automático)
  - Declaradas Preparadas (numérico, automático)
  - Denegadas (numérico, automático)
- Total Resuelto (numérico, automático)
- Pendientes al Final (numérico, automático)

Además el sistema permitirá mostrar todos los sujetos demandantes, que son los siguientes:

- Grupos, Uniones y Empresas Estatales (texto, automático)
- Empresas Estatales subordinadas a un mismo Organismo o Grupo Empresarial (texto, automático)
- Unidades Presupuestadas (texto, automático)
- Empresas Extranjeras (texto, automático)
- Empresas Mixtas (texto, automático)

- Sociedades Mercantiles Cubanas (texto, automático)
- Cooperativas Agropecuarias (texto, automático)
- Unidades Básicas de Producción Cooperativa (texto, automático)
- Agricultores Pequeños (texto, automático)
- Otras personas naturales (texto, automático)
- Total de Diligencias Previas (texto, automático)

**Prioridad:** Alta

#### **RF. 78 Registrar Período de Búsqueda**

El sistema permitirá registrar el período de tiempo que define la búsqueda del reporte deseado, para ello solicita los siguientes datos:

- Período de tiempo
  - Desde - Fecha Inicio (fecha)
  - Hasta - Fecha Fin (fecha)

**Prioridad:** Alta

#### **RF.81 Mostrar Parámetros para Exportar el Reporte**

El sistema permitirá al usuario seleccionar el formato, la cantidad de páginas y otras opciones en que se desee ver el reporte, para ello muestra los siguientes datos:

- Formato (HTML, PDF, CSV, EXCEL)
- Límite de Salida (5, 10, 15)
- Imprimir
- Exportar

**Prioridad:** Media

## **2.4 Descripción de actores del sistema**

Los actores de Reportes son los mismos para todos los módulos que éste abarca, sólo varía el presidente que será de Sala o Sección, según la instancia en la que se encuentre, TPP o TMP, respectivamente. A continuación se muestra la descripción de cada actor y su función dentro del subsistema Reportes, en la materia Económica.

**Tabla 1: Descripción de los actores del sistema**

Actor	Descripción
Secretaria	<p>Actor encargado de tramitar las decisiones del tribunal. Podrá realizar, dentro de su competencia, las siguientes acciones en el sistema:</p> <ul style="list-style-type: none"> <li>• Buscar los reportes</li> <li>• Visualizar los reportes</li> <li>• Exportar los reportes</li> </ul>
Juez Ponente	<p>Actor encargado de llevar los procesos en el tribunal. Podrá realizar, dentro de su competencia, las siguientes acciones en el sistema:</p> <ul style="list-style-type: none"> <li>• Buscar los reportes</li> <li>• Visualizar los reportes</li> <li>• Exportar los reportes</li> </ul>
Presidente de Sala	<p>Actor encargado de controlar sus reportes y los de otros usuarios. Podrá realizar, dentro de su competencia, las siguientes acciones en el sistema:</p> <ul style="list-style-type: none"> <li>• Buscar los reportes</li> <li>• Visualizar los reportes</li> <li>• Exportar los reportes</li> </ul>
Revisor	<p>Es una generalización de los actores Secretaria, Presidente de sala y Juez Ponente. Podrá realizar las siguientes acciones en el sistema:</p> <ul style="list-style-type: none"> <li>• Buscar los reportes</li> <li>• Visualizar los reportes</li> <li>• Exportar los reportes</li> </ul>

## 2.5 Diagrama de casos de uso del sistema

En el diagrama de casos de uso del sistema que se muestra en la **Fig. 12**, perteneciente al módulo económico se muestran los reportes que se visualizan en dicha materia y la relación de los actores que interactúan con el sistema.

En este diagrama se puede apreciar el uso adecuado de los patrones estudiados anteriormente, el patrón múltiples actores roles comunes se puede evidenciar en el actor Revisor, el cual generaliza los demás actores que interactúan con los reportes debido a que no realizan ninguna acción en específico dentro del sistema, **Fig. 13**, y el patrón de inclusión concreta se puede apreciar estableciendo la dependencia del caso base Listar Reportes, **Fig. 14**. Para cada uno de los módulos del SIT ha quedado definido un diagrama de casos de uso similar con los reportes especificados en cada materia (TPC, 2011).

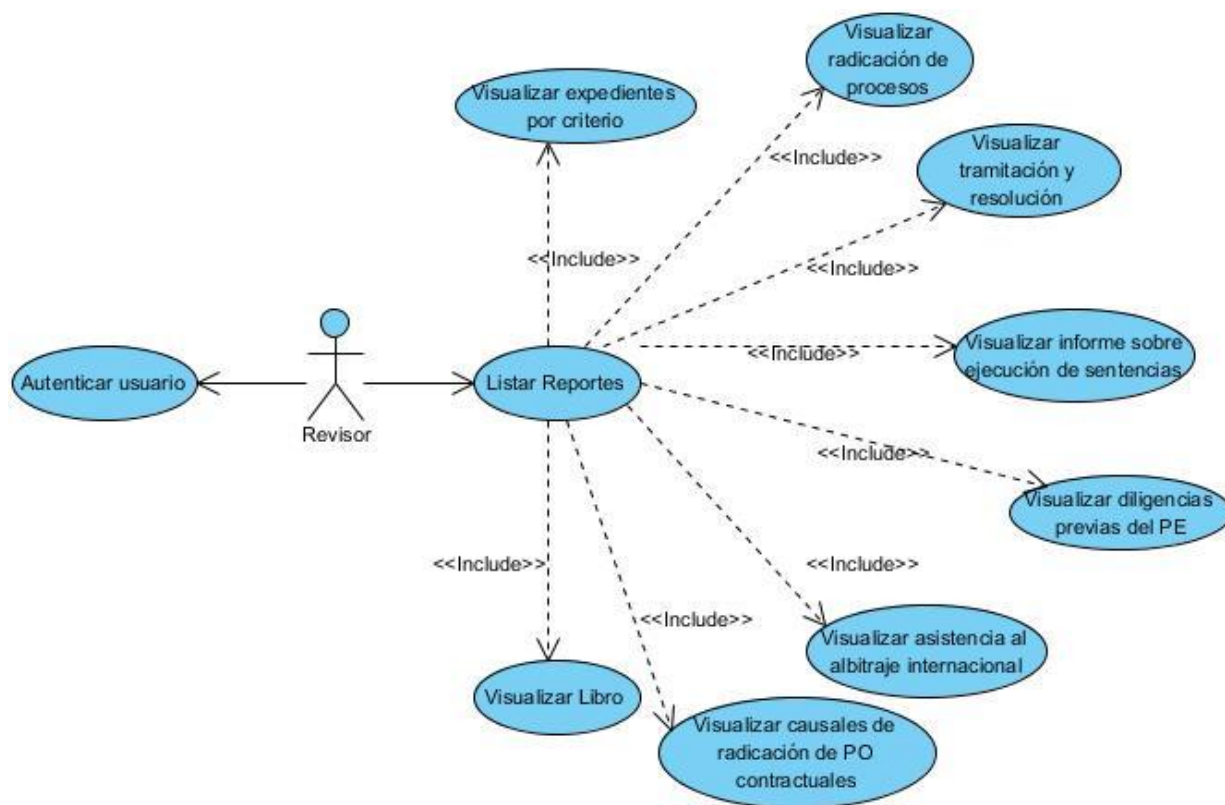


Fig.12. Modelo de casos de uso del sistema del módulo económico

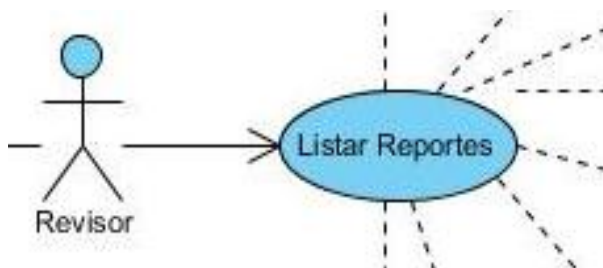


Fig.13. Presencia del patrón múltiples actores roles comunes

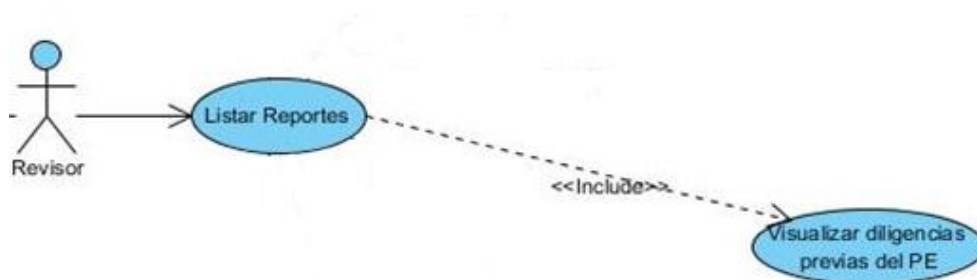


Fig.14. Presencia del patrón de inclusión concreta

## 2.6 Descripción de casos de uso del sistema

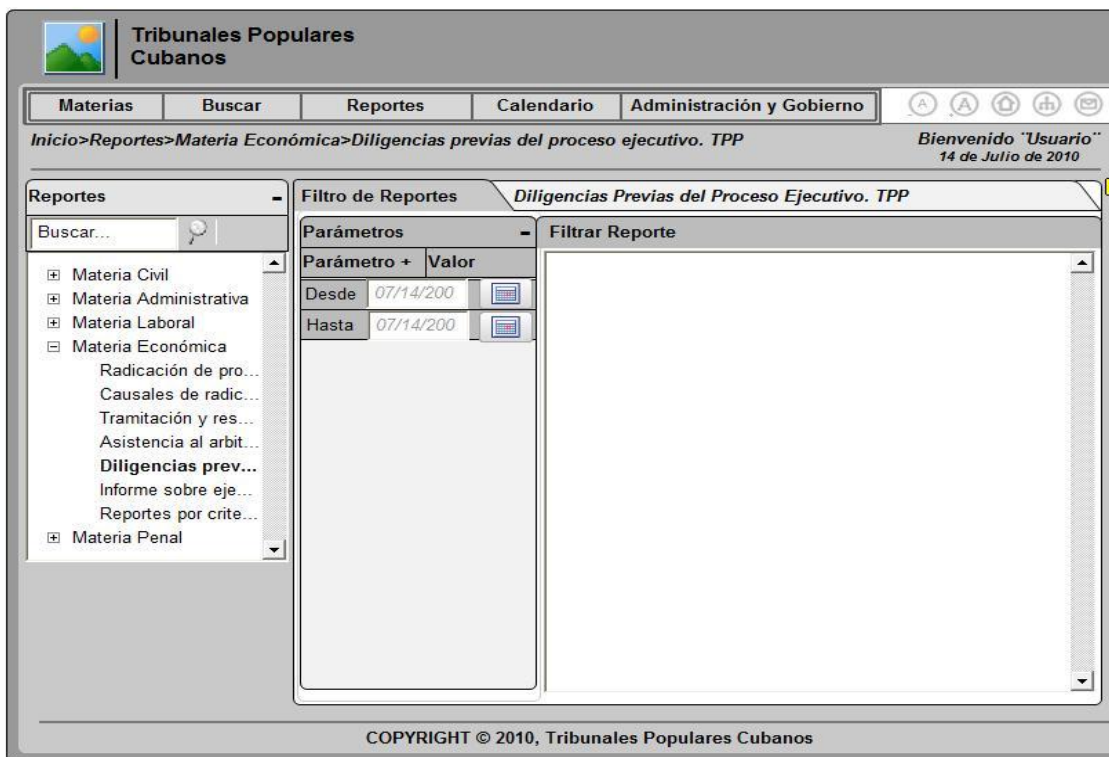
A continuación se muestra la descripción del caso de uso Visualizar Reporte de Diligencias Previas del Proceso Ejecutivo, esta información se encuentra en el documento Modelo de casos de uso del sistema referente al módulo económico, artefacto generado durante la fase de inicio luego de realizadas las actividades de identificación y clasificación de los requisitos correspondientes al flujo de trabajo levantamiento de requisitos que propone RUP. La descripción de todos los casos de uso del sistema se ha realizado utilizando pautas similares al de la **Tabla 2**, y se encuentra en cada modelo de casos de uso del sistema de los distintos módulos que utilizan la generación de reportes en el expediente de proyecto (TPC, 2011).

**Tabla 2: Descripción de Casos de Uso del Sistema**

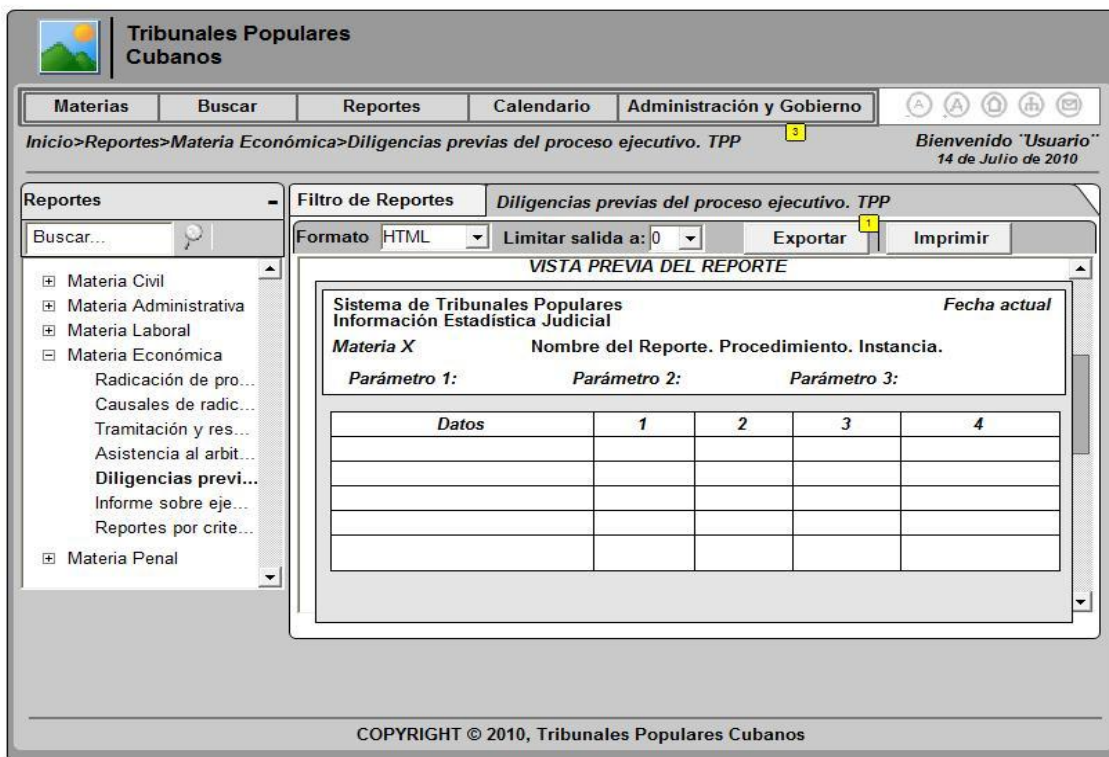
<b>Caso de Uso:</b>	Visualizar Reporte Diligencias Previas del Proceso Ejecutivo.
<b>Actores:</b>	Revisor.
<b>Resumen:</b>	El caso de uso se inicia cuando el revisor selecciona de los reportes que se muestran en cada materia, el que desee visualizar según sus permisos. Para ello debe introducir los parámetros que desee,

	el sistema mostrará al revisor los datos del reporte seleccionado de acuerdo a los parámetros introducidos. El caso de uso termina cuando finalmente se muestra el reporte con las preferencias que el revisor seleccionó.
<b>Precondiciones:</b>	El sistema debe estar instalado y ejecutándose correctamente. El Revisor debe estar autenticado con los permisos necesarios.
<b>Referencias</b>	RF.44, RF.78, RF.81
<b>Prioridad</b>	Alta
<b>Flujo Normal de Eventos</b>	
<b>Sección “Principal”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1- El revisor selecciona el reporte que desea visualizar.	2- El sistema permite al revisor introducir el(los) parámetro(s) por el(los) cual(es) desea filtrar el reporte seleccionado anteriormente: <ul style="list-style-type: none"> <li>• Período de tiempo (Obligatorio) <ul style="list-style-type: none"> <li>○ Desde (Fecha Inicio)</li> <li>○ Hasta (Fecha Fin)</li> </ul> </li> </ul>
3- El revisor selecciona la pestaña con el nombre del reporte.	4- El sistema muestra una nueva interfaz en la cual se visualiza el reporte filtrado según los parámetros señalados, además de varias opciones que permiten modificar la salida del reporte: <ul style="list-style-type: none"> <li>• Formato. <ul style="list-style-type: none"> <li>○ HTML</li> <li>○ PDF</li> <li>○ CSV</li> <li>○ EXCEL</li> </ul> </li> <li>• Límite de Salida. <ul style="list-style-type: none"> <li>○ 5</li> <li>○ 10</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ 15</li> <li>• Imprimir</li> <li>• Exportar</li> </ul>
5- El revisor selecciona el formato en que desea visualizar el reporte. 6- El revisor selecciona la opción Exportar.	7- El sistema muestra el reporte fuera de la interfaz en el formato deseado.  Terminando así el CU.
<b>Flujos Alternos</b>	
<b>Flujo alternativo 1 “Imprimir reporte”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
6.1- El revisor selecciona la opción Imprimir.	7- El sistema imprime el reporte. Terminando así el CU.
<b>Flujo alternativo 2 “Introducir datos incorrectos y/o dejar campos vacíos”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
3.1- El revisor introduce datos incorrectos o deja campos en blanco.	3.2- El sistema muestra una ventana informando que: “Los datos introducidos son incorrectos y/o ha dejado campos sin llenar”, y el botón Aceptar.
3.3- El revisor da clic en la opción Aceptar. Continúa el flujo normal de eventos en el paso 4.	
<b>Flujo alternativo 3 “Salir del Sistema”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
*. El revisor decide salir del sistema seleccionando la opción cerrar.	*. El sistema cierra la interfaz.
<b>Prototipos de Interfaz</b>	




Prototipo de Interfaz: Diligencias Previas del Proceso Ejecutivo. (Principal)





*Prototipo de Interfaz: Diligencias Previas del Proceso Ejecutivo.*

<b>Sistema de Tribunales Populares</b> Información Estadística Judicial Materia Económico Tribunal Provincial Popular de: Playa      Período Desde: 3-4-2010      Hasta: 5-4-2010										viernes, 16 de marzo de 2012	
<b>Diligencias Previas del Proceso Ejecutivo. TPP.</b>											
Sujeto Demandante	Pendientes al Inicio	Radicadas			Total a Resolver	Resolución				Total resuelto	Pendientes al final
		Total	Confesión de deuda	Reconoc de firma		Inadmisibles	Desistidas	Declaradas preparadas	Denegadas		
Grupos, Uniones y Empresas Estatales											
Empresas Estatales subordinadas a un mismo Organismo o Grupo Empresarial.											
Unidades Presupuestadas.											
Empresas Extranjeras.											
Empresas Mixtas.											
Sociedades Mercantiles Cubanas											
Cooperativas Agropecuarias.											
Unidades Básicas de Producción Cooperativa											
Agricultores Pequeños											
Otras personas naturales											
Total de Diligencias Previas											
Cerrar 											

*Prototipo de Interfaz: Diligencias Previas del Proceso Ejecutivo.(Reporte)*

Poscondiciones      El sistema muestra el reporte.

## 2.7 Especificación de requisitos no funcionales

En el módulo Reportes del proyecto TPC se identificaron un total de 16 requisitos no funcionales que están en correspondencia con las propiedades o cualidades que el producto debe tener para que sea atractivo, usable, rápido y confiable.

### 2.7.1 Apariencia o Interfaz Externa

El producto final debe tener una interfaz fácil de usar y amigable con un ambiente acorde a los principios de trabajo de los TPC y sus respectivas Instancias. Estará diseñado para la resolución deseada por el usuario, aunque debe de soportar un estándar de 800 x 600 píxeles. Debe tener imágenes acordes a las funciones que se ejercen en las Instancias de los TPC.

### **2.7.2 Usabilidad**

El diseño será limpio y claro, lo que permitirá la facilidad de navegación. Además de que se podrá contar con una ayuda del módulo en línea, perteneciente al generador dinámico para guiar en el uso de la interfaz.

### **2.7.3 Fiabilidad**

El sistema deberá estar disponible las 24 horas durante los 7 días de la semana, se le deben realizar mantenimientos preventivos en un horario estipulado o los fines de semana para no afectar la disponibilidad en el horario laboral. Solo se accederá a la base de datos desde la aplicación y los usuarios del sistema que posean los permisos suficientes serán los únicos que tendrán acceso a las funcionalidades. Se deberá garantizar la consistencia de los datos a través de la realización de comprobaciones y validaciones automáticas en todos los casos posibles.

### **2.7.4 Requisitos de Software**

El servidor para instalar la aplicación debe tener GNU/Linux o preferiblemente Ubuntu 8.4 o superior como Sistema Operativo (SO). Debe tener los paquetes: apache2, php5, libapache2-mod-php5, php5-cli, php5-mysql, php5-pgsql, php5-sqlite, php5-xsl, php5-gd. El servidor donde se instalará la Base de Datos del sistema debe tener como SO a GNU/Linux preferentemente Ubuntu GNU/Linux 8.04 o superior, la versión 8.3 del PostgreSQL o superior que debe estar bien configurada para aceptar conexiones vía TCP/IP utilizando el método de autenticación por md5 y PAdmin III.

### **2.7.5 Seguridad**

Para lograr el acceso a la información de las personas autorizadas se debe crear una jerarquía de usuarios, los cuales deben ser manejados por administradores a cada uno de los niveles. Las acciones que podrá realizar un usuario estarán determinadas por los roles que tendrá el usuario y el cual pertenecerá a una entidad específica. La información que maneja el sistema siempre va ser restringida de acuerdo a la entidad donde se originó, permitiendo solo el acceso de la misma a dicha entidad. De manera general después de ser integrado el GDR al Sistema de Informatización de Tribunales la seguridad del mismo estará determinada por las

características del marco de trabajo y los elementos de la solución informática definidos para cada uno de los subsistemas que la componen (TPC, 2011).

## 2.8 Modelo de diseño

En el diseño se determina la arquitectura general del sistema adaptando la especificación realizada en la etapa anterior. En esta fase se establece el comportamiento dinámico del sistema, es decir, cómo debe reaccionar ante los acontecimientos. El resultado obtenido de la etapa de Diseño facilita la posterior implementación, pues proporciona la estructura básica del sistema y cómo los diferentes componentes actúan y se relacionan entre sí.

En el flujo de trabajo de Análisis y Diseño, definido por RUP, se describe cómo se debe implementar el sistema, como resultado de un análisis realizado a los requerimientos no funcionales del mismo. El diseño no debe tener ambigüedades para que el modelo final que se obtenga sea suficiente para la implementación. El modelo de diseño es el resultado más importante de este flujo de trabajo, además describe la realización de los casos de uso (Pressman, 2007).

### 2.8.1 Diagramas de clases del diseño

Anteriormente se ha mencionado el uso de los diferentes subsistemas que conforman al GDR 1.7.0, haciéndose énfasis en el área Diseñador de reportes porque permite el diseño, generación y publicación de los reportes y es una de las más utilizadas. En la **Fig. 15** se representa el diagrama de clases para la realización de los casos de uso relacionados con la etapa de diseño mediante el Diseñador de reportes. El fichero Factory.js es el encargado de construir el workflow.js que envía las peticiones al controlador. Entre las entidades utilizadas se encuentra Category, que guarda la categoría a la cual pertenece un reporte, Report, guarda el XML de definición del reporte y Template, se refiere a las plantillas que pueden estar asociadas a algunos reportes ( Infante Frometa, y otros, 2009).

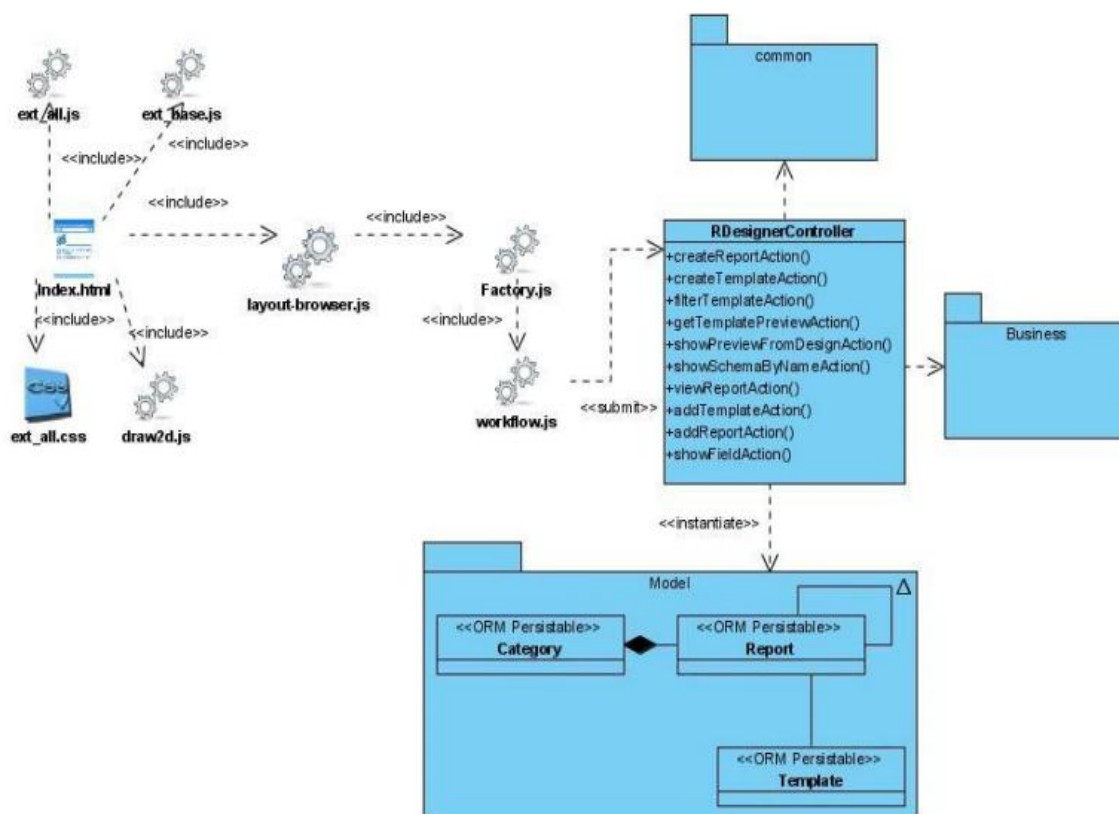


Fig.15. Diagrama de clases diseñador de reportes.

Otra de las áreas con la que se interactúa directamente para visualizar y obtener los informes, las interfaces y otros servicios después de ser diseñados e implementados es el Visor de reportes. El controlador de la capa de presentación, `layout_browser.js`, recibe las peticiones para crear las interfaces de este módulo, carga la clase principal `main.js` que invoca a `report.js`, `parameters.js` y `report_viewer.js` que son los encargados de construir las interfaces correspondientes y realizar las peticiones necesarias al controlador según las necesidades del usuario. El controlador del negocio recibe dichas solicitudes, ejecuta las acciones representadas en la **Fig. 16** en la clase `RViewerController`, que para su realización requiere la interacción con las clases del modelo, los objetos de la lógica de negocio y funciones y librerías comunes que se encuentran en el paquete `common` ( Infante Frometa, y otros, 2009).

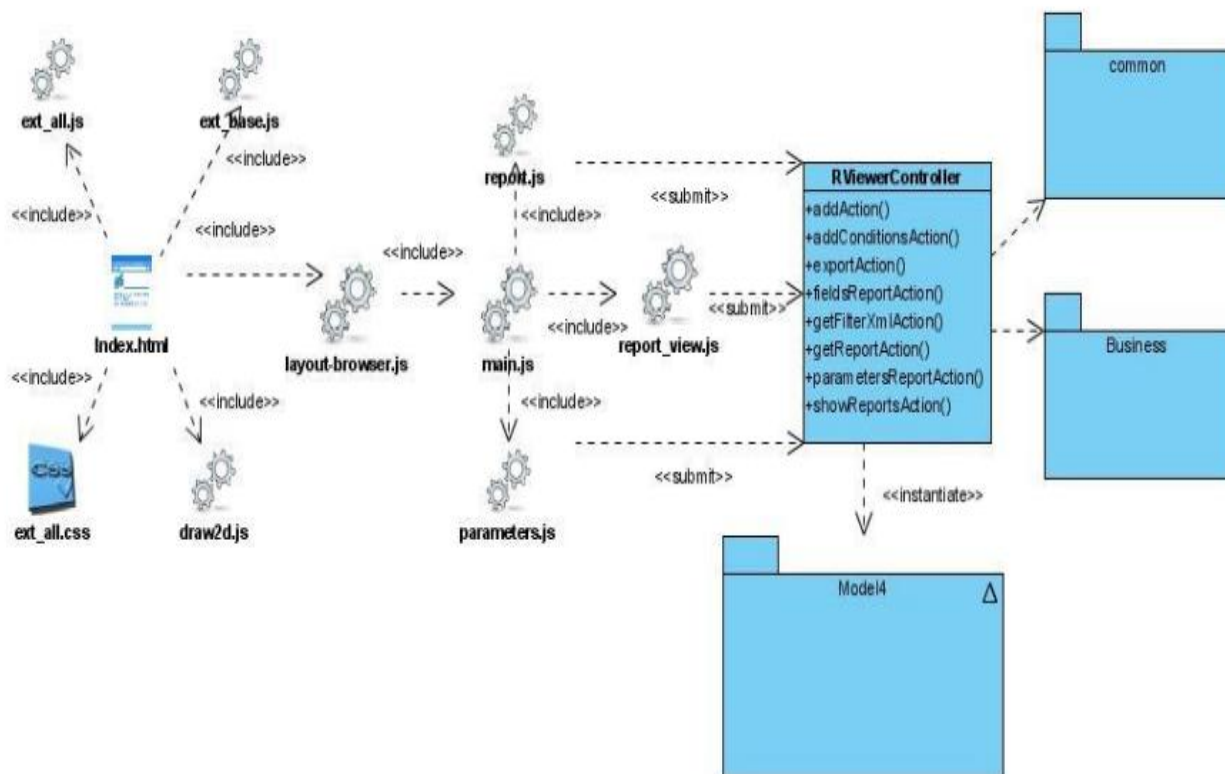


Fig.16. Diagrama de clases Visor de reportes.

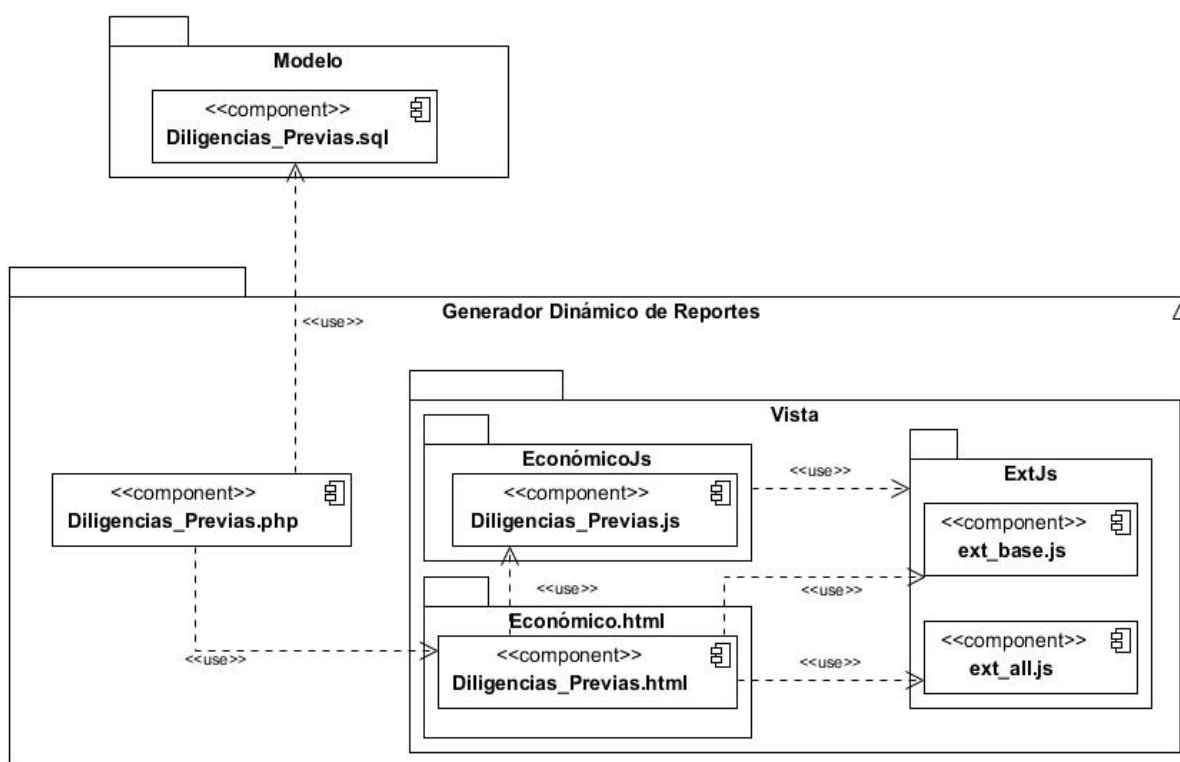
El uso de la herramienta GDR deja prácticamente inexistente la realización de un diseño que se ajuste a la descripción del flujo de actividades que se realizan para obtener los reportes, por lo que es importante destacar en este punto, que el presente trabajo hace uso de este diseño, sin proponer variación alguna, porque las funcionalidades del diseñador y el visor de reportes como herramientas, son indispensables para realizar los casos de uso levantados en esta disciplina independientemente de que el funcionamiento interno de dicho software no esté contenido en el alcance de la investigación.

## 2.9 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Proveen una vista arquitectónica de alto nivel del sistema y ayudan a los desarrolladores a visualizar el camino de la implementación. Son utilizados para mostrar las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de la compilación, dependencias entre elementos de

implementación y los correspondientes elementos de diseños que son implementados. Los componentes representan los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas y las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente se refiere a los servicios ofrecidos por otro componente (Pressman, 2007).

De acuerdo con la arquitectura que brinda soporte a la aplicación, se elaboró el diagrama de componentes que se muestra en la **Fig. 17**. En él se puede apreciar la interrelación entre cada uno de los componentes utilizados en solución a las funcionalidades del Módulo de Reportes.



**Fig.17. Modelo de componentes para el CU Visualizar Reporte Diligencias Previas del Proceso Ejecutivo**

### 2.9.1 Descripción del diagrama de componentes

El componente Diligencias\_Previas.js es el que interactúa con los componentes de la Librería ExtJs para crear una instancia de la vista del reporte Diligencias Previas y de esta manera interactuar con el controlador indicado para atender las peticiones que se hacen desde la vista que permiten obtener el reporte. Todo este proceso se realiza en el Generador Dinámico de

Reportes de forma transparente al que diseña los informes, haciéndose más evidente la petición del controlador al componente `Diligencias_Previas.sql` que se encuentra en el modelo y donde se define la función implementada en PL/Pgsql a través de la cual se interactúa con la BD para devolver los datos necesarios del reporte en cuestión.

## 2.10 Diagrama de despliegue

Un Diagrama de Despliegue es un grafo de nodos unidos por conexiones de comunicación que muestra las relaciones de estos nodos que componen el sistema y el reparto de los componentes sobre los mismos. Los diagramas de despliegue revelan la configuración en funcionamiento del sistema, incluyendo su hardware y su software. Para cada componente se deben documentar las características técnicas demandadas, el tráfico de red esperado y el tiempo de respuesta requerido. Estos diagramas describen la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software (Alva, 2012).

A continuación se muestra en la **Fig. 18** el Diagrama de Despliegue del Sistema de Informatización de los Tribunales Populares Cubanos donde se puede apreciar la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. La máquina cliente se conectará al servidor de aplicaciones web, encargado de manejar las funcionalidades del sistema y a una impresora desde donde se podrán imprimir los diferentes reportes que se visualicen, el servidor de aplicaciones web se conectará al servidor de base de datos, de donde obtendrá la información necesaria para el cumplimiento de las funcionalidades. Los requerimientos de hardware y los componentes que se ejecutan en cada nodo quedan especificados en la versión 1.0 del artefacto modelo de despliegue (TPC, 2012).

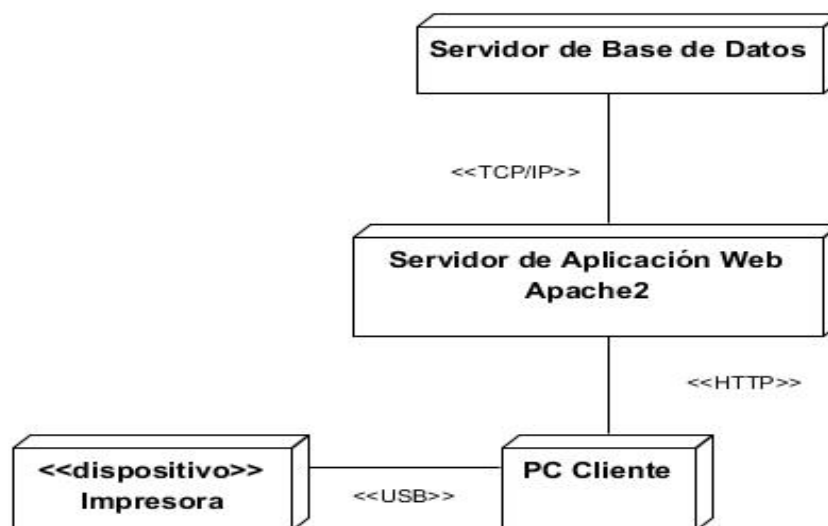


Fig.18. Modelo de despliegue

## 2.11 Desarrollo de los casos de uso

El desarrollo de los Casos de Uso definidos en cada materia se ha realizado siguiendo los pasos que se describen posteriormente y haciendo uso de los mismos componentes del Generador Dinámico de Reportes. Después de realizada la descripción de los casos de uso, donde se obtienen los prototipos de interfaz correspondientes se utiliza la herramienta GDR en su versión 1.7.0 para diseñar dicho modelo en el área diseñador de reportes de la cual se realizó un análisis en el Capítulo 1.

El diseño del informe se hace en total correspondencia con el prototipo definido en el caso de uso ya que el mismo se realiza de forma gráfica, especificándose en cada caso las medidas estándares para cada parte en la que se divide el documento. Se puede tener un mejor control en el trabajo verificándose la vista previa del reporte cuando se termine el trabajo con las etiquetas y campos de texto. Posterior al diseño se obtiene de la fuente de datos la función implementada en el lenguaje PL/PGSQL que devuelve la información estadística o no estadística que le corresponde al modelo y se comprueba la veracidad de los datos en el área de trabajo Visor de Reportes, la cual está conformada por dos componentes, los reportes existentes y el visor; ésta es el área principal con la que se relaciona el usuario, posee varias opciones que le permiten realizar las búsquedas y el filtrado con mayor rapidez para exportar posteriormente el reporte en los diferentes formatos que se proponen.



Como resultado de la descripción anterior en la **Fig. 19** se muestra el reporte Diligencias Previas del Proceso Ejecutivo en formato HTML.

Sujeito Demandante	Pendientes al Inicio	Radicadas			Total a Resolver	Resolución				Total resuelto	Pendientes al final
		Total	Confesión de deuda	Reconoc de firma		Inadmisibles	Desistidas	Declaradas preparadas	Denegadas		
Grupos, Uniones y Empresas Estatales	10	5	2	3	15	6	0	1	6	13	2
Empresas Estatales subordinadas a un mismo Organismo o Grupo Empresarial.	6	2	0	2	8	4	1	1	0	6	2
Unidades Presupuestadas.	3	1	1	0	4	2	0	0	2	4	0
Empresas Extranjeras.	9	6	5	1	15	6	2	1	6	15	0
Empresas Mixtas.	1	3	1	2	4	2	0	0	2	4	0
Sociedades Mercantiles Cubanas	5	7	4	3	12	3	2	0	5	10	2
Cooperativas Agropecuarias.	1	2	1	1	3	0	0	3	0	3	0
Unidades Básicas de Producción Cooperativa	0	3	3	0	3	3	0	0	0	3	0
Agricultores Pequeños	4	5	2	3	9	5	3	1	0	9	0
Otras personas naturales	0	4	1	3	4	1	0	0	1	2	2
<b>Total de Diligencias Previas</b>	<b>39</b>	<b>38</b>	<b>20</b>	<b>18</b>	<b>77</b>	<b>32</b>	<b>8</b>	<b>7</b>	<b>22</b>	<b>69</b>	<b>8</b>

**Fig.19. Diseño del Reporte Diligencias Previas del Proceso Ejecutivo**

### 2.11.1 Código fuente

Siguiendo las normas del lenguaje estructurado en bloques PL/Pgsql vistas también en el Capítulo 1, se realizó la codificación de las consultas de la siguiente manera, vigente para todos los Casos de Uso implementados como se muestra en el ejemplo de la **Fig. 20**, la misma es representativa del Reporte Diligencias Previas del Proceso Ejecutivo.

```

Editor SQL Constructor Gráfico de Consultas
Borrar Eliminar Todos

SELECT ts.denominacion AS tipo_sujeto,
SUM(CASE WHEN (nt.idnestadoorigen = 1 OR nt.idnestadoorigen = 2
OR nt.idnestadoorigen = 3 OR nt.idnestadoorigen = 4) THEN 1 ELSE 0 END) AS pendientes_inicio,
COUNT(cau.idncausal) AS total_radicadas,
SUM(CASE WHEN (cau.idncausal = 3 OR cau.idncausal = 1) THEN 1 ELSE 0 END) AS confesion_deuda,
SUM(CASE WHEN (cau.idncausal = 3 OR cau.idncausal = 2) THEN 1 ELSE 0 END) AS reconoc_firma,
(SUM(CASE WHEN (nt.idnestadoorigen = 1 OR nt.idnestadoorigen = 2
OR nt.idnestadoorigen = 3 OR nt.idnestadoorigen = 4) THEN 1 ELSE 0 END) + COUNT(cau.idncausal)) AS total_resolver,

SUM(CASE WHEN trj.idntiporesolucionjudicial = 7 THEN 1 ELSE 0 END) AS inadmisibles,
SUM(CASE WHEN trj.idntiporesolucionjudicial = 9 THEN 1 ELSE 0 END) AS desistidas,
SUM(CASE WHEN trj.idntiporesolucionjudicial = 5 THEN 1 ELSE 0 END) AS declaradas_preparadas,
SUM(CASE WHEN trj.idntiporesolucionjudicial = 6 THEN 1 ELSE 0 END) AS denegadas,
COUNT(trj.idntiporesolucionjudicial) AS total_resuelto,

SUM(CASE WHEN (nt.idnestadoorigen = 1 OR nt.idnestadoorigen = 2
OR nt.idnestadoorigen = 3 OR nt.idnestadoorigen = 4) THEN 1 ELSE 0 END) AS pendientes_final

FROM admgov.ntiposujeto ts

LEFT JOIN economico.depersonajuridica pj
ON ts.idntiposujeto = pj.idntiposujeto
LEFT JOIN economico.dedemandante d
ON pj.idpersona = d.idpersona
LEFT JOIN economico.dedemandanteescrito de
ON d.iddemandante = de.iddemandante
LEFT JOIN economico.deescrito c
ON de.idescrito = c.idescrito
LEFT JOIN economico.deescritopromocional ep
ON c.idescrito = ep.idescrito
LEFT JOIN economico.netramiteescrito te
ON c.idncausal = te.idncausal
LEFT JOIN economico.necausal cau
ON ep.idncausal = cau.idncausal
LEFT JOIN economico.deexpediente e
ON d.idexpediente = e.idexpediente
LEFT JOIN economico.detrामite tr
ON te.idncausal = tr.idncausal
LEFT JOIN economico.deresolucionjudicial rj
ON tr.idncausal = rj.idncausal
LEFT JOIN economico.netiporesolucionjudicial trj
ON rj.idntiporesolucionjudicial = trj.idntiporesolucionjudicial
LEFT JOIN economico.netramite nt
ON tr.idncausal = nt.idncausal

GROUP BY ts.denominacion
    
```

Fig.20. Código fuente del reporte Diligencias Previas del Proceso Ejecutivo

## 2.12 Interfaces del sistema

Los trabajadores de la estadística judicial, después de estar autenticados en el sistema solo pueden acceder al subsistema Reportes (Recuperaciones) desde la pestaña Materias, como muestra la interfaz de la **Fig.21** y dentro del subsistema solo podrá acceder a los modelos correspondientes a la materia en la cual presta servicios.

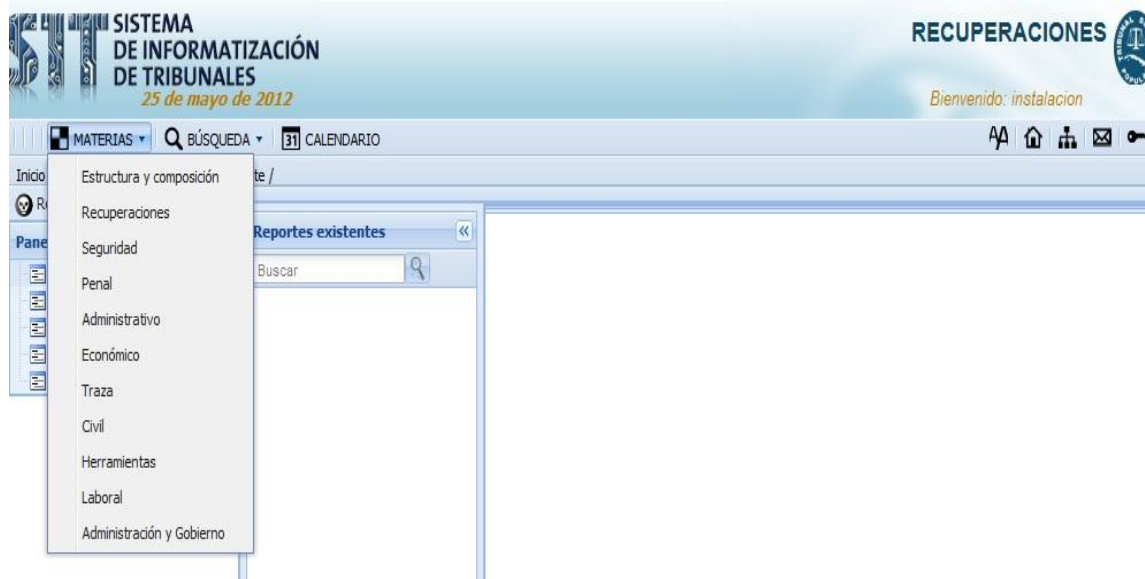


Fig.21. Interfaz principal del subsistema Reportes (Recuperaciones)

El acceso a los informes estadísticos y no estadísticos es simple y rápido, una vez dentro del módulo se le presenta al usuario solamente el área Visor de reportes donde se encuentran los mismos separados por materia.

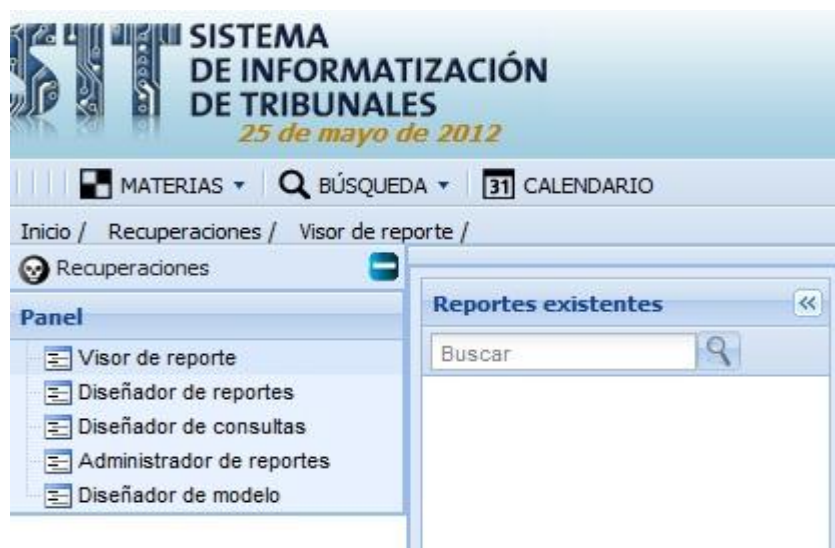


Fig.22. Interfaz de acceso al Visor de reporte

## **Conclusiones del capítulo**

En este capítulo se ha logrado llevar a la práctica lo estudiado en el anterior, se han conceptualizado los principales objetos del sistema para la realización del diagrama de clases del Modelo de Dominio y la comprensión del mismo. Se ha mostrado el diagrama de Casos de Uso del módulo Económico y posteriormente las especificaciones de los requisitos funcionales del Caso de Uso Diligencias Previas del Proceso Ejecutivo que pertenece a dicho módulo y del cual se mostró su descripción, diagrama de componentes, diseño y código fuente. Además se plantearon los requisitos no funcionales con los que debe cumplir el sistema, la descripción de los actores que interactúan con los reportes, el diagrama de despliegue del mismo; así como las principales interfaces con las que interactúa el usuario revisor.

## **CAPÍTULO 3: VALIDACIÓN Y PRUEBAS**

### **Introducción**

En el desarrollo del software, las posibilidades de error son innumerables, las mismas pueden manifestarse en la mala especificación de los requisitos funcionales, uso indebido de las estructuras de datos o simplemente errores al documentar los procesos. Para tratar de evitar estas dificultades se hace necesario ir acompañado de alguna actividad que garantice la calidad, por lo que la prueba es un elemento crítico para encontrar errores y lograr que se cumpla con las expectativas del cliente. En el presente capítulo se abordan varias actividades que son imprescindibles para que el software finalice con la calidad requerida, como las etapas de validación y gestión de los requisitos y las pruebas que se le realizan una vez que estos conforman los casos de uso, por lo que en el desarrollo de este capítulo se especifican las pruebas que serán hechas al software y se describen las irregularidades encontradas en las mismas.

### **3.1 Iteraciones**

Para comprobar la calidad del software se realizaron dos iteraciones de prueba donde se validó la funcionalidad del sistema de acuerdo a los requisitos definidos. El objetivo fundamental de estas iteraciones es encontrar no conformidades y registrarlas para posteriormente satisfacerlas.

### **3.2 Validación de requisitos**

La validación de los requisitos en el subsistema Reportes se realizó mediante la técnica de prototipos, revisión de requisitos y posteriormente la generación de casos de prueba, ya que se hace imprescindible encontrar los errores en los requerimientos antes de la implementación de los mismos.

#### **3.2.1 Prototipos funcionales**

Los prototipos son un método de validación ampliamente utilizado en muchas disciplinas, se utilizan fundamentalmente, para comprobar la corrección y completitud de la especificación de requisitos. Para validar los requisitos, mediante esta técnica, se realizaron diferentes tareas.

**Seleccionar quién evaluará el prototipo:** los usuarios que participaron en la validación de los prototipos funcionales del proyecto fueron un grupo de trabajadores estadísticos de los tribunales que manejan la información contenida en los reportes, además de los diferentes integrantes del equipo de trabajo del proyecto, como son programadores, arquitectos, analistas, etc.

**Desarrollar y ejecutar los escenarios de validación:** para evaluar correctamente el prototipo, se identificaron una serie de escenarios de forma tal que se pudiesen visualizar las diferentes funcionalidades de los prototipos de la forma más real posible, estos fueron ejecutados por los usuarios, estableciéndose siempre una adecuada orientación por parte del analista hacia el mismo, de manera tal que éste tenga conocimiento de diferencia entre el producto final y el prototipo para que no se muestre descontento.

**Documentar los problemas:** simplemente se confeccionó la lista de problemas encontrados, pidiéndole a los diferentes usuarios que documentaran los problemas o no conformidades que pudiesen tener los prototipos, de esta forma se obtuvo la lista de problemas. Después se procede a corregir y retroalimentar al prototipo para que refleje los cambios.

De modo general se presentaron los modelos elaborados durante la especificación de requisitos a grupos especializados en los procesos, a fin de validar si el análisis realizado responde a las necesidades y aspiraciones del cliente. Se presentaron además los documentos de Especificación de Requisitos de Software y Modelo del Sistema correspondientes al análisis, los que fueron entregados al cliente luego de ser sometidos a una revisión técnica formal por el equipo de calidad del proyecto (TPC, 2012).

En un primer encuentro con el cliente se detectaron seis no conformidades pertenecientes al análisis, una de ellas de recomendación, mientras que en las pruebas a las funcionalidades fueron encontradas cinco, para un total de doce no conformidades, algunas de ellas son por ejemplo: la necesidad de corregir errores ortográficos en los prototipos y documentos y la estandarización de los encabezados de los reportes que se muestran, las mismas se plasmaron en el documento de No Conformidades del proyecto y una vez corregidas todas y cada una de ellas, se estableció un segundo encuentro para la nueva presentación de los artefactos, finalmente cumplidas las necesidades del cliente se generó el Acta de Aceptación (Ver Anexo 2), documento en el cual queda constancia de la satisfacción del cliente con el trabajo realizado por el equipo de desarrollo.

### 3.2.2 Matriz de trazabilidad

La gestión de requisitos en Reportes consistió en llevar un control sobre los cambios que pudieron sufrir los mismos, esta tarea se ha llevado a cabo a lo largo de todo el desarrollo de la IR siguiendo las prácticas específicas de la gestión de requisitos definidas en el Plan de Desarrollo de Ingeniería de Requisitos del proyecto. Durante esta etapa se realizaron varias actividades que contribuyeron a lograr una mayor calidad de lo que se quiere construir, una de las actividades que permiten el cumplimiento de lo planteado anteriormente es mantener la trazabilidad bidireccional de los requisitos, o sea su rastreabilidad, clave para conseguir una exitosa gestión de los requisitos, por lo que se aplicó la técnica de la matriz de trazabilidad de requisitos, (Ver Anexo 3), a casos de uso del sistema, con el objetivo principal de asegurar que cada requisito especificado estuviera reflejado en al menos un caso de uso. Esta matriz, contenida en el documento Gestión de Requisitos del proyecto, abarca todos los requisitos pertenecientes al subsistema Reportes y a su vez los casos de uso que son definidos por los mismos, agrupados según la materia a la que corresponden (TPC, 2012).

### 3.3 Pruebas

Las pruebas son un instrumento adecuado para determinar el estado de la calidad de un producto en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos especificados. En cada una de las etapas de desarrollo de un software se hace necesaria la realización de las pruebas ya que a partir de ellas es posible controlar que los productos cumplan los requisitos mínimos de operabilidad.

El proceso de pruebas ayuda al cliente a refinar, concretar, identificar, corregir fallos u omisiones en las funcionalidades y favorece la comunicación entre el cliente y el equipo de desarrollo. Las pruebas no pueden asegurar la ausencia de defectos; sólo pueden demostrar que existen errores (Oré B, 2008).

#### 3.3.1 Diseño de casos de prueba

El objetivo del diseño de casos de prueba es crear un conjunto de casos de prueba que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos. Para diseñar un caso de prueba, se selecciona una característica del sistema o componente que se está probando, luego se selecciona un conjunto de entradas que ejecutan

dicha característica, documenta las salidas esperadas o rangos de salida, y donde sea posible se diseña una prueba automatizada que prueba que las salidas reales y esperadas son las mismas (Sommerville, 2005).

### 3.3.2 Pruebas de caja negra

Las pruebas de caja negra se centran principalmente en los requisitos funcionales del software permitiendo obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos (Oré B, 2008).

Se hace factible aplicar caja negra, pues permite la comprobación de los resultados que salen en el reporte, ya que el objetivo no consiste en probar funcionalmente todos los métodos procedimentales del producto, como requiere la prueba de caja blanca, sino analizar las principales funcionalidades en correspondencia con el cumplimiento de los requisitos; además de que la prueba de caja negra intenta encontrar errores en estructuras de datos o en accesos a las bases de datos externas que es una de los pasos que se hacen necesario validar.

Las pruebas de caja negra, intentan encontrar errores de las siguientes categorías:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de comportamiento o desempeño.
- Errores de inicialización y término.

#### 3.3.2.1 Técnica de partición de equivalencia

Dentro del método de caja negra la técnica de la **partición de equivalencia** es una de las más efectivas ya que permite examinar los valores válidos e inválidos de las entradas existentes en el software. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así el número de clases de prueba que hay que desarrollar. Para comprobar la calidad del producto desarrollado, o sea, si este presenta defectos graves o cumple con los requisitos necesarios, se realiza un caso de prueba por cada



caso de uso, de manera que se pueda abarcar todos los requisitos que el sistema debe cumplir (Pressman, 2005).

El diseño de casos de prueba para partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana. Las clases de equivalencia se definen de acuerdo con las siguientes directrices:

1. Si una condición de entrada especifica un rango, se definen una clase de equivalencia válida y dos no válidas.
2. Si una condición de entrada requiere un valor específico, se definen una clase equivalencia válida y dos no válidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se definen una clase de equivalencia válida y otra no válida.
4. Si una condición de entrada es booleana, se definen una clase de equivalencia válida y otra no válida.

Procedimiento para la técnica de partición equivalente:

1. Se identifican clases de equivalencia válida (CEV) y no válidas (CENV).
2. Asignar un número único a cada clase de equivalencia.
3. Escribir casos de prueba hasta que sean cubiertas todas las CEV, intentando cubrir en cada caso tantas CEV como sea posible.
4. Para cada CENV, escribir un caso de prueba, cubriendo en cada caso una CENV (McGraw-Hill, 2007).

### 3.3.3 Casos de pruebas diseñados

A continuación se muestran los casos de prueba que han sido diseñados con el fin de cubrir todas las posibilidades, estos corresponden a los casos de uso donde se describen los valores de entrada y los resultados observados aplicando el método de prueba de caja negra y utilizando la técnica de partición de equivalencia (TPC, 2012).

#### Descripción general

El caso de uso se inicia cuando el revisor selecciona de los reportes que se muestran en cada materia, el que desee visualizar según sus permisos. Para ello debe introducir los parámetros que desee, el sistema mostrará al revisor los datos del reporte seleccionado de acuerdo a los parámetros introducidos. El caso de uso termina cuando finalmente se muestra el reporte con las preferencias que el revisor seleccionó.

**Condiciones de ejecución:**

El sistema debe estar instalado y ejecutado correctamente.

El revisor debe estar autenticado con los permisos necesarios.

**Tabla 3: Secciones a probar en el caso de uso**

Nombre de la sección	Escenarios de prueba de la sección	Descripción de la funcionalidad
<b>SC1:</b> Visualizar Diligencias Previas del Proceso Ejecutivo	<b>EC1:</b> Visualizar reportes	El revisor selecciona el reporte que desea visualizar y el sistema muestra la interfaz del reporte seleccionado, permitiéndole al revisor seleccionar la pestaña "Filtro de reportes", elegir el rango de fechas (campo de selección) y el nombre del tribunal (lista desplegable). El revisor selecciona la pestaña con el nombre del reporte y realiza el <b>EC2 o EC3</b> .
	<b>EC2:</b> Modificar salida del reporte.	El revisor selecciona el formato y el límite de salida con que desea visualizar el reporte y realiza el <b>EC3</b> .
	<b>EC3:</b> Exportar reporte exitosamente.	El revisor selecciona el botón "Exportar" y el sistema muestra una nueva interfaz en la cual se visualiza el reporte filtrado según los parámetros señalados.
	<b>EC4:</b> Exportar reporte con fallo.	El sistema muestra un cuadro de

		diálogo mostrando el siguiente mensaje: <i>“Debe llenar los campos de filtrado”</i> .
	<b>EC5:</b> Imprimir reporte.	El revisor selecciona la opción Imprimir.

**Tabla 4: Descripción de variables**

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Fecha inicio	Campo de selección	No	Seleccionar una fecha.
2	Fecha fin	Campo de selección	No	Seleccionar una fecha.
3	Nombre del TP	Lista desplegable	No	Seleccionar un nombre.
4	Formato	Lista desplegable	Si	Seleccionar un formato.
5	Limitar salida a: (0 = no limitar)	Campo de texto	Si	Solo permite introducir valores numéricos.

**Tabla 5: SC1: Reporte diligencias previas del proceso ejecutivo**

ID del escenario	Escenario	Fecha inicio	Fecha fin	Nombre del TP	Formato	Limitar salida a: (0 = no limitar)	Respuesta del sistema	Resultado de la prueba
EC1	Visualizar reportes	N/A	N/A	N/A	N/A	V/"10"	El revisor selecciona el reporte que desea visualizar y el sistema muestra la interfaz del reporte seleccionado, permitiéndole al revisor seleccionar la pestaña "Filtro de reportes" y elegir el rango de fechas (campo de selección). El revisor selecciona la pestaña con el nombre del reporte y realiza el <b>EC2</b> o <b>EC3</b> .	Satisfactorio
EC2	Modificar salida del reporte	N/A	N/A	N/A	N/A	V/"10"	El revisor selecciona el formato y el límite de salida con que desea visualizar el reporte y realiza el <b>EC3</b> .	Satisfactorio
EC3	Exportar reporte exitosamente	N/A	N/A	N/A	N/A	V/"10"	El revisor selecciona el botón "Exportar" y el sistema muestra una nueva interfaz en la cual se visualiza el reporte filtrado según los parámetros señalados.	Satisfactorio

EC4	Exportar reporte con fallo	N/A	N/A	N/A	N/A	V/"10"	El sistema muestra un cuadro de diálogo mostrando el siguiente mensaje: <i>"Debe llenar los campos de filtrado"</i> .	Satisfactorio
EC5	Imprimir reporte	N/A	N/A	N/A	N/A	V/"10"	El revisor selecciona la opción Imprimir.	Satisfactorio

### 3.4 Registro de errores y dificultades detectadas

Durante la primera iteración se detectaron varias no conformidades tanto en la documentación como en la aplicación correspondientes a la implementación de las funcionalidades del sistema, al igual que en la fase de análisis las mismas fueron corregidas para luego ser presentadas en una segunda iteración, en la que se aprobaron por completo los documentos y funcionalidades, quedando satisfacción entre el cliente y el equipo de desarrollo. A continuación se muestra una tabla donde se recogen algunas de las dificultades y errores encontrados:

**Tabla 6: No conformidades de la aplicación**

Elemento	N o.	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Se recomienda	Estado NC	Respuesta equipo de desarrollo
EC1: Visualizar reporte	7	La aplicación muestra un aviso debido a que no se obtienen los	Los tipos de datos declarados inicialmente son incompatibles.	Prueba	Funcionalidad			Resuelta	Corregir los tipos de datos de modo que coincidan con los definidos

		datos de las tablas.							en el modelo de datos.
<b>EC3:</b>		El reporte mostrado en la nueva interfaz presenta irregularidades en los bordes de las tablas.	Etiquetas con medidas mal definidas.	Prueba	Funcionalidad			Resuelta	Corregir las medidas de las etiquetas con las que se conforman las tablas.

### 3.5 Análisis general de resultados

Las NC (no conformidades) detectadas durante la primera iteración fueron erradicadas en una segunda, lográndose la correcta y satisfactoria solución de los defectos e irregularidades que poseía el sistema. Las pruebas realizadas para ello contribuyeron a mejorar la calidad del mismo, estas no hacen al sistema libre de errores, pero garantizan su mejor funcionamiento, como queda demostrado a través de los resultados obtenidos. El gráfico de la **Fig.23** muestra la cantidad de no conformidades encontradas en ambas iteraciones y las clasificaciones de las mismas, su análisis evidencia que fueron solucionadas en una segunda iteración las fallas encontradas. Posterior al mismo en la **Fig.24** se profundiza en las no conformidades significativas de documentación y aplicación, mientras que el gráfico final, **Fig.25**, muestra la cantidad de errores detectados en la primera iteración en las pruebas de caja negra.

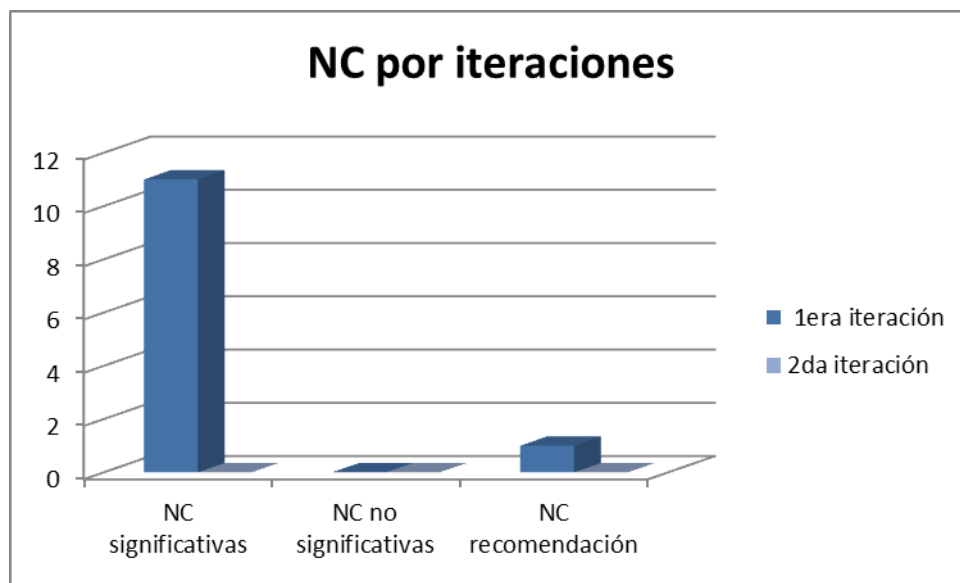


Fig.23. Gráfico de NC encontradas por iteraciones.

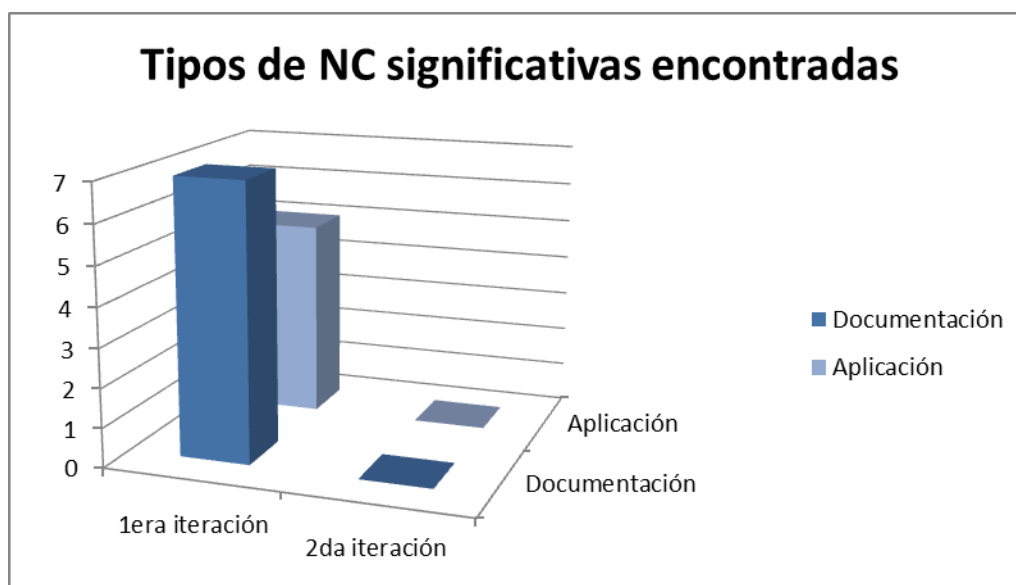


Fig.24. Gráfico tipo de NC significativas encontradas por iteraciones.



Fig.25. Gráfico de resultados de pruebas de caja negra por módulo.

### 3.6 Validación de las variables de la investigación

La investigación desarrollada plantea como idea a defender que: “Si se desarrollan los reportes de los procesos definidos en la fase I, se puede lograr una estandarización, integridad y control de la información de la estadística judicial”.

A continuación se evalúan las variables estandarización, integridad y control con el uso del subsistema Reportes del SIT en los Tribunales Populares Cubanos.

Tabla 7: Variables de la investigación

Variables	Generación de Reportes sin el uso del SIT	Generación de Reportes con el uso del SIT
<b>Estandarización</b>	Existen aplicaciones de manera aislada, que no abarcan todas las instancias, materias y procedimientos correspondientes al sistema de tribunales.	<ul style="list-style-type: none"> <li>• Un único sistema incluye todas las instancias, materias y procedimientos que utilizan los reportes.</li> <li>• Se sigue un mismo patrón,</li> </ul>



		formato y ruta de acceso para el manejo y exportación de los reportes.
<b>Integridad</b>	La información se registra en grupos por lo que en muchas ocasiones se repite y se crean dependencias innecesarias, el acceso a los datos cuando el cúmulo de información es mucho se hace complejo y por tanto comprobar la veracidad de los mismos es un proceso complicado.	<ul style="list-style-type: none"> <li>• Obtención, búsqueda y supervisión de la información estadística de forma automática.</li> </ul>
<b>Control</b>	La consulta de los datos contenidos en los reportes sólo se realiza en un determinado período.	<ul style="list-style-type: none"> <li>• Se eliminan los períodos fijos, permite acceder a los datos en cualquier intervalo de tiempo.</li> </ul>

La comparación mediante las variables establecidas, de la generación y manejo de los reportes con el uso del sistema o no, **Tabla 3**, es el resultado de una entrevista realizada con la especialista encargada, en la cual se mostró la aplicación en ejecución, así como la documentación necesaria. Producto de la satisfacción entre cliente y equipo de desarrollo se realizó dicho análisis, quedando plasmado que una vez el sistema esté completamente desplegado y ejecutándose cumple con las funcionalidades requeridas.

### Conclusiones del capítulo

En el presente capítulo se logró la validación de los requisitos mediante la erradicación de los errores y defectos encontrados permitiendo la aceptación de los artefactos por parte del cliente. Se evaluó el cumplimiento de las funcionalidades del sistema de acuerdo al análisis de las pruebas de caja negra, que fueron aplicadas mediante la técnica de partición de equivalencia,

lográndose con las mismas la calidad requerida. Resulta importante explicar que los casos de pruebas presentados no son los únicos diseñados para las aplicaciones ya que el objetivo ha sido plasmar la forma de realizar los mismos.

## **CONCLUSIONES GENERALES**

La manipulación de datos y su presentación de forma dinámica constituye un elemento fundamental para la gestión de la información correspondiente a la estadística judicial en los Tribunales Populares Cubanos y es por ello que se ha desarrollado una primera versión del Módulo de Reportes del Sistema de Informatización de Tribunales, el cual brinda soporte a los restantes módulos que integran el SIT y permite el manejo y visualización de los datos almacenados.

- Se realizó un estudio de los conceptos fundamentales que se manejan en torno a los reportes para lograr entender los procesos que se realizan.
- Se analizaron y fundamentaron las tecnologías definidas en la arquitectura del SIT basando su estudio en las ventajas que ofrecían a un desarrollo integrado.
- Se obtuvo el modelo de dominio correspondiente a los procesos que se realizan en el Sistema de Tribunales y las instancias definidas en la fase 1 referente a los reportes, lo que permitió una mejor comprensión de los conceptos con que trabajan los usuarios y con los que trabajará la aplicación.
- Se obtuvo la especificación de los requisitos funcionales y no funcionales a partir de los métodos empíricos utilizados, con el objetivo de lograr un mayor entendimiento entre clientes y desarrolladores como premisa para lograr un producto con la calidad requerida.
- Se realizaron los artefactos correspondientes definidos en el modelo de implementación, obteniéndose como resultado, los diagramas de casos de uso para cada materia, los diagramas de componentes y el diagrama de despliegue.
- Se probaron las funcionalidades del Módulo de Reportes a través de los diseños de casos de prueba y pruebas de caja negra, con el objetivo de verificar el grado de satisfacción de los requerimientos de software.
- Se obtuvo como resultado una primera versión del Módulo de Reportes para el SIT permitiendo con la misma la estandarización, integridad y control de la información de la estadística judicial cumpliéndose así con el objetivo general propuesto.

## **RECOMENDACIONES**

Como resultado del sistema implementado se hacen las siguientes recomendaciones:

- Continuar con el proceso de desarrollo del sistema a fin de concluir las funcionalidades identificadas que no se satisfacen con esta primera versión del módulo de Reportes para el SIT.

## **BIBLIOGRAFÍA**

1. **Alva, Eduardo Rivera. 2012.** Scribd. *Arquitectura de Software II. Diagrama de Componentes y Despliegue*. [En línea] 2012. [Citado el: 26 de Marzo de 2012.]  
<http://www.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes-y-Despliegue>.
2. **Axure. 2012.** Axure. *You'll love using Axure RP*. [En línea] [Citado el: 15 de febrero de 2012.]  
<http://www.axure.com/>.
3. **Christopher, Alexander. 1979.** *The Timeless Way of Building*. 1979.
4. **Colectivo de Profesores de DATEC. 2010.** *Manual de Usuario del Generador Dinámico de Reportes V.1.5.1*. UCI. : s.n., 2010.
5. **Departamento de Sistemas Informáticos y Computación, UPV. 2010.** Entorno Virtual de Aprendizaje. *Rational Unified Process (RUP)*. [En línea]  
<http://eva.uci.cu/mod/resource/view.php?id=9303&subdir=/Metodologias/RUP>.
6. **Durán Toro, Amador y Bernández Jiménez, Beatriz. octubre de 2000.** *Metodología para la Elicitación*. Sevilla : Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla., octubre de 2000. Informe Técnico LSI-2000-10.
7. **Hernández León, Rolando Alfredo y Coello González, Sayda. 2011 .** *El proceso de investigación científica*. La Habana : Editorial Universitaria, 2011 . ISBN 978-959-161307-3.
8. **IEEE. 1998.** *IEEE Std 1233:1998. Guide for Developing System Requirements Specification*. . s.l. : Institute of Electrical and Electronics Engineers, 1998. ISBN/ 0-7381-1723-4.
9. **Implementación y debugging, Capítulo 1. 2012.** Entorno Virtual de Aprendizaje. EVA. [En línea] [Citado el: 23 de 1 de 2012.]  
[http://eva.uci.cu/file.php/161/Documentos/Materiales\\_complementarios/UD\\_1\\_Procesos/Sobre\\_Modelos/Ciclo\\_de\\_vida\\_del\\_software.pdf](http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Sobre_Modelos/Ciclo_de_vida_del_software.pdf).
10. **Infante Frometa, Jenny y Hernández Hernández , Yasmany . 2009.** *Arquitectura de Software para el Sistema de Gestión de Reportes Dinámicos* . Habana : s.n., 2009.
11. **J. Solís, Camilo, A. Cabrera, Armando y G. Figueroa, Roberth.** Entorno Virtual de Aprendizaje. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*. [En línea]  
[http://eva.uci.cu/file.php/161/Documentos/Materiales\\_complementarios/UD\\_1\\_Procesos/Metodologias/METODOLOGIAS\\_TRADICIONALES\\_VS.\\_METODOLOGIAS\\_AGILES.pdf](http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/METODOLOGIAS_TRADICIONALES_VS._METODOLOGIAS_AGILES.pdf).
12. **Kendall, K. y Kendall, J. 1997.** *Análisis y Diseño de Sistemas. Tercera Edición*. México : s.n., 1997.

13. **Kruchten, P. 2004.** *The rational unified process: an introduction.* . s.l. : Addison-Wesley Professional, 2004.
14. **Larman, Craig. 1999.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Prentice Hall, Mexico : AREA: UNIVERSITARIOS, 1999. ISBN: 970-17-0261-1.
15. **McGraw-Hill. 2007.** *Ingeniería del Software, Un enfoque Práctico.* s.l. : Sexta Edición, pág 436., 2007.
16. **Mora, Sergio Luján. 2002.** *Programación de aplicaciones web: historia, principios básicos y clientes web.* Alicante : s.n., 2002.
17. **Oré B, Alexander. 2008.** CalidadySoftware.com. *CalidadySoftware.com.* [Online] 2008. [Cited: 4 16, 2012.] [http://www.calidadysoftware.com/testing/pruebas\\_funcionales.php](http://www.calidadysoftware.com/testing/pruebas_funcionales.php).
18. **Paradigm, Visual. 2012.** Visual Paradigm. [En línea] [Citado el: 29 de enero de 2012.] <http://www.visual-paradigm.com/>.
19. **Piloto, Niurisleidy Reyes y Martínez, Raúl Cambar. 2010.** *Indicadores que determinan la seguridad de las bases de datos realizadas en PostgreSQL en cuanto a la configuración, diseño, arquitectura y codificación.* La Habana : UCI, 2010.
20. **Pressman, Roger S. 2005.** *Ingeniería del Software. Un enfoque práctico.* La Habana : Félix Varela, 2005.
21. **Populares, Tribunales Supremos. 2010.** *Documentación del Sistema Estadístico de las materias.* La Habana : s.n., 2010.
22. **PostgreSQL-es. 2012.** PostgreSQL-es. *Sobre PostgreSQL.* [En línea] [http://www.postgresql.org/es/sobre\\_postgresql](http://www.postgresql.org/es/sobre_postgresql).
23. **Quintuña Churo, B.R. y V.C. García Alpala. 2010.** *Sistema generador de reportes dinámicos para web, configurable para las plataformas de bases de datos más conocidas.* 2010.
24. **Reyes Lomeli, Walter Ivan. 2007.** *Optimización de la Web del I.T.M, Aplicación Servidor Web.* México : s.n., 2007.
25. **Rodríguez Durán, A. 2012.** *IMPACTO DEL GENERADOR DINÁMICO DE REPORTES EN LOS SISTEMAS DE GESTIÓN DE INFORMACIÓN.* 2012.
26. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2000.** *EL Proceso Unificado de Desarrollo de Software.* España : Addison Wesley, 2000. ISBN: 8478290362.
27. **Silva, Andrés.** Ingeniería de requisitos. [En línea] <http://is.ls.fi.upm.es/docencia/swcritico/Requisitos.pdf>.

28. **Sommerville, Ian. 2005.** *Ingeniería del Software. Séptima Edición.* Madrid : Pearson Educación, S.A., 2005. ISBN: 84-7829-074-5.
29. **SQL Server 2008. 2011.** SQL Server 2008. *Reporting.* [En línea] 2011. [Citado el: 16 de diciembre de 2011.] <http://www.microsoft.com/sqlserver/2008/en/us/reporting.aspx>.
30. **TPC, Equipo de Desarrollo del Proyecto. 2011.** *Expediente de Proyecto.* 2011.
31. **UML. 2012.** Unified Modeling Lenguaje. *UML Resource Page.* [En línea] [Citado el: 20 de febrero de 2012.] [http://www.omg.org/gettingstarted/what\\_is\\_uml.htm](http://www.omg.org/gettingstarted/what_is_uml.htm).
32. **Villanueva, Isabel. 2000.** *Elicitación de requisitos en sistemas de gestión orientados a procesos1.* España: s.n., 2000.
33. **Visual Paradigm for UML 8.0 Enterprise Edition. 2011.** Visual Paradigm for UML - UML tool for software application development. [En línea] 2011. <http://www.visual-paradigm.com/product/vpuml/>.
34. **Young, Ralph R. 2004 .** *The Requirements Engineering.* Boston, London. : ARTECH HOUSE, INC., 2004 . MA 02062

## **GLOSARIO DE TÉRMINOS**

### **B**

**BPMN:** Business Process Modeling Notation (Notación de Modelado de Procesos del Negocio).

### **C**

**CASE:** Computer Aided Software Engineering (Ingeniería del Software Asistida por Computadoras).

**Casos de Uso:** Técnica de captura, funcionalidad que debe cumplir el sistema.

### **H**

**Hito:** Es un punto de referencia que marca un evento importante de un proyecto y se usa para supervisar el progreso del mismo.

### **I**

**IR:** Ingeniería de Requisitos

### **M**

**MSF:** Microsoft Solution Framework (Marco de Solución de Microsoft).

**Multiplataforma:** Capacidad para funcionar en más de un sistema operativo con similares características y sin que su funcionalidad varíe en exceso.

### **N**

**NC:** No Conformidades

### **P**

**Paquetes:** El término se refiere a cierto software de aplicación diseñado para atender necesidades sectoriales. Un paquete integrado contiene un conjunto de programas para atender diversas necesidades.

**PGAdmin III:** Aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL.

**Prototipos no funcionales:** Se utilizan principalmente para probar ciertos aspectos del diseño dentro de los objetivos del proyecto.

### **R**

**RUP:** Rational Unified Process (Proceso Unificado de Desarrollo).

### **S**

**SIT:** Sistema de Informatización de Tribunales, nombre correspondiente a la solución informática que se desarrolla.

**SOO:** Sistema Orientado a Objetos, es un conjunto de objetos colaborativos que interactúan entre sí y que son persistentes por lo que deben ser guardados en una base de datos.



**Stakeholders:** Trabajadores del negocio u otros interesados que interactúan de una forma u otra con el sistema.

**U**

**UML:** Unified Modeling Language (Lenguaje de Modelado Unificado).

## ANEXOS

**Anexo 1:** “Primera entrevista para elicitación de requisitos correspondientes al módulo Reportes realizada en el Tribunal Supremo Popular.”

Los resultados de esta entrevista contribuirán a desarrollar la elicitación, análisis y especificación, con el objetivo de transformar las necesidades del cliente a un lenguaje comprensible por los desarrolladores contribuyendo así a facilitar el desarrollo de cada módulo que utiliza los resultados.

1. ¿Cuáles son los datos por los que se requiere realizar la búsqueda de manera general y en específico de cada reporte?
2. ¿Se puede definir un encabezado estándar para todos los reportes sin que se afecte la estructura del mismo?
3. ¿Qué quieren decir cada uno de los campos que se encuentran en las tablas de los reportes?
4. ¿A qué materia pertenecen los reportes que no tienen el título bien definido?
5. ¿Cuántos procedimientos posee cada materia de la cual se va a exportar el reporte?
6. ¿El nombre del tribunal que muestra el reporte en el encabezado corresponde al que pertenece el revisor estadístico?
7. ¿Cuáles son los reportes que se consultan con más frecuencia?
8. ¿Qué acciones realiza cada trabajador con los reportes?

**Anexo 2:** Acta de aceptación generada durante la validación de requisitos.



## Acta de aceptación

**ACTA DE ACEPTACIÓN**

En cumplimiento del **Convenio de colaboración con el Tribunal Supremo Popular** y en función de la ejecución del Proyecto de Informatización para la Gestión de los Tribunales Populares Cubanos Fase I, se hace entrega del producto que se relaciona a continuación.

Lista de productos que serán aceptados:

- Subsistema Reportes Fase I
  - Especificación de Requisitos de Software
  - Modelo de Casos de Uso del Sistema

<b>Entrega</b> Proyecto de Informatización de los TPC	<b>Recibe</b> Tribunal Supremo Popular
<b>Nombre y Apellidos:</b> Alayna Pantoja Trincado	<b>Nombre y Apellidos:</b> Araceli Alonso Lorenzo
<b>Cargo:</b> Analista del Módulo	<b>Cargo:</b> Analista funcional (Jefa Departamento Estadística Judicial. Tribunal Supremo Popular)
<b>Firma:</b> 	<b>Firma:</b> 
<b>Comentarios:</b>	

**Representante Parte Suministradora Proyecto de Informatización de los TPC**

**Nombre y Apellidos:** Chavelys Tellez Larramendi

**Cargo:** Analista Principal

**Firma:** 

**Fecha:** 29/mayo/2012

### Anexo 3: Matriz de trazabilidad del módulo económico

Materia Económica	RF38	RF39	RF40	RF41	RF42	RF43	RF44	RF45	RF46	RF47	RF48	RF78	RF79	RF80	RF81
CU 10. Visualizar Reporte Expedientes por Criterio								X	X			X			X
CU 11. Visualizar Reporte Radicación de Procesos	X											X			X
CU 12. Visualizar Reporte Causales de Radicación de Procesos Ordinarios Contractuales		X										X			X
CU 13. Visualizar Reporte Asistencia al Arbitraje Internacional			X									X			X
CU 14. Visualizar Reporte Tramitación y Resolución.				X	X							X			X
CU 15. Visualizar Reporte Informe sobre Ejecución de Sentencias						X						X			X
CU 16. Visualizar Reporte Diligencias Previas del Proceso Ejecutivo							X					X			X
CU 17. Visualizar Libro en Materia Económica										X	X	X			X