

*Universidad de las Ciencias
Informáticas
Facultad 3*



*Título: Módulo Diccionario Jurídico para el
Sistema de Informatización de la Gestión de la
Fiscalía.*

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

Autor: Dannier Rabelo Martín

Tutor: Ing. Vlamir Rodríguez Fernández

Co-tutora: Ing. Yanet Rodríguez López

La Habana, Cuba

Junio, 2012

*Módulo Diccionario Jurídico para el Sistema de Informatización
de la Gestión de la Fiscalía.*

Declaración de Autoría:

Declaro ser el único autor del presente trabajo de diploma y autorizo al Centro de Gobierno Electrónico de la Universidad de las Ciencias Informáticas los derechos de hacer con el presente trabajo el uso pertinente del mismo.

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

Dannier Rabelo Martín

Autor

Ing. Vlamir Rodríguez Fernández

Ing. Yanet Rodríguez López

Tutor

Tutora

*Módulo Diccionario Jurídico para el Sistema de Informatización
de la Gestión de la Fiscalía.*

Agradecimientos:

Agradezco mis padres y a mis abuelos, porque ellos son un ejemplo digno para mí en mi formación y ellos me han enseñado cual es la manera de ser una persona de bien.

A mis tutores Vladimir, Yanet y a Leanet por la paciencia que tuvieron en el momento de ayudarme a realizar este trabajo que es de tanta importancia para mí.

A la Universidad de las Ciencias Informáticas por brindarme los medios para mi formación como profesional y a todos los profesores que me enseñaron a lo largo de la carrera y en especial a la profesora Dariela que ella fue como una madre para mí en el primer año en la escuela.

A mis hermanos Dayron, Gustavo, Rafa, José y Felipe por estar compartiendo conmigo todo lo que viví en la universidad y a todos mis compañeros que en algún momento de la carrera estuvieron ahí compartiendo.

A todas aquellas personas que de algún modo me ayudaron a la realización de este trabajo.

A todos, muchas gracias.

*Módulo Diccionario Jurídico para el Sistema de Informatización
de la Gestión de la Fiscalía.*

Dedicatoria:

Dedico este trabajo a mis padres, mis abuelos, mis tíos que sin ellos y sin su enseñanza yo no hubiera podido regalarles este momento tan feliz en mi vida.

En especial a mi hermanita de mi alma que ella es todo para mí y ella ha sido mi fuerza y mi espíritu para salir adelante en los momentos difíciles que se me han presentado a lo largo de la carrera.

Dannier Rabelo martin

Resumen:

La Fiscalía General de la República de Cuba (FGR), es la institución rectora del sistema judicial cubano y el órgano del Estado responsable del control y conservación de la legalidad dentro del sistema social, cuyo funcionamiento está basado en el cumplimiento estricto de la Constitución y las leyes, creadas y aprobadas por los diferentes órganos del Estado, las distintas entidades y el pueblo en general en pleno ejercicio de la democracia.

El presente trabajo de diploma tiene como su principal objetivo desarrollar un diccionario electrónico que facilite una rápida consulta de los términos que son utilizados por el personal de la fiscalía para elaborar cualquier documento que necesite. Para el desarrollo del mismo se hizo un estudio sobre las herramientas y tecnologías utilizada en la elaboración de la solución. Se presentan los artefactos generados en los flujos que brinda la metodología RUP, obteniendo como resultado una herramienta que satisface las necesidades del cliente.

*Módulo Diccionario Jurídico para el Sistema de Informatización
de la Gestión de la Fiscalía.*

Índice

Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	4
1.1. Diccionarios electrónicos.....	4
1.2. Metodología de desarrollo de software.....	4
1.3. Lenguaje de Programación.....	6
1.3.1. Lenguaje de programación PHP.....	6
1.3.2. Lenguaje HTML.....	7
1.3.3. Lenguaje CSS.....	8
1.3.4. Lenguaje JavaScript.....	8
1.4. Servidores de Base de Datos.....	8
1.5. Servidor Web Apache 2.....	10
1.6. IDE de Desarrollo.....	11
1.7. Marco de trabajo de desarrollo en PHP.....	13
1.8. Herramienta Case.....	15
1.9. Pruebas de Calidad.....	15
1.9.1. Prueba de Caja Negra.....	15
1.9.2. Prueba de Caja Blanca.....	16
Capítulo 2. Propuesta de la Solución.....	18
2.1. Características de la herramienta.....	18
2.2. Definición de Requisitos.....	18
2.2.1. Requisitos Funcionales.....	18
2.2.2. Requisitos no Funcionales.....	19
2.3. Modelado del Sistema.....	20
2.3.1. Definición de los actores del sistema.....	20
2.3.2. Diagrama de Casos de Uso del Sistema.....	20
2.3.3. Especificación de los Casos de Uso del Sistema.....	21

*Módulo Diccionario Jurídico para el Sistema de Informatización
de la Gestión de la Fiscalía.*

2.3.4.	Especificación de los Casos de Uso.....	21
2.4.	Análisis.....	27
2.4.1.	Modelo de Clases del Análisis.	28
2.4.2.	Diagrama de interacción.....	28
Capítulo 3. Diseño e Implementación.		31
3.1.	Diagramas de clases del Diseño.....	31
3.2.	Patrones Utilizados.	35
3.2.1.	Patrones GRASP.....	36
3.2.2.	Patrones GoF.....	37
3.3.	Modelo físico de la base de datos.	38
3.4.	Implementación.....	41
3.4.1.	Estándares de Codificación.....	41
3.4.2.	Diagrama de Componentes.	43
3.4.3.	Diagrama de Despliegue.....	45
3.4.4.	Seguridad.....	47
Capítulo 4. Validación de la propuesta de solución.		48
4.1.	Pruebas Caja Negra.	48
4.1.1.	Buscar Término.	48
4.1.2.	Descripciones de las variables.....	49
4.1.3.	Registrar Solicitud.....	50
4.1.4.	Modificar Solicitud.....	51
4.1.5.	Eliminar Solicitud.	52
4.1.6.	Descripciones de las variables.....	52
4.2.	Pruebas Unitarias.	53
Conclusiones Generales.		57
Bibliografía.		58

*Módulo Diccionario Jurídico para el Sistema de Informatización
de la Gestión de la Fiscalía.*

Anexos.....60

Introducción.

Desde la antigüedad, el hombre sintió la necesidad de definir el significado de las palabras para su comunicación. Las primeras compilaciones de palabras que se hicieron a modo de diccionario datan del siglo VII a.n.e, pero desde aquellos tiempos hasta nuestros días, los diccionarios han evolucionado, convirtiéndose en una de las principales fuentes de conocimiento. Existen diferentes clasificaciones de acuerdo al idioma, temáticas a tratar y geografía.

Un diccionario, según la Real Academia de la Lengua Española, se define como un “libro en el que se recogen y explican de forma ordenada voces de una o más lenguas, de una ciencia o de una materia determinada.” (ESPAÑOLA). Dicha compilación suele incluir el significado, la etimología, la ortografía, la pronunciación y la separación silábica de los conceptos, aunque no siempre presenta toda esta información. La lexicografía es la ciencia encargada de elaborar los diccionarios. De todas formas, hay que destacar que los datos presentados suelen variar según el tipo de diccionario. En general, la obra se publica en forma de libro impreso, aunque también existen versiones electrónicas.

Un diccionario electrónico consiste en una lista de palabras con información gramatical codificada, la cual puede ser utilizada por programas.

La Universidad de las Ciencias Informáticas (UCI) tiene como uno de sus objetivos la informatización de la sociedad cubana; una de las acciones que se llevan a cabo, como acuerdo entre la Fiscalía General de la República (FGR) y la UCI, es la informatización de los procesos fiscales con el objetivo de crear una solución que condicione un incremento en la calidad de la tramitación, una supervisión y control en tiempo real de los procesos judiciales, una reducción de los términos de las actividades y diligencias practicadas, una utilización más óptima de la fuerza fiscal y una disminución de la utilización del papel.

La FGR es el órgano del estado al que corresponde como objetivos fundamentales, el control y la preservación de la legalidad sobre la base de la vigilancia del estricto cumplimiento de la constitución, las leyes y demás disposiciones legales, por los organismos del estado, entidades económicas, sociales y por los ciudadanos.

Actualmente no existe una vía de que dichos términos, necesarios en la elaboración de los documentos, sean consultados de manera rápida y precisa, facilitando la correcta utilización de un lenguaje jurídico, por tal motivo existe la necesidad de gestionar los términos judiciales que son utilizados en el trabajo diario del personal de

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

la fiscalía, para que estos puedan estar disponibles para su consulta en el momento de elaborar cualquier documento que se necesite en la fiscalía.

Teniendo en cuenta lo explicado anteriormente se plantea resolver el siguiente problema: el uso correcto del lenguaje jurídico a la hora de elaborar los documentos se está viendo afectado por no disponerse de una vía para el acceso rápido y preciso a los términos.

Se encuentra enmarcado el proceso de desarrollo de software como objeto de estudio.

Para resolver el problema de la investigación, el presente trabajo de diploma tiene como objetivo general: desarrollar el módulo Diccionario Jurídico para el Sistema de informatización de la gestión de la Fiscalía, que realice la gestión de los términos jurídicos permitiendo un acceso rápido y preciso en su posterior consulta.

Encontrándose dentro del campo de acción el proceso de desarrollo de software en herramientas de diccionarios en línea.

Se plantea como idea a defender que con la informatización de la gestión de los términos judiciales utilizados en el trabajo diario en la fiscalía, se posibilitará la consulta de dichos términos posteriormente.

Para cumplir con el objetivo general se elaboraron los siguientes objetivos específicos:

- Elaborar el marco teórico de la investigación.
- Elaborar la propuesta de la solución.
- Validar los artefactos generados en la propuesta.

Para darle cumplimiento a dichos objetivos específicos, se elaboraron las siguientes tareas de la investigación:

- Recopilación de la información necesaria para la investigación.
- Selección de la información relevante para la investigación.
- Confección del marco teórico.
- Obtención de los artefactos necesarios en la investigación.
- Validación de la propuesta generada.

Métodos Científicos

Teóricos:

Analítico – Sintético: este método se utilizó como punto de partida en la investigación, debido a que ayudó a identificar un grupo de necesidades que habían dentro de la

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

fiscalía, con el desarrollo de una herramienta capaz de facilitar el trabajo de la gestión de una serie de procesos que se realizan en la misma.

Inductivo – Deductivo: permitió identificar la necesidad de una herramienta para la automatización del trabajo, ya que cuando no existía, el trabajo se dificultaba demasiado.

Hipotético – Deductivo: dio la posibilidad de identificar los factores que ayudarían a mejorar todo el trabajo que se realiza en las fiscalías con los términos jurídicos.

Modelación: se trabajó con dicho método para la obtención de los artefactos del sistema.

Empírico:

Entrevista: este método permitió obtener la información acerca de las necesidades del cliente, requisitos y funcionalidades que la aplicación debería tener.

Medición: este método se utilizó con el objetivo de realizar las pruebas necesarias para identificar si la herramienta cumple con los requisitos.

La investigación está estructurada de la siguiente forma:

Capítulo 1. Fundamentación Teórica: Se realiza un estudio con los diccionarios electrónicos, las herramientas y tecnologías para desarrollar la aplicación.

Capítulo 2. Propuesta de la Solución: Se presenta la propuesta que se quiere desarrollar. Se muestran la especificación de los requisitos funcionales y no funcionales con los que debe cumplir el sistema.

Capítulo 3. Diseño e Implementación: Se elabora el modelo de diseño, el cual da paso al modelo de implementación para dar solución al problema.

Capítulo 4. Validación de la propuesta de solución: Se muestran los resultados obtenidos con las pruebas realizadas a la solución.

Capítulo 1. Fundamentación Teórica.

En el siguiente capítulo se abordarán los aspectos teóricos que fundamentan el objetivo de la investigación. Entre estos aspectos se encuentran la metodología de desarrollo, el lenguaje de programación, el servidor de aplicaciones web, el marco de trabajo en que se desarrollará, así como los tipos de prueba que serán aplicadas posteriormente a la implementación. La aplicación a desarrollar forma parte de la solución Sistema de Informatización de la Gestión de las Fiscalías, todas las herramientas y tecnologías utilizadas en la construcción del módulo fueron definidas por el equipo de arquitectura del proyecto.

1.1. Diccionarios electrónicos.

“Las lenguas cambian de continuo, y lo hacen de modo especial en su componente léxico. Por ello los diccionarios nunca están terminados: son una obra viva que se esfuerza en reflejar la evolución registrando nuevas formas y atendiendo a las mutaciones de significado” (Real Academia, 2012). En el mundo existen diferentes diccionarios electrónicos y se utilizan de acuerdo las necesidades del cliente. Los diccionarios electrónicos son dispositivos modernos y ligeros con una gran variedad de funciones, referencias gramaticales y posee base de datos que contienen gran cantidad de información.

WordReference. Es un sitio web que tiene variedad de diccionarios, está el bilingüe que brinda sus servicios en varios idiomas gratuitamente a los clientes y también proporciona el diccionario de sinónimos y antónimos para todas las personas que necesitan consultar dichos términos. Este sitio está publicado en la dirección de <http://www.wordreference.com>.

RAE. Es un sitio web que tiene el propósito de mantener al alcance de las personas el significado de las palabras para su uso en el momento que se requiera, también el de fijar las voces y vocablos de la lengua castellana en su mayor propiedad, elegancia y pureza. Dicho sitio está publicado en la dirección <http://www.rae.com>.

Diccionariojuridico.mx. Este sitio trata de presentar una visión general y fragmentaria de la evolución del Derecho. El Diccionario Jurídico se integra por términos de la doctrina, legislación y jurisprudencia. Este sitio está publicado en la dirección <http://www.diccionariojuridico.mx/index.php>.

1.2. Metodología de desarrollo de software.

En el mundo de la informática existen dos tipos de metodologías, las ágiles y las robustas, cada una de ellas tiene características distintas, y de acuerdo a estas

características son aplicadas en los diferentes proyectos, según las condiciones que se requieran.

El término ágil nace en febrero del 2001 en una reunión celebrada en Utah, EEUU, aplicado al desarrollo de software. Los que estaban presentes en esa reunión fueron los creadores de una organización dedicada a promover los conceptos relacionados con el desarrollo ágil del software, donde el punto de partida fue el Manifiesto Ágil y se valoran los siguientes aspectos: (Abrahamsson, et al., 2002)

- El individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- El desarrollo de software más que conseguir una buena documentación.
- La colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan.

Estos valores anteriores inspiran a que de ahí se deriven las características fundamentales de las metodologías ágiles que son las que las diferencian de las metodologías robustas.

Como metodología robusta o tradicional, se encuentra el Proceso Unificado Racional (RUP, por sus siglas en inglés) que junto al lenguaje de modelado UML, llevan a cabo la representación de la documentación del software y recopilación de la información necesaria para la confección del mismo.

Consiste en un conjunto de actividades necesarias para transformar los requerimientos del usuario en el sistema de software. Está especializado para diversos tipos de software de sistemas, diversas áreas de aplicación, diferentes tipos de organizaciones y diferentes tamaños de proyectos.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. (Jacobson, et al., 2000)

El ciclo de vida de RUP es el siguiente:

- Inicio: define el alcance y objetivos del proyecto.
- Elaboración: plan de proyecto, especificación de características y arquitectura base.
- Construcción: construir y operar el producto.
- Transición: garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Las características principales de RUP son:

- Dirigido por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.
- Tiene una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Incluye artefactos (que son los productos tangibles del proceso).
- Incluye roles (es el papel que desempeña una persona en un determinado momento).

Unas de las principales ventajas de esta metodología son:

- Mitigación temprana de posibles altos riesgos.
- Progreso visible en las etapas tempranas.
- El conocimiento adquirido en una iteración puede aplicarse de iteración a iteración.
- Los usuarios están involucrados continuamente.

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Es por ello que se escogió como la metodología idónea para el desarrollo de la herramienta en cuestión, pues permite detallar la documentación del proceso de desarrollo, tema de vital importancia, pues la información que se maneja es de carácter sensible. Aunque en la utilización de dicha metodología no se utilicen todos artefactos definidos en cada una de sus fases, se realizará un flujo básico donde se expondrán los artefactos esenciales de cada uno de los flujos. Los artefactos que se pueden mencionar son la especificación de requisito del software, modelo de sistema, modelo de diseño y el modelo de diseño de la base de datos.

1.3. Lenguaje de Programación.

1.3.1. Lenguaje de programación PHP.

A diferencia de otros lenguajes de programación, PHP (Hipertexto Pre-Processor) se creó específicamente para la generación de páginas web, muy utilizadas en la actualidad. Desde la aparición de PHP5 se ha incrementado el número de usuario de este lenguaje de programación. Con el mismo se ofrece la posibilidad de hacer programas orientados a objetos, la lectura de archivos XML de forma sencilla, la utilización de base de datos y la implementación de servicios, el envío de correos electrónicos o la utilización del protocolo FTP. También ofrece librerías que ayudan a la implementación de archivos RSS (Rich Site Summary), servicios Web XML-RPC

que es un protocolo de llamada a procedimientos remoto que funciona a través del protocolo HTTP (Granado, 2004). Con la integración de varias librerías externas permite la generación de documento en formato pdf.

Ventajas de la utilización de PHP: (The PHP Group, 2012)

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados extensiones). Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución. Tiene manejo de excepciones (desde PHP5).

Por la gran cantidad de características que posee dicho lenguaje se convierte en la elección más ideal para la creación de la herramienta.

1.3.2. Lenguaje HTML.

HTML (HyperText Markup Language) es lenguaje de marcado de hipertexto, con él se escriben las páginas web. Se utiliza para representar la estructura y el contenido en forma de texto. La forma que se realiza es a través de “etiquetas”, y cada etiqueta tiene una forma de representación, que comienza y termina con angulares ('<', '>'). A través de este lenguaje los navegadores web pueden mostrar toda la información que se encuentra en dicho documento de formato HTML y en estos documentos también puede incluir otros lenguajes (JavaScript, CSS) permitiendo agregar más funcionalidades para mejores prestaciones. (W3C, 2012)

1.3.3. Lenguaje CSS.

CSS (Cascading Style Sheets) es hoja de estilo en cascada, este lenguaje se utiliza para definir la presentación de un documento estructurado por HTML o XML. El contenido de la hoja de estilo facilita al creador la modificación de su estructura cuando sea necesario, sin tener la necesidad de alterar el contenido de la página web. A través de la estructura que utiliza, se puede realizar todo lo relacionado con la presentación que se encuentra dentro de una página web. (w3c, 2012)

1.3.4. Lenguaje JavaScript.

Es un lenguaje de programación utilizado para crear pequeñas acciones dentro de una página web. Con la utilización del mismo se puede crear efectos en las páginas y definir interactividad con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutar todo el código que se encuentra embebido en los documentos HTML o en los documentos que se encuentran en otra ubicación que tienen dicho código de Javascript, permite la validación de la información que se requiera. Permite la programación tanto de pequeños scripts, como programas más grandes. Es un lenguaje dinámico que responde a eventos. Se encarga de detectar eventos dentro de una página web y dar respuesta a dichos eventos. Según lo planteado por el autor David Flanagan en su libro "JavaScript: The Definitive Guide" JavaScript posee un tipo de dato de punto flotante, operadores aritméticos que trabajan con él y un complemento de las normas de punto flotante de las funciones matemáticas. (Flanagan, 1997). Además en el libro "Guía de aprendizaje Javascript" se expone que puede crear páginas HTML personalizadas sobre la marcha, dependiendo de las acciones ejecutadas por el usuario. Javascript puede asegurar que la información introducida en los formularios por los usuarios sea correcta y si requieren hacer algún tipo de cálculos con los datos introducidos lo podemos realizar a través de las funcionalidades que brinda dicho lenguaje. Permite crear una interfaz de usuario activa, lo que ofrece retroalimentación a los visitantes según navegan por sus páginas. (NEGRINO, et al., 2000)

1.4. Servidores de Base de Datos.

PostgreSQL es un sistema gestor de bases de datos objeto-relacional, distribuidos bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado. Dicho gestor de base de datos tiene una característica muy esencial como es la de multiplataforma, que puede ser utilizado en varios sistemas operativos (Linux, Windows). El mismo utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para mejorar

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

A continuación se muestra un gráfico que ilustra de manera general los componentes más importantes de PostgreSQL. (The PostgreSQL Global Development Group , 2012)

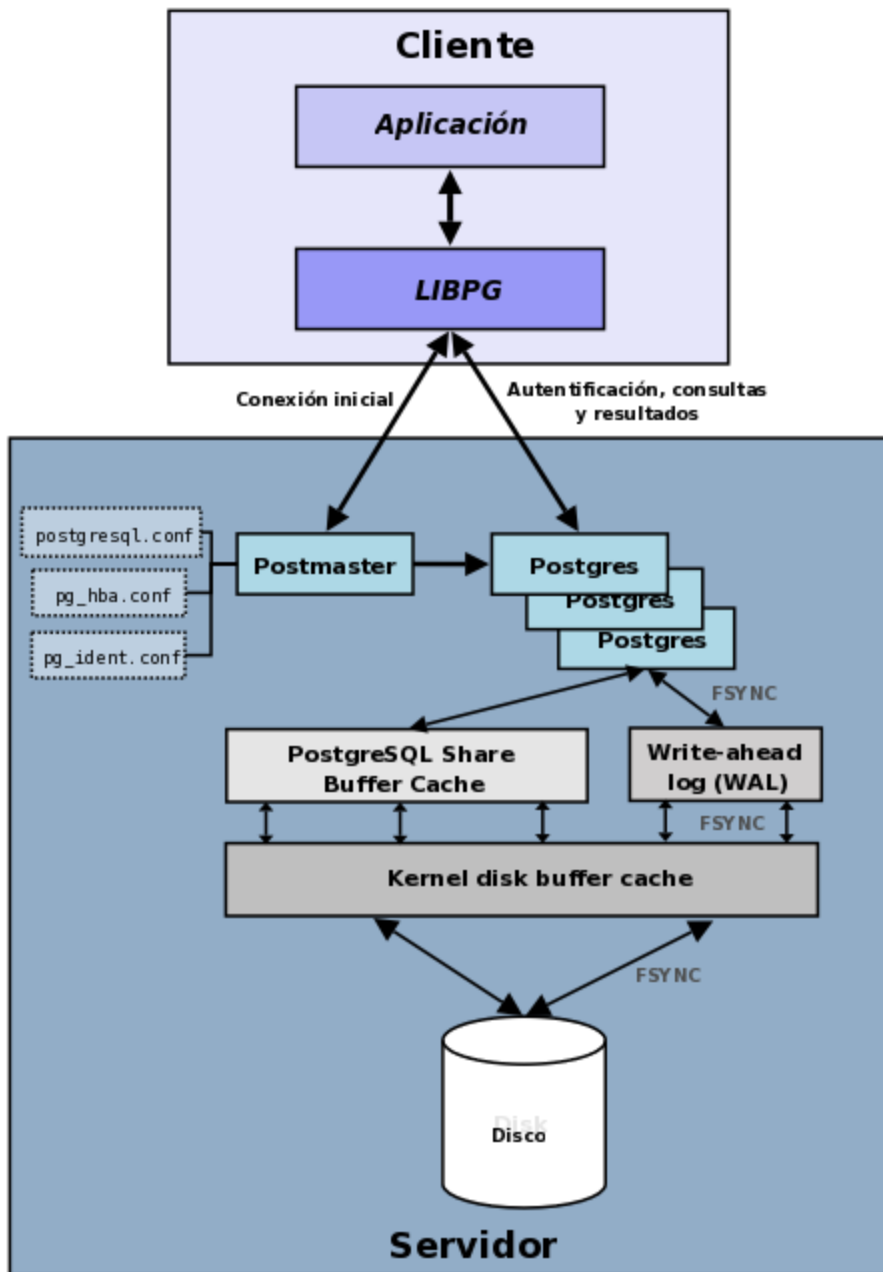


Figura1. Arquitectura de PostgreSQL. (The PostgreSQL Global Development Group , 2012)

- **Aplicación cliente:** esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.
- **Demonio postmaster:** este es el proceso principal de PostgreSQL. Encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargarán de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- **Ficheros de configuración:** los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf.
- **Procesos hijos postgres:** se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- **PostgreSQL share buffer cache:** memoria compartida usada por PostgreSQL para almacenar datos en caché.
- **Write-Ahead Log (WAL):** componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).
- **Kernel disk buffer cache:** caché del disco del sistema operativo.
- **Disco:** disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

1.5. Servidor Web Apache 2.

El servidor apache es un servidor web HTTP de código abierto, potente y flexible para plataforma Unix (BSD, GNU/Linux), Microsoft Windows y Macintosh que implementa el protocolo HTTP. “Según los estudios realizados, es el principal servidor de la Web desde hace diez años. La transparencia y diversidad son las principales características de Apache. El código fuente es totalmente abierto. Su arquitectura modular, construida sobre pequeños núcleos, se adapta a las necesidades específicas de cada usuario”. (Kew, 2008). El diseño modular de apache permite a los administradores de sitios web seleccionar que características se van a incluir en el servidor, así como los módulos que se van a cargar, ya sea al compilar o al ejecutar el servidor. Es un servidor que trabaja con gran cantidad de Perl, PHP y otros lenguajes script. Permite personalizar la respuesta ante un posible error que se pueda dar en el servidor, es posible configurarlo para que se ejecute un script cuando ocurra un error en concreto. Presenta una alta configurabilidad en la creación y gestión de los logs, a través de los

cuales se puede tener un mejor control con lo que sucede en el servidor. (The Apache Software Foundation, 2012)

1.6. IDE de Desarrollo.

El IDE NetBeans es un reconocido entorno de desarrollo integrado, disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles, utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy and Grails y C/C++. (Oracle, 2012)

El proyecto de NetBeans ofrece documentación y recursos de formación exhaustivos, así como una amplia selección de complementos de terceros.

El IDE NetBeans 6.9 introduce JavaFX Composer, una herramienta de diseño para la creación de aplicaciones gráficas JavaFX, parecido al constructor de aplicaciones gráficas Swing para aplicaciones Java SE. Otras notoriedades incluyen la interoperatividad OSGi para aplicaciones de plataforma NetBeans, y la compatibilidad para desarrollar paquetes OSGi (Open Services Gateway Initiative) con Maven; compatibilidad para el SDK de JavaFX 1.3.1, con el marco de trabajo Symphony y RoR (Ruby on Rails) 3.0; así como mejoras en el editor Java, Depurador Java, seguimiento de incidencias, y muchas más. (Oracle, 2012)

El IDE Netbeans 6.9 tiene más prestaciones como son las que se muestran a continuación: (Oracle, 2012)

OSGi

- Desarrollo de paquetes OSGi con Maven.
- Utilización de paquetes OSGi en una aplicación RCP de NetBeans.

Plataformas

- Interoperatividad OSGi (desarrollo y utilización de paquetes OSGi en aplicaciones basadas en plataformas).
- Compatibilidad OSGi para Felix 2.0.3, compatibilidad experimental para Equinox.
- Conversión de módulos NetBeans a paquetes OSGi, y ejecutarlos en un contenedor OSGi sin modificación.
- Edición con marcas, para proyectos basados en las plataformas de Ant y de Maven.

- Aplicaciones basadas con la plataforma NetBeans capaces de generar instaladores para los sistemas operativos más comunes.
- Compatibilidad mejorada para la utilización de aplicaciones y la conexión a bases de datos.

JavaFX

- JavaFX Composer para superficies de usuario parecidas a formularios, con control y acceso a diversas fuentes de datos.
- Añadidos y mejorados consejos en el editor y en la reestructuración.
- Paletas mejoradas de JavaFX: formas, colores, efectos y gráficas.
- Formateo de código mejorado.

Java EE

- Compatibilidad para CDI (Contexts and Dependency Injection) (JSR-299).
- Compatibilidad con Spring Framework 3.0.
- Inclusión del Servidor GlassFish OSE (Open Source Edition) 3.0.1.
- Regeneración de entidades JPA tras un cambio en la base de datos.
- Compatibilidad para servicios web REST para aplicaciones RCP.
- Java.
- Agrupación de puntos de interrupción en el Depurador de Java, histórico de parámetros adjuntados al depurador.
- Compatibilidad para procesadores de anotaciones para el editor, configurable en las propiedades del proyecto.
- Compatibilidad para nuevos Applets y Web Starts (JNLP).
- Navegación mejorada en el analizador de trazas de pila y URLs (ir a la acción del método Overridden/Implemented).

Lenguajes Web: HTML, CSS, JavaScript

- Reestructuración y búsqueda de usos para CSS y lenguajes parecidos a HTML.
- Autocompletado y enlace para atributos id y class.
- Reestructuración de estilos en línea de CSS.

Java Card

- Creación de servlets Java Card Connected 3, applets extendidos y applets clásicos.
- Compatibilidad retrospectiva con Java Card 2.2.x y Java Card Classic.
- Depuración y creación de múltiples Plataformas/Dispositivos Java Card.
- Está acorde con la Implementación de Referencia de Java Card.

PHP

- Soporte para el marco de trabajo de Symfony.
- Anotaciones "Overrides/Implements" y "Is Overridden/Implemented".
- Nuevo formateador con más reglas de formateo.
- Ruby.
- Compatibilidad Ruby on Rails 3.0.

C/C++

- Integración de pruebas unitarias con proyectos C/C++.
- Desarrollo remoto mejorado, incluyendo la descarga y navegación en ficheros remotos.
- Compatibilidad mejorada para objetivos makefile y fortran.
- Navegación mediante hiperenlaces entre métodos overridden y clases.

General

- Muchas más sugerencias de edición para Java y Ruby.
- Corrector ortográfico en el Editor.
- Opción del menú de ayuda para reportar fallos y de lentitud que afecta al IDE NetBeans.
- Compatibilidad para diversos servidores de trabajo en equipo basados en Kenai (por ejemplo: kenai.com y netbeans.org).

1.7. Marco de trabajo de desarrollo en PHP.

Symfony simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, él proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, Symfony facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (Potencier, et al., 2008)

El marco de trabajo Symfony está diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. (Potencier, et al., 2008)

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Él

es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas (Unix, Linux) como en plataformas Windows. (Potencier, et al., 2008)

Las bases de datos siguen una estructura relacional. PHP 5 y Symfony por el contrario son orientados a objetos. Por este motivo, para acceder a la base de datos como si fuera orientada a objetos, es necesaria una interfaz que traduzca la lógica de los objetos a la lógica relacional. Esta interfaz se denomina "mapeo de objetos a bases de datos" (ORM, de sus siglas en inglés "object-relational mapping"). La principal ventaja que aporta el ORM es la reutilización, permitiendo llamar a los métodos de un objeto de datos desde varias partes de la aplicación e incluso desde diferentes aplicaciones. (Potencier, et al., 2008)

Symfony trae incluido el marco de trabajo Lime que proporciona el soporte para las pruebas unitarias, es más eficiente que otros marcos de trabajo de pruebas de PHP y tiene las siguientes ventajas: (Potencier, et al., 2008)

- Ejecuta los archivos de prueba en un entorno independiente para evitar interferencias entre las diferentes pruebas. No todos los marcos de trabajo de pruebas garantizan un entorno de ejecución "limpio" para cada prueba.
- Las pruebas de Lime son fáciles de leer y sus resultados también lo son. En los sistemas operativos que lo soportan, los resultados de Lime utilizan diferentes colores para mostrar de forma clara la información más importante.
- Symfony utiliza Lime para sus propias pruebas y su "regression testing", por lo que el código fuente de Symfony incluye muchos ejemplos reales de pruebas unitarias y funcionales.

La versión que se desea utilizar en la confección de la herramienta es la 1.3, la cual tiene muchos rasgos que las distinguen de sus versiones anteriores. Dicha versión tiene incluida el trabajo con "Mailer" basado en la librería "SwiftMailer 4.1" que se utiliza para enviar correo a través de la aplicación que se esté realizando, utilizando el método `composeAndSend()`. En la parte de seguridad, cuando se crea una nueva aplicación con la tarea `generate:app`, las opciones de seguridad ahora están activadas por defecto. En dicha versión aparecen también los llamados `widgets`¹ que a través de ellos los programadores pueden acceder a los atributos de los objetos que se crean en la aplicación y a través de los validadores se puede garantizar que la información ingresada en la solución este correcta. Aparecen también las clases formulario que en

¹ el submarco de trabajo de los widgets incluye las clases que muestra el código HTML de los campos del formulario.

ellas se encuentra todo la programación del lado del cliente para que esté más organizada, en dicha clase se encuentran los lenguaje de programación como son: PHP, JavaScript, CSS, HTML que son los que se utilizan del lado del cliente.

1.8. Herramienta Case.

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. La misma fue concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de probada utilidad para el analista. Fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos. (Pressman, 2005)

Es un software que trabaja sobre múltiples plataformas (Windows, Linux). El diseño de la herramienta está centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. Con la herramienta se puede hacer la ingeniería directa e inversa, posee una licencia comercial. (Pressman, 2005)

Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y su mantenimiento. Aumenta el conocimiento informático de una empresa, ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la ingeniería del software. (Pressman, 2005)

1.9. Pruebas de Calidad.

Las pruebas de calidad son procesos que permiten comprobar y mostrar la calidad de un producto de software. A través de ellas se puede identificar donde se encuentran los posibles errores de implementación, calidad o de usabilidad que presenta el software al que son aplicadas. Básicamente es una fase en el desarrollo del software.

1.9.1. Prueba de Caja Negra.

La prueba de Caja Negra se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten 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. La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca. Muchos autores consideran que estas pruebas permiten encontrar funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a las bases de datos externas, errores de rendimiento, errores de inicialización y terminación. Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa, según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa responda bien. Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están: (Pressman, 2002)

- **Técnica de la Partición de Equivalencia:** esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- **Técnica del Análisis de Valores Límites:** esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- **Técnica de Grafos de Causa-Efecto:** es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. 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. (Pressman, 2002)

1.9.2. Prueba de Caja Blanca.

La prueba de Caja Blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que: (Pressman, 2002)

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

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática. (Pressman, 2002)

Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecute por lo menos una vez, cada sentencia del programa. (Pressman, 2002)

En este capítulo se ha mostrado la información necesaria sobre toda la tecnología a utilizar en el desarrollo de la herramienta.

Con el desarrollo de este capítulo se llevó a cabo un estudio sobre toda la tecnología, herramientas y metodologías que permitirán el desarrollo de la aplicación. Se selecciona a RUP como metodología de desarrollo, PHP5 como lenguaje de programación, Symfony 1.3 como marco de trabajo para el desarrollo, Netbeans 6.9 como IDE de desarrollo, Apache 2 como servidor web, PostgreSQL como sistema de gestor de base de datos y Visual Paradigm como herramienta CASE.

Capítulo 2. Propuesta de la Solución.

En el presente capítulo se abordarán aspectos sobre la solución del problema, en el mismo se definirán los requisitos funcionales y no funcionales, así como las clases del análisis, incluyendo sus diagramas y la descripción de las funcionalidades que presentará la herramienta.

2.1. Características de la herramienta.

La herramienta Diccionario Jurídico dispondrá de un buscador de términos que podrían ser categorizados por temas y subtemas relacionados con el ámbito jurídico y que son tributarias a necesidades del fiscal en la toma de decisiones.

La herramienta permitirá incluir nuevos términos por vía de una solicitud que se realiza a la comisión que evaluará la inclusión o no del mismo. El fallo de la comisión será informado al solicitante a través del servicio de correo electrónico que tiene incorporado la herramienta. Además la misma deberá integrarse a la herramienta de avisos que se encuentra funcional en la FGR la cual informa a la comisión que existen nuevos términos a evaluar. Deberá existir un contador que mantenga actualizada la cantidad de términos existentes, las solicitudes que se están procesando y mostrará los 10 términos incluidos recientemente, entre otros.

2.2. Definición de Requisitos.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Mientras que los requerimientos no funcionales son las propiedades o cualidades que el producto debe tener, debe pensarse en estos atributos como las características que hacen al producto atractivo, usable, rápido o confiable. (Jacobson, et al., 2000)

2.2.1. Requisitos Funcionales.

RF1. Mostrar términos.

RF2. Buscar términos.

RF3. Realizar solicitud de inclusión de nuevo término.

RF4. Modificar solicitud de inclusión de nuevo término.

RF5. Eliminar solicitud de inclusión de nuevo término.

RF6. Notificar fallo de la comisión.

RF7. Notificar a dirección.

RF8. Registrar miembro comisión evaluativa.

RF9. Modificar miembro comisión evaluativa.

RF10. Eliminar miembro comisión evaluativa.

RF11. Registrar término.

RF12. Adicionar tema.

RF13. Modificar tema.

RF14. Eliminar tema.

Las técnicas utilizadas para la validación de los requisitos fueron:

- La validación de los requisitos con el cliente. Para validar los requisitos se realizó una reunión de trabajo, el día 10 de abril del 2011 donde se le mostró al cliente el listado de requisitos siendo aprobados por el mismo.
- Prototipado. Dicha técnica consiste en la revisión de los prototipos de interfaz donde se evidencia los requisitos funcionales, el cliente en reunión de trabajo, el día 10 de abril del 2011 aprobaron los prototipos.

2.2.2. Requisitos no Funcionales.

Confidencialidad.

La información manejada por la herramienta estará protegida a través del protocolo Https por donde viaja la información cifrada y encriptada utilizando SSL.

Integridad.

La información manejada por la herramienta será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, lo mismo se logra con la validación de los datos por los script que se encuentran en las clases pertinentes y los validadores que genera Symfony, de la misma forma será considerada igual, la fuente o autoridad de los datos.

Confiabilidad.

Se garantizará la consistencia de los datos haciendo comprobaciones a través de los permisos y los roles que se asignan, con ellos se podrá tener la información necesaria, si fue alterada cualquier información y por quién fue.

Seguridad.

Los discos duros que tendrán los servidores tendrán contraseña para proteger la información sensible que se guarda. Existen los firewall de aplicación, ModScurity es la herramienta que se utiliza para ello, es la encargada de administrar toda la transferencia de datos y también se encuentran los firewall de red para mayor protección.

Apariencia o interfaz externa.

Las ventanas de la herramienta tendrán claro y bien estructurados los datos, y al mismo tiempo permitirán la interpretación correcta de la información.

Mostrar mensajes de errores en la introducción de datos de una forma sencilla y explicativa, la entrada de datos incorrecta será detectada claramente por la herramienta.

Todos los textos y mensajes serán mostrados en español.

El funcionamiento será diseñado de modo intuitivo, y que requiera de información mínima.

Hardware.

PC Servidor: Mínimo Micro Quad Core, 2 GB de Ram, 2GB de espacio libre en disco duro.

PC Cliente: Mínimo Pentium 4, 512 Mb de memoria RAM.

Software.

Sistema operativo Linux, Apache para servidor web, PostgreSQL como servidor de base de datos.

2.3. Modelado del Sistema.

2.3.1. Definición de los actores del sistema.

Actor	Descripción
Fiscal	Es el actor del sistema que realiza las peticiones de inclusión de nuevos términos.
Consultante	Es el actor que interviene en el procesamiento de las solicitudes.
Jefe de comisión	Además de intervenir en el procesamiento, toma decisiones y registra las solicitudes.
Administrador	Es el encargado de crear, modificar o eliminar los miembros de la comisión evaluativa.

Tabla 1. Definición de los actores del sistema.

2.3.2. Diagrama de Casos de Uso del Sistema.

“El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El dicho modelo de casos de uso sirve como acuerdo clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.” (Jacobson, et al., 2000)

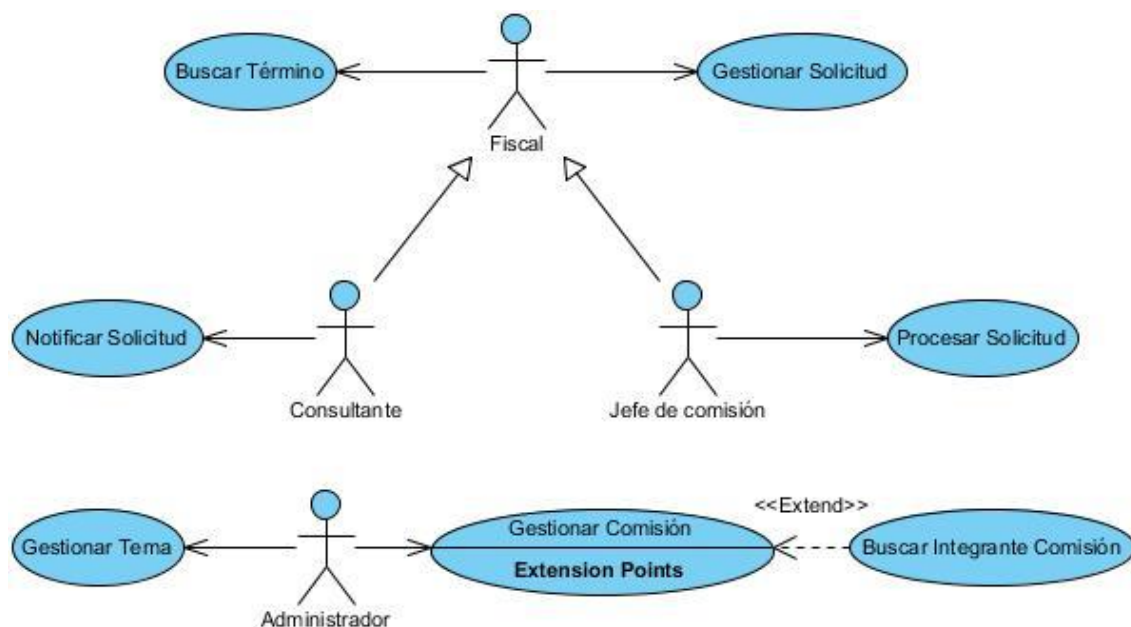


Figura 2. Diagrama de caso de uso del sistema.

2.3.3. Especificación de los Casos de Uso del Sistema.

Para entender la funcionalidad, no es suficiente con la representación gráfica del Diagrama de casos de uso. La especificación de casos de uso brinda los detalles de la secuencia de las acciones, el estado inicial y los posibles estados finales, además de cuándo comienza y termina el caso de uso. Describe explícitamente qué debe hacer el sistema, separando las responsabilidades del sistema y la de los actores.

A continuación se presenta una muestra de la descripción de los casos de uso más significativos con los que cuenta el sistema que se desarrolla, el resto de las descripciones de los casos de uso se pueden encontrar en los artefactos generados.

2.3.4. Especificación de los Casos de Uso.

2.3.4.1. Buscar Término.

Caso de Uso:	Buscar Término
Actores:	Fiscal (Inicia)
Resumen:	El caso de uso inicia cuando el Fiscal busca un término deseado. El sistema muestra el término y los datos del mismo, terminando el caso de uso.
Precondiciones:	El actor debe estar autenticado con los permisos necesarios.
Referencias	RF1, RF2
Prioridad	Alta
Flujo Normal de Eventos	

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

Sección 1: Buscar término.	
Acción del Actor	Respuesta del Sistema
1- Necesita consultar un término y da clic en la opción Buscar Términos del menú principal.	2- Muestra una interfaz con las siguientes opciones para filtrar según la necesidad. <ul style="list-style-type: none"> ➤ Nombre del término. ➤ Tema. ➤ Subtema. ➤ Términos Recientes. ➤ Términos en Procesamiento. ➤ Términos con Imágenes.
3- Selecciona ninguna o varias de las opciones del filtro.	4- En caso de que existan términos que se correspondan con el criterio de búsqueda, el sistema los muestra. 5- En caso de que no se especifique ningún criterio de búsqueda, ver <i>Flujo Alterno 1</i> . 6- En caso de que no existan términos que se correspondan con el criterio de búsqueda, ver <i>Flujo Alterno 2</i> .
7- Selecciona el término deseado.	8- Muestra los datos del término seleccionado.
9- Presiona la opción Cerrar.	10- Finaliza el caso de uso.
Prototipo de Interfaz	
<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 10px;"> Nombre del Término <input style="width: 200px; height: 20px;" type="text"/> </div> <div style="display: flex; justify-content: space-around; width: 100%;"> <div style="display: flex; align-items: center;"> Tema <input style="width: 150px; height: 20px;" type="text"/> </div> <div style="display: flex; align-items: center;"> Sub-tema <input style="width: 150px; height: 20px;" type="text"/> </div> </div> <div style="display: flex; justify-content: space-around; width: 100%; margin-top: 10px;"> <div style="display: flex; align-items: center;"> <input type="checkbox"/> Términos Recientes </div> <div style="display: flex; align-items: center;"> <input type="checkbox"/> Términos en Procesamientos </div> </div> <div style="display: flex; justify-content: space-around; width: 100%; margin-top: 10px;"> <div style="display: flex; align-items: center;"> <input checked="" type="checkbox"/> Términos con Imágenes </div> </div> <div style="display: flex; justify-content: center; margin-top: 20px;"> <div style="border: 1px solid gray; padding: 5px 20px; margin: 0 10px;">Buscar</div> <div style="border: 1px solid gray; padding: 5px 20px; margin: 0 10px;">Cancelar</div> </div> </div>	

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

Flujos Alternos 1	
Acción del Actor	Respuesta del Sistema
	1- Muestra todos los términos que se encuentran en la base de datos.
Flujos Alternos 2	
Acción del Actor	Respuesta del Sistema
	1- El sistema muestra el siguiente mensaje: "No se encontró ningún término que responda a los criterios seleccionados."
2- Ejecuta el paso 3 del Flujo Normal de Eventos.	

Tabla 2. Especificación del caso de uso Buscar Término.

2.3.4.2. Gestionar Solicitud.

Caso de Uso:	Gestionar Solicitud
Actores:	Fiscal (Inicia)
Resumen:	El caso de uso inicia cuando el Fiscal solicita la inclusión, modificación o eliminación de un término, dicha solicitud llega a la comisión de evaluación y finaliza el caso de uso.
Precondiciones:	El actor debe estar autenticado con los permisos necesarios.
Referencias	RF3, RF4, RF5
Prioridad	Alta
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción Propuesta de Nuevo Término del menú principal.	2- Muestra el listado de solicitudes realizadas por el fiscal que aún no han sido aprobadas o rechazadas por la comisión.
3- En caso de que seleccione la opción Nuevo, ver Sección 1: Registrar Solicitud.	5- El sistema guarda la solicitud y la incluye en el listado de solicitudes realizadas por el solicitante.
4- En caso de que seleccione una solicitud, ver Sección 2: Modificar Solicitud.	
Prototipo de Interfaz	

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 50%;">Término</th> <th style="width: 50%;">Fecha de Creación</th> </tr> <tr> <td>Abandono</td> <td>29/03/2011</td> </tr> <tr> <td>Absolución</td> <td>05/04/2011</td> </tr> <tr> <td>Absolución judicial</td> <td>05/04/2011</td> </tr> <tr> <td>Abstención delictiva</td> <td>04/06/2011</td> </tr> </table>	Término	Fecha de Creación	Abandono	29/03/2011	Absolución	05/04/2011	Absolución judicial	05/04/2011	Abstención delictiva	04/06/2011	<table style="width: 100%;"> <tr> <td style="border: 1px solid gray; padding: 2px 10px;">Nuevo</td> <td style="border: 1px solid gray; padding: 2px 10px;">Cerrar</td> </tr> </table>	Nuevo	Cerrar
Término	Fecha de Creación												
Abandono	29/03/2011												
Absolución	05/04/2011												
Absolución judicial	05/04/2011												
Abstención delictiva	04/06/2011												
Nuevo	Cerrar												
Sección 1: Registrar Solicitud													
Acción del Actor	Respuesta del Sistema												
	<p>1- Muestra una interfaz para introducir los siguientes datos:</p> <ul style="list-style-type: none"> ➤ Término (obligatorio). ➤ Concepto (obligatorio). ➤ Comentario doctrinal. ➤ Imagen. ➤ Tema asociado ➤ Sinónimos. ➤ Término relacionado. 												
<p>2- Introduce los datos y presiona la opción Guardar.</p>	<p>3- Valida los campos obligatorios:</p> <p>4- En caso de que se deje de introducir al menos un campo obligatorio, <i>ver Flujo Alterno 1.</i></p> <p>5- En caso de que estén completos la totalidad de los campos obligatorios el sistema guarda la solicitud, muestra la pantalla que contiene el listado de solicitudes realizadas por el solicitante incluyendo dicha solicitud.</p>												
	<p>6- Finaliza el caso de uso.</p>												
Prototipo de Interfaz													

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

Término *

Concepto *

Comentario Doctrinal

Imagen Tema Asociado

Sinónimos
Términos

Término Relacionado
Términos

Sección 2: Modificar Solicitud

Acción del Actor	Respuesta del Sistema
	1- Muestra los datos de la solicitud.
2- En caso de que desee modificar la solicitud introduce las modificaciones deseadas y selecciona la opción Guardar.	4- Valida los campos obligatorios: 5- En caso de que se deje de introducir al menos un campo obligatorio, <i>ver Flujo Alterno 1</i> .
3- En caso de que desee eliminar la solicitud selecciona la opción Eliminar, <i>ver Sección 3</i> .	6- En caso de que estén completos la totalidad de los campos obligatorios el sistema actualiza la solicitud, muestra la pantalla que contiene el listado de solicitudes realizadas por el solicitante actualizando dicha solicitud.
	7- Finaliza el caso de uso.

Prototipo de Interfaz

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

Término *

Concepto *

Comentario Doctrinal

Imagen Tema Asociado

Sinónimos
Términos

Término	
Absolución	<input type="button" value="Eliminar"/>

Término Relacionado
Términos

Término	
Absolución	<input type="button" value="Eliminar"/>

Sección 2: Eliminar Solicitud

Acción del Actor	Respuesta del Sistema
	1- Muestra un mensaje de confirmación de la operación.
2- Selecciona la opción Aceptar.	3- Elimina la solicitud, muestra la pantalla que contiene el listado de solicitudes realizadas por el solicitante eliminando dicha solicitud.
	4- Finaliza el caso de uso.

Flujo Alternativo 1

Acción del Actor	Respuesta del Sistema
	1- Muestra el campo que debe ser completado con el siguiente mensaje "El campo "X" es requerido"
2- Introduce el dato correspondiente.	3- Ejecuta el paso 3 del Flujo Normal de Eventos de la Sección 1: Registrar

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

	Solicitud o ejecuta el paso 4 del Flujo Normal de Eventos de la Sección 2: Modificar Solicitud.
Prototipo de Interfaz	
Poscondiciones Sección 1	Se registra una nueva solicitud.
Poscondiciones Sección 2	Se actualiza una nueva solicitud.
Poscondiciones Sección 3	Se elimina una nueva solicitud.

Tabla 3. Especificación del caso de uso Gestionar Solicitud.

2.4. Análisis.

Durante esta etapa en el desarrollo del software, se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema. (Jacobson, et al., 2000).

2.4.1. Modelo de Clases del Análisis.

El siguiente modelo brinda una información resumida de las clases que intervienen en el desarrollo de la herramienta. El resto de los diagramas de clases del análisis se encuentra en los artefactos generados.

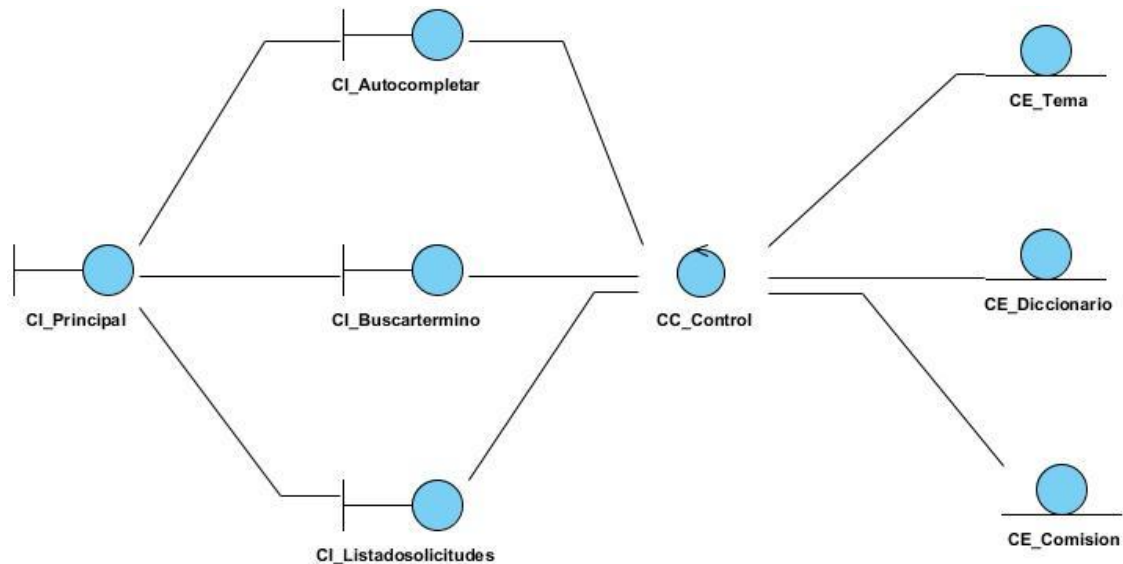


Figura 3. Diagrama de clases del análisis del caso de uso Buscar Término.

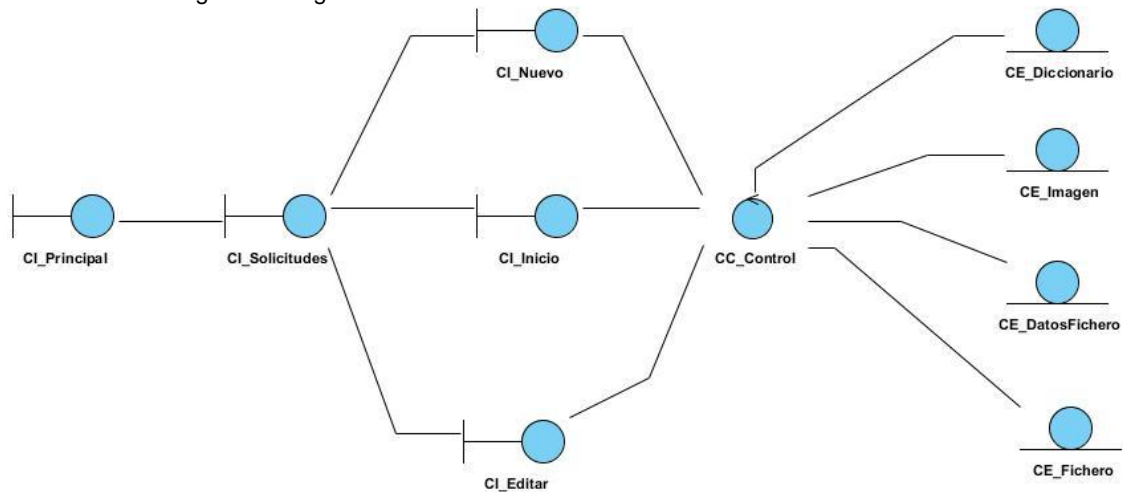


Figura 4. Diagrama de clases del análisis del caso de uso Gestionar Solicitud.

2.4.2. Diagrama de interacción.

El diagrama de interacción muestra el flujo de mensaje y operaciones que se realiza en algunas de las funcionalidades que presenta la aplicación. El resto de los diagramas de secuencias se encuentra en los artefactos generados.

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

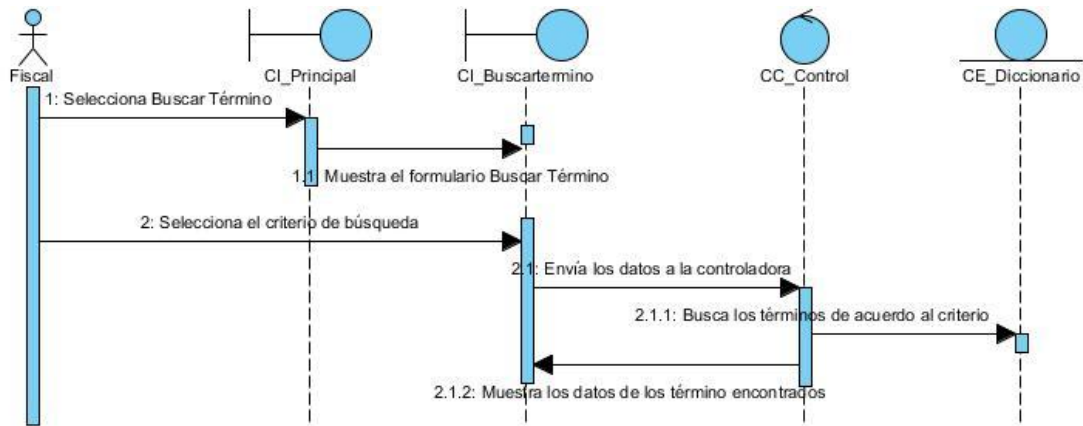


Figura 5. Diagrama de secuencia del caso de uso Buscar Término.

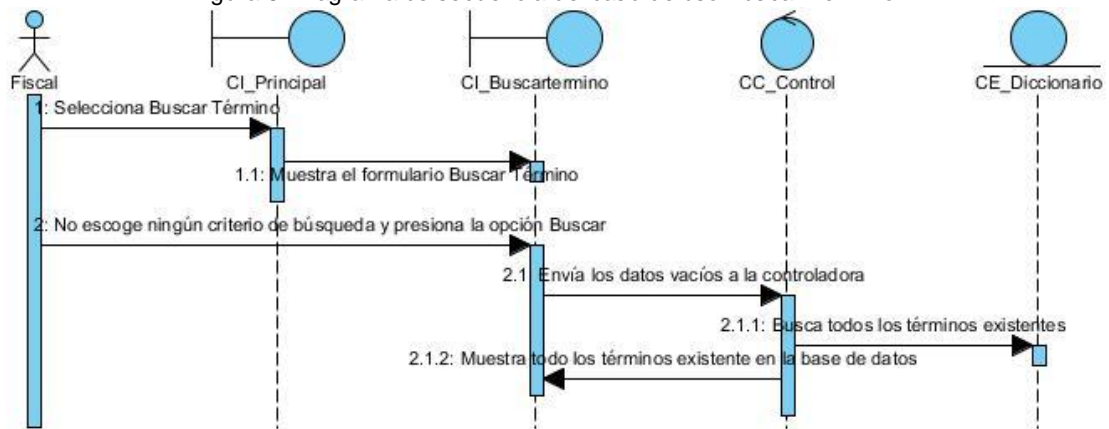


Figura 6. Diagrama de secuencia del Flujo Alternativo 1 del caso de uso Buscar Término.

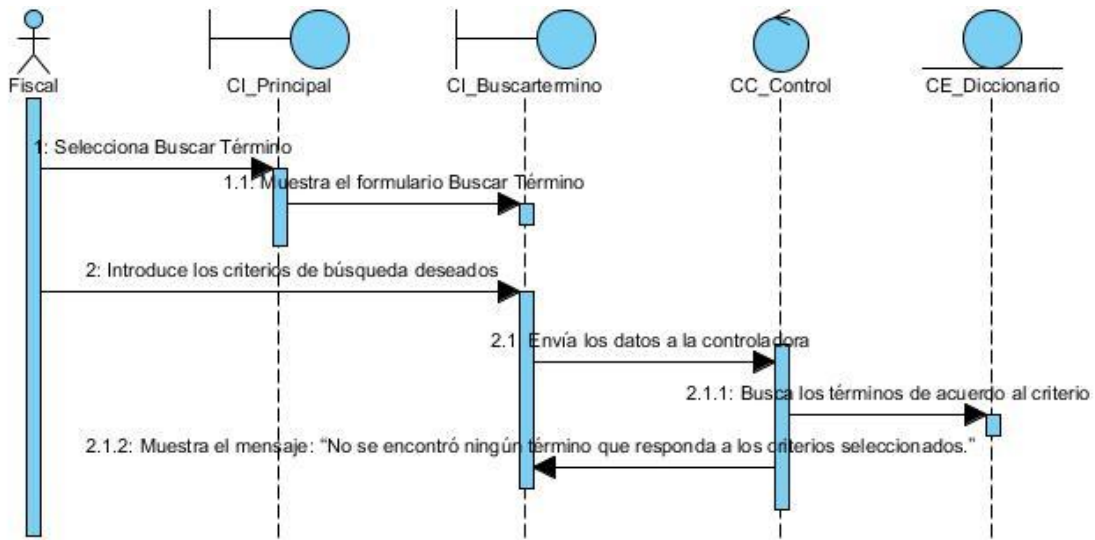


Figura 7. Diagrama de secuencia del Flujo Alternativo 2 del caso de uso Buscar Término.

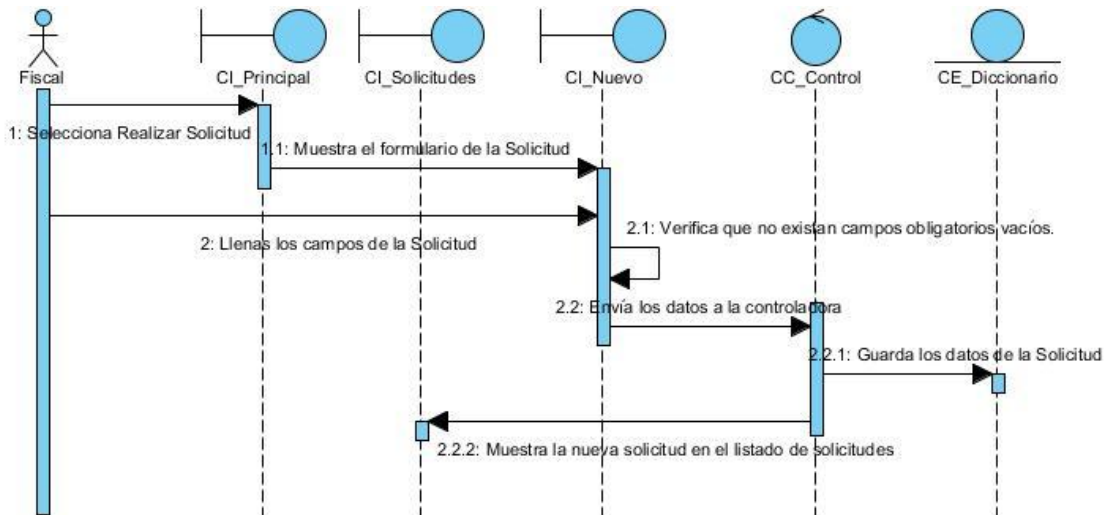


Figura 8. Diagrama de secuencia de la Sección 1: Registrar Solicitud del caso de uso Gestionar Solicitud.

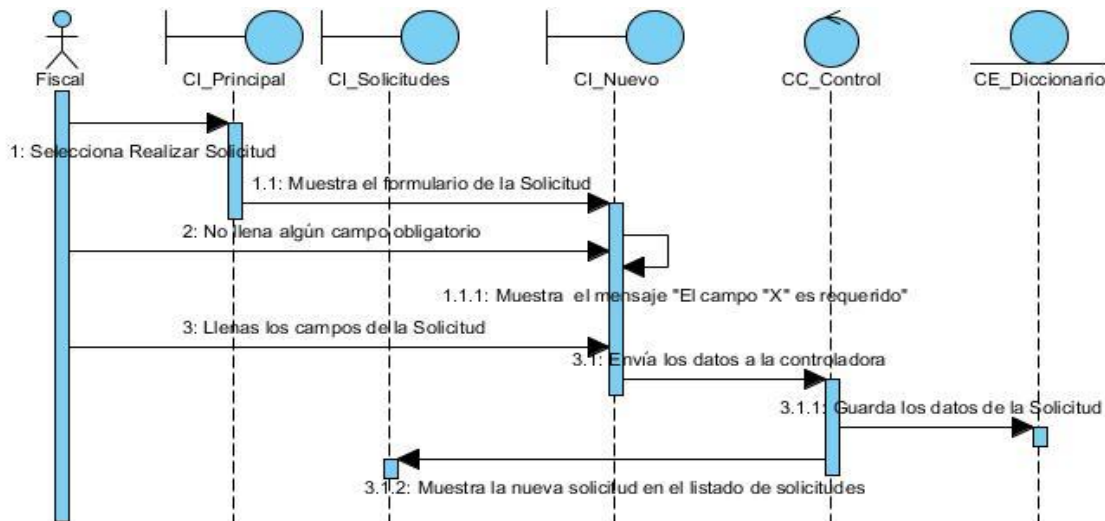


Figura 9. Diagrama de secuencia del Flujo Alternativo 1 del caso de uso Gestionar Solicitud.

En este capítulo se realizaron los pasos iniciales para el desarrollo de la solución. Se obtuvieron los requisitos funcionales y no funcionales, así como la especificación de los casos de usos del sistema correspondientes, los diagramas de clases del análisis y de secuencia que son importantes para el diseño y programación que se realizarán en el capítulo siguiente.

Capítulo 3. Diseño e Implementación.

En el presente capítulo se realizará una descripción de los principales patrones de diseño utilizados en el desarrollo de la solución, además se realizarán los principales artefactos pertenecientes a los flujos de diseño e implementación.

3.1. Diagramas de clases del Diseño.

Un diagrama de diseño es un conjunto de artefactos relacionados entre sí, para mostrar gráficamente las relaciones que existen entre las distintas clases que intervienen en el desarrollo de la aplicación.

El modelo de diseño es un modelo de objeto que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. (Jacobson, et al., 2000)

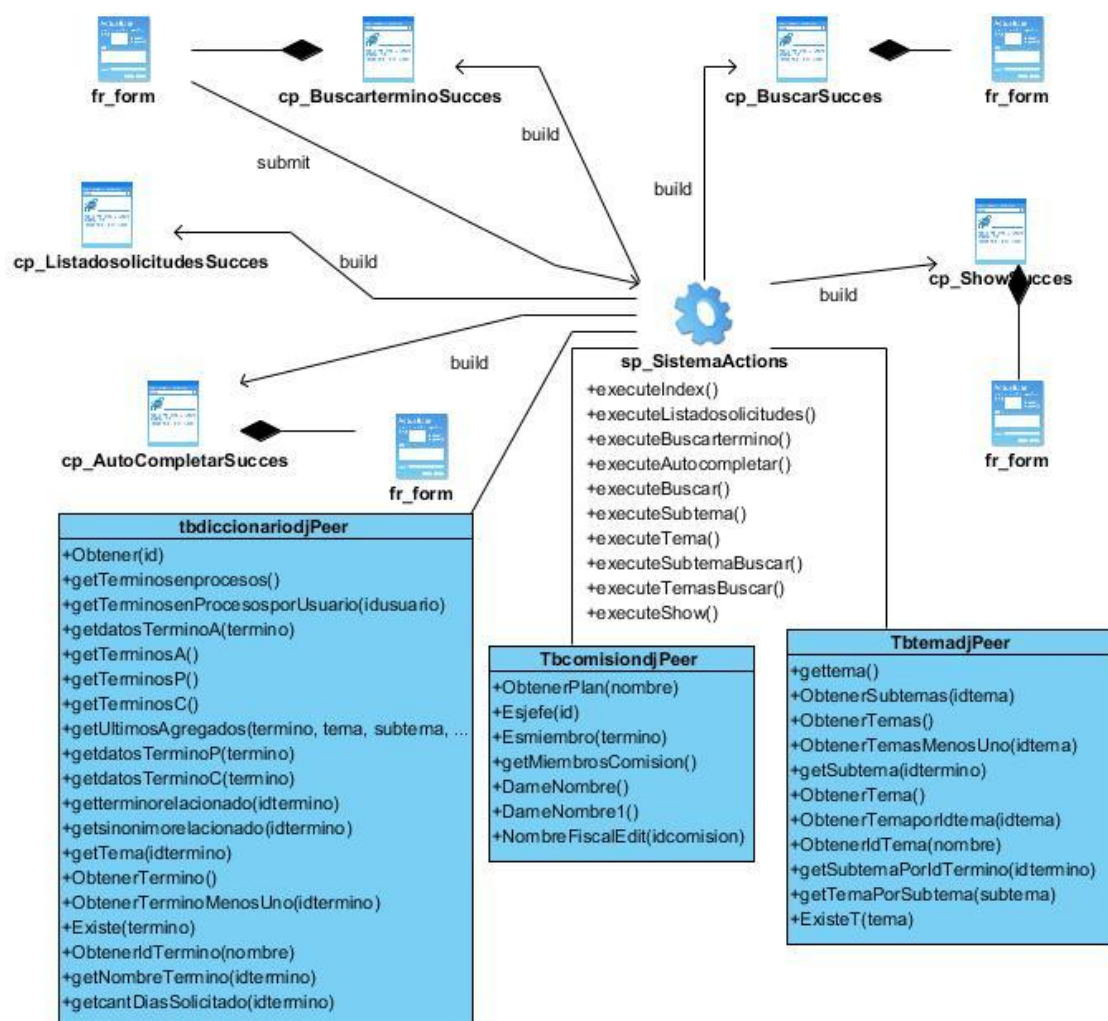


Figura 10. Diagrama de clases del diseño del caso de uso Buscar Término.

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

A continuación se muestra una breve descripción de los métodos de la clase controladora del diagrama de clases del diseño del caso de uso Buscar Término.

Buscar Término	
Nombre: diccionariojuridActions	
Nombre	Descripción
executeIndex()	Es el método encargado de ejecutar varias consultas a la base de datos, para inicializar las variables necesarias para brindar información a la interfaz.
executeListadosolicitudes()	Retorna una lista de las solicitudes realizadas y que no han sido procesadas.
executeBuscartermino()	Se envía hacia la vista una lista de temas obtenidos de la base de datos.
executeAutocompletar()	Este método tiene la responsabilidad de autocompletar las palabras cuando se está escribiendo el nombre del término en el criterio de búsqueda.
executeBuscar()	Se realiza la búsqueda del término según el criterio seleccionado y se envía hacia la vista.
executeSubtema()	En este método se realiza la búsqueda del subtema relacionado al término que se desea visualizar.
executeTemas()	En este método se realiza la búsqueda del tema relacionado al término que se desea visualizar.
executeSubtemaBuscar()	Se responsabiliza con el envío hacia la vista del listado de subtemas, dado el identificador tema con el que se relacionan.
executeTemasBuscar()	Se envía hacia la vista de los temas.
executeShow()	Dado el identificador de un término se envía los datos relacionado al mismo.

Tabla 5. Descripciones de los métodos de la clase controladora Buscar Término.

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

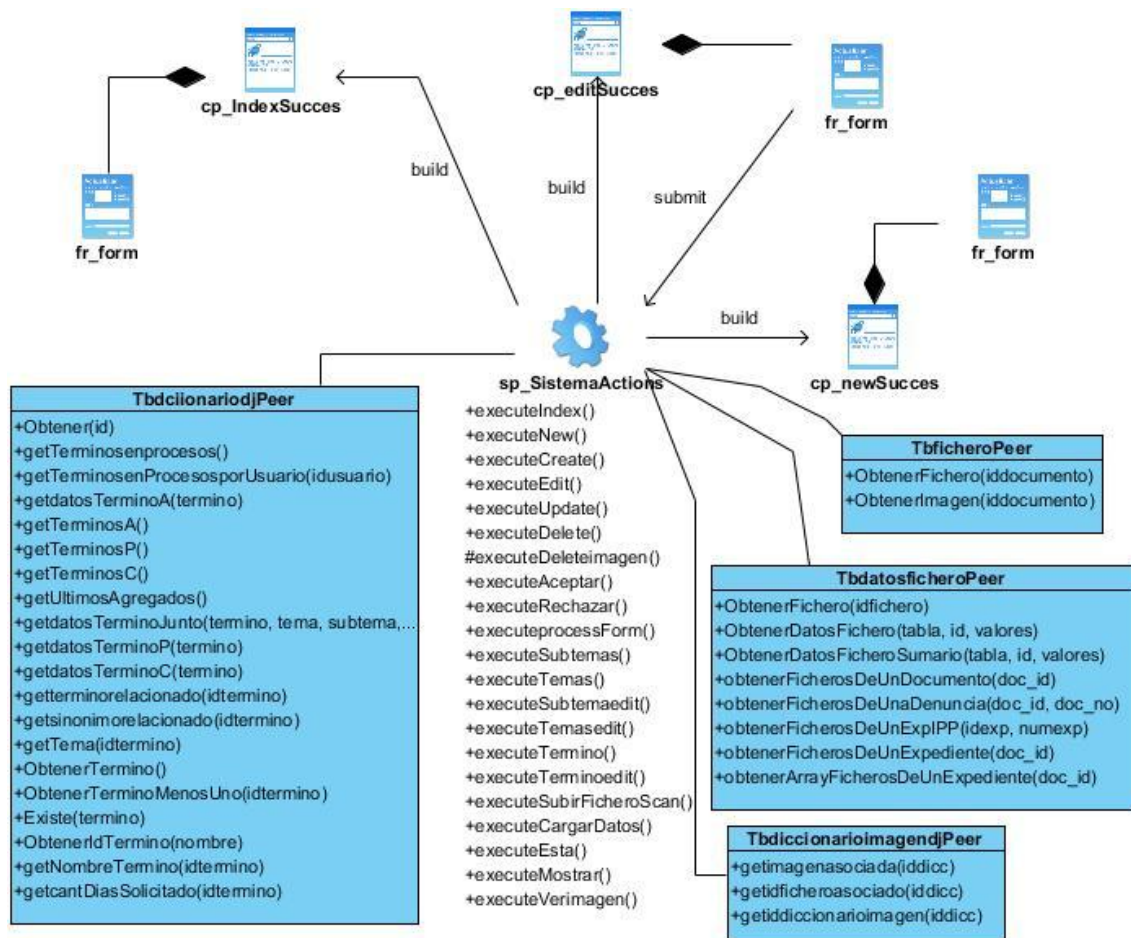


Figura 11. Diagrama de clases del diseño del caso de uso Gestionar Solicitud.

A continuación se muestra una breve descripción de los métodos de la clase controladora del diagrama de clases del diseño del caso de uso Gestionar Solicitud.

Gestionar Solicitud	
Nombre: realizarsolicitudActions	
Nombre	Descripción
executeIndex()	Dado el identificador del fiscal se envía hacia la vista la lista de términos realizados por él.
executeNew()	Crea el formulario tbdiccionarioDjForm() y carga los datos necesarios para la creación de una nueva solicitud de inclusión de término.
executeCreate()	En este método se re-direcciona hacia el método executeNew().
executeEdit()	Este método permite editar la solicitud de término una vez insertada.
executeUpdate()	En este método se re-direcciona hacia el

*Módulo Diccionario Jurídico para el Sistema de Informatización
de la Gestión de la Fiscalía.*

	método Edit.
executeDelete()	Este método permite el borrado de una solicitud de un término y re-direcciona hacia el excuteIndex().
executeDeleteimagen()	Este método proporciona la posibilidad de eliminar la imagen asociada al término.
executeAceptar()	Este método tiene la responsabilidad de cambiar el estado del término a "Aceptado" y envía por correo la notificación de la acción realizada al solicitante.
executeRechazar()	Envía por correo la notificación de que la solicitud realizada fue rechazada.
processForm()	Este método es el encargado de guardar los datos que son introducidos en la aplicación en la base de datos.
executeSubtema()	Dado el identificar de un tema envía hacia la vista una lista de los subtemas.
executeTemas()	Este método permite enviar hacia la vista la lista de temas.
executeSubtemaedit()	Este método permite cargar el subtema que tiene la solicitud cuando se quiera mostrar los datos de ella.
executeTemasedit()	Permitir cargar el tema que tiene la solicitud cuando se quiera mostrar los datos de ella.
executeTermino()	Este método permite cargar el término que tiene la solicitud cuando se quiera mostrar los datos de ella.
executeTerminoedit()	Permitir cargar los términos que son sinónimos y relacionados de la solicitud cuando se quiera mostrar los datos de ella.
executeSubirFicheroScan()	Este tiene la responsabilidad de guardar la imagen en la base de datos.
executeCargarDatos()	Este método permite cargar los datos de la imagen dado el identificador del término.
executeMostrar()	Este método permite mostrar la imagen en el formulario.
executeEliminar()	Este método permite eliminar la imagen.

executeVerImagen()	Este método permite ver la imagen en una ventana auxiliar.
--------------------	--

Tabla 6. Descripciones de los métodos de la clase controladora Gestionar Solicitud.

3.2. Patrones Utilizados.

Symfony está basado en un patrón clásico del diseño web, conocido como estilo arquitectónico MVC, que está formado por tres niveles: (Potencier, et al., 2008)

- ✓ El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ✓ La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- ✓ El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

El uso del marco de trabajo que utiliza MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por él. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. Aplicar el patrón MVC a una aplicación resulta bastante útil. La implementación que realiza Symfony de la arquitectura MVC incluye varias clases como son:

sfController: es la clase del controlador y se encarga de decodificar la petición y transferirla a la acción correspondiente.

sfRequest: guarda todos los elementos que integran la petición (parámetros, cookies, cabeceras)

sfResponse: posee las cabeceras de la respuesta y los contenidos. El contenido de este objeto se convierte en la respuesta HTML que se remite al usuario.

El solitario de contexto (que se obtiene mediante `sfContext::getInstance()`): guarda una referencia a todos los objetos que constituyen el núcleo de Symfony y puede ser accedido desde cualquier parte de la aplicación.

Symfony toma lo mejor de la arquitectura MVC y la realiza de modo que el desarrollo de aplicaciones sea rápido y sencillo. En el controlador se encuentran las acciones, las cuales son el núcleo de la aplicación, pues contienen toda la lógica de la misma. Estas acciones utilizan el modelo y precisan las variables para la vista. Al realizarse una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición.

La vista es la encargada de originar las páginas que son mostradas como resultado de las acciones, donde se encuentra el layout, que es común para todas las páginas de la aplicación. La vista en Symfony está conformada por varias partes, preparadas cada una de ellas especialmente para ser fácilmente transformable por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones.

En el modelo se encuentran las clases, que son generadas de forma automática según la estructura de la base de datos. En Symfony, el acceso y la modificación de los datos que se almacenan en la base de datos, se realiza mediante objetos. Propel es el motor generador que se encarga de la generación automática para construir sus clases, creando la estructura y generando el código de las mismas. A medida que el desarrollo de un proyecto va avanzando, puede ser necesario agregar métodos y propiedades personalizadas en los objetos del modelo, esto trae consigo que se aumenten las tablas o columnas. Así mismo, cada vez que se modifica, se deben regenerar las clases del modelo de objetos. Si se añaden los métodos personalizados en las clases que se generan, cada vez que se vuelvan a generar esas clases estos métodos se borrarían.

Las clases con nombre Base son generadas directamente a partir del esquema. Estas clases no deberían modificarse, ya que cada vez que se genera el modelo, todas las clases se borran. Por otro lado, las clases de objetos propias que heredan de las clases con nombre Base no se modifican, es por ello que son estas clases en las que se añaden los métodos propios. Las mismas heredan todos los métodos de la clase padre correspondiente, pero no son afectadas por las modificaciones en el esquema. (Potencier, et al., 2008)

3.2.1. Patrones GRASP.

Experto: es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión de la librería Propel para mapear la Base de Datos. Symfony utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos (Peer del Modelo) poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.

Creador: en la clase Actions se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase Actions es “creador” de

dichas entidades. Ejemplos de algunas funciones utilizadas en la clase Actions son: doSelect (), retrieveByPK (), doSelectOne ().

Alta Cohesión: Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Actions, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties.

Bajo Acoplamiento: la clase Actions hereda únicamente de sfActions para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Controlador: todas las peticiones Web son manipuladas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases sfFrontController, sfWebFrontController, sfContext, los "actions" y el index.php del ambiente. (Potencier, et al., 2008)

3.2.2. Patrones GoF.

Solitario: la clase sfRouting que tiene el método getInstance, utiliza el controlador frontal (sfWebFrontController) y se encarga de enrutar todas las peticiones que se hagan a la aplicación. El solitario sfRouting precisa otros métodos muy útiles para la gestión manual de las rutas: ClearRoutes (), hasRoutes (), getRoutesByName ().

Comando: este patrón se observa en la clase sfWebFrontController, en el método dispatch(). Esta clase está por defecto y es la encargada de establecer el módulo y la acción que se va a usar según la petición del usuario. Este patrón se aplica además en la clase sfRouting, que está desactivada por defecto y procede según las necesidades del administrador del sistema donde se aplique el marco de trabajo, la cual se puede activar o desactivar. En este método es parseada la URL con el objetivo de precisar los parámetros de la misma y de esta forma saber el Actions que debe responder a la petición. (Potencier, et al., 2008)

Decorador: la clase abstracta sfView, es padre de todas las vistas que contienen un decorador para permitir agregar funcionalidades dinámicamente. El archivo nombrado layout.php es el que contiene el layout de la página. Este archivo, conocido también

como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este procedimiento es una implementación del patrón Decorador. (Potencier, et al., 2008)

3.3. Modelo físico de la base de datos.

El modelo físico de la base de datos muestra la relación que existe entre las entidades que están presentes en la herramienta, como se estructuran las tablas y los atributos que guardan cada una.

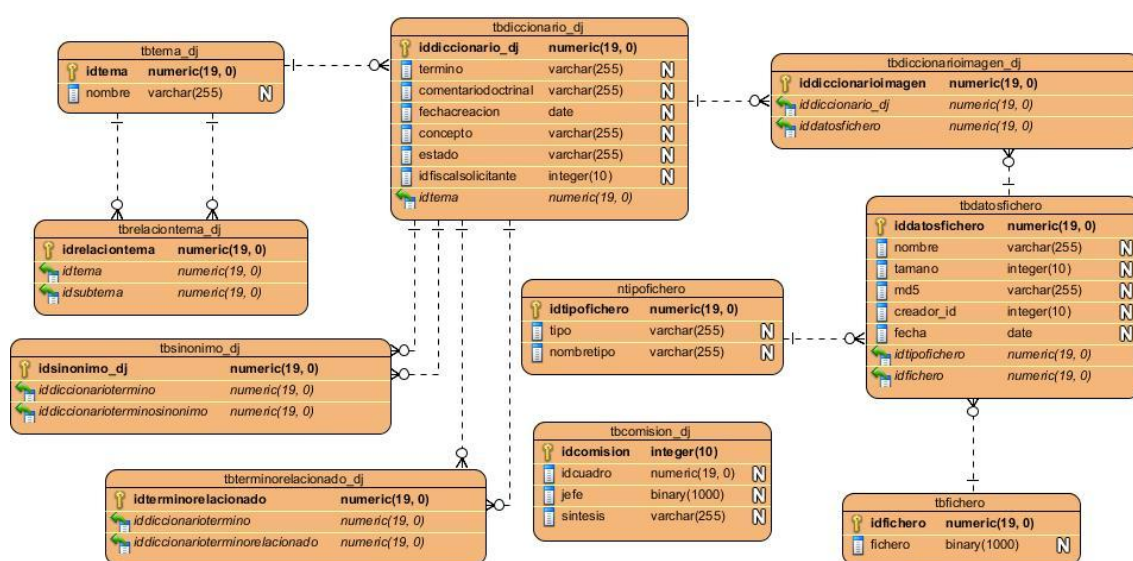


Figura 12. Modelo físico de la base de datos.

A continuación se muestra una breve descripción de las tablas y atributos.

Nombra de la tabla	Descripción	Atributo	Descripción
tbtema_dj	Guarda los temas que son insertados en la solución.	idtema	Es la llave primaria de la tabla.
		nombre	Nombre del tema insertado.
tbrelaciontema_dj	Guarda la relación de cuáles son los subtemas de los temas insertados.	idrelaciontema	Es la llave primaria de la tabla.
		idtema	Es llave foránea de la tabla tbtema_dj que representa al tema en esta relación.
		idsubtema	Es llave foránea de la

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

			tabla tbtema_dj que representa al subtema en esta relación.
Tbdiccionario_dj	Guarda las características de los términos que son introducidos en la solución.	Idiccionario_dj	Es la llave primaria de la tabla.
		término	Nombre del término, este campo es obligatorio.
		comentariodoctrinal	Datos importantes que no están contemplados en el concepto.
		fechacreacion	Fecha de creación del término.
		concepto	Concepto del término, este campo es obligatorio.
		estado	Estado en que se encuentra el término dentro de la solución.
		idfiscalsolicitante	Guarda el id del fiscal que está ingresando el término.
		idtema	Llave foránea de la tabla tbtema_dj, guardando el tema en el cual está incluido el término.
Tbsinonimo_dj	Guarda los términos que tienen sinónimos.	Idsinonimo_dj	Es la llave primaria de la tabla.
		iddiccionariotermino	Son llaves foráneas de la tabla tbdiccionario_dj, representa la relación entre término que son sinónimo.
		Idcccionariotermino sinonimo	

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

Tbtermino relacionado_dj	Guarda los términos que tienen relación con otros.	ldtermino relacionado	Es la llave primaria de la tabla.
		iddiccionariotermino	Son llaves foráneas de la tabla tbdiccionario_dj, representa la relación que existen entre dos términos.
		Iddiccionariotermino relacionado	
tbdatosfichero	Esta tabla lo que tiene guardado es toda la información de los ficheros.	iddatosfichero	Es la llave primaria de la tabla.
		nombre	Nombre del fichero que se está guardando.
		idtipofichero	Es la llave foránea de una tabla ntipofichero que controla el tipo de fichero que se está guardando.
		tamaño	Tamaño que se lleva el fichero ingresado en la herramienta.
		md5	Forma de encriptación del fichero.
		creador	Identificador del usuario que guarda el fichero.
		fecha	Fecha en que se guardo el fichero
		idfichero	Es llave foránea de la tabla tbfichero.
		tbdiccionario imagen_dj	Representa la relación de mucho a mucho entre las tablas tbdiccionario_dj y tbdatosfichero.
iddiccionario_dj	Es llave foránea de la tabla tbdiccionario_dj.		
iddatosficheros	Es llave foránea de la tabla tbdatosfichero.		
tbfichero	Guarda el fichero.	idfichero	Es la llave primaria

			de la tabla.
		fichero	Almacena el fichero.
tbcomision_dj	Guarda toda la información sobre los miembros de la comisión.	ldcomision	Es la llave primaria de la tabla.
		idcuadro	Es llave foránea de la tabla tbcuadro.
		jefe	Guarda la información si es jefe o no.
		sintesis	Una breve descripción sobre la persona.
ntipofichero	Tabla nomenclador que guarda la información de los ficheros que puede almacenar en el Sistema de informatización de la gestión de la fiscalía.	idtipofichero	Es la llave primaria de la tabla.
		tipo	Tipo de fichero (.doc, .jpg, .pdf).
		nombretipo	Nombre del tipo de fichero.

Tabla 4. Descripción de las tablas del modelo físico de la base de datos.

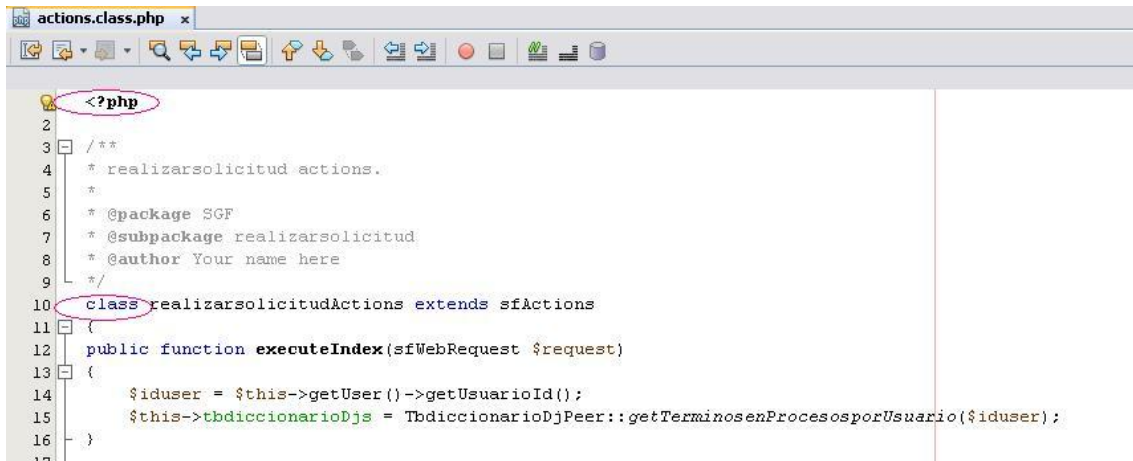
3.4. Implementación.

3.4.1. Estándares de Codificación.

Los estándares de códigos utilizados están recogidos en el PHP Development Best Practices de los autores (Naberezny, et al., 2006) en el que se recogen directrices sobre cómo realizar una correcta implementación. Algunos de estos estándares creados por la compañía desarrolladora de PHP Zend se explican a continuación.

La indentación debe ser a cuatro espacios sin caracteres de tabulación. Esto es debido a que ciertos Entorno de Desarrollo Integrado IDE (sigla en inglés de integrated development environment) de desarrollo introducen caracteres de tabulación cuando indentan un texto automáticamente.

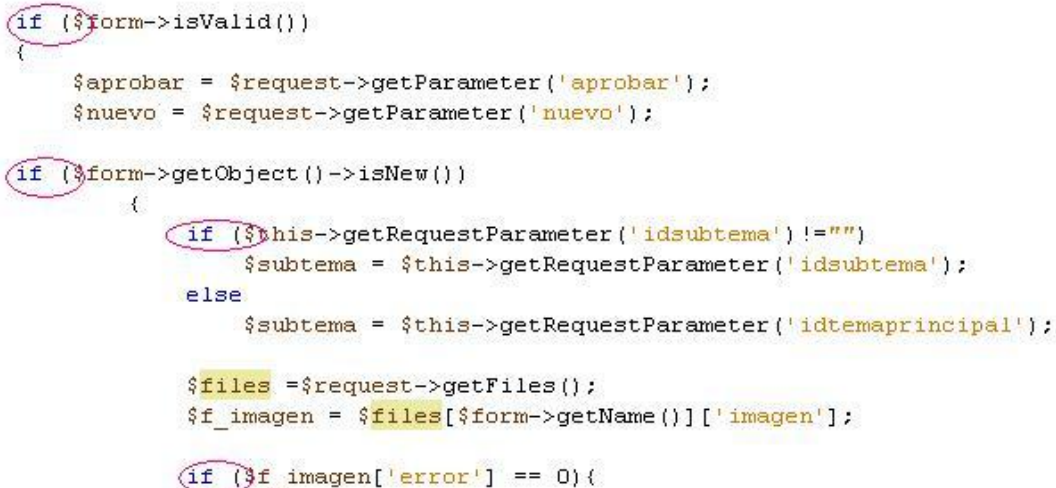
Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.



```
actions.class.php x
<?php
/**
 * realizarsolicitud actions.
 *
 * @package SGF
 * @subpackage realizarsolicitud
 * @author Your name here
 */
class realizarsolicitudActions extends sfActions
{
    public function executeIndex(sfWebRequest $request)
    {
        $iduser = $this->getUser()->getUsuarioId();
        $this->tbdiccionarioDjs = TbdiccionarioDjPeer::getTerminosenProcesosporUsuario($iduser);
    }
}
```

Figura 13. Indentación del código fuente.

Las estructuras de control deben tener un espacio entre la palabra reservada de la estructura y el signo de apertura de paréntesis para distinguir entre las llamadas de las funciones y el signo de llaves debe estar sobre la línea de la estructura.



```
if ($form->isValid())
{
    $aprobar = $request->getParameter('aprobar');
    $nuevo = $request->getParameter('nuevo');

    if ($form->getObject()->isNew())
    {
        if ($this->getRequestParameter('idsubtema') != "")
            $subtema = $this->getRequestParameter('idsubtema');
        else
            $subtema = $this->getRequestParameter('idtemaprincipal');

        $files = $request->getFiles();
        $f_imagen = $files[$form->getName()]['imagen'];

        if ($f_imagen['error'] == 0){

```

Figura 14. Como escribir las estructuras de control.

Las funciones deben ser llamadas sin espacios entre el nombre de la función, el signo de paréntesis y el primer parámetro; espacios entre cada coma por parámetro y sin espacios entre el último paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).



```
$this->getUser()->setFlash('notice', 'Sus cambios han sido guardados');
```

Figura 15. Llamada a función.

El estilo de los comentarios debe ser como el estilo de comentarios para C (`/* */` ó `//`), no debe de utilizarse el estilo de comentarios de Perl (`#`).

```
7  /**
8   * gestionarDiccionario actions.
9   *
10  * @package   SGF
11  * @subpackage gestionarDiccionario
12  * @author    Dannier
13  * @version   SVN: $Id: actions.class.php 3335 2007-01-23 16:19:56Z fabien $
14  */
```

Figura 16. Comentarios.

Cuando se incluya un archivo de dependencia incondicionalmente se utiliza `require_once` y cuando sea condicionalmente, se utiliza `include_once`.

```
require 'lib/model/om/BaseTbdiccionarioDj.php';
```

Figura 17. Inclusión de archivos dependientes.

Siempre se usan las etiquetas `<?php?>` para abrir un bloque de código. No se utiliza el método de etiquetas cortas, porque esto depende de las directivas de configuración en el archivo `PHP.INI` y hace que el script no sea tan portable.

```
<?php foreach ($sinonimos as $sinonimo):{ ?>
```

Figura 18. Uso de las etiquetas de PHP.

Para la nomenclatura de las clases, variables y métodos se utiliza Lower Camel Case.

```
class realizarSolicitudActions extends sfActions
{
```

Figura 19. Nombre de la clase.

Los archivos con código PHP, deben de ser guardados en formato ASCII utilizando la codificación ISO-8859-1. El formato ASCII con codificación ISO-8859-1, es el formato en que se guardan los archivos de texto plano (.txt). La razón de este estándar es que determinados editores HTML (en especial Dreamweaver), agregan códigos de carácter extraño de salto de línea (como si se tratara de un archivo binario) y esto puede ocasionar que el intérprete de PHP, encuentre problemas a la hora de leer el script.

3.4.2. Diagrama de Componentes.

Muestra la dependencia que existe entre los componentes, que incluyen archivos, bibliotecas compartidas y módulos.

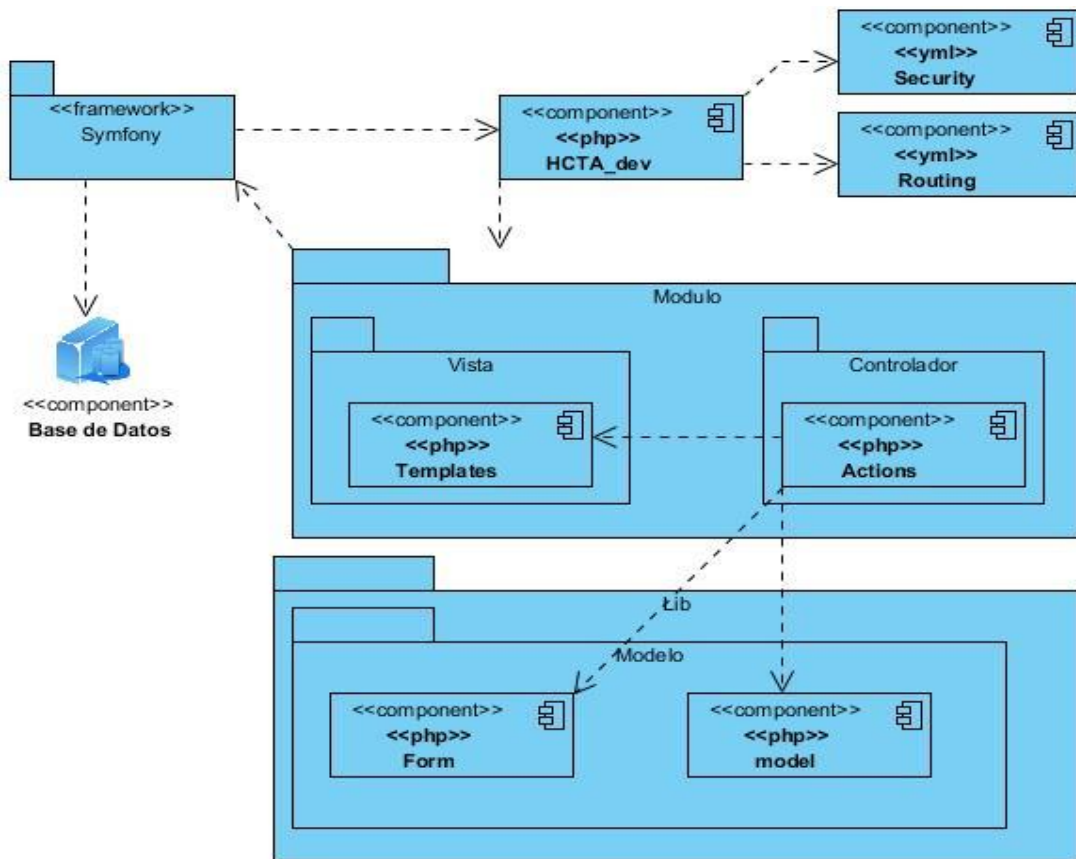


Figura 20. Diagrama de componente general de la aplicación.

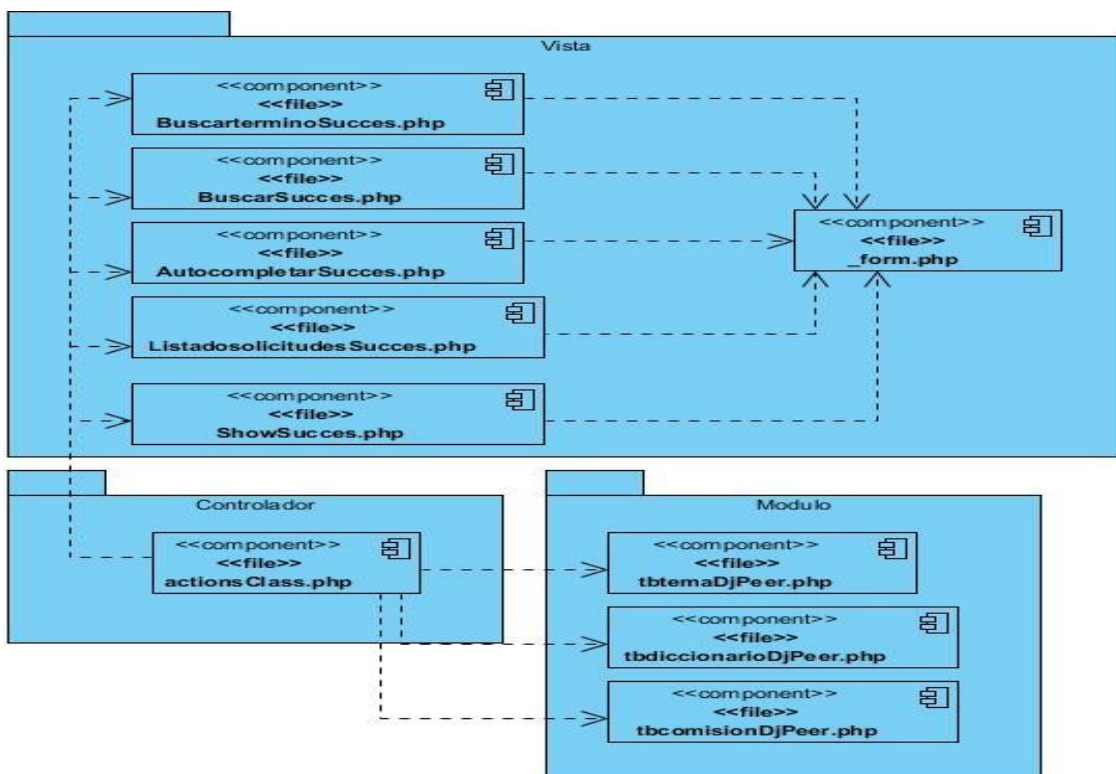


Figura 21. Diagrama de componente del caso de uso Buscar Término.

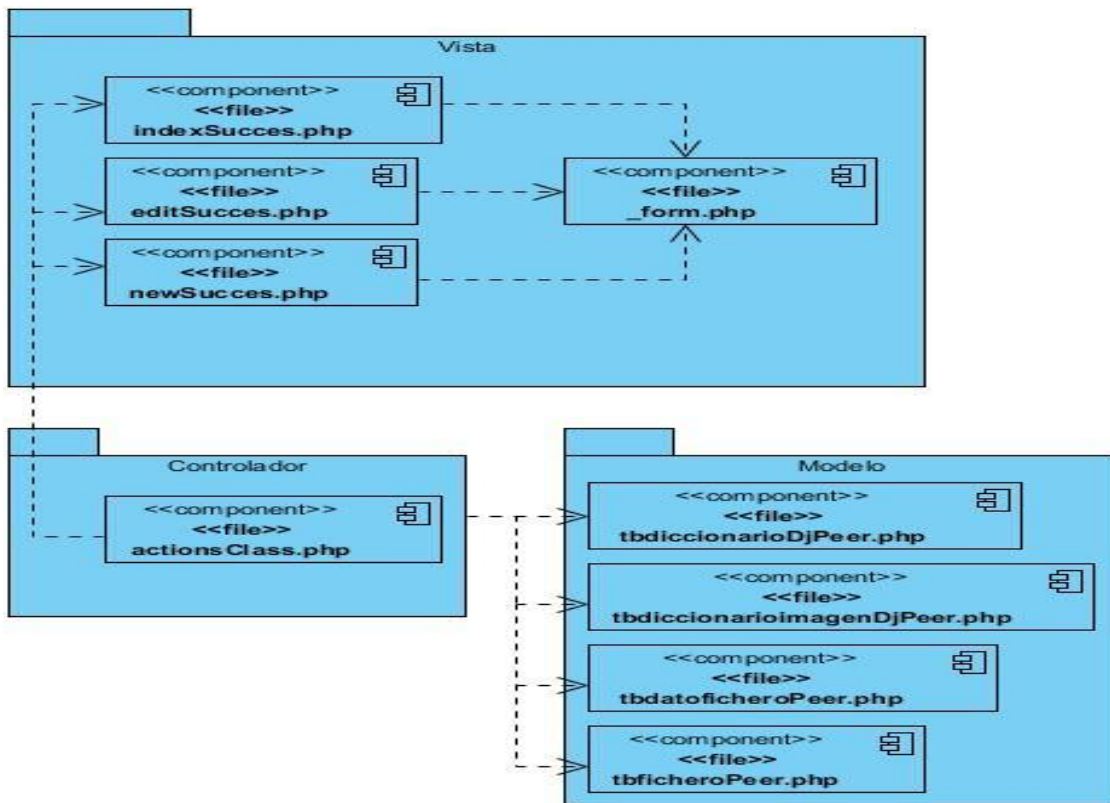


Figura 22. Diagrama de componente del caso de uso Gestionar Solicitud.

3.4.3. Diagrama de Despliegue.

Muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.

En la FGR es hacia donde se dirige todo el flujo de información proveniente de las fiscalías provinciales y municipales, esto se realiza utilizando herramientas de replicación de bases de datos, garantizando el acceso a la información de manera jerárquica. Por ello se utilizan dos servidores de bases de datos en forma de clúster para evitar sobrecargas, facilitar el procesamiento y almacenar altas cantidades de información.

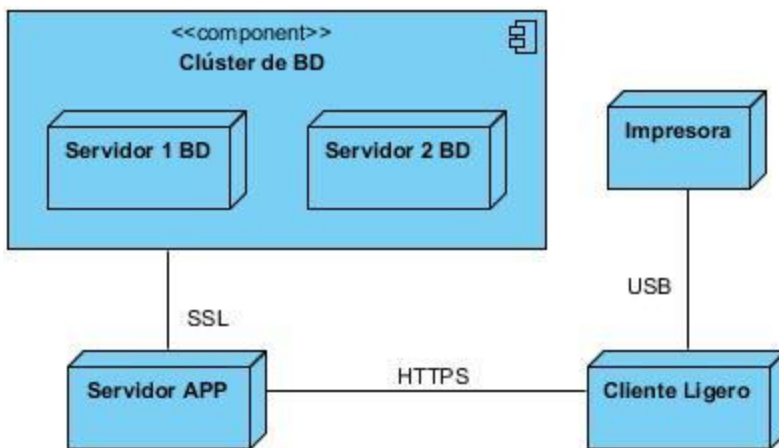


Figura 23. Diagrama de despliegue de la FGR.

En las fiscalías provinciales se utiliza un solo servidor de bases de datos pero se mantiene separado del servidor de aplicaciones porque la carga de transacciones de todos los municipios existentes en la provincia puede provocar problemas de rendimiento.

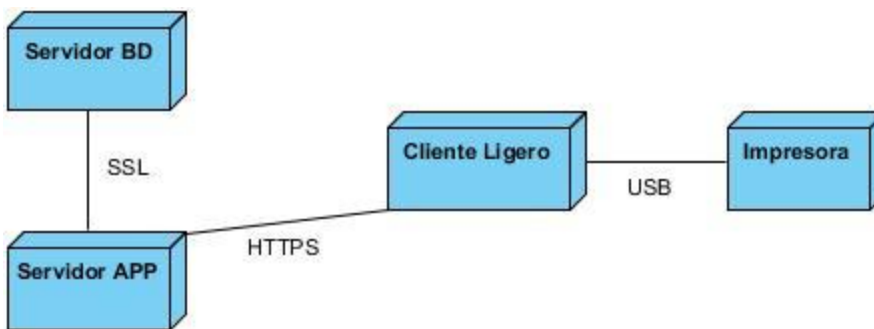


Figura 24. Diagrama de despliegue en la Fiscalía Provincial.

En las fiscalías municipales con el fin de ahorrar recursos y debido a que no se sobrecarga la aplicación por ser la instancia más específica se instalan las herramientas en un solo nodo formando un servidor de aplicación y de base de datos conjunto, capaz de funcionar de forma óptima.

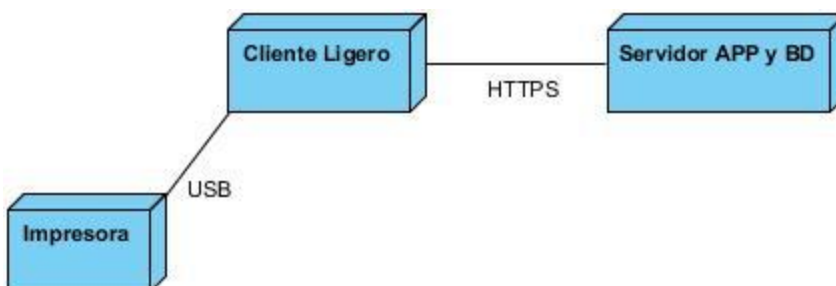


Figura 25. Diagrama de despliegue en la Fiscalía Municipal.

3.4.4. Seguridad.

La seguridad es un pilar importante en el desarrollo de la herramienta, ya que restringe el acceso a ejecutar las diferentes acciones a los usuarios, estos tienen determinados privilegios y necesitan autenticarse antes de comenzar a usar el sistema informático. Antes de ser ejecutada, cada acción pasa por un filtro especial que verifica si el usuario actual tiene privilegios de acceder a la acción requerida. En Symfony, los privilegios están compuestos por dos partes:

- Las acciones seguras requieren que los usuarios estén autenticados.
- Las credenciales son privilegios de seguridad agrupados bajo un nombre y que permiten organizar la seguridad en grupos.
- Para restringir el acceso a una acción se crea y se edita un archivo de configuración YAML llamado, `security.yml` en el directorio `config/` del módulo. En este archivo, se pueden especificar los requerimientos de seguridad que los usuarios deberán satisfacer para cada acción o para todas las acciones. (Potencier, et al., 2008)

3.4.4.1. Protección frente a ataques XSS y CSRF.

Se mostrará cómo la tarea `generate:app` incluye automáticamente varios mecanismos de seguridad necesarias para proteger a la aplicación frente a las dos vulnerabilidades más extendidas en la web. La primera medida de seguridad que se activa es la protección frente a los ataques de tipo XSS (Cross Site Scripting). Esto significa que por defecto las plantillas aplican el mecanismo de escape a los valores de todas las variables. Si tratas por ejemplo de incluir código HTML en la descripción de una oferta de trabajo, verás que cuando Symfony muestra los detalles de la oferta, las etiquetas se ven tal y como están escritas y no se interpretan como etiquetas HTML. La segunda es la protección contra los ataques de tipo CSRF (Cross Site Request Forgery), que la medida de seguridad es todos los formularios incluyen un campo oculto llamado `_csrf_token`. (Potencier, 2009)

En este capítulo se realizó una breve descripción de los patrones utilizado en el desarrollo de la aplicación. Además se reflejaron los artefactos pertenecientes a los flujos de diseño e implementación, tales como: modelo de diseño, modelo de componentes y el modelo de despliegue, así como los estándares de codificación utilizados en la implementación de la herramienta.

Capítulo 4. Validación de la propuesta de solución.

El siguiente capítulo aborda la validación de la herramienta, mostrando el resultado de las pruebas de caja negra para probar las funcionalidades del software, así como sus pruebas unitarias.

4.1. Pruebas Caja Negra.

4.1.1. Buscar Término.

Escenario	Descripción	Buscar Término	Tema	Subtema	Términos Recientes	Términos en procesamiento	Términos con Imágenes	Respuesta del sistema	Flujo central
EC 1.1 Buscar Término	Buscar un término correctamente (conociendo el término)	V/Agi tación	NA	NA	NA	NA	NA	El sistema muestra una tabla con el nombre del término y la fecha de creación. (Dando clic sobre el término se pueden ver los datos del mismo).	Módulo Diccionario Jurídico, opción Buscar Término, botón Buscar, clic sobre el término.
EC 1.2 Buscar Término	Buscar término correctamente (sin conocer el término)	NA	V/Derecho de Acción	NA	NA	NA	NA	El sistema muestra una tabla con el nombre de los términos que aparecen según el criterio seleccionado y la fecha de creación. (Dando clic sobre el término se pueden ver los	Módulo Diccionario Jurídico, opción Buscar Término, botón Buscar, clic sobre el término.

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

								datos del mismo).	
EC 1.3 Buscar Término	Buscar término dejando los campos vacíos	I/vacía	I/vacía	I/vacía	I/vacía	I/vacía	I/vacía	El sistema muestra una tabla con todos los términos existentes en la Base de Datos.	Módulo Diccionario Jurídico, opción Buscar Término, botón Buscar.
EC 1.4 Buscar Término	Buscar término sin resultados	V/Pri sione ro	NA	NA	NA	NA	NA	El sistema muestra el siguiente mensaje: "No se encontró ningún término que responda a los criterios seleccionados"	Módulo Diccionario Jurídico, opción Buscar Término, botón Buscar.
EC 1.5 Buscar Término	Se decide no realizar la búsqueda.	NA	NA	NA	NA	NA	NA	El sistema cierra la interfaz actual y muestra la interfaz principal.	Módulo Diccionario Jurídico opción Buscar Términos, clic en la opción Cerrar.

Tabla 7. Caso de prueba del caso de uso Buscar Término.

4.1.2. Descripciones de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Buscar Término	Campo de texto	Sí	No se pueden entrar números ni otros caracteres que no sean letras (a-z).

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

2	Tema	Lista desplegable	Sí	La introducción de este dato no admite error ninguno.
3	Subtema	Lista desplegable	Sí	La introducción de este dato no admite error ninguno.
4	Términos Recientes	Campo de selección	Sí	La introducción de este dato no admite error ninguno.
5	Términos en Procesamiento	Campo de selección	Sí	La introducción de este dato no admite error ninguno.
6	Términos con Imágenes	Campo de selección	Sí	La introducción de este dato no admite error ninguno.

Tabla 8. Descripciones de las variables del caso de uso Buscar Término.

4.1.3. Registrar Solicitud.

Escenario	Descripción	Término	Concepto	Comentario Doctrinal	Temas	Subtemas	Sinónimos	Término Relacionado	Respuesta del sistema	Flujo central
EC 1.1 Nueva Solicitud	Realizar una solicitud correctamente	V/Abuso de menores	V/Todo perjuicio o mal, con impunidad inicial y prolongada por lo común, de que se hace víctima a la juventud o a la infancia, aprovechándose de su inexperiencia o explotando sus pasiones.	NA	NA	NA	NA	NA	El sistema guarda los datos introducidos en la Base de Datos.	Módulo Diccionario Jurídico, opción Propuesta de Nuevo Término, botón Nuevo.

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

EC 1.2 Nueva Solicitud	Realizar una solicitud existente	V/Abuso de menores	V/Todo perjuicio o mal, con impunidad inicial y prolongada por lo común, de que se hace víctima a la juventud o a la infancia, aprovechándose de su inexperiencia o explotando sus pasiones.	NA	NA	NA	NA	NA	El sistema muestra a el siguiente mensaje de error: "El término que usted ha introducido ya existe. Por favor introduzca un término nuevo."	Módulo Diccionario Jurídico, opción Propuesta de Nuevo Término, botón Nuevo.
EC 1.3 Nueva Solicitud	Realizar solicitud dejando campos vacíos	V/Absoluto	I/Vacío	NA	NA	NA	NA	NA	El sistema muestra a el siguiente mensaje de error: El campo "Concepto" es requerido	Módulo Diccionario Jurídico, opción Propuesta de Nuevo Término, botón Nuevo.

Tabla 9. Caso de prueba de la Sección 1: Registrar Solicitud del caso de uso Gestionar Solicitud.

4.1.4. Modificar Solicitud.

Escenario	Descripción	Término	Concepto	Comentario Doctrinal	Temas	Subtemas	Sinónimos	Término o Relacionado	Respuesta del sistema	Flujo central
EC 1.1 Actualizando Solicitud	Actualizar una solicitud	V/Abuso de menores	V/Todo perjuicio o mal, con impunidad inicial y prolongada por lo común,	NA	V/Menores	NA	NA	NA	El sistema guarda en la Base de Datos, los datos que han sido	Módulo Diccionario Jurídico, opción Propuesta de Nuevo Término

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

			de que se hace víctima a la juventud o a la infancia, aprovechándose de su inexperiencia o explotando sus pasiones.						modifica dos.	o, clic sobre un término de la tabla.
--	--	--	---	--	--	--	--	--	---------------	---------------------------------------

Tabla 10. Caso de prueba de la Sección 2: Modificar Solicitud del caso de uso Gestionar Solicitud.

4.1.5. Eliminar Solicitud.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Actualizando Solicitud	Una vez realizada una solicitud, si ésta no ha sido procesada aún, el solicitante puede proceder a eliminarla.	El sistema elimina la solicitud de la Base de Datos.	Módulo Diccionario Jurídico, opción Propuesta de Nuevo Término, clic sobre un término de la tabla, botón Eliminar.

Tabla 11. Caso de prueba de la Sección 2: Eliminar Solicitud del caso de uso Gestionar Solicitud.

4.1.6. Descripciones de las variables.

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Término	Campo de texto	No	No se pueden entrar números ni otros caracteres que no sean letras (a-z).
2	Concepto	Campo de texto	No	La introducción de este dato no admite error ninguno.
3	Comentario Doctrinal	Campo de texto	Sí	La introducción de este dato no admite error ninguno.
4	Temas	Lista desplegable	Sí	La introducción de este dato no admite error ninguno.

5	Subtemas	Lista desplegable	Sí	La introducción de este dato no admite error ninguno.
7	Sinónimos	Lista desplegable	Sí	La introducción de este dato no admite error ninguno.
8	Término relacionado	Lista desplegable	Sí	La introducción de este dato no admite error ninguno.

Tabla 12. Descripciones de las variables del caso de uso Gestionar Solicitud.

Una vez realizados los casos de prueba se encontraron cinco errores ortográficos y diez errores de validación los cuales fueron corregidos en su totalidad, al entrar a una segunda iteración fue liberada la aplicación por Calisoft el 23 de junio del 2011, el acta de liberación se podrá encontrar en los artefactos generados.

4.2. Pruebas Unitarias.


Para realizar las pruebas unitarias se utilizó la librería LimeTest que proporciona Symfony, fue necesario crear las clases en la carpeta test\unit que se encuentra en el directorio del framework utilizado, donde el nombre de la clase terminará con el sufijo Test y dentro de ella se programará el código para las pruebas. Las funciones utilizadas para demostrar la correcta estructura del código fueron *is* y *isnt*. Las pruebas se realizaron a las clases del modelo.

```

1  <?php
2  /*
3   * To change this template, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  include(dirname(__FILE__) . '/../bootstrap/Propel.php');
7  $t = new lime_test(3, new lime_output_color());
8
9  $posible_respuesta= TbcomisionDjPeer::getMiembrosComision();
10
11 $t->is($posible_respuesta[1]['nombre'], 'Jose', 'Comprobando que el nombre sea igual.');
```

Figura 26. Prueba unitaria al método getMiembrosComision().

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.



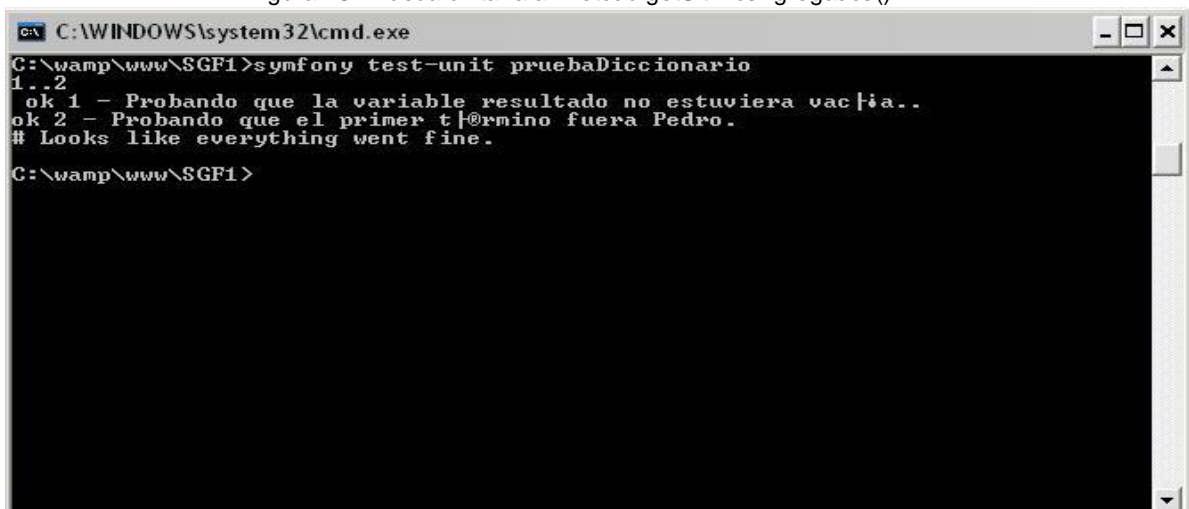
```
C:\WINDOWS\system32\cmd.exe
C:\wamp\www\SGF1>symfony test-unit miembroComision
1..3
ok 1 - Comprobando que el nombre sea igual.
ok 2 - Comprobando que no es jefe.
ok 3 - Comprobando que el Id de comisi|n son iguales.
# Looks like everything went fine.
C:\wamp\www\SGF1>
```

Figura 27. Respuesta del método getMiembrosComision().



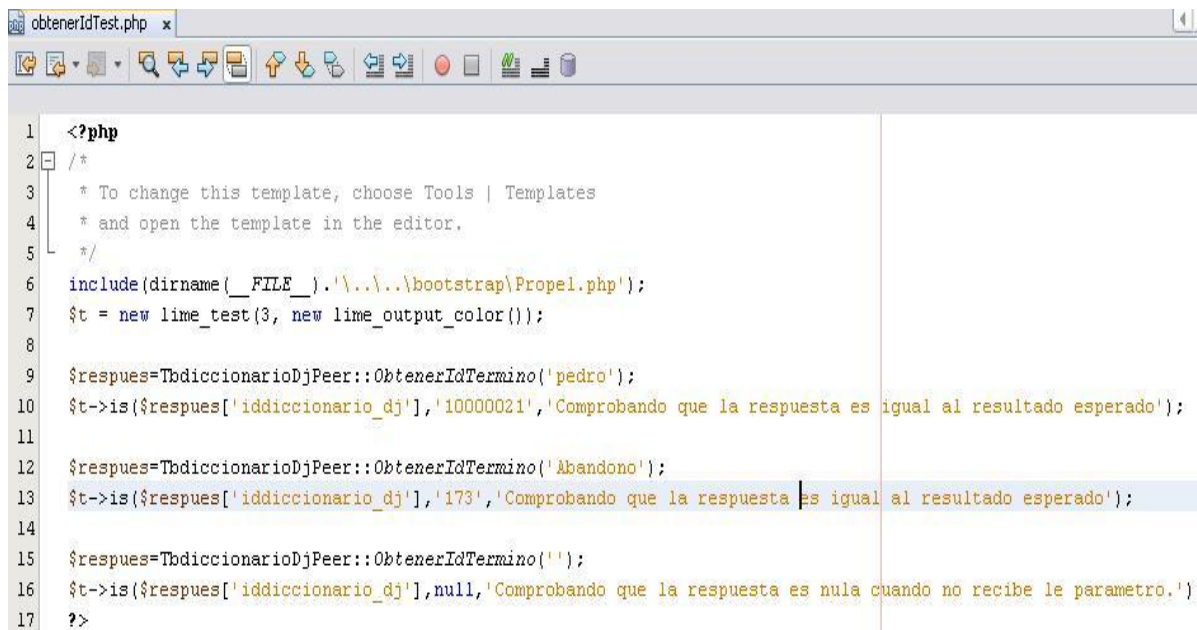
```
pruebaDiccionarioTest.php
1 <?php
2 /*
3  * To change this template, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 include(dirname(__FILE__).'\..\..\bootstrap\Propel.php');
7
8 $t = new lime_test(2, new lime_output_color());
9
10 $resultado=TbdiccionarioDjPeer::getUltimosAgregados();
11
12 $t->isnt($resultado, NULL, 'Probando que la variable resultado no estuviera vacía..');
13
14 $t->is($resultado[0]['termino'], 'pedro', 'Probando que el primer término fuera Pedro.');
```

Figura 28. Prueba unitaria al método getUltimosAgregados().



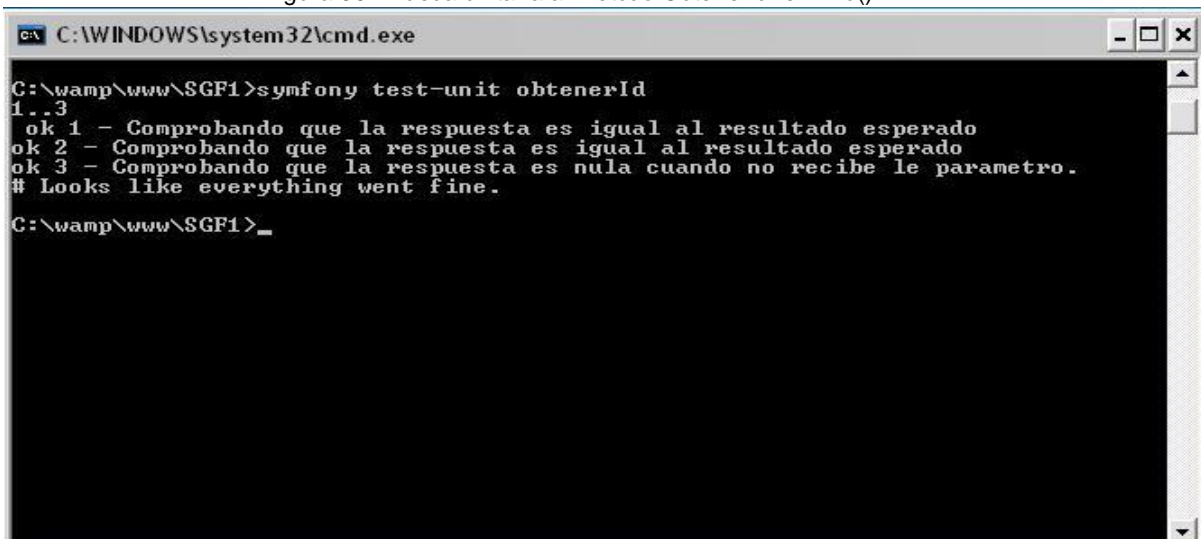
```
C:\WINDOWS\system32\cmd.exe
C:\wamp\www\SGF1>symfony test-unit pruebaDiccionario
1..2
ok 1 - Probando que la variable resultado no estuviera vacía..
ok 2 - Probando que el primer término fuera Pedro.
# Looks like everything went fine.
C:\wamp\www\SGF1>
```

Figura 29. Respuesta del método getUltimosAgregados().



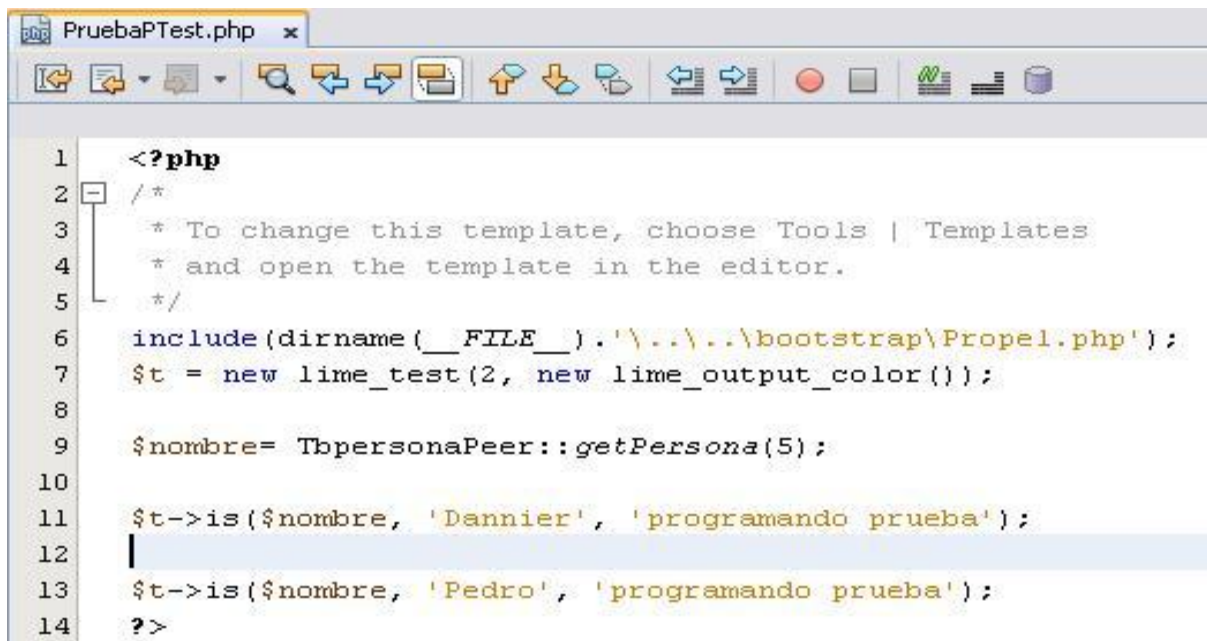
```
1 <?php
2 /*
3  * To change this template, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 include(dirname(__FILE__).'..\..\bootstrap\Propel.php');
7 $t = new lime_test(3, new lime_output_color());
8
9 $respues=TbdiccionarioDjPeer::ObtenerIdTermino('pedro');
10 $t->is($respues['iddiccionario_dj'],'10000021','Comprobando que la respuesta es igual al resultado esperado');
11
12 $respues=TbdiccionarioDjPeer::ObtenerIdTermino('Abandono');
13 $t->is($respues['iddiccionario_dj'],'173','Comprobando que la respuesta es igual al resultado esperado');
14
15 $respues=TbdiccionarioDjPeer::ObtenerIdTermino('');
16 $t->is($respues['iddiccionario_dj'],null,'Comprobando que la respuesta es nula cuando no recibe le parametro. ');
17 ?>
```

Figura 30. Prueba unitaria al método ObtenerIdTermino().



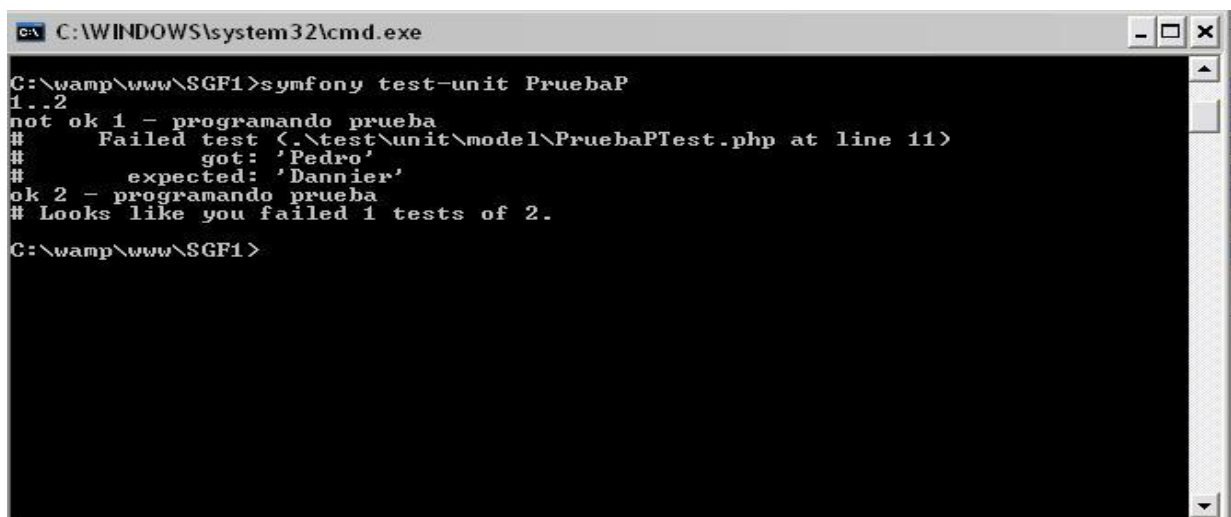
```
C:\WINDOWS\system32\cmd.exe
C:\wamp\www\SGF1>symfony test-unit obtenerId
1..3
ok 1 - Comprobando que la respuesta es igual al resultado esperado
ok 2 - Comprobando que la respuesta es igual al resultado esperado
ok 3 - Comprobando que la respuesta es nula cuando no recibe le parametro.
# Looks like everything went fine.
C:\wamp\www\SGF1>_
```

Figura 31. Respuesta del método ObtenerIdTermino().



```
1 <?php
2 /*
3  * To change this template, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 include(dirname(__FILE__).'..\..\bootstrap\Propel.php');
7 $t = new lime_test(2, new lime_output_color());
8
9 $nombre= TpersonaPeer::getPersona(5);
10
11 $t->is($nombre, 'Dannier', 'programando prueba');
12
13 $t->is($nombre, 'Pedro', 'programando prueba');
14 ?>
```

Figura 32. Prueba unitaria al método getPersona(\$id).



```
C:\WINDOWS\system32\cmd.exe
C:\wamp\www\SGF1>symfony test-unit PruebaP
1..2
not ok 1 - programando prueba
# Failed test C:\test\unit\model\PruebaPTest.php at line 11>
# got: 'Pedro'
# expected: 'Dannier'
ok 2 - programando prueba
# Looks like you failed 1 tests of 2.
C:\wamp\www\SGF1>
```

Figura 33. Respuesta del método getPersona(\$id).

En este capítulo se hicieron las pruebas de caja negra y las pruebas unitarias para validar la solución, mostrando los distintos resultados arrojados en cada una de ellas.

Conclusiones Generales.

Con el desarrollo del presente trabajo se concluye afirmando que:

- ✓ El estudio de las tecnologías y metodologías definidas por el equipo de arquitectura del proyecto permitió apropiarse de los conocimientos necesarios para el desarrollo de la solución. Siendo RUP la metodología de desarrollo, PHP5 como lenguaje de programación, Symfony 1.3 como framework de desarrollo, Netbeans 6.9 como IDE de desarrollo, Apache 2 como servidor web, PostgreSQL como sistema de gestor de base de datos, Visual Paradigm como herramienta CASE.
- ✓ La definición de los requisitos funcionales y no funcionales sirvió de punto de partida para la elaboración de los artefactos del análisis y poder capturar cuales eran las necesidades del cliente que debía cubrir la solución.
- ✓ Desarrollar haciendo uso de la metodología RUP permitió guiar el proceso de forma organizada obteniendo los diferentes artefactos tales como modelo de diseño, modelo de componentes y el modelo de despliegue propuesto para cada una de las fases.
- ✓ La validación del sistema mediante la aplicación de las pruebas unitarias y de caja negra arrojó resultados satisfactorios, demostrándose que la solución desarrollada responde a las necesidades del cliente, quedando validada con el acta de liberación del producto por parte de Calisoft.

Bibliografía.

Abrahamsson, Pekkan, et al. 2002. *Agile software development methods*. s.l. : Vtt Publications, 2002. ISSN: 12350621, ISBN: 9513860094.

ESPAÑOLA, REAL ACADEMIA. REAL ACADEMIA ESPAÑOLA. REAL ACADEMIA ESPAÑOLA.

[Online] [Cited: 06 22, 2012.]

http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=diccionario.

Fiscalía General de la República de Cuba. 2006. Fiscalía General de la República de Cuba.

Fiscalía General de la República de Cuba. [Online] 2006. [Cited: 11 16, 2011.]

<http://www.cuba.cu/gobierno/cuba.htm>.

Flannagan, David. 1997. *JavaScript: The Definitive Guide*. s.l. : OReilly, 1997. ISBN: 1-56592-235-2.

Granado, Luis Miguel Cabezas. 2004. *PHP 5*. s.l. : ANAYA MULTIMEDIA, 2004. ISBN:

9788441517851 978-84-415-1785-1 8441517851 84-415-1785-1.

Grupo Soluciones GSINNOVA. 2012. Grupo Soluciones GSINNOVA. *Grupo Soluciones*

GSINNOVA. [Online] 2012. [Cited: 2 10, 2012.]

<http://www.rational.com.ar/herramientas/rup.html>.

H, José Canos, Leteller, Patricio and Penadés, M Carmen. *Métodologías Ágiles en el Desarrollo de Software*. Valencia : s.n.

Jacobson, Ivar, Booch, Grady and Rumbauch, James. 2000. *El proceso unificado de desarrollo de software*. Madrid : s.n., 2000. ISBN 84-7829-036-2.

Kew, Nick. 2008. *Desarrollo de módulos y aplicaciones con Apache*. España : ANAYA

MULTIMEDIA, 2008. 9788441523289 978-84-415-2328-9 8441523282 84-415-2328-2.

Naberezny, Mike and O'Phinney, Matthew Weier. 2006. *Best Practices of PHP Development*.

s.l. : Zend Technologies, 2006.

NEGRINO, TOM and SMITH, DORI. 2000. *Guía de aprendizaje JavaScript 3ª edición*. Madrid :

Pearson Educación, 2000. 84-205-3091-3.

Oracle. Java. *Java*. [Online] [Cited: 1 15, 2012.]

http://www.java.com/es/download/faq/java_javascript.xml.

—. **2012.** Netbeans. *Netbeans*. [Online] 2012. [Cited: 2 5, 2012.]

http://netbeans.org/community/releases/69/index_es.html.

Poppendieck, Mary and Poppendieck, Tom. 2003. *Lean software development: an agile*

toolkit. [ed.] 2003 Addison-Wesley Professional. 2003. ISBN 0321150783, 9780321150783.

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

Potencier, Fabien and Zaninotto, François. 2008. *La guía definitiva de Symfony*. Francia : s.n., 2008.

—. *The Definitive Guide to Symfony*. s.l. : Apress . 978-1590597866.

Potencier, Fabien. 2009. *El tutorial Jobeet*. Francia : s.n., 2009.

—. **2007.** libroweb.es. *libroweb.es*. [Online] 2007. [Cited: 2 10, 2012.] <http://www.librosweb.es/>.

Pressman, Roger. 2002. *Ingeniería de Software, un enfoque práctico*. Quinta edición. s.l. : McGraw-Hill Companies, 2002. ISBN: 8448132149.

—. **2005.** *Ingeniería del Software: Un Enfoque Práctico*. s.l. : McGraw-Hill, 2005. ISBN: 9701054733.

Real Academia Española. Real Academia Española. *Real Academia Española*. [Online] [Cited: 10 14, 2011.] <http://buscon.rae.es/drae/>.

Real Academia. 2012. Real Academia. *Real Academia*. [Online] 2012. [Cited: 5 25, 2012.] <http://buscon.rae.es/drae/>.

Ruggerberg, Carlos Subirats. 1987. Sociedad Española para el Procesamiento de Lenguaje Natural. *Sociedad Española para el Procesamiento de Lenguaje Natural*. [Online] 1987. [Cited: 11 16, 2011.] <http://www.sepln.org/revistaSEPLN/revista/5/5-Pag63.pdf>.

Sklar, David. 2005. *Introducción a PHP 5*. s.l. : ANAYA MULTIMEDIA, 2005. ISBN: 9788441518032 978-84-415-1803-2 8441518033 84-415-1803-3.

2012. Symfony. *Symfony*. [Online] 2012. [Cited: 2 10, 2012.] <http://www.symfony.es/documentacion/las-novedades-de-symfony-1-3-y-1-4/>.

The Apache Software Foundation. 2012. The Apache Software Foundation. *The Apache Software Foundation*. [Online] 2012. [Cited: 2 15, 2012.] <http://www.apache.org/>.

The PHP Group. 2012. PHP. *PHP*. [Online] 2012. [Cited: 05 22, 2012.] <http://www.php.net>.

The PostgreSQL Global Development Group . 2012. PostgreSQL. *PostgreSQL*. [Online] 2012. [Cited: 1 16, 2012.] <http://www.postgresql.org>.


w3c. 2012. W3C. *W3C*. [Online] 2012. [Cited: 05 22, 2012.] <http://www.w3.org/Style/CSS/>.

W3C. 2012. W3C. *W3C*. [Online] 2012. [Cited: 05 22, 2012.] <http://www.w3c.com>.


WordReference. 2012. WordReference. *WordReference*. [Online] 2012. [Cited: 5 25, 2012.] <http://wordreference.com/english/AboutUs.aspx>.

Anexos.

Anexo 1. Acta de liberación del Software.



Validez desconocida
Digitally signed by Tayché Capote García
Date: 2011.07.13 10:30:43 CDT
Reason: Document oficial
Location: Cuba



Acta de Liberación de Productos Software

Fecha de liberación: 23 de junio de 2011

Emitida a favor de: Proyecto Sistema de Gestión Fiscal.

1. Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas
Módulo Inspección	1.3	0	3	Pruebas Funcionales
Módulo Sumario	1.3	0	3	Pruebas Funcionales
Módulo Ordinario	1.3	0	3	Pruebas Funcionales
Módulo Procesos Confiscatorios	1.3	0	3	Pruebas Funcionales
Módulo Gestión Documental	1.3	0	3	Pruebas Funcionales
Módulo Diccionario Jurídico	1.3	0	3	Pruebas Funcionales
Sistema Gestión Fiscal Integrado	1.3	0	3	Pruebas de Funcionales de Integración

Yadra Machado Peña

Nombre y Apellidos
Responsable Callsoft

Dallin Benavides Jorge

Nombre y Apellidos
Proyecto Sistema de Gestión Fiscal.

Figura 34. Acta de Liberación SGF app-firmado.

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

Anexo 2. Prototipo de interfaz de la página principal de la aplicación.

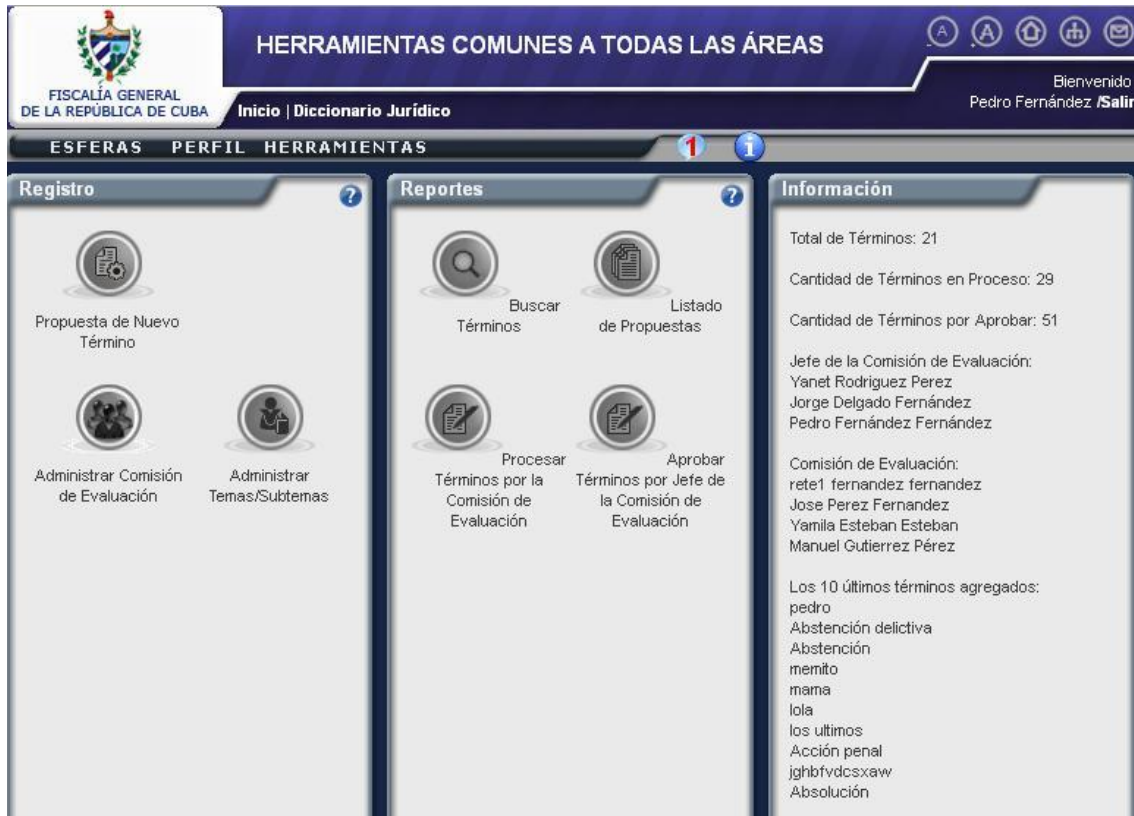


Figura 35. Prototipo de interfaz página principal de la aplicación.

Anexo 3. Prototipo de interfaz de la página buscar término.



Figura 36. Prototipo de interfaz de la página buscar término.

Módulo Diccionario Jurídico para el Sistema de Informatización de la Gestión de la Fiscalía.

Anexo 4. Prototipo de interfaz de la página realizar solicitud.

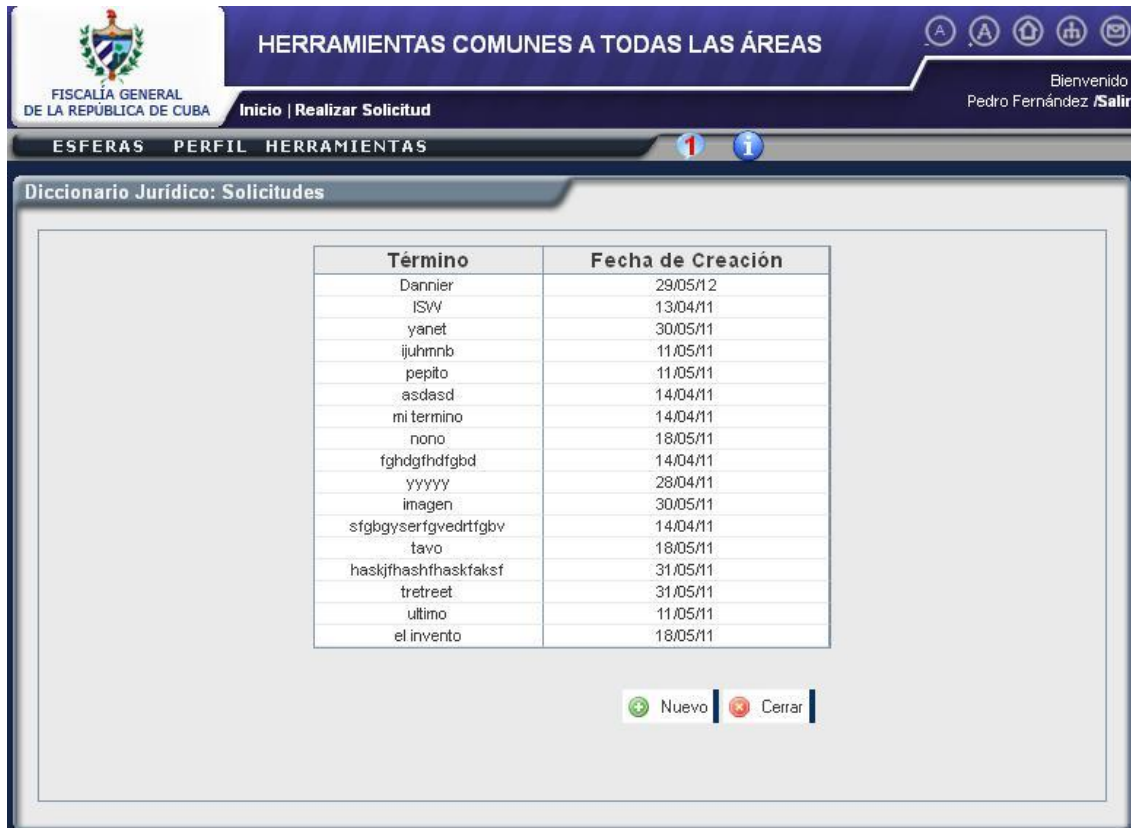


Figura 37. Prototipo de interfaz de la página realizar solicitud.