

Universidad de las Ciencias Informáticas

Facultad 3



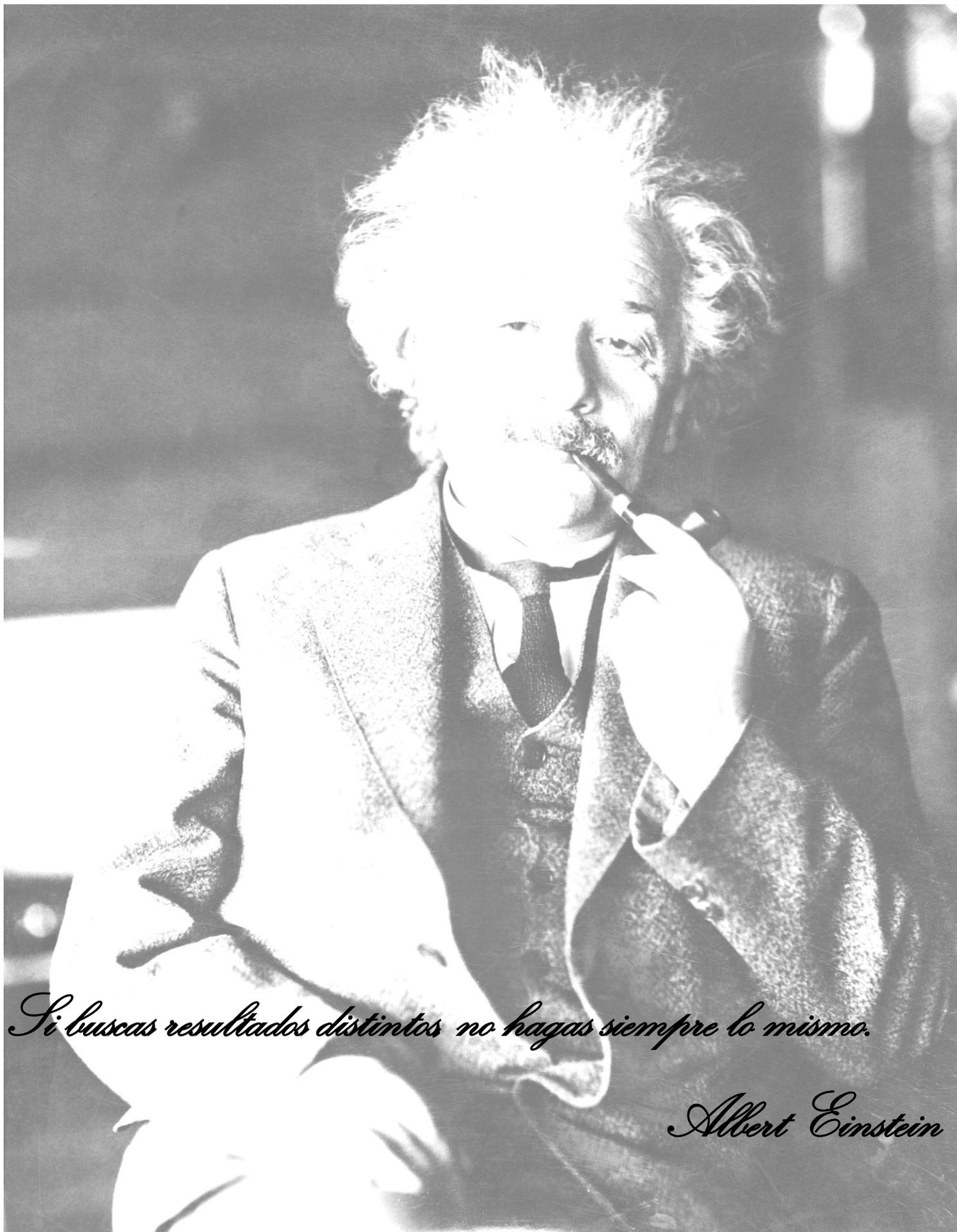
Título: Implementación del Sistema Integral de Documentación e Información Judicial para el Centro Nacional de Documentación e Información Judicial

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

Autores: Daysi María Granado García
Ana Ivette Ferrer Hernández

Tutor: Yanet Pérez Valcarcel
Co-tutores: Heiler Fabars Corrales
Yuniel Rodríguez Bello

Ciudad de la Habana, junio de 2011



Si buscas resultados distintos no hagas siempre lo mismo.

Albert Einstein

Declaración de Autoría

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

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

Autores

Daysi María Granado García

Ana Ivette Ferrer Hernández

Tutor

Yanet Pérez Valcarcel

Co-tutores

Heiler Fabars Corrales

Yuniel Rodríguez Bello

Datos de Contacto

Autor: Daysi María Granado García

Correo electrónico: dmgranado@uci.cu

Autor: Ana Ivette Ferrer Hernández

Correo electrónico: aiferrer@uci.cu

Tutor: Ing. Yanet Pérez Valcárcel

Correo electrónico: yperezva@uci.cu

Cotutor: Ing. Heiler Fabars Corrales

Correo electrónico: hfabars@uci.cu

Cotutor: Ing. Yuniel Rodríguez Bello

Correo electrónico: yrbello@uci.cu

Agradecimientos de Daisy

Gracias a **Alys y Nelly**, más que hermanas, han sido como dos madres para mí, ustedes son todo lo que yo quiero ser en mi vida, sin sus consejos, su apoyo, sus regaños, el esfuerzo que han hecho para que esté aquí y la fe que han puesto en mí, nunca hubiera llegado a ser quien soy, las adoro, son lo máximo para mí.

Gracias a mis padres, **Daisy y Lázaro**, por enseñarme lo que quiero ser en mi vida y por formarme como soy.

Gracias a toda mi familia por preocuparse todo el tiempo.

Gracias a mi tatico, mi **Julio**, por estar a mi lado incondicionalmente en los buenos y malos momentos de esta universidad, como mi amigo y como mi pareja, supiste darme sostén y me enseñaste a ser fuerte, te quiero mucho mi amor, espero compartir juntos lo que nos depara la vida.

Gracias a **Cary, Mirta y Alvarito** por preocuparse por mí y hacerme sentir como parte de su familia.

Gracias a mis amistades de toda la vida, **Alianne, Félix y Sellés**, a pesar de estar lejos siempre los he sentido cerca.

Gracias a las amistades de la UCI que han sido mi familia estos 5 años, **Enma, Lary, Rey, Ruby y Linnet** por darme siempre ánimo y estar ahí cuando los he necesitado, no los voy a olvidar nunca, son lo mejor de la UCI.

Gracias a **Yuniel y a Alien** por ser tan pacientes con nosotras.

Gracias a las personas que me rodean y se preocupan por mí, **Cealys, Mariela, Nany, Alianis Mayté, Lexys, Kariné** y otras más que no alcanzarían ni las 80 páginas para nombrarlas.

Agradecimientos de Nany

Agradecimientos

Primeramente agradecer a mis amistades que se convirtieron en una familia que no olvidaré jamás, por estar conmigo y apoyarme en cada momento. Los quiero mencionar porque realmente merece la pena arriesgarme a que se me quede alguno con tal de que pueda dejar sus nombres plasmados en este documento.

A Lexys, Mayté, Alianis, Kariné y Daysi gracias por acogerme con tanto cariño cuando la vida me dió un golpe duro y sufría amargas decepciones, doy gracias por haberlas conocido y haber compartido con ustedes tantos momentos inolvidables. A mis niños, mis dos hermanos Raiko y Leo los quiero mucho muchísimo no saben cuánto, gracias por cubrirme, protegerme y comprenderme. A mi novio que sin él no sería ni la sombra de lo que soy ahora, gracias por su cariño y su paciencia.

A los nuevos que fueron entrado en mi vida poco a poco a Sandrita y a Dini, a Claudia, a Mary.

Gracias a mi familia, mis abuelitos, mi tía Elsa, mi hermano al cual quiero dar el ejemplo para que se sienta orgulloso y siga los buenos pasos, es una de las cosas que más deseo en la vida.

No puedo dejar de mencionar a mis otras amistades, quienes me ayudaron incondicionalmente, al Kiki, una persona excepcional, a Yubisel, a Allien.

A todas las chicas del team de futsal, a mi fanática Isabel, gracias a mi profe Ariel, el mejor.

¡A mis amistades de los primeros años... Yenma, Yuliet, Aliemny por su preocupación!

Gracias a mis tutores, Heyler, Yanet y Yuniel por habernos brindado su apoyo y habernos aguantado en todos los estados posibles de ebullición.

Finalmente gracias a todos los que de una forma u otra hicieron que este día fuera tan especial como había soñado.

Dedicatoria

Daysi

Este trabajo es lo mínimo que puedo ofrecer a mis hermanas Alys y Nelly a cambio de todo el esfuerzo que han hecho, pero es solo el comienzo, no éste trabajo, mi vida entera está dedicada a ustedes y a mis padres, se merecen mucho más, todo lo mejor del mundo, las quiero mucho,

Nany

Le dedico mi título, mis esfuerzos y mi vida a la persona más importante de este mundo, a quien le debo más de lo que pueda pagar jamás, por su entrega, su amor, su dulzura. Por ser madre y padre, por ser todo ella. Por hacer de mi vida un paseo tan extraordinario, lleno de cariño y de alegría aun cuando las cosas pintan de color oscuro, a mi luz, mi ejemplo, mi faro, mi estrella...

A la mamá más linda que existe, mi mamita.

Resumen

Actualmente, el acelerado incremento de la información junto a la acumulación desordenada de datos que existe en el mundo provoca que procesos esenciales como las búsquedas, el almacenamiento y la gestión se tornen complejos y engorrosos, lo cual requiere aplicaciones de software enfocadas a hacer búsquedas y recolección de la información, con las cuales las entidades puedan acceder de forma rápida y objetiva a los datos almacenados. Debido a lo anterior expuesto, toda empresa tiene como objetivo fundamental progresar a través del desarrollo de las Tecnologías de la Información y las Comunicaciones.

En Cuba, la Universidad de las Ciencias Informáticas y otras empresas productoras de software se ocupan de hacer realidad la informatización de la sociedad al trabajar arduamente en el desarrollo de sistemas informáticos para las empresas e instituciones nacionales.

Una de las instituciones que es necesario informatizar es el Centro Nacional de Documentación e Información Judicial, perteneciente al Tribunal Supremo Popular, donde se realiza la prestación de servicios de búsqueda de información judicial tanto a órganos judiciales como a la población en general.

La institución no cuenta con un sistema informático que desarrolle sus procesos fundamentales como son la prestación de servicios y los procesos de gestión. En el presente trabajo de diploma se realiza la implementación de un Sistema Integral de Documentación e Información Judicial para dicho centro, que permitirá almacenar la información manejada facilitando su seguimiento, búsqueda y control, potenciando así de manera sustancial los procesos fundamentales que allí se realizan.

Los estudios realizados permitieron una correcta implementación del sistema, validado a través de pruebas de software.

PALABRAS CLAVE: implementación, desarrollo, software, documentación, información jurídica

Índice General

AGRADECIMIENTOS.....	VI
DEDICATORIA	VIII
RESUMEN.....	IX
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
INTRODUCCIÓN.....	5
1.1 RESEÑA HISTÓRICA DE LOS SERVICIOS EN LAS BIBLIOTECAS.....	5
1.2 CONCEPTOS ASOCIADOS A LOS SISTEMAS DE DOCUMENTACIÓN E INFORMACIÓN JUDICIAL.....	6
INFORMÁTICA JURÍDICA	6
1.3 SOFTWARE EXISTENTE VINCULADO A SISTEMAS DE DOCUMENTACIÓN E INFORMACIÓN JUDICIAL	7
1.3.1 Sistemas Internacionales.....	7
1.3.2 Sistemas Nacionales	9
1.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE	9
1.4.1 Proceso Unificado de Desarrollo.....	10
1.4.2 Programación Extrema	11
1.5 LENGUAJE DE MODELADO.....	12
1.5.1 Lenguaje Unificado de Modelado.....	12
1.6 LENGUAJES DE PROGRAMACIÓN.....	13
1.6.1 Lenguajes del lado del servidor	13
1.6.2 Lenguajes del lado del cliente.....	15
1.7 HERRAMIENTAS CASE.....	17
1.7.1 Visual Paradigm para UML 6.4	17
1.7.2 Rational Rose Enterprise Edition	18
1.8 FRAMEWORKS DE DESARROLLO	19
1.8.1 Symfony 1.4.....	19
1.8.2 Zend Framework.....	20
1.8.3 Codelgniter	20
1.8.4 Framework de interfaz de usuario Ext JS 3.0	21
1.9 IDE	21
1.9.1 NetBeans	21
1.9.2 Eclipse	22
1.9.3 Zend Estudio.....	23
1.10 SERVIDOR WEB.....	23
1.10.1 Apache 2.2.6	24
1.10.2 Roxen.....	24
1.11 SISTEMA GESTOR DE BD.....	25
1.11.1 PostgreSQL 8.4.....	25
1.11.2 MySQL	26

CONCLUSIONES DEL CAPÍTULO	26
2 CAPÍTULO 2: DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA	27
INTRODUCCIÓN	27
2.1 VALORACIÓN DE LOS ARTEFACTOS PROPUESTOS POR LOS ANALISTAS	27
2.1.1 Requisitos Funcionales	27
2.1.2 Requisitos no Funcionales	28
2.1.3 Diagrama de Casos de Uso del Sistema	29
2.1.4 Diagrama de Componentes	30
2.1.5 Diagrama de Clases del Diseño.....	31
2.1.6 Modelo de Datos.....	31
2.2 ESTÁNDARES DE CODIFICACIÓN	33
2.3 ARQUITECTURA.....	35
2.3.1 Estrategias de Integración	35
2.3.2 Patrones GRASP implementados.....	36
2.3.3 Patrones GoF que utiliza Symfony.....	37
2.4 DESCRIPCIÓN DE LAS CLASES.....	38
2.4.1 Clases Controladoras	38
2.4.2 Clases del Modelo	39
2.4.3 Clases de la Interfaz	40
2.5 PROPUESTA DE SOLUCIÓN.....	40
2.5.1 Tratamiento de errores y validación en la entrada de datos	45
2.5.2 DIAGRAMA DE DESPLIEGUE.....	46
CONCLUSIONES DEL CAPÍTULO.....	47
3 CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	48
INTRODUCCIÓN.....	48
3.1 PRUEBAS	48
3.1.1 Pruebas de Caja Blanca	49
3.1.2 Pruebas de Caja Negra	55
CONCLUSIONES DEL CAPÍTULO.....	59
CONCLUSIONES GENERALES	60
RECOMENDACIONES.....	61
BIBLIOGRAFÍA.....	62
ANEXOS.....	¡ERROR! MARCADOR NO DEFINIDO.
GLOSARIO DE TÉRMINOS	65

Índice de Ilustraciones

Ilustración 1. Requisitos no funcionales del producto.....	29
Ilustración 2. Diagrama de Casos de Uso del Sistema.....	29
Ilustración 3. Diagrama de Componentes.....	30
Ilustración 4. Caso de Uso Gestionar Boleta de Sala de Lectura.....	31
Ilustración 5. Modelo de Datos del Sistema.....	32
Ilustración 6. Tabla Publicación Seriada y sus relaciones.....	33
Ilustración 7. Colaboración entre clases definidas por la arquitectura.....	36
Ilustración 8. Interfaz para el usuario general.....	41
Ilustración 9. Interfaz Gestionar Documento.....	42
Ilustración 10. Interfaz para gestionar descriptores.....	43
Ilustración 11. Interfaz para gestionar usuario.....	44
Ilustración 12. Interfaz para gestionar boletas.....	45
Ilustración 13. Diagrama de Despliegue.....	47
Ilustración 14. Prueba de Caja Blanca.....	49
Ilustración 15. Notación para las Instrucciones de secuencia, if y while.....	50
Ilustración 16. Algoritmo InsertarUsuario().....	51
Ilustración 17. Grafo de flujo asociado al algoritmo insertarUsuario().....	52
Ilustración 18. Resultado de la prueba realizada.....	54
Ilustración 19. Prueba de Caja Negra.....	55

Índice de Tablas

Tabla 1. Insertar Legislación.....	28
Tabla 2. Descripción de la Clase Controladora.....	39
Tabla 3. Descripción de las clases Entidad.....	40
Tabla 4. Descripción de la Clase Insertar Legislación.....	40
Tabla 5. Tabla de los caminos recorridos.....	53
Tabla 6. Secciones a probar en el CU_ Agregar Ficha de Entrada de Información.....	56
Tabla 7. Descripción de variables. SC 1. Agregar ficha.....	57
Tabla 8. Matriz de Datos SC 1. Agregar ficha.....	58
Tabla 9. Registro de defectos y dificultades detectados.....	59

Introducción

En la actualidad es inmenso el volumen de información y de datos que rodea al hombre en su día a día, la acumulación de los mismos va en acelerado incremento con el paso del tiempo, esto causa que su almacenamiento, búsqueda y gestión se convierta en un proceso complejo si no se hace un buen uso de las Tecnologías de la Información y las Comunicaciones¹. Por lo expresado anteriormente, se hace necesario tener aplicaciones de software, enfocadas a hacer búsquedas y recolección de la información, con el objetivo de que las entidades puedan acceder de forma rápida y precisa a los datos almacenados. Por tales motivos, toda empresa, institución u organización tiene como premisa la informatización y el desarrollo de las TIC como un proceso indispensable para lograr su progreso.

Cuba no está exenta a este proceso global, por lo que ha sido interés permanente del Estado el progreso de las TIC en todas las ramas de la sociedad. Como un proyecto de desarrollo, que tiene como pilares la justicia social, la equidad y solidaridad, se ha diseñado e iniciado la aplicación de estrategias que permiten convertir las TIC en instrumentos a disposición del avance y las profundas transformaciones revolucionarias (Visbal, 2009). Numerosos son los logros que se han alcanzado hasta el momento como resultado de los grandes esfuerzos realizados por el gobierno cubano. Se han planificado metas ambiciosas que están a la altura de los países del primer mundo y hoy en día no están muy lejos de alcanzarse. Un ejemplo de ello es que la Industria Cubana del Software, con la participación de la Universidad de las Ciencias Informáticas² y otras empresas productoras de software del país, están haciendo realidad esas metas al trabajar arduamente en la informatización de las organizaciones e instituciones nacionales.

Una de las instituciones que es necesario informatizar es el Centro Nacional de Documentación e Información Judicial³, el cual es una biblioteca que posee los fondos más ricos y completos del país en cuanto a información jurídica al contar con documentos de gran valor histórico. La creación de la biblioteca constituye un paso importante y de gran utilidad para la administración de justicia, actividad en que el acceso a la información se torna cada día más complejo y por tanto, requiere de una ejecución rápida, objetiva y precisa para facilitar la acertada toma de decisiones con la prontitud requerida.

El CENDIJ brinda un conjunto de servicios al público como son: la búsqueda de información en formato duro y formato digital, los préstamos de libros tanto internos como externos, así como el uso de la

¹ En lo adelante TIC.

² En lo adelante UCI.

³ En lo adelante CENDIJ.

sala de lectura. Además de los servicios que se prestan, también se realizan una serie de actividades como son la compra de libros, las donaciones y el procesamiento de la información en general. Los fondos que posee el centro son: libros, folletos y publicaciones seriadas, los cuales se encuentran en formato duro y también se cuenta con disposiciones del consejo de gobierno, legislaciones, sentencias, investigaciones y doctrinas, en formato digital.

El CENDIJ tiene como procesos fundamentales la prestación de servicios y los procesos de gestión. Los procesos son cada vez más complejos porque consumen mucho tiempo al realizar manualmente la búsqueda de la documentación existente en la organización. Esto hace que se genere un volumen significativo de documentación, lo que trae como consecuencia que se vuelvan procesos lentos y que afecten de manera directa la satisfacción de los clientes. También la entidad cuenta con un registro de constancia de los fondos documentales existentes en la misma; si ésta información sufriera alguna pérdida o daño se afectaría su control. Por todo lo expuesto anteriormente, los procesos mencionados constituyen la prioridad ante la necesaria informatización de las funciones del CENDIJ al no existir un software que gestione la documentación y agilice las prestaciones de servicio.

A raíz de la situación planteada se traza el siguiente **problema a resolver**: La lentitud en los procesos de gestión y prestación de servicios causa insatisfacción en los clientes del CENDIJ.

Por lo tanto el **objeto de estudio** se enmarca en el proceso de Desarrollo del Software y para dar solución al problema planteado se determinó como **objetivo general** implementar un Sistema Integral de Documentación e Información Judicial⁴ para agilizar los procesos de gestión y prestación de servicios que se realizan en el CENDIJ.

Se delimita como **campo de acción** el proceso de desarrollo de software para aplicaciones web y se defiende como **idea** que la implementación del Sistema Integral de Documentación e Información Judicial potenciará los procesos y las prestaciones de servicio en el mismo.

Para dar cumplimiento a lo antes expuesto se desglosaron como **objetivos específicos** los siguientes:

- Realizar un estudio de las principales tendencias sobre las tecnologías y herramientas existentes para el desarrollo de aplicaciones web.
- Realizar el análisis del diseño de clases y el modelo de datos.
- Implementar un Sistema Integral de Documentación e Información Judicial a partir de los artefactos obtenidos en fases anteriores.

⁴ En lo adelante SIDIJ.

- Validar el Sistema Integral de Documentación e Información Judicial a través de pruebas de software de Caja Blanca y Caja Negra.

Entre los **métodos de investigación** existentes se utilizaron los siguientes:

Métodos teóricos:

- Analítico-Sintético: Se estudian los servicios que se prestan en el CENDIJ, como también se usa la documentación existente de cada uno de estos servicios y formas de llevarlos a cabo.
- Sistémico: Se tiene en cuenta el problema como un todo, con el propósito de que se desarrolle un sistema flexible y robusto que cumpla con los requisitos exigidos.
- Modelación: Se utiliza fundamentalmente para realizar el modelo de todos los artefactos que se generan en el flujo de trabajo de implementación.

Métodos empíricos:

- Experimento: El método se utiliza para evaluar el rendimiento del sistema a construir.

El documento está estructurado de la siguiente manera:

Capítulo 1 Fundamentación Teórica

Se realiza la fundamentación teórica de la investigación donde se incluye una breve reseña histórica de los servicios en las bibliotecas, además se hace un estudio de la trascendencia de los sistemas informáticos relacionados con la gestión de la informática jurídica a nivel internacional y nacional. Se expone una breve descripción de las principales tecnologías tanto del lado del cliente como del servidor. Este capítulo también incluye las metodologías en conjunto con las herramientas que se utilizaron para la solución del problema y se abordan los conceptos asociados a software de gestión jurídica para lograr un mejor entendimiento en el desarrollo del nuevo sistema.

Capítulo 2: Descripción y análisis de la solución propuesta

Se hace una valoración de los artefactos de Especificación de Requisitos, Diagrama de Casos de Uso del Sistema, Diagrama de Clases del Diseño y del Modelo de Datos, todos ellos obtenidos en la fase de Análisis y Diseño. Se describen también las principales clases a utilizar en la implementación del Sistema Integral de Documentación e Información Judicial y se expone la propuesta de solución con su respectivo tratamiento de errores y Diagrama de Componentes.

Capítulo 3: Validación de la solución propuesta

Se incluye en este capítulo un estudio de los conceptos fundamentales de las pruebas de software con sus clasificaciones. Se aplican casos de pruebas para validar la implementación de los componentes obtenidos. Los casos de pruebas aplicados permiten validar la solución propuesta al detallar las no conformidades resultantes en cada iteración.

Capítulo 1: Fundamentación Teórica

Introducción

En la presente investigación existen conceptos asociados a los sistemas de gestión que pueden resultar desconocidos y son de vital importancia para la comprensión de la misma. Ayuda en esta investigación, el conocimiento del estado actual de dichos sistemas y el análisis de sus distintas formas de clasificación, todo esto enfocado a la utilización de la información jurídica.

En este capítulo se hará además una introducción a la historia de los servicios prestados por las bibliotecas. Se describen brevemente las técnicas fundamentales de programación, también se hace un estudio de las tecnologías y herramientas utilizadas mundialmente en el desarrollo de software.

1.1 Reseña Histórica de los Servicios en las Bibliotecas

En la actualidad el manejo de la información en Internet como alternativa universal de comunicación, refleja nuevas tendencias en la gestión de la información que aparecen en los diferentes espacios.

Las prestaciones de servicios⁵ son esenciales, sin ellas, las bibliotecas serían sólo un almacén de libros. Las prestaciones permiten dar el enlace entre las colecciones y los usuarios ofreciéndoles un mejor acceso a la información (Green, 1876). El concepto más antiguo del trabajo de referencia moderno se le atribuye a Samuel Swett Green en su libro titulado "Personal relations between librarian and readers" publicado en 1876, documento que refleja la defensa que hace Swett sobre la importancia del servicio personalizado y la orientación en los centros de información. Hasta ese momento las bibliotecas sólo se enfocaban en obtener y organizar materiales, esperando que los usuarios fueran autónomos en su búsqueda de información. Los servicios bibliotecarios surgen a partir de la sistematización de los materiales que conforman las colecciones y de la necesidad que existía de mostrar la información, al tener como objetivo aumentar y enriquecer los fondos en contenido. Debido a la sistematización, surgen servicios como los de préstamos y consultas, los cuales son parte de los servicios fundamentales que se realizan en el CENDIJ. Las prestaciones de servicios específicas de ese centro son: el servicio de sala de lectura, de préstamo externo de documentos, referencia, preparación y entrega de información digitalizada, de búsqueda de información en internet, disseminación selectiva de la información y preparación, como también la entrega de noticieros.

⁵ Son los servicios que se llevan a cabo en una biblioteca.

Capítulo 1: Fundamentación Teórica

1.2 Conceptos asociados a los Sistemas de Documentación e Información Judicial

Los conceptos expuestos a continuación se han extraído del Diccionario de Términos Jurídicos bajo recomendación de especialistas del CENDIJ.

Informática Jurídica

La Informática Jurídica consiste en una ciencia que forma parte de la Informática, es la especie en el género y se aplica sobre el Derecho, de manera que se dé el tratamiento lógico y automático de la información legal.

Informática Jurídica Documental

Constituye el área más antigua de la informática jurídica. Ésta parte de la informática jurídica se desarrolla con la finalidad de expandir los sistemas de documentación automatizada.

Búsqueda Jurídica Documental

En los sistemas de informática jurídica documental se busca la creación de bancos de datos jurídicos que se refiera a cualquier fuente del derecho con la finalidad de poder establecer una relación entre el usuario y la computadora, la cual se torna interactiva, haciendo posible de que el usuario interroge o enfoque palabras para que la computadora realice la función de búsqueda de las mismas.

Software de Gestión

Comprende las aplicaciones diseñadas para sustituir procedimientos comerciales o administrativos, desarrolladas especialmente para adecuarse a los diferentes requisitos de la organización.

Doctrina

Se entiende por doctrina jurídica la opinión de los juristas prestigiosos sobre una materia concreta. En el campo del Derecho, una doctrina jurídica es un concepto que sustentan los juristas y que influye en el ordenamiento jurídico.

Doctrina Legal

La doctrina establecida por los tribunales al interpretar y aplicar de modo reiterado e idéntico las leyes y que referida al Tribunal Supremo, configura la jurisprudencia.

Legislación

Acción y efecto de legislar. Conjunto de leyes por las cuales se gobierna un Estado o una materia determinada. Ciencias de las leyes.

Capítulo 1: Fundamentación Teórica

Sentencia

Acto del Órgano jurisdiccional que pone término al proceso resolviendo todas las cuestiones litigiosas planteadas por las partes. Según el tipo de acción ejercitada se habla de sentencias declarativas, constitutivas o de condena.

Descriptor

Es cada uno de los términos o expresiones escogidos entre un conjunto de sinónimos o casi sinónimos para representar (en calidad de término preferido), generalmente de manera unívoca, un concepto susceptible de aparecer con cierta frecuencia en los documentos indizables y en las consultas que se realicen.

Flujo ascendente

Concepto que categoriza algunos tipos de documentos como investigaciones que no llegan a tener el rótulo de informe científico pero cuyo nivel de profundidad merece que sean tenidos en cuenta, como por ejemplo tesis, actas de congresos, boletines, cuadernos de trabajo, informes técnicos, autobiografías, programas de computación, entre otros. Es el conjunto de documentos de tirada limitada y de circulación restringida que no pueden obtenerse en los canales habituales de venta.

Fondo General

Es el conjunto de documentos que conforman la colección de la biblioteca del CENDIJ cuya temáticas no pertenecen a la especialidad de Derecho y Justicia. (Jurídicos, 1999)

1.3 Software existente vinculado a Sistemas de Documentación e Información Judicial

En el mundo, el desarrollo de las tecnologías va en conjunto al incremento de sistemas para la informatización de los procesos en las distintas esferas. La necesidad de comprender y analizar la trascendencia de los sistemas informáticos relacionados con la gestión de la informática documental en el ámbito internacional y nacional es un factor clave para la realización de un nuevo sistema.

1.3.1 Sistemas Internacionales

Legislaw es un directorio orientado a facilitar la localización de sitios web con contenidos jurídicos, es una guía que contribuye a simplificar el acceso a información desarrollada por terceros. Este sitio facilita a estudiantes, profesionales, entidades, organismos y usuarios en general el acceso a páginas donde es posible hallar cientos de leyes nacionales y locales, fallos jurisprudenciales en las más diversas

Capítulo 1: Fundamentación Teórica

disciplinas jurídicas, opiniones doctrinarias, acceso a sistemas de autoconsulta⁶ de causas judiciales y administrativas, a Bases de Datos⁷ de interés práctico, a sitios gubernamentales y a muchas otras páginas de interés jurídico. (Legislaw, 1999)

- **DOC Expert:** Este sistema puede ser implementado en modalidad monousuario (en una sola computadora), para usuarios que diariamente reciben gran cantidad de información de diversas fuentes y desean disponer de una herramienta que ordene dicha información y luego les permita recuperarla en el momento que sea necesario sin costosas pérdidas de tiempo. El sistema se compone de dos Módulos: Centro de Control y Administración de Documentos. (DocExpert, 2011)
- **SiDIJ (Sistema Documental de Informática Jurídica):** El proyecto consiste en convertir varios documentos de jurisprudencia y doctrina de diversos formatos electrónicos a un formato documental uniforme, generar una base de datos con los documentos convertidos e indexados en 1000 discos compactos, realizar un control de calidad de la información previo a su publicación y montar un motor de búsquedas que permita la consulta de la jurisprudencia desde cualquier computadora en la que se esté el disco compacto. (Jurídica, 2011)
- **Jurisoft (Sistemas de Informática Jurídica):** Es un sistema con vocación de servicio integral para cubrir necesidades de la actividad diaria de los abogados como BD Documentales, Bibliografía, Infraestructura Informática, Servicios de Internet, Software de Gestión y Seguridad Informática. (Jurisoft, 2011)
- **Lex-Doctor 9.0:** Es un software que permite opcionalmente gestionar las BD a través de Internet y cambiar el motor de datos que gestiona la información. Este software ha sido desarrollado por Sistemas Jurídicos SRL, una empresa que desarrolla y comercializa exclusivamente sistemas de computación para ser aplicados a la actividad jurídica y está ubicada como líder en el mercado latinoamericano en la producción de dichos software. (Doctor, 2011)
- **Level Kmaleón:** Level Kmaleón pertenece a Level Programs, una empresa de informática de Barcelona, España. Desde sus inicios ésta empresa se especializó en el mundo jurídico, por lo que la mayoría de las aplicaciones que ha desarrollado están orientadas a la gestión de despachos profesionales: abogados, economistas y asesores. (Kmaleon, 2010)

⁶ Es un sistema donde cualquier usuario puede realizar una consulta y el sistema devuelve una respuesta de una base de conocimientos existente.

⁷ En lo adelante BD.

1.3.2 Sistemas Nacionales

- **Software para el Control de Contratos:** DESOFT ha trabajado por ir informatizando todos sus procesos internos del país. En Ciego de Ávila existe una aplicación para el control de los contratos a la que se le hizo una primera mejora. En esa aplicación los abogados son los encargados de registrar los clientes e insertar los contratos. El número del contrato es invariable, pero el software permite incorporar detalles como son: pagos, plazos de entrega y detalles sobre la venta. Los especialistas de ventas y los subgerentes tienen su sesión para hacer las consultas que necesiten. También existe la posibilidad de hacer reportes y exportar a una tabla en Excel. (Silva, 2010)

Aunque el Sistema Jurídico Cubano tuvo sus orígenes en el Sistema Jurídico Romano - Francés (Taboada, 2010), del cual descienden también los sistemas de gestión de otros países mencionados anteriormente, sus realidades actuales en el plano jurídico difieren totalmente de la cubana, razón por la cual se dificulta la adaptación de dichos sistemas a uno cubano. Todos ellos se rigen por las leyes judiciales de sus respectivos países por lo que sus módulos no se corresponden con las leyes socialistas establecidas en Cuba.

En Cuba es necesario un sistema que refleje la realidad jurídica cubana desarrollado por especialistas con herramientas de código abierto que permitan su mantenimiento, adaptaciones y versiones posteriores, aunque el Software para el Control de Contratos de Ciego de Ávila es realizado en este país, no cumple con los requisitos necesarios ni es posible la adaptación sus módulos, por ello se considera que se debe realizar un sistema que cumpla con las características necesarias para satisfacer lo referente a la gestión de documentación e información jurídica y las prestaciones de servicios que puedan ser utilizados en cualquier biblioteca judicial.

1.4 Metodologías de Desarrollo de Software

Las metodologías de desarrollo son un conjunto de procedimientos, técnicas, herramientas y un soporte documental formal que ayuda a los desarrolladores a realizar un nuevo software. Son útiles para garantizar la validez según los requisitos iniciales y al minimizar las pérdidas de tiempo en el proceso de generación de software. (Metodologías, 2011)

1.4.1 Proceso Unificado de Desarrollo⁸

RUP es un proceso de desarrollo propuesto por Rational Software Corporation, resultado del esfuerzo de las tres últimas décadas en desarrollo de software y de la experiencia de sus creadores Ivar Jacobson, Grady Booch y James Rumbaugh. Este proceso proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo.

Es una metodología basada en un pequeño grupo de principios claves: el equipo de un proyecto de software debe planificar el desarrollo, debe conocer hacia donde se dirige, debe documentar el proyecto de una manera perdurable y extensible.

RUP está definido por tres características fundamentales:

- **Dirigido por Casos de Uso**

Los casos de uso constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.

- **Centrado en la arquitectura**

La arquitectura involucra los elementos más significativos del sistema y está influenciada por plataformas de software, sistemas operativos, manejadores de BD, protocolos, consideraciones de desarrollo como sistemas heredados y requisitos no funcionales.

- **Iterativo e Incremental**

Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un pequeño proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

La metodología RUP divide en 4 fases el desarrollo del software:

- **Inicio**

En la fase inicial se establecen los objetivos para el ciclo de vida del producto. Se establece el caso del negocio con el fin de delimitar el alcance del sistema.

- **Elaboración**

En Elaboración se plantea la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requisitos funcionales, al manejar los riesgos que interfieran con los objetivos del sistema.

⁸ En lo adelante, RUP, Rational Unified Process.

- **Construcción**

En Construcción se alcanza la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, el cual es frecuentemente llamado versión beta.

- **Transición**

En Transición se realiza la entrega del producto operado una vez realizadas las pruebas de aceptación por un grupo especial de usuarios al efectuarse los ajustes y correcciones que sean requeridos. (Electrónica, 2010)

RUP incluye artefactos y roles, con dichas características ha hecho que junto con el UML, constituya la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. (Martínez, 2010)

1.4.2 Programación Extrema⁹

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software al promover el trabajo en equipo, preocuparse por el aprendizaje de los desarrolladores y propiciar un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP en dos: sin cubrir los detalles técnicos y de implantación de las prácticas.

Las prácticas fundamentales de la metodología XP son:

- **Entregas pequeñas**

La idea es producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad pretendida para el mismo.

⁹eXtreme Programming en lo adelante, XP.

- **Refactorización**

La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. También mejora la estructura interna del código sin alterar su comportamiento externo (Fowler, 2010).

- **Propiedad colectiva del código**

Cualquier programador puede cambiar una parte del código en un momento dado. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código. (Penadés, 2006)

Al realizar un estudio de las principales metodologías de desarrollo de software se decidió optar por RUP al adaptarse a las características necesarias y complejidad del sistema informático a desarrollar. Es una metodología flexible que genera abundante documentación, lo cual contribuye a un mejor entendimiento entre los integrantes del equipo de trabajo, aspecto imprescindible para facilitar el desarrollo de una segunda versión del sistema.

1.5 Lenguaje de Modelado

Un modelo de un sistema software está construido en un lenguaje de modelado, como el UML. El modelo tiene semántica y notación, puede adoptar varios formatos que incluyen texto y gráficos. El modelo pretende ser más fácil de usar para ciertos propósitos que el sistema final (James Rumbaugh, 1998).

1.5.1 Lenguaje Unificado de Modelado¹⁰

UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. Está considerado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. Pretende unificar la experiencia pasada sobre técnicas de modelado. UML incluye conceptos semánticos, notación y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está hecho premeditadamente para ser utilizado en

¹⁰ En lo adelante UML.

Capítulo 1: Fundamentación Teórica

herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero la intención es que sea útil en un proceso de desarrollo iterativo. El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos. (James Rumbaugh, 1998)

1.6 Lenguajes de Programación

Las máquinas en general y las computadoras en particular necesitan de un lenguaje propio para poder interpretar las instrucciones que se les dan y para que se les pueda controlar su comportamiento. El lenguaje que permite esta relación con las computadoras es el de programación.

En este sentido, hay que diferenciarlo del lenguaje informático, con el que se le suele confundir, al ser éste mucho más amplio, abarcando desde los lenguajes de programación hasta los que dan formato a los textos, como el HTML¹¹.

Asimismo, el lenguaje de programación está conformado por una serie de reglas sintácticas y semánticas que serán utilizadas por el programador y a través de las cuales creará un programa o subprograma. Por otra parte, las instrucciones que forman dicho programa son conocidas como código fuente. (Lanzillotta, 2004)

1.6.1 Lenguajes del lado del servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a BD, conexiones en red y otras tareas para crear la vista final que se enviará al cliente, el cual solamente recibe una página con el código HTML. Como la página resultante contiene únicamente código HTML, es compatible con la mayoría de los navegadores.

1.6.1.1 Java

Java es un lenguaje orientado a objeto. Fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas web.

En Java se pueden realizar los applets¹², que son aplicaciones especiales, los cuales se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor web, por lo general los applets son programas pequeños y de propósitos específicos.

¹¹ Hyper Text Markup Language o Lenguaje de Marcado de Hipertexto.

¹² Ver glosario de términos.

Capítulo 1: Fundamentación Teórica

Otra de las utilidades de la programación en Java es el desarrollo de aplicaciones, que son programas que se ejecutan en forma independiente, es decir, se pueden realizar aplicaciones como un procesador de palabras, una hoja de cálculos o una aplicación gráfica. Permite la modularidad por lo que se pueden hacer rutinas individuales que sean usadas por más de una aplicación.

La programación en Java permite el desarrollo de aplicaciones bajo el esquema de cliente servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente y de esta forma logra distribuir el trabajo a realizar. (Programación, 2009)

1.6.1.2 PHP5.3¹³

Es un lenguaje de programación del lado del servidor gratuito, rápido, de código abierto, con una gran librería de funciones y mucha documentación. Es independiente de la plataforma, puesto que existe un módulo de PHP para la mayoría de los servidores web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, al permitir trasladar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún esfuerzo.

En cuanto a la seguridad, es importante el hecho de que en muchas ocasiones PHP permite configurar el servidor de modo que se aprueben o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde los cálculos matemáticos complejos hasta el tratamiento de conexiones de red.

Algunas de las más importantes capacidades de PHP son: compatibilidad con las BD más comunes, como MySQL, mSQL, Informix y Oracle. Incluye funciones para el envío de correo electrónico, adjuntar archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales. (Carrero, 2011)

Para el desarrollo del sistema se escogió PHP debido a que es un lenguaje de programación del lado del servidor gratuito e independiente de la plataforma, rápido y su desarrollo no necesita grandes capacidades de hardware. Teniendo en cuenta los servidores, una aplicación en PHP no demanda tanto volumen de memoria de máquina como podría solicitar una aplicación en Java con sus servidores que

¹³ PHP es el acrónimo de Hipertext Preprocesor.

Capítulo 1: Fundamentación Teórica

podieran necesitar varios procesadores y gran capacidad de memoria RAM, además, al usar Java en aplicaciones web, el acceso a las páginas se demora más por la sobrecarga que genera la máquina virtual, por ello se necesitan mejoras de recursos de hardware y no es posible para Cuba tener acceso a la última versión de la máquina virtual.

1.6.2 Lenguajes del lado del cliente

Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. Pero la página no se verá bien si el ordenador cliente no tiene instalados los plugin adecuados. El código, tanto del hipertexto como de los scripts, es accesible a cualquiera y ello puede afectar a la seguridad. (Mordevre, 2008)

1.6.2.1 HTML

HTML fue creado por Tim Berners-Lee a partir del SGML (Standard Generalized Markup Language) o lenguaje de marcación generalizado. El HTML nació para producir todo tipo de documentos estructurados.

En sus inicios el HTML creaba documentos muy básicos, pero muy pronto con el auge de Internet, comienza también su desarrollo. Así en sus comienzos los navegadores de Internet solo funcionaban en modo texto. Luego todos se fueron mejorando a un ritmo extraordinario debido al desarrollo de la red de redes (Internet).

Como ya se ha expuesto, HTML es un lenguaje de programación especializado en crear páginas web, para ello se utilizan una serie de etiquetas. Todo documento creado con HTML tiene una estructura claramente definida. Siempre se comienza con la etiqueta <HTML>, que es la que comprende a toda la página web. Tiene dos secciones básicas bien diferenciadas: la cabecera y el cuerpo, que se corresponden con las etiquetas <head> y <body> respectivamente. La cabecera puede contener información, siempre lleva el título del documento HTML encerrado por la etiqueta <title>, mientras que en el cuerpo se localiza todo el contenido de la página web, sea texto, imágenes, sonido, hipervínculos, video, entre otros. (La Revista Informática, 2006)

1.6.2.2 CSS¹⁴

Las CSS son una tecnología desarrollada con el fin de separar la presentación de la estructura del HTML. Funcionan aplicando reglas de estilo a los elementos HTML, entre las que incluyen tamaño, color de fondo, color del texto, posición de los elementos, márgenes, tipos de letra, quedando de esta manera todo lo que tiene que ver con la parte gráfica de la web, separada completamente de la estructura del HTML.

Este lenguaje desarrollado por la W3C, ha venido haciéndose cada vez más importante entre los diseñadores, gracias a la facilidad de uso y a sus flexibles resultados. Aprender a conocer CSS brinda como resultado un mejor flujo de trabajo, mayor organización del código, menos peso en las páginas y mayor flexibilidad a los cambios.

Los tres principales elementos en el desarrollo de CSS son:

- **Atributos**
- **Valores**
- **Selectores** (Spirate, 2011)

1.6.2.3 JavaScript

El JavaScript es un lenguaje interpretado, lo que significa que no necesita ser compilado. Proviene del Java y se utiliza principalmente para la creación de páginas web. Es una mezcla entre Java y HTML. Su creador fue Brendan Eich. Al principio se le llamó Mocha, más tarde LiveScript, hasta que en el año 1995 se le llamó JavaScript. Es un lenguaje muy diferente del Java porque es orientado a objetos, no tiene herencia, al contrario del Java que sí la tiene, sino que el JavaScript es más bien un lenguaje orientado a eventos.

Otra diferencia entre ambos lenguajes es que mientras con Java se puede crear aplicaciones autónomas como son los applets, el JavaScript es un lenguaje que se incorpora dentro de la página web, formando parte del código HTML, sin el que no puede existir.

Se puede incluir el código del JavaScript en cualquier página web o documento HTML, desde el punto de vista cliente como servidor en documentos PHP, Asp, entre otros. Este código va incluido dentro de las etiquetas de HTML, de esta manera: `<SCRIPT> </ SCRIPT>`. El JavaScript es un lenguaje que diferencia entre mayúsculas o minúsculas y los espacios en blanco también los tiene en cuenta. (La Revista Informática, 2006)

¹⁴ Cascading Style Sheets, Hojas de Estilo en Cascada.

1.6.2.4 JSON¹⁵

JSON es un formato ligero de intercambio de datos. Está basado en un subconjunto del lenguaje de programación JavaScript. Es un formato de texto que completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl y Python. Estas propiedades hacen que JSON sea un lenguaje magnífico para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes es conocido como un objeto, registro, estructura, diccionario, tabla, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras. (JSON, 2011)

1.7 Herramientas CASE¹⁶

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores durante todos los pasos del ciclo de vida de desarrollo de un software. CASE también se define como un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. (Scribd, 2011)

1.7.1 Visual Paradigm para UML 6.4

Visual Paradigm es una de las herramientas UML CASE considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones.

Fue creado para el ciclo vital completo del desarrollo de software, el cual lo automatiza y acelera al permitir la captura de requisitos, análisis, diseño e implementación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y la definición de BD a partir del esquema de las clases.

¹⁵ JavaScript Object Notation, Notación de Objetos de JavaScript.

¹⁶ Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador.

Capítulo 1: Fundamentación Teórica

Esta herramienta permite aumentar la calidad del software a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa al ayudar 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. (VP, 2011)

Permite además invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñado para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. (EcuRed, 2011)

1.7.2 Rational Rose Enterprise Edition

Es una herramienta CASE que permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo del software. (Teresita Rojas, 2011) Rational Rose permite completar una parte de los flujos fundamentales de RUP tales como:

- Captura de requisitos (parcialmente).
- Análisis y diseño (completamente).
- Implementación (como ayuda).
- Control de cambios y gestión de configuración (parcialmente).

Entre las características principales de Rational se pueden destacar que:

- Admite como notaciones: UML, OMT y Booch.
- Permite desarrollo multiusuario.
- Genera documentación del sistema.
- Está disponible en múltiples plataformas. (Edition, 2011)

Es evidente la completitud de las tareas que aporta Visual Paradigm, que además maneja todas las fases de los ciclos de vida de RUP, metodología usada para guiar los pasos de la construcción del software en cuestión, por lo que se ha hecho una correcta elección por parte de los analistas del proyecto SIDIJ al seleccionar esta herramienta. Además, en la UCI se cuenta con una licencia y se han incrementado los niveles de aceptación por su robustez, usabilidad y portabilidad.

1.8 Frameworks de Desarrollo

Los objetivos principales que persigue un framework son acelerar el proceso de desarrollo, reutilizar código ya existente y promover prácticas de desarrollo como el uso de patrones. Un framework web, por tanto, se puede definir como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. (Carrero, 2011)

1.8.1 Symfony 1.4

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, 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.

Symfony ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de los gestores de BD como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix o Linux como en plataformas Windows.

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Es fácil de instalar y configurar en la mayoría de plataformas.
- Es independiente del sistema gestor de BD.
- Es sencillo de usar en la mayoría de casos pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Está basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Tiene un código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Es fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony está basado en un patrón conocido como MVC¹⁷, formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

¹⁷ Modelo Vista Controlador.

Capítulo 1: Fundamentación Teórica

- 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. (Potencier, 2007)

1.8.2 Zend Framework

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas instituciones han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y Strikelron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework.

Es un framework de código abierto para desarrollar aplicaciones web y servicios web. Usa código totalmente orientado a objetos. En él, cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "uso a voluntad" (use-at-will).

Ofrece un gran rendimiento, una abstracción de base de datos fácil de usar y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. (Gonzales, 2010)

1.8.3 CodeIgniter

CodeIgniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que se debe seguir para obtener provecho de la aplicación. Esto marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado como también más fácil de crear y mantener. CodeIgniter implementa un estilo MVC, que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales. (Alvarez, 2009)

Se ha decidido utilizar Symfony por ser un framework muy completo diseñado para optimizar el desarrollo de aplicaciones web al poseer herramientas y clases dirigidas a informatizar las tareas comunes de una aplicación y permitirle al desarrollador enfocarse en los aspectos específicos de la misma.

1.8.4 Framework de interfaz de usuario Ext JS 3.0

Ext JS es una librería de JavaScript para el desarrollo de aplicaciones web interactivas que usa tecnologías como AJAX¹⁸, DHTML¹⁹ y DOM²⁰. Originalmente fue construida como una extensión de la biblioteca YUI²¹, en la actualidad puede usarse como extensión para las bibliotecas jQuery²². Desde la versión 1.1 puede ejecutarse como una aplicación libre.

Ext JS proporciona una interfaz cuya potencia radica en gran colección de componentes para el diseño de interfaces del lado del cliente. Tiene incluidos la mayoría de los controles de los formularios web conteniendo tablas para mostrar datos y elementos semejantes a la programación de escritorio como los formularios, paneles, barras de herramientas, menús y muchos otros. Dentro de su librería de componentes incluye componentes para el manejo de datos, lectura de XML²³, lectura de datos JSON e implementaciones basadas en AJAX. Presenta el uso de JavaScript con una programación orientada a objetos.

Una de las grandes ventajas es que permite crear aplicaciones complejas utilizando componentes predefinidos, así como un manejador de layouts²⁴, por esto provee una experiencia consistente sobre cualquier navegador. (Cobas, 2011) Ext JS soporta todos los navegadores web más importantes. (Sclavos, 2011)

1.9 IDE²⁵

Un IDE es una aplicación compuesta por un conjunto de herramientas para programar, la cual puede estar dedicada de forma exclusiva a un solo lenguaje de programación o bien a poder utilizarse para varios. Un IDE es un entorno de programación que ha integrado diferentes aplicaciones, tales como: editores de código, compiladores, depurador, entre otras. (M., 2010)

1.9.1 NetBeans

El IDE NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de

¹⁸ Ashynchronous JavaScript And XML.

¹⁹ Es el HTML Dinámico.

²⁰ Es el Document Object Model o DOM , Modelo de Objetos del Documento

²¹ Yahoo User Interface, conocido como el I framework CSS de Yahoo!.

²² Ver glosario de términos.

²³ Extensible Markup Language, Lenguaje de Etiquetado Extensible.

²⁴ Ver glosario de términos.

²⁵ Integrated Development Environment, Entorno de Desarrollo Integrado.

Capítulo 1: Fundamentación Teórica

programación. Existe además un número importante de módulos para extender el IDE NetBeans, el cual es un producto libre y gratuito sin restricciones de uso.

El NetBeans es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE²⁶, web, EJB²⁷ y aplicaciones móviles) (Estrada., 2011).

Todas las funciones son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente. Sun Studio, Sun Java Studio Enterprise y Sun Java Studio Creator de Sun Microsystems han sido todos basados en el IDE NetBeans. (Duarte, 2008)

La integración del NetBeans con Symfony permite desarrollar aplicaciones de forma más sencilla y productiva. En primer lugar, es posible crear nuevos proyectos y aplicaciones directamente desde el IDE. También se pueden ejecutar todas las tareas de Symfony, incluso pasándole argumentos y opciones, visualizando el resultado sin necesidad de utilizar una consola de comandos externa. (Symfony.es, 2009)

NetBeans permite crear aplicaciones Web con PHP 5 y además viene con soporte para Symfony, un gran framework con estilo MVC, escrito en PHP.

1.9.2 Eclipse

Eclipse es una plataforma de software de código abierto y que ha sido muy utilizada para desarrollar IDEs, pero del mismo modo se puede usar para otros tipos de aplicaciones cliente. Lo mejor de Eclipse es que tiene una gran comunidad de usuarios extendiendo constantemente las aplicaciones. (tu Función, 2007)

Eclipse fue desarrollado originalmente por IBM Canadá. Actualmente es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. En noviembre del 2001, se formó un consorcio para el desarrollo futuro de Eclipse como código abierto.

Esta plataforma, típicamente ha sido usada para desarrollar IDE, como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse y que son

²⁶ Java 2 Platform, Standard Edition

²⁷ Enterprise JavaBeans.

usados también para desarrollar el mismo Eclipse. Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.

1.9.3 Zend Estudio

Zend Studio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. (tu Función, 2007)

Este IDE está destinado a desarrolladores profesionales. Es propietario, compatible con las plataformas Linux, MAC y Windows e integrado para el lenguaje de programación PHP. Permite agilizar el desarrollo web y simplificar proyectos complejos. Tiene como características un buen completamiento de código, coloreado en la sintaxis del código, administración avanzada de proyectos, múltiples lenguajes, incorpora el Framework de Zend, PHP Documentor y manual de PHP.

Entre sus ventajas está que agiliza el trabajo, cuenta con un buen depurador, muchas opciones que permiten un desarrollo profesional de nuestras aplicaciones pero tiene de desventajas que requiere licencia de pago, no incluye editor visual HTML y es un poco complejo.

Aprovechando las facilidades que brinda NetBeans, la directiva del proyecto SIDIJ ha decidido utilizar el mismo como entorno de desarrollo de la aplicación que se desea desarrollar. Es multiplataforma, compatible con el lenguaje seleccionado y se le pueden añadir un sinnúmero de plugins²⁸ que ayudan al desarrollo de aplicaciones web. Zend Studio pese a las grandes ventajas que proporciona, requiere licencia de pago y no incluye editor visual HTML.

1.10 Servidor Web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP²⁹ o el protocolo HTTPS (la versión cifrada y autenticada). Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

1. Espera peticiones en el puerto TCP/IP³⁰ indicado (el estándar por defecto para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso.

²⁸ Ver glosario de términos.

²⁹ HyperText Transfer Protocol, Protocolo de Transferencia de Hipertexto.

³⁰ Transmission Control Protocol/Internet Protocol, Protocolo de control de transmisión/Protocolo de Internet.

4. Envía el recurso utilizando la misma conexión por la que recibió petición.
5. Vuelve al segundo punto.

Un servidor web que siga el esquema anterior cumple todos los requisitos básicos de los servidores HTTP, aunque sólo podrá servir ficheros estáticos. (Docencia Universitaria, 2010)

1.10.1 Apache 2.2.6

Apache es un programa de servidor web robusto de código abierto, cuya implementación se ha realizado de forma colaborativa con prestaciones, características y funcionalidades equivalentes a las de cualquier servidor comercial. El proyecto está bajo el control de un grupo de voluntarios de todo el mundo que, sirviéndose de Internet para comunicarse, desarrollan el programa y la documentación relacionada. A estos voluntarios se les conoce como el Apache Group. Además de esos voluntarios, otras personas han contribuido al proyecto desarrollando código o documentación y aportando ideas. (Scribd, 2011)

- Se ejecuta en varios de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar sus capacidades. Existen muchos módulos para Apache que son adaptables a éste.
- Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. También trabaja con Java y páginas jsp³¹ al poseer todo el soporte que se necesita para tener páginas dinámicas.
- Es posible configurarlo para que ejecute un determinado script³² cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs³³. Permite la creación de ficheros de log a medida del administrador, de éste modo puede tener un mayor control sobre lo que sucede en el servidor. (Identi, 2011)

1.10.2 Roxen

Roxen es un servidor web de licencia GNU³⁴, desarrollado por un grupo sueco que después fundaría la empresa Roxen Internet Services. Este servidor destaca por su cantidad de funcionalidades. Está desarrollado en el lenguaje Pike, ofrece muchos módulos que permiten el desarrollo de sitios web dinámicos, sin más herramienta que él mismo. Entre sus características principales están las siguientes:

³¹ JSP es el acrónimo de Java Server Pages.

³² Ver glosario de términos.

³³ Ver glosario de términos.

³⁴ GNU es un acrónimo recursivo (ver glosario de términos) que significa GNU No es Unix.

- Es multiplataforma: puede ejecutarse en Windows, Linux, MAC OS/X, Solaris.
- Posee código libre. (Scribd, 2011)

A pesar de que Roxen tiene una gran cantidad de funcionalidades, los analistas del proyecto SIDIJ en conjunto con la directiva se declinaron por el uso de Apache, que además de ser seguro, es uno de los más utilizados en el mundo. Es tan potente como flexible, libre, de código abierto y también posee módulos configurables de forma tal que se pueda decidir cuáles de estos serán ejecutados en el servidor.

1.11 Sistema Gestor de BD

Un Sistema Gestor de Base de Datos³⁵ es un conjunto de programas que permiten crear y mantener una BD asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes. (Collector, 2004)

1.11.1 PostgreSQL 8.4

Es un sistema de gestión de BD objeto relacional, distribuido bajo licencia BSD³⁶ y con su código fuente disponible libremente. Es el SGBD de código abierto más potente del mercado y en sus últimas versiones está a la par de otras BDs comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

A continuación se exponen algunas de las características más importantes de PostgreSQL:

- Es una base de datos segura.
- Tiene integridad referencial.
- Tiene múltiples métodos de autenticación.
- Acceso encriptado vía SSL.
- Provee actualización integrada momentánea (pg_upgrade).
- Posee documentación completa. (PostgreSQL-es, 2010)

³⁵ En lo adelante SGBD.

³⁶ Ver glosario de términos.

1.11.2 MySQL³⁷

La historia del MySQL se remite a principios de la década de 1980 cuando programadores de IBM lo desarrollaron para contar con un código de programación que permitiera generar múltiples BD para empresas y organizaciones de diferentes tipos.

Una de las características de MySQL es que permite recurrir a BD multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requisitos. Por otro lado, es conocido por desarrollar alta velocidad en la búsqueda de datos e información.

La UCI pertenece a la Comunidad Iberoamericana de PostgreSQL, una organización que alienta el desarrollo con esa herramienta, por lo que su uso es condicionado por la propia tendencia existente en la Universidad. Se debe señalar que respecto a la selección del SGBD se hace mayor alusión en la tesis de los autores: Yamila Zamora Mena y Eugenio Ávila Osoria titulada: “Proceso de migración de datos para el Sistema Integral de Documentación e Información Judicial (SIDIJ) del Tribunal Supremo Popular”. Por todo lo referido anteriormente, se toma la decisión de utilizar como SGBD a PostgreSQL.

Conclusiones del Capítulo

En el desarrollo de este capítulo se realizó un estudio de varios sistemas relacionados con la gestión jurídica en el ámbito mundial. También se observó una breve historia de los servicios en una biblioteca y los conceptos asociados al problema y se realizó un análisis de las herramientas a utilizar en la implementación del sistema para el CENDIJ. Todas las herramientas seleccionadas están en consonancia con las recomendaciones para el centro CEGEL del Informe Tecnológico de la Producción en la UCI en el año 2010 desarrollado por Pedro Piñeiro y Maikel Arcia.

Se adoptará la metodología de desarrollo RUP y se trabajará con Visual Paradigm para UML 6.4 como herramienta CASE para la especificación y el diseño de la aplicación. El lenguaje de programación del lado del servidor que se usará es PHP 5.3, como framework de desarrollo se recurrirá a Symfony 1.4 utilizando la librería Ext JS en su versión 3.0 y como servidor web se manipulará a Apache 2.2.6. Además se utilizará con una arquitectura basada en capas

³⁷ My Structured Query Language, Lenguaje de Consulta Estructurado.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

Capítulo 2: Descripción y Análisis de la Solución Propuesta

Introducción

En el presente capítulo se relacionan varios aspectos de la implementación del SIDIJ. Se comienza haciendo una valoración crítica de los artefactos propuestos en la fase de Análisis y Diseño que antecede a esta investigación. Se exponen los componentes que se generan durante en el flujo de implementación del sistema para facilitar la comprensión del funcionamiento del mismo. Por último se hace una descripción de clases y operaciones utilizadas, además, describe los estándares de codificación aplicados para una mejor legibilidad del código.

2.1 Valoración de los artefactos propuestos por los analistas

La ingeniería de requisitos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema. Los requisitos para un sistema de software por su parte determinan lo que hará el sistema y definen las restricciones de su operación e implementación. (Soto, 2010)

Uno de los artefactos generados durante esta etapa es la Especificación de Requisitos, propuesto por los analistas del sistema para describir los servicios que debe ofrecer el mismo y las limitaciones asociadas a su funcionamiento. En dicho artefacto se detallaron los requisitos funcionales y no funcionales obtenidos a partir de la extracción de información de las actividades que se desean informatizar en el CENDIJ.

Los artefactos recibidos reúnen los parámetros necesarios, lo cual permite seleccionar la información necesaria para establecer la funcionalidad a alcanzar con el sistema y reflejarse todas las necesidades de los clientes para ésta versión del producto.

2.1.1 Requisitos Funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer. Estos requisitos dependen del tipo de software que se desarrolle, de los posibles clientes y del enfoque general tomado por la organización al redactar los requisitos. Cuando se expresan como requisitos del usuario, habitualmente se escriben en una forma bastante abstracta. Sin embargo, los requisitos funcionales del sistema describen con detalle la función de éste, sus excepciones, entradas y salidas. (Sommerville, 2005)

Capítulo 2: Descripción y Análisis de la Solución Propuesta

La información detallada se aprecia en el documento de Especificación de Requisitos del proyecto SIDIJ en su versión 1.1, a continuación se presenta un ejemplo.

RF21 Insertar Documento: El sistema debe permitir insertar un documento, en dependencia del tipo de documento se establecen los siguientes campos:

Insertar Legislación

El sistema permitirá entrar los datos a la aplicación para insertar un nuevo documento de tipo Legislación.

Datos	Tipo	Obligatorio
Tipo de Norma	(string)	(Obligatorio)
Fecha de emisión	(date)	(Obligatorio)
Gaceta oficial	(string)	(Obligatorio)
Mes	(string)	(Obligatorio)
Fecha de entrada	(date)	(Obligatorio)
Especialista	(string)	(Obligatorio)
Descripción	(string)	(No Obligatorio)
Descriptor	(string)	(Obligatorio)
Órgano emisor	(string)	(Obligatorio)
Materia	(string)	(Obligatorio)

Tabla 1. Insertar Legislación.

2.1.2 Requisitos no Funcionales

Son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada, salida y las representaciones de datos que se utilizan en las interfaces del sistema. (Sommerville, 2005) En el documento de Especificación de Requisitos del proyecto SIDIJ en su versión 1.1 se encuentra mayor información sobre los requisitos no funcionales, de los cuales se presenta también un ejemplo:

Capítulo 2: Descripción y Análisis de la Solución Propuesta

7 Requisitos del producto

- ❖ **RNF14** Para que un cliente de la aplicación pueda ejecutar procesos, en línea, considerados en el sistema, el punto de acceso deberá cumplir con los siguientes requisitos mínimos: 512 Mb de memoria y 20 Gb de disco duro.
- ❖ **RNF15** Para que se pueda ejecutar la aplicación el servidor debe contar con los requisitos mínimos: 1 Gb de memoria, 120 de disco duro y un procesador Pentium IV.

Ilustración 1. Requisitos no funcionales del producto.

2.1.3 Diagrama de Casos de Uso del Sistema

En el Documento de Casos de Uso del Sistema en su versión 1.2 se aprecia la descripción del sistema en su totalidad en términos de casos de uso y la interacción con los actores del sistema. Este documento se hace imprescindible para la posterior implementación de las funcionalidades al mostrarse en él de una manera más específica las características que van a ser implementadas.

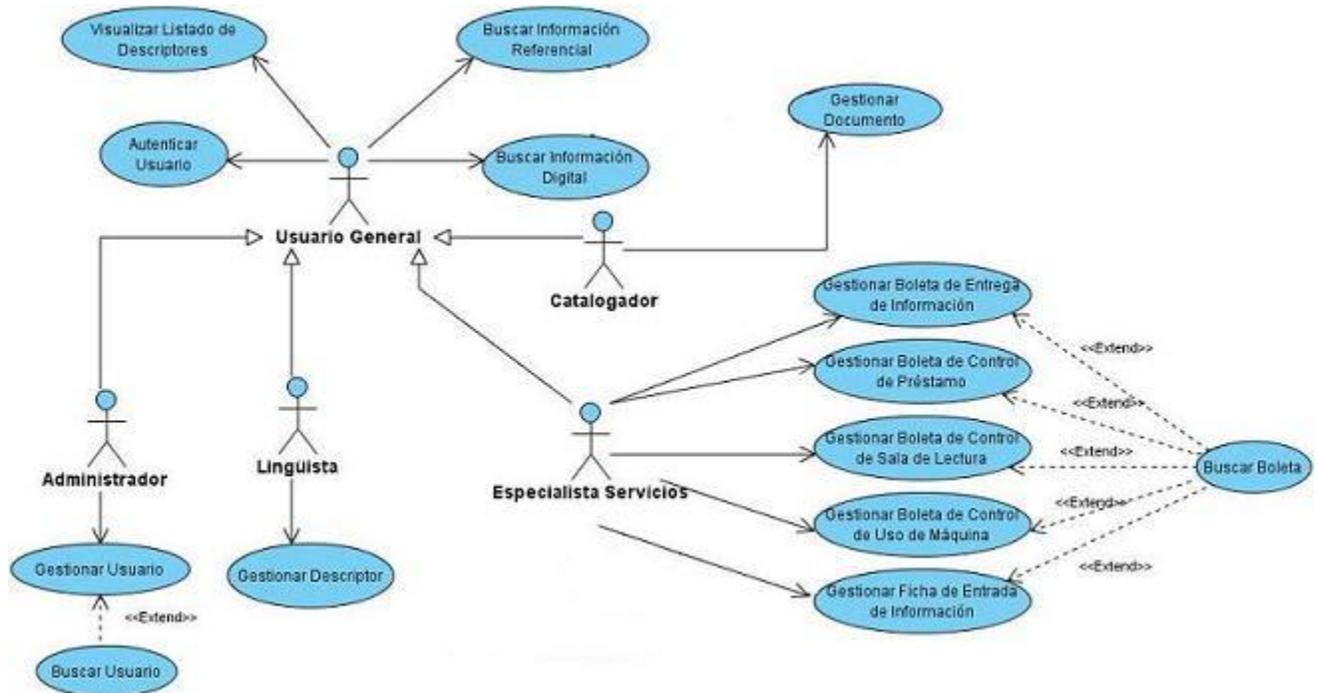


Ilustración 2. Diagrama de Casos de Uso del Sistema.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

2.1.4 Diagrama de Componentes

Un diagrama de componentes muestra el sistema dividido por componentes, así como la relación entre éstos. Un componente es una parte física de un sistema (módulo, base de datos, programa o ejecutable), es decir, la materialización de una o más clases.

En el siguiente diagrama se representan los componentes del SIDIJ, donde se evidencian las relaciones entre los componentes, desde el acceso a través del controlador frontal, la clase actions.php, cada una de las interfaces de la aplicación y las clases del negocio, así como los elementos de configuración.

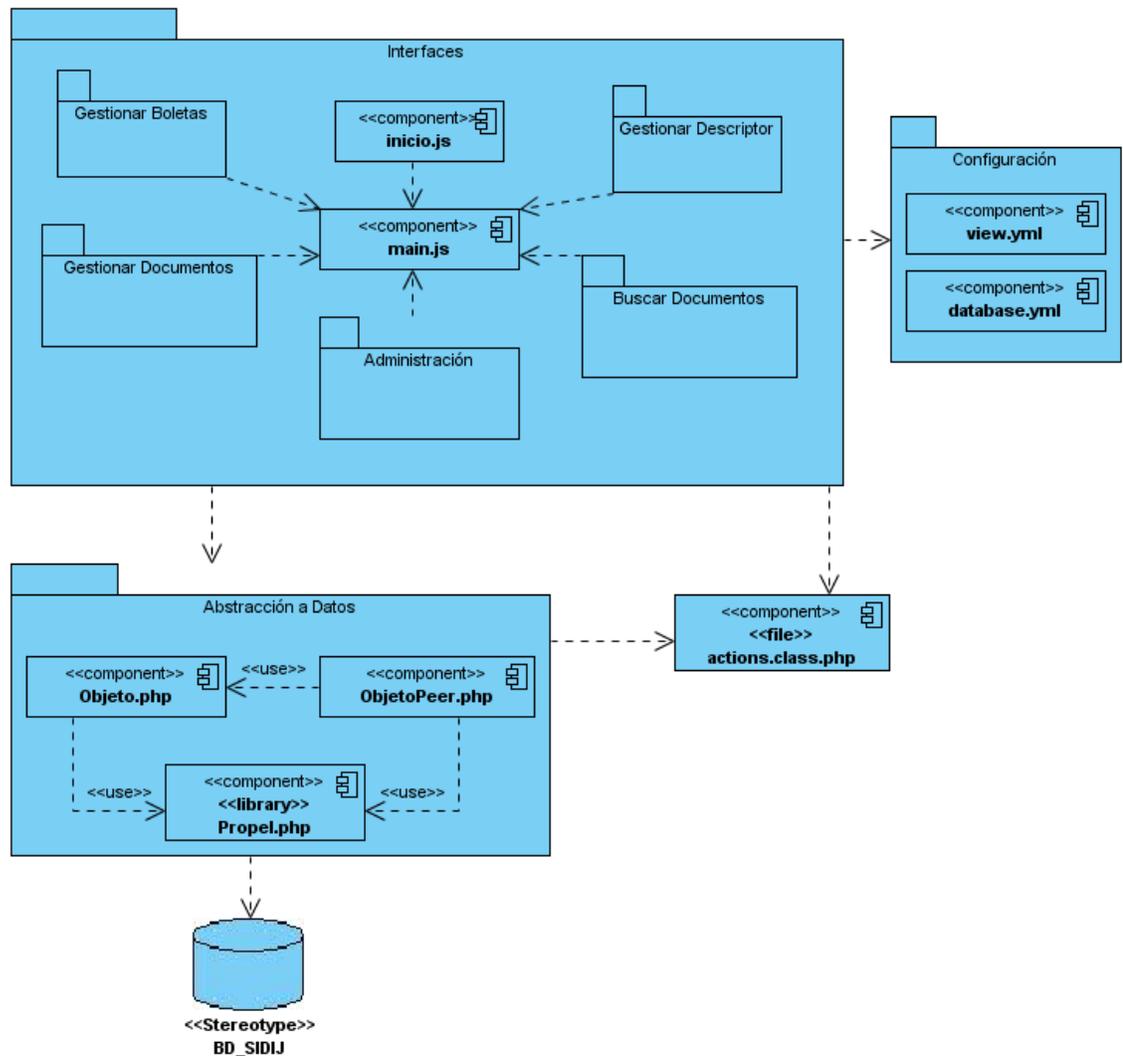


Ilustración 3. Diagrama de Componentes.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

2.1.5 Diagrama de Clases del Diseño

Es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a la implementación. Representa a los casos de uso en el dominio de la solución. A continuación se muestran como ejemplo los Diagramas de Clases de Diseño definidos en la tesis de Análisis y Diseño titulada: "Análisis y Diseño del Sistema Integral de Documentación e Información Judicial para el Tribunal Supremo Popular" de los autores: Anais Louit Moreno y Yordanka Carralero Domínguez, la cual es un artefacto de entrada al flujo de trabajo de implementación:

- Diagrama de clases del Caso de Uso Gestionar Boleta Sala Lectura

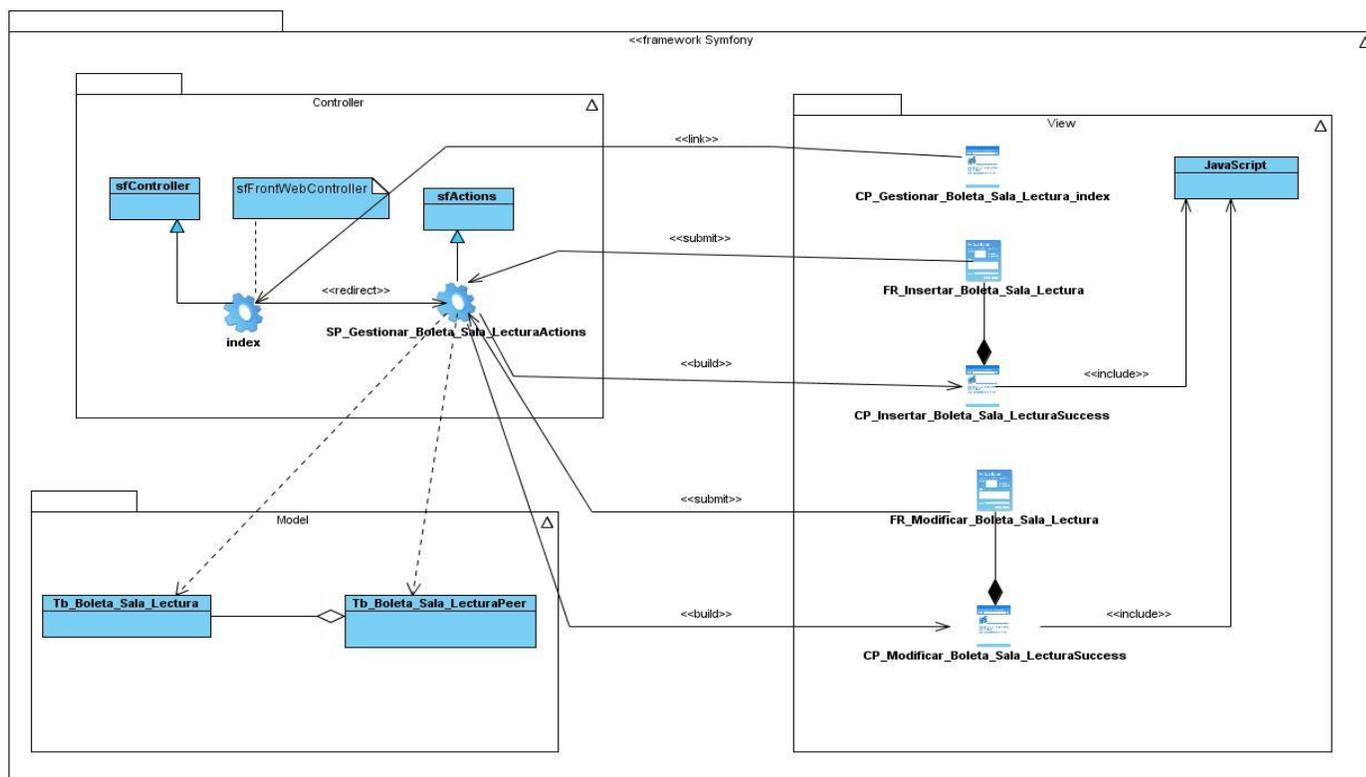


Ilustración 4. Caso de Uso Gestionar Boleta de Sala de Lectura.

2.1.6 Modelo de Datos

Un modelo de datos describe la representación física y lógica de los datos constantes utilizados por la aplicación. Estará compuesto por las entidades que pasarán a ser las tablas de la base de datos que será utilizada por el sistema en la implementación.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

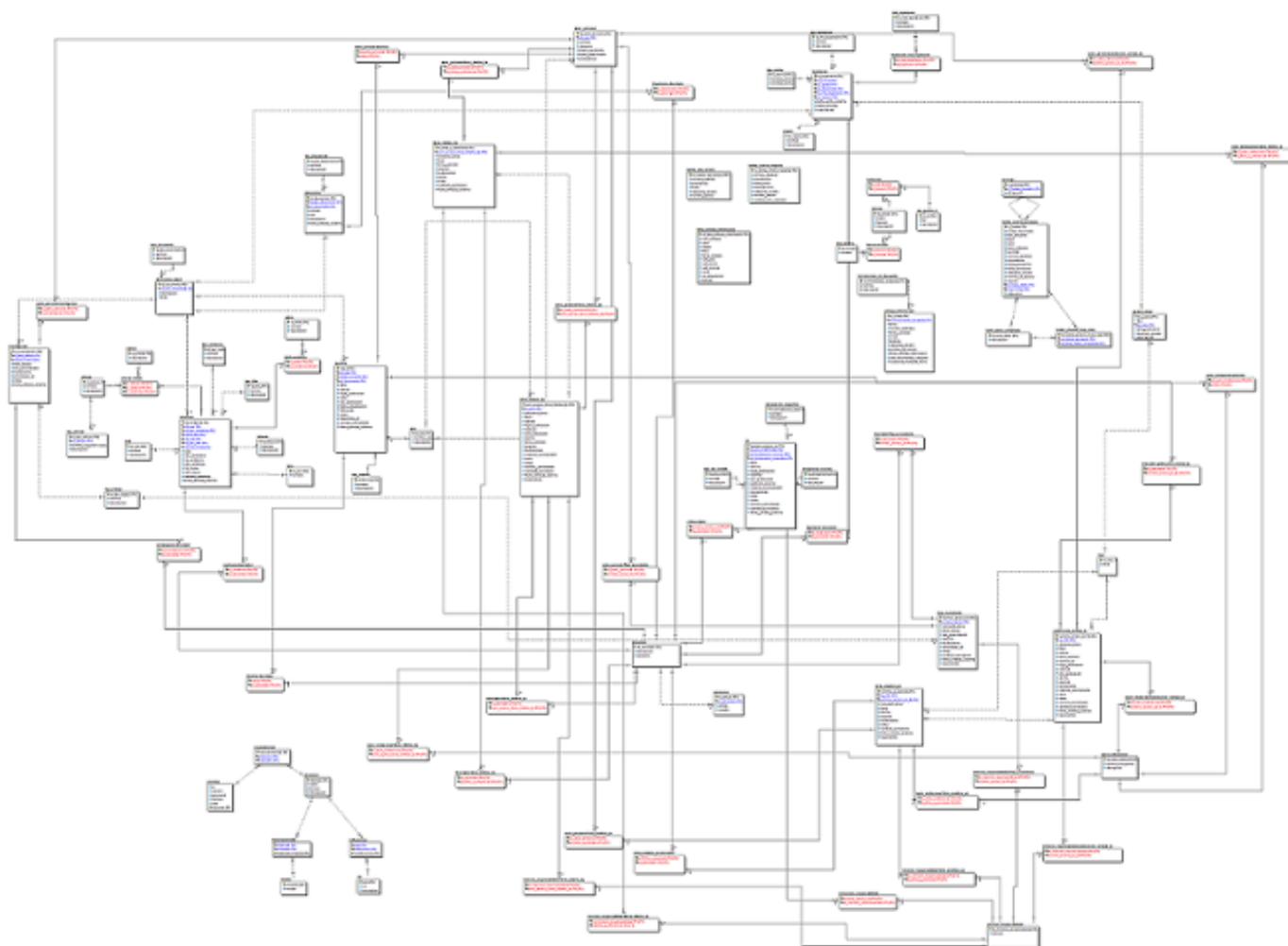


Ilustración 5. Modelo de Datos del Sistema.

La BD del SIDIJ está constituida por un total de 86 tablas incluyendo las tablas generadas por relaciones de muchos a muchos y 393 atributos.

Para la realización de ésta primera versión del proyecto se trabajó principalmente con las tablas relacionadas a usuarios para la administración, descriptores como también boletas para la gestión de las mismas con documentos de tipo referencial para la inserción y búsquedas específicamente siendo las tablas más usadas. A continuación se expone un ejemplo de la tabla publicaciones seriadas una de las más complejas por la cantidad de relaciones y atributos que posee.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

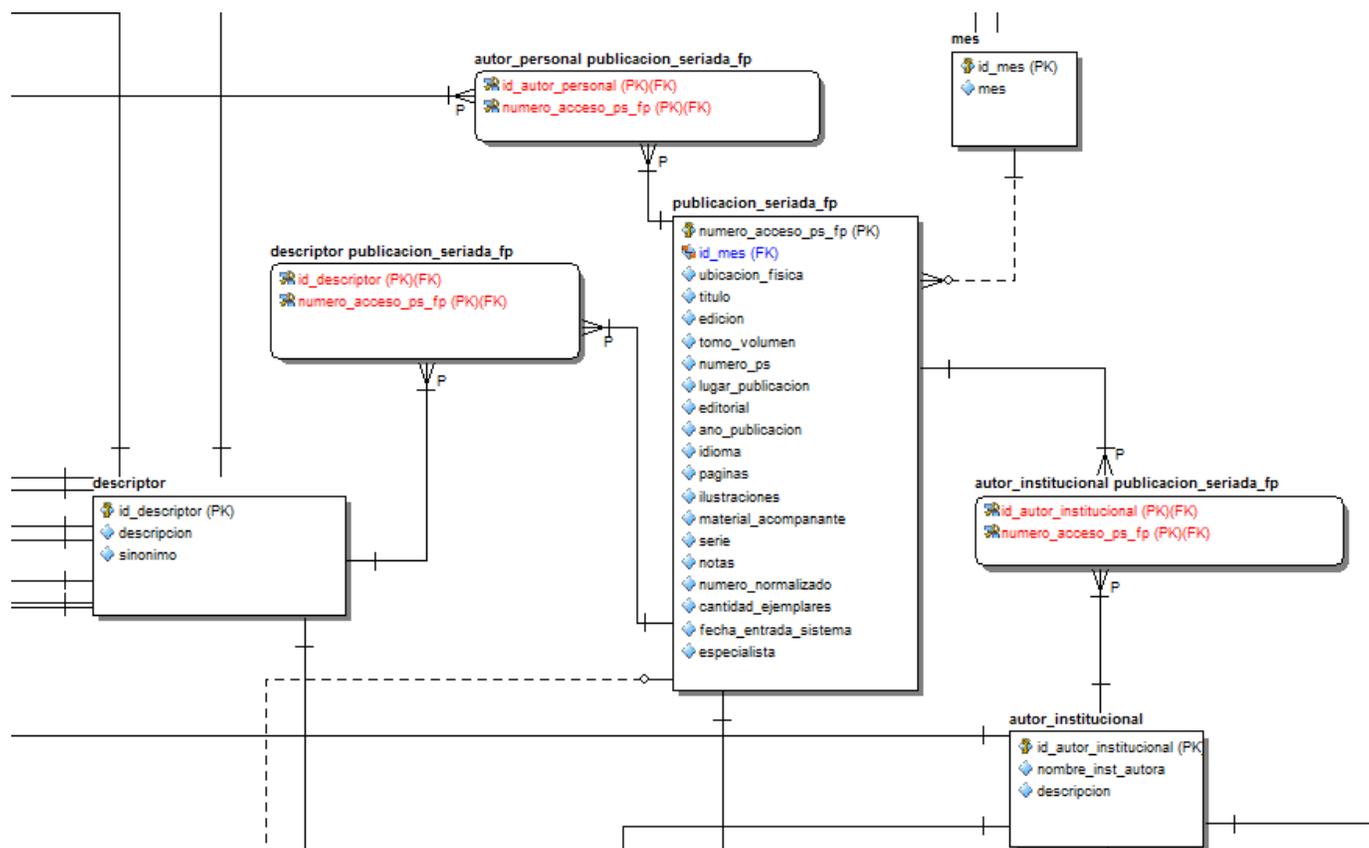


Ilustración 6. Tabla Publicación Seriada y sus relaciones.

Para un estudio y análisis más profundo del modelo, diseño y migración de datos realizado, referirse a la tesis de los autores: Yamila Zamora Mena y Eugenio Ávila Osoria titulada: “Proceso de migración de datos para el Sistema Integral de Documentación e Información Judicial (SIDIJ) del Tribunal Supremo Popular” la cual constituye un artefacto de entrada al flujo de implementación.

2.2 Estándares de Codificación

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad constituye un factor fundamental en el desarrollo del software.

Todos los nombres de acciones deben estar en la nomenclatura CamelCase, la cual consiste en escribir frases o palabras compuestas eliminando los espacios intermedios y poniendo en minúscula la

Capítulo 2: Descripción y Análisis de la Solución Propuesta

primera letra y en mayúscula las demás primeras letras de cada palabra contigua, que inicia por la palabra `execute`, el nombre de las clases expresado en esa misma notación, además del uso “Peer” como sufijo en el caso de las clases de abstracción a datos. Los nombres de las variables deben expresar claramente el contenido de la misma.

- **Nomenclatura de las clases**

Los nombres de las clases son escritos iniciando la palabra con minúscula, en caso de que exista más de una palabra estas comienzan en mayúscula sin utilizar caracteres especiales entre una palabra y otra. Ejemplo:

```
class sidijActions extends sfActions
```

- **Nomenclatura según el tipo de clases de Symfony**

- `sfActions` es la clase del controlador. Se encarga de decodificar la petición y transferirla a la acción correspondiente.
- `sfRequest` almacena todos los elementos que forman la petición (parámetros, cookies³⁸ y cabeceras).
- `sfResponse` contiene las cabeceras de la respuesta y los contenidos. El contenido de este objeto se transforma en la respuesta HTML que se envía al usuario.

- **Nomenclatura de las funciones**

El nombre a emplear para las funciones que están en las clases de acceso a datos Peer es descriptivo y comienza siempre por `get`, une las palabras y están en mayúscula.

Ejemplo:

```
public static function getOrganos()
```

- **Nomenclatura de las variables**

La vía a emplear para nombrar las variables es escribir siempre en minúscula la primera palabra y si es compuesto el nombre las siguientes palabras comienzan con mayúscula.

Ejemplo:

```
$tipoTrabajo = $this->getRequestParameter('tipoTrabajoInsFA');
```

- **Nomenclatura de las constantes**

Se declaran de igual manera que las variables.

³⁸ Ver glosario de términos.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

- **Nomenclatura de los atributos**

El nombre a emplear para los atributos los declara el Symfony por defecto.

2.3 Arquitectura

La arquitectura está basada en el estilo MVC que separa el código del programa en tres capas: la lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador.

Symfony simplifica este proceso: toma lo mejor del estilo MVC y lo implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo. En primer lugar, el controlador frontal y el layout son comunes para todas las acciones de la aplicación. Se pueden tener varios controladores y varios layouts, pero solamente es obligatorio tener uno de cada uno. El controlador frontal es un componente que sólo tiene código relativo al MVC, por lo que no es necesario crear uno, al generarlo Symfony de forma automática.

Symfony también hace que se generen las clases de la capa del modelo, en función de la estructura de datos de la aplicación. La librería Propel se encarga de esta generación automática, creando la estructura básica de las clases y generando el código necesario. Cuando Propel encuentra restricciones de claves foráneas (o externas) o cuando encuentra datos de tipo fecha, crea métodos especiales para acceder y modificar esos datos, por lo que la manipulación de los datos es muy sencilla. La abstracción de la base de datos es completamente transparente para el programador, al realizarse mediante PDO³⁹. Así, si se cambia el SGBD en cualquier momento, no se debe reescribir ni una línea de código, por ser necesario solo modificar un parámetro en un archivo de configuración. (Potencier, 2007)

2.3.1 Estrategias de Integración

La aplicación está definida por 3 capas: capa de presentación, negocio y acceso a datos al posibilitar ésta arquitectura un trabajo más seguro y rápido. La integración vertical consta de cuatro nodos de integración, el que está situado entre la vista y el controlador, el que está entre el controlador y el modelo, el que vincula el modelo con el ORM ⁴⁰Propel y el que se encuentra entre Propel y la BD. Todo el código dentro de un mismo componente utiliza llamadas a métodos o eventos de forma

³⁹ PHP Data Objects.

⁴⁰ Object-Relational mapping, Mapeo Objeto Relacional.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

directa.

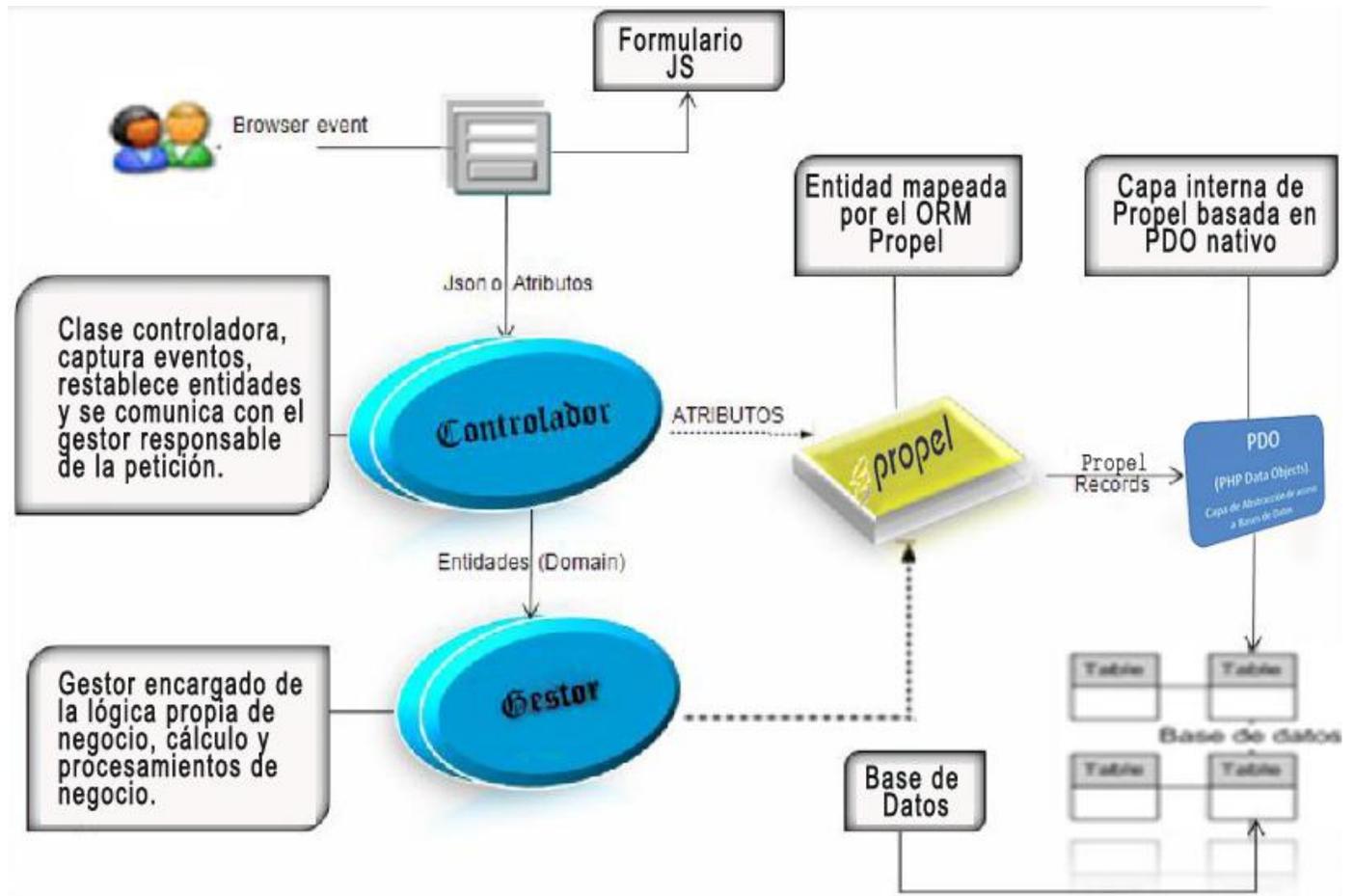


Ilustración 7. Colaboración entre clases definidas por la arquitectura.

2.3.2 Patrones GRASP⁴¹ implementados

- **Creador**

En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase acciones es creador de dichas entidades.

- **Experto**

Este es uno de los más utilizados, puesto que Propel es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

⁴¹ General Responsibility Assignment Software Patterns, Patrones de Asignación de Responsabilidades.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

- **Alta Cohesión**

Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo, la clase actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

- **Controlador**

Todas las peticiones Web son manejadas por un solo controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el de un módulo con la URL ⁴²entrada por el usuario.

- **Bajo Acoplamiento**

La clase actions hereda solamente de sfActions para lograr un bajo acoplamiento de clases.

2.3.3 Patrones GoF que utiliza Symfony

Instancia Única (Singleton): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el controlador frontal hay una llamada a sfContext::createInstance(). El objeto sfContext contiene referencias a todos los objetos internos de Symfony como la petición, la respuesta, el usuario, entre otros. Debido a que sfContext actúa como Singleton, se puede hacer uso de la instrucción sfContext::getInstance() para obtenerlo en cualquier punto de la aplicación y por tanto, para tener acceso a los objetos internos de Symfony (Potencier, 2009).

Patrón Controlador Frontal (Front Controller): La aplicación de este patrón se manifiesta en la capa de controladores, donde existe una estructura de clases bien definida. Una de sus principales funciones es definir un solo punto de entrada para las peticiones de los usuarios, permitiendo un mejor control del flujo de eventos del sistema. La clase sfController se encarga de decodificar la petición y transferirla a la acción correspondiente.

Envoltorio (Decorator): Añade funcionalidad a una clase, dinámicamente. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el

⁴² Uniform Resource Locator, Localizador de Recurso Uniforme.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación de dicho patrón.

Objeto compuesto (Composite): Permite tratar objetos compuestos como si se tratase de uno simple. Sirve para construirlos a partir de otros más simples y similares entre sí, por la composición recursiva y a una estructura en forma de árbol. Esto simplifica el tratamiento de los objetos creados, porque al poseer todos ellos una interfaz común, se tratan todos de la misma manera. (Potencier, 2007)

2.4 Descripción de las Clases

2.4.1 Clases Controladoras

La clase controladora es responsable de ejecutar una determinada lógica en función de las acciones que se producen en una aplicación.

En este ejemplo se menciona y describen los métodos usados en la clase `action.php` para lograr insertar satisfactoriamente un nuevo documento de tipo Legislación.

Insertar Legislación	
Nombre: <code>sidijActions</code>	
Nombre	Descripción
<code>executeComboxNormas</code>	Se obtienen todos los tipos de norma existentes mediante el método <code>TipoNormaPeer::getNormas()</code> y se envían a la vista.
<code>executeComboxGacetaOficial</code>	Se obtienen todos los tipos de gacetas oficiles existentes mediante el método <code>GacetaOficialPeer::getGacetasOficiales()</code> y se envían a la vista.
<code>executeComboxMeses</code>	Se obtienen todos los meses mediante el la invocación del método <code>MesPeer::getMeses()</code> y se envían a la vista.
<code>executeComboxEspecialistas</code>	Se obtienen todos los usuarios del sistema mediante el la invocación del método <code>UsuariosPeer::getUsuarios()</code> y se envían a la vista.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

executeComboxDescriptores	Se obtienen todos los descriptores existentes mediante el la invocación del método DescriptorPeer::Descriptores() y se envían a la vista.
executeComboxOrganoEmisor	Se obtienen todos los órganos del sistema mediante el la invocación del método OrganoPeer::getOrganos() y se envían a la vista.
executeComboxMaterias	Se obtienen todos las materias mediante el la invocación del método OrganoPeer::getMaterias() y se envían a la vista.
executeInsertarLegislacion	En este método se obtienen todos los parámetros que vienen desde la vista y se inserta una nueva Legislación mediante la llamada al método LegislacionPeer::insertarLegislacion(\$parametros).

Tabla 2. Descripción de la Clase Controladora.

2.4.2 Clases del Modelo

Las clases entidad pertenecientes al modelo, manejan la información que poseen una larga vida y que a menudo son conceptos. Se encargan de modelar la información del sistema y el comportamiento asociado a una información.

En este ejemplo se mencionan las clases del modelo que tienen relación con la clase Legislación y que por tanto se hizo necesario el trabajo con ellas para obtener la información requerida.

Tipo de Clase: Entidad
Nombre
tb_tipo_norma
tb_gaceta_oficial
tb_mes
tb_usuarios

Capítulo 2: Descripción y Análisis de la Solución Propuesta

tb_descriptores
tb_organo
tb_materias
tb_legislacion

Tabla 3. Descripción de las clases Entidad.

2.4.3 Clases de la Interfaz

Permiten la interacción del usuario con el sistema mediante un formulario donde se le muestran los datos que deben ser introducidos por el mismo para realizar las diferentes acciones que serán manejadas en el controlador.

Insertar Legislación	
Nombre	Descripción
InsertarLegislacion.js	<p>Crea toda la estructura visual mediante la utilización del framework para interfaz de usuario Ext js.</p> <p>Esta vista permite la entrada de los datos necesarios para insertar una legislación. Se realizan validaciones de campos y se muestran mensajes de error y confirmación al hacer alguna acción que lo requiera.</p>

Tabla 4. Descripción de la Clase Insertar Legislación.

2.5 Propuesta de Solución

El usuario general o común, entendiéndose usuario no autenticado, tiene la posibilidad de navegar en el sistema y realizar funcionalidades como visualizar el listado de Descriptores, hacer búsquedas de Información tanto Digital como Referencial y autenticarse en el mismo.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

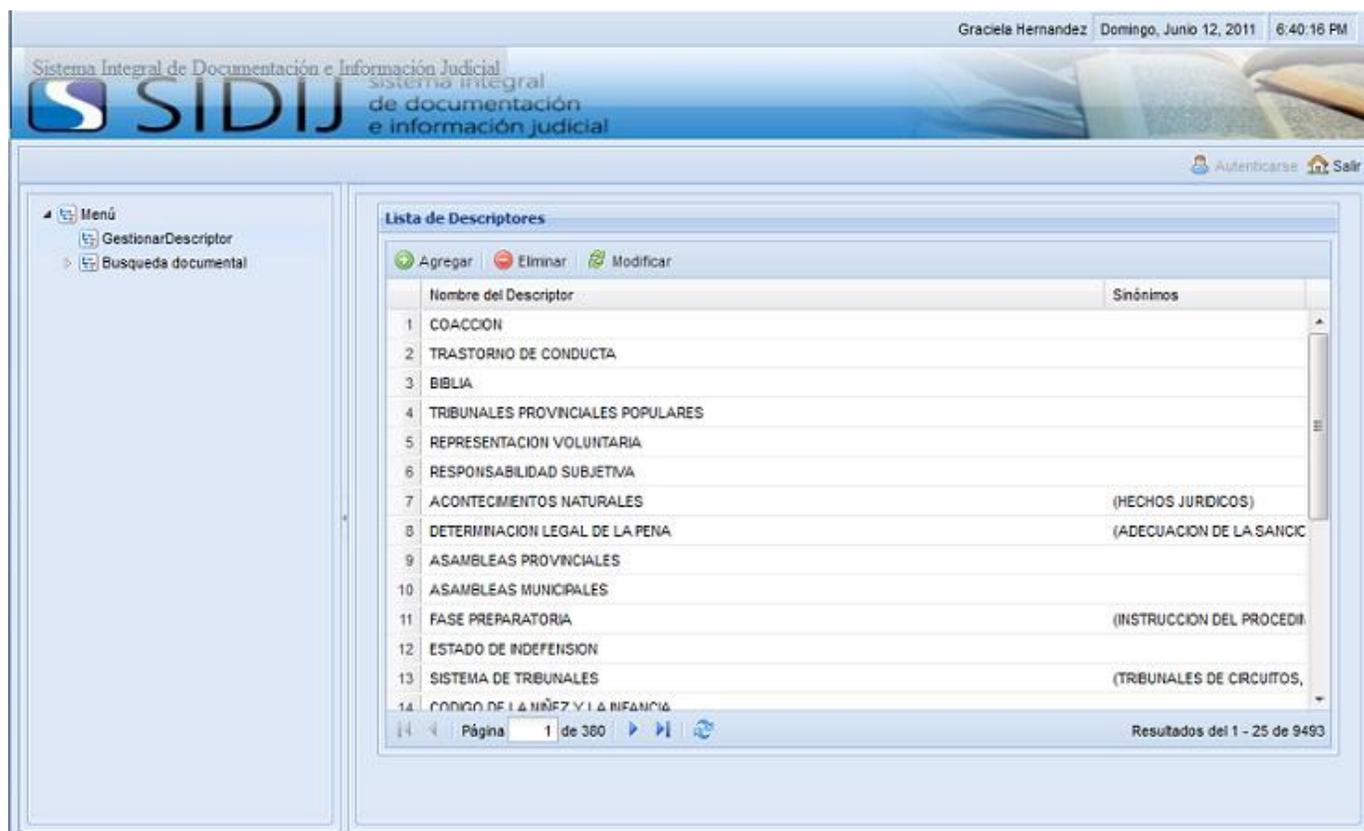


Ilustración 8. Interfaz para el usuario general.

Una vez autenticado el usuario se inicia la sesión y el sistema verificará los privilegios que posee, permitiéndole acceder solo a las funcionalidades que tenga acceso.

Si el usuario autenticado tiene la autoridad de gestionar parcialmente los documentos existentes en el sistema, es porque está asociado a un Catalogador. En este caso el sistema da la posibilidad de insertar, modificar y buscar un documento que puede ser digital o referencial. Según la opción que se seleccione se muestra un listado de todos los documentos de ese tipo existentes en la BD y se habilitan las opciones antes mencionadas.

Capítulo 2: Descripción y Análisis de la Solución Propuesta



Ilustración 9. Interfaz Gestionar Documento.

Si puede Gestionar los Descriptores, entonces está asociado a un Lingüista. En este caso el sistema muestra el listado de descriptores con las opciones de Agregar, Eliminar y Modificar un descriptor seleccionado.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

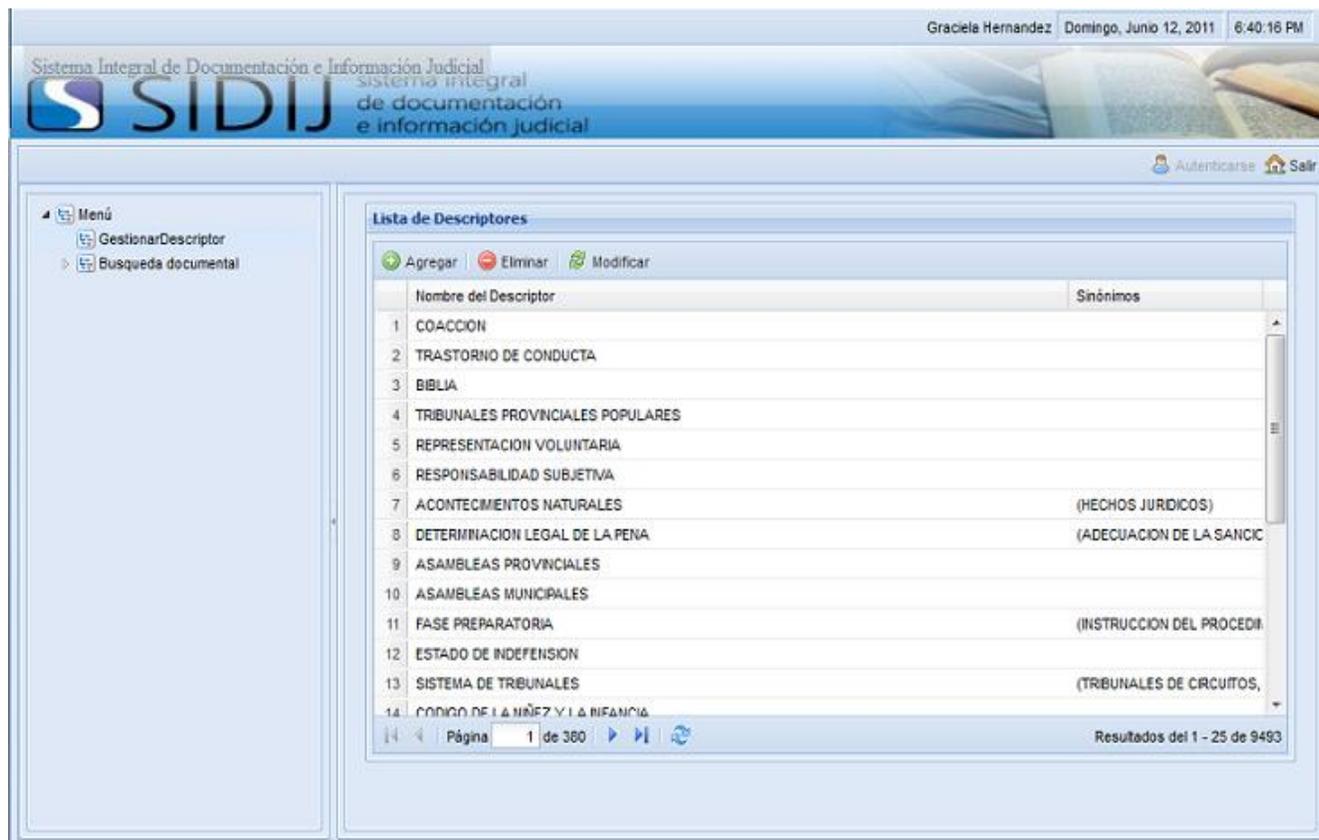


Ilustración 10. Interfaz para gestionar descriptores.

De igual manera si se habilita la Administración, el sistema brinda la posibilidad de Gestionar Usuarios, si se selecciona esta opción se muestra el listado de usuarios existentes y debajo la lista de funciones por usuario según el que se seleccione, además de adicionar un usuario y asociarle las funcionalidades a las cuales tendrá acceso posteriormente, además de modificarlos, eliminarlos y/o buscarlos.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

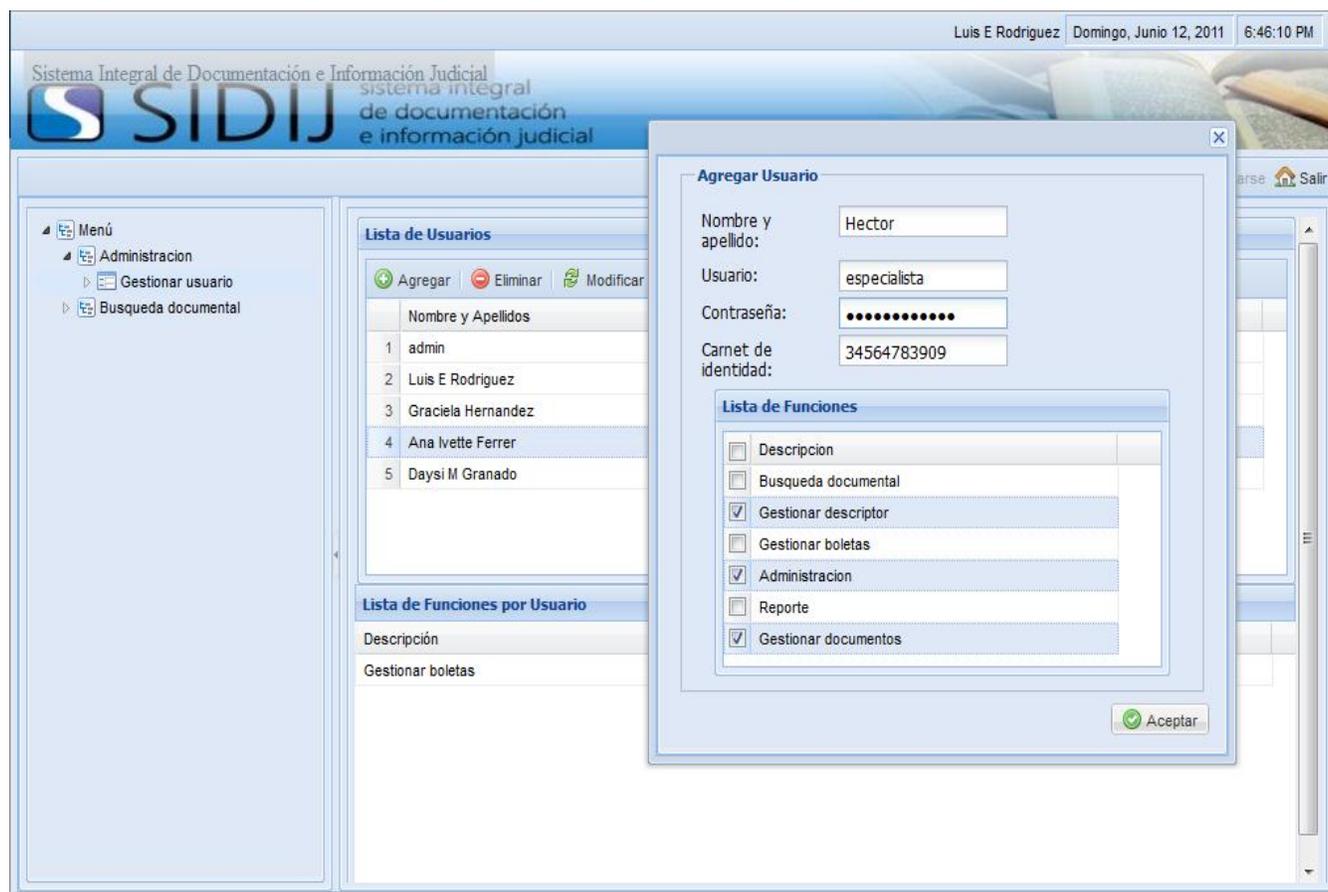


Ilustración 11. Interfaz para gestionar usuario.

En el caso de habilitarse la opción Gestionar Boletas, el sistema muestra todos los tipos de boletas existentes, al seleccionar un tipo de boleta determinada se muestra el listado con todas las boletas existentes de ese tipo y se le permite al usuario Agregar, Modificar y Buscar las mismas.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

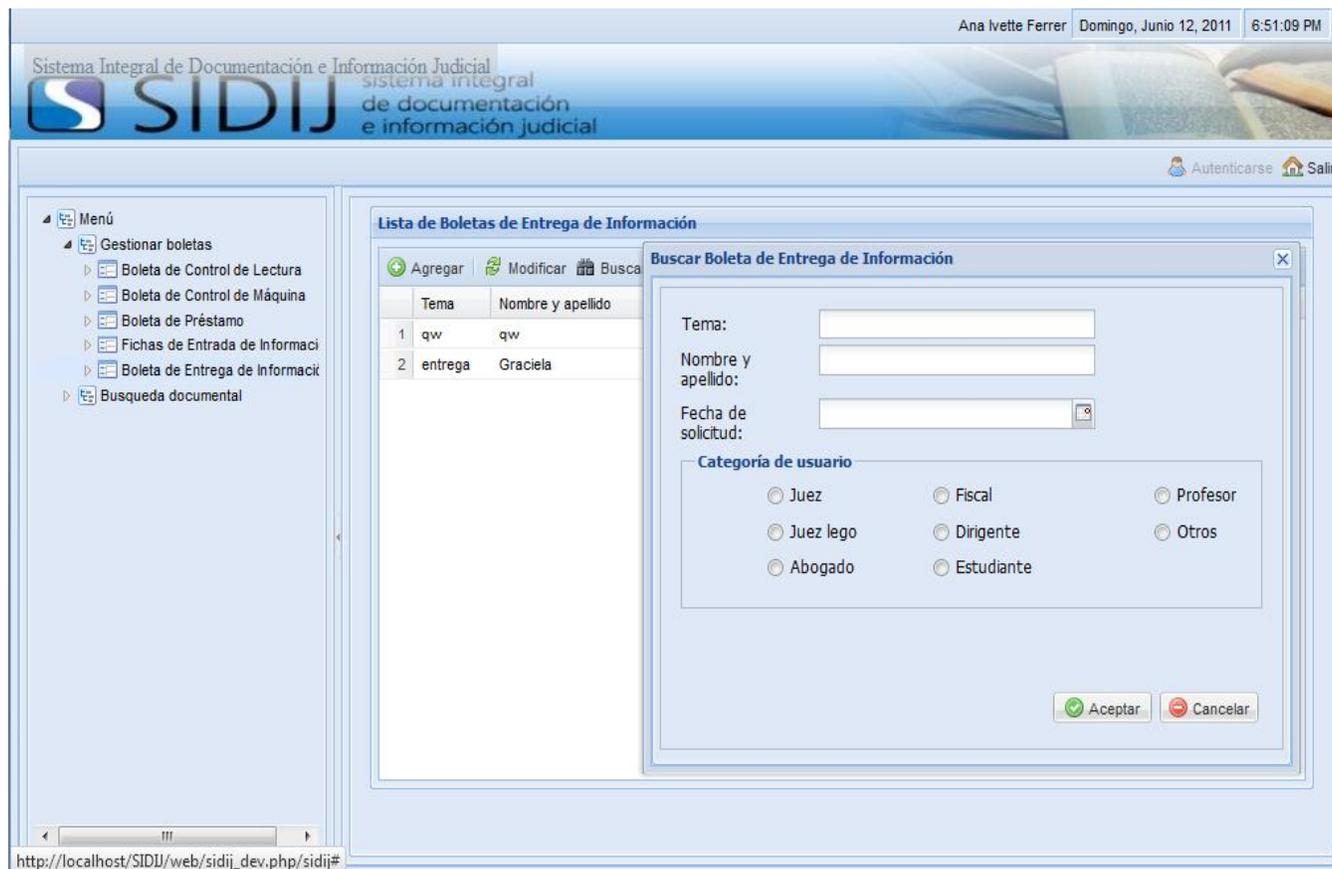


Ilustración 12. Interfaz para gestionar boletas.

2.5.1 Tratamiento de errores y validación en la entrada de datos

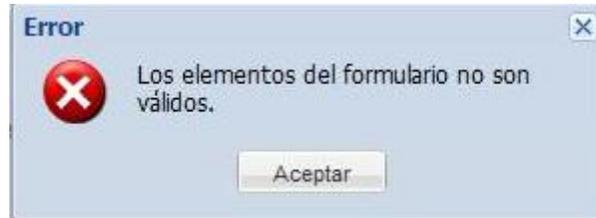
Se debe tomar en cuenta un adecuado tratamiento de errores para el funcionamiento del software, ejemplo de ello es cuando el sistema captura todas aquellas excepciones que son lanzadas y se les realiza un tratamiento para que no deje de funcionar.

Esta es la vía mediante la cual se da solución a las problemáticas de los lanzamientos de excepciones que surgen según las peticiones del usuario, al tomar los tipos de errores lanzados y mostrar de manera visible mensajes de confirmación al usuario, esto le informará de lo que está erróneo y como corregirlo.

A través de la interfaz Web el usuario asumirá un papel en la introducción de la información, para esto se contará con cuadros de opción, menú de selección y textos autocompletables lo cual facilitará la entrada de datos. Los datos que necesiten ser adicionados por el usuario se validarán mediante funciones o expresiones regulares que prueben que sean válidos y que el cuadro de texto no esté vacío si es

Capítulo 2: Descripción y Análisis de la Solución Propuesta

obligatorio llenarlo. Si hay un error en los datos le aparecerá un mensaje en pantalla al usuario indicándole el error.



2.5.2 Diagrama de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.

Algunas de sus características son:

- Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.
- Los nodos poseen relaciones que representan medios de comunicación entre ellos, tales como Internet, intranet, bus y similares.
- El modelo de despliegue puede describir diferentes configuraciones de red, incluidas las configuraciones para pruebas y para simulación. (James Rumbaugh, 1998)

La aplicación estará desplegada del siguiente modo:

- Nodo⁴³ PC⁴⁴ – Cliente: se encontrará el sistema operativo Windows o Linux y los navegadores web Mozilla Firefox o Internet Explorer mediante los cuales los clientes tendrán acceso al sistema.
- Nodo Servidor Web Apache: se encontrará todo lo relacionado con la aplicación. Estarán agrupados los archivos a través de los cuales el usuario logra acceder al sistema, además se encuentra contenida toda la información específica de cada registro, sus clases. También almacena la configuración general del proyecto, las clases y librerías externas, todo el código común de las aplicaciones del sistema y los plugins de instalación de la aplicación.
- Nodo Servidor BD PostgreSQL: serán almacenados los datos de la aplicación, se guardará el modelo de objetos del sistema.

⁴³ Ver glosario de términos.

⁴⁴ Personal Computer, computadora personal.

Capítulo 2: Descripción y Análisis de la Solución Propuesta

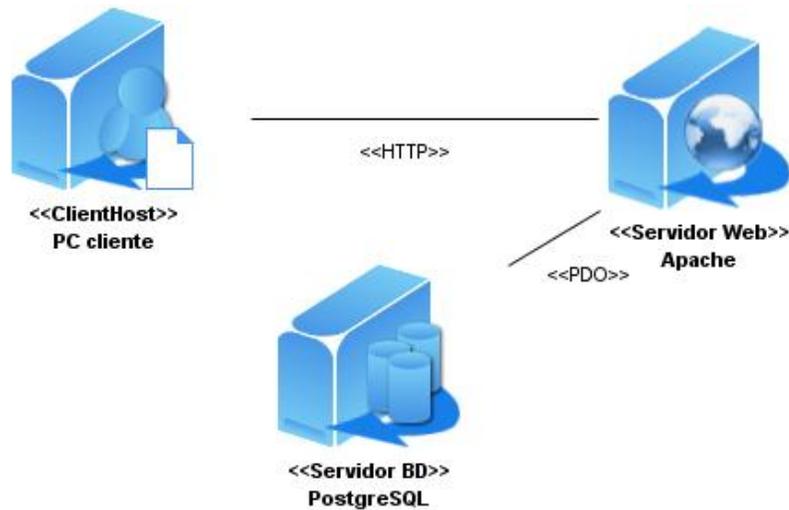


Ilustración 13. Diagrama de Despliegue.

Conclusiones del capítulo

En el capítulo se aprecia el diagrama de casos de uso del sistema y otros artefactos recibidos de la fase anterior, lo cual brinda una idea de la complejidad de la solución. Se explican los componentes que fueron utilizados para el desarrollo del sistema siempre teniéndose en cuenta la arquitectura definida. Todo lo anterior permitió la implementación de las interfaces necesarias mediante un proceso adecuado. La descripción de las clases permitió tener una visión para la puesta en práctica de los patrones usados. De manera general, se implementó la primera versión del SIDIJ para facilitar un sistema informático que acelere los procesos fundamentales del CENDIJ y cumpliéndose de ésta forma el objetivo del presente trabajo de diploma.

Capítulo 3: Validación de la Solución Propuesta

Capítulo 3: Validación de la Solución Propuesta

Introducción

Una de las características típicas del desarrollo de software basado en el ciclo de vida es la realización de controles periódicos, normalmente coincidiendo con los hitos del proyecto o la terminación de documentos. Estos controles pretenden una evaluación de la calidad de los productos generados (especificación de requisitos, documentos de diseño) para poder detectar posibles defectos cuanto antes. Sin embargo, todo sistema o aplicación, independientemente de estas revisiones, debe ser probado mediante su ejecución controlada antes de ser entregado al cliente. Estas ejecuciones o ensayos de funcionamiento, posteriores a la terminación del código del software, se denominan habitualmente pruebas. (Pressman, 2005)

Por ejemplo, verificar el código de un módulo significa comprobar si cumple lo marcado en la especificación de diseño donde se describe. La validación es el proceso de evaluación de un sistema o de uno de sus componentes durante el proceso de desarrollo o al final de éste para determinar si satisface los requisitos especificados. Validar una aplicación implica comprobar si satisface los requisitos marcados por el usuario. Se puede recurrir a la clásica explicación informal de Boehm de estos conceptos:

Verificación: ¿se está construyendo correctamente el producto?

Validación: ¿se está construyendo el producto correcto?

3.1 Pruebas

En la etapa de prueba del software se crean una serie de casos de prueba que intentan encontrar errores en el software desarrollado. La prueba requiere que se descarten ideas preconcebidas sobre la calidad o corrección del mismo. Para la validación de los resultados que se obtendrán después de haber desarrollado el software, se ha decidido realizar las pruebas de Caja Negra y Caja Blanca, al asegurar así la completitud de los requisitos funcionales, parte fundamental en el desarrollo de un software.

Según Pressman, *“las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente”* (Pressman, 2005)

Capítulo 3: Validación de la Solución Propuesta

Objetivos de la prueba

- Brindar un mayor nivel de confiabilidad en los productos que se van generando.
- Detectar fallas o errores.
- Aumentar la calidad del producto final. La prueba no puede asegurar la ausencia de errores, sólo puede demostrar si existen defectos en el software. (Yaimí Márquez Alpízar, 2007)

Para la validación de los resultados que se obtendrán después de haber desarrollado el software, la directiva del proyecto ha decidido realizar las pruebas de Caja Negra y Caja Blanca y así se asegurará la completitud de los requisitos, parte fundamental en el desarrollo de un software.

3.1.1 Pruebas de Caja Blanca

La prueba de código se basa en examinar la lógica del programa. Para ejecutarlos se desarrollan casos de prueba que produzcan la ejecución de cada posible ruta del programa o módulo, considerándose una ruta como una combinación específica de condiciones manejadas por un programa. (Yaimí Márquez Alpízar, 2007)

Las pruebas de Caja Blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas.

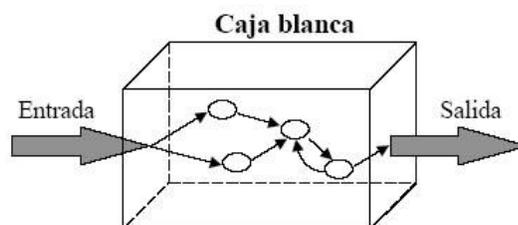


Ilustración 14. Prueba de Caja Blanca.

Tipos de prueba de caja blanca:

- Prueba de Condición: Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.
- Prueba de Flujo de Datos: Se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.

Capítulo 3: Validación de la Solución Propuesta

- Prueba de Bucles: Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.

- Prueba del Camino Básico: 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 grupo 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. (Yaimí Márquez Alpízar, 2007)

Los pasos que se siguen para aplicar esta técnica son:

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

Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecutan por lo menos una vez cada sentencia del programa. Para aplicar la técnica del camino básico se debe introducir la notación para la representación del flujo de control, éste puede representarse por un grafo de flujo en el cual:

- Cada nodo del grafo corresponde a una o más sentencias de código fuente.
- Todo segmento de código de cualquier programa se puede traducir a un grafo de flujo.
- Se calcula la complejidad ciclomática del grafo.

Para construir el grafo se debe tener en cuenta la notación para las instrucciones.

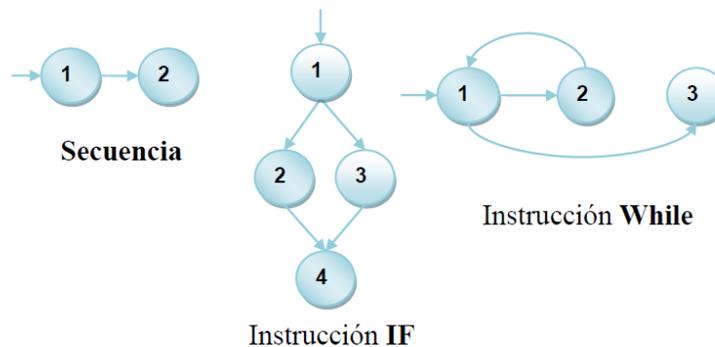


Ilustración 15. Notación para las Instrucciones de secuencia, if y while.

Para realizar la prueba de caja blanca específicamente la prueba del camino básico es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar. A continuación se

Capítulo 3: Validación de la Solución Propuesta

enumeran las sentencias de código del procedimiento realizado sobre el método insertarUsuario(\$nombre, \$pass, \$ci, \$idfunciones, \$user) el cual se encarga de adicionar un usuario a la BD y cargar del mismo las funciones a las que tiene acceso.

```
public static function insertarUsuario($nombre, $pass, $ci, $idfunciones, $user)
{
    $funciones = FuncionesPeer::getFuncionesIDfuncion($idfunciones); (1)
    $bandera = false; (1)
    if (!empty($idfunciones) || !empty($nombre) || !empty($pass) || !empty($user)) (2)
    {
        $usuario = new Usuarios(); (3)
        $usuario->setNombre($nombre); (3)
        $usuario->setPassword($pass); (3)
        $usuario->setCi($ci); (3)
        $usuario->setUsuario($user); (3)
        $usuario->save(); (3)
        $idusuario = $usuario->getIdusuario(); (3)
        foreach ($idfunciones as $idfuncion) (4)
        {
            $usuariofunciones = new Usuariofuncion(); (5)
            $usuariofunciones->setIdfuncion($idfuncion); (5)
            $usuariofunciones->setIdusuario($idusuario); (5)
            $usuariofunciones->save(); (5)
        }
        $bandera = true; (6)
    }
    return $bandera; (7)
}
```

Ilustración 16. Algoritmo InsertarUsuario().

Componentes del grafo de flujo

Nodos: son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

Aristas: son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.

Capítulo 3: Validación de la Solución Propuesta

Regiones: son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento.

En la siguiente figura se muestra el grafo de flujo asociado al código anterior.

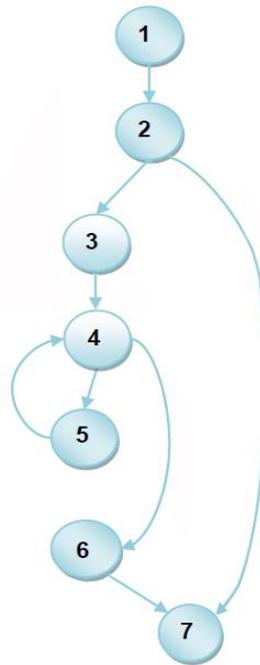


Ilustración 17. Grafo de flujo asociado al algoritmo insertarUsuario()

Cálculo de la complejidad ciclomática a partir de un segmento de código

La complejidad ciclomática es una métrica de software extremadamente útil pues proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite mínimo para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. Para efectuar el cálculo de la complejidad ciclomática del código es necesario tener varios parámetros como son la cantidad total de aristas del grafo, cantidad total de nodos para la siguiente fórmula:

$$V(G) = (A - N) + 2$$

$$V(G) = (8 - 7) + 2$$

$V(G) = 3$ Siendo "A" la cantidad total de aristas y "N" la cantidad total de nodos. Se puede usar también:

Capítulo 3: Validación de la Solución Propuesta

$$V(G) = P + 1$$

$$V(G) = 2 + 1$$

$V(G) = 3$ Siendo “P” la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = R$$

$V(G) = 3$ Siendo “R” la cantidad total de regiones.

Para cada fórmula “V(G)” representa el valor del cálculo. El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es de 3, lo que significa que existen 3 posibles caminos por donde el flujo puede circular, por lo que se deben realizar al menos 4 casos de pruebas para el procedimiento tratado. Seguidamente es necesario representar los caminos básicos por los que puede recorrer el flujo. En estas representaciones se subrayan los elementos de cada camino que los hacen independientes a los demás

En la siguiente tabla se muestran los caminos básicos.

Número	Caminos básicos
1	1-2-3-4-6-7
2	1-2-3-4-5-6-7
3	1-2-7

Tabla 5. Tabla de los caminos recorridos.

Después de haber extraído los caminos básicos del flujo se procede a ejecutar los casos de pruebas para este procedimiento, es necesario realizar al menos un caso de prueba por cada camino básico. Para realizarlos es preciso cumplir con las siguientes exigencias:

- Descripción: Se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.
- Condición de ejecución: Se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.
- Entrada: Se muestran los parámetros que entran al procedimiento.
- Resultados Esperados: Se expone el resultado que se espera devuelva el procedimiento.

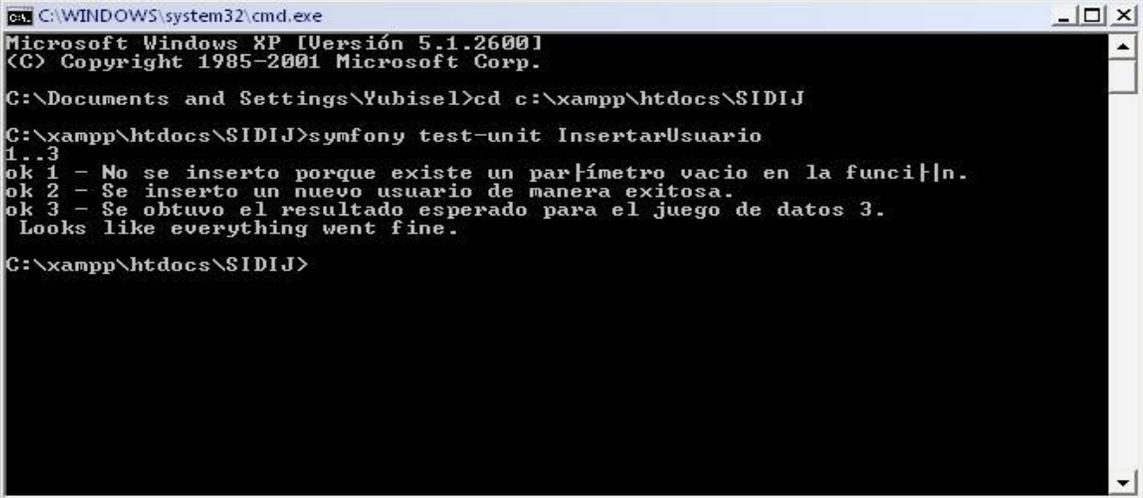
Capítulo 3: Validación de la Solución Propuesta

Lime framework para pruebas en Symfony

En PHP existen muchos framework para crear pruebas unitarias y Symfony a su vez incluye su propio framework llamado Lime. (Potencier, 2007) Lime proporciona soporte para la realización de pruebas unitarias, alguna de las ventajas que posee son:

- Ejecuta los archivos de prueba en un entorno independiente para evitar interferencias entre las diferentes pruebas.
- Las pruebas de Lime son fáciles de leer y sus resultados también lo son. Utiliza colores de forma clara e inteligente para distinguir la información más importante.
- Symfony utiliza Lime para sus propias pruebas y por lo que el código fuente de Symfony incluye muchos ejemplos reales de pruebas unitarias y funcionales.
- Está escrito con PHP, es muy rápido y está bien diseñado internamente. Consta únicamente de un archivo, llamado lime.php, y no tiene ninguna dependencia.

Se utiliza este framework para comparar los resultados obtenidos con los esperados por el desarrollador. Después de haberse introducido 3 casos de pruebas con valores incorrectos, vacíos y completos en el método insertarUsuario() del ejemplo anterior se obtuvo el siguiente resultado.



```
ca. C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Yubisel>cd c:\xampp\htdocs\SIDIJ

C:\xampp\htdocs\SIDIJ>symfony test-unit InsertarUsuario
1..3
ok 1 - No se inserto porque existe un parámetro vacío en la función.
ok 2 - Se inserto un nuevo usuario de manera exitosa.
ok 3 - Se obtuvo el resultado esperado para el juego de datos 3.
Looks like everything went fine.

C:\xampp\htdocs\SIDIJ>
```

Ilustración 18. Resultado de la prueba realizada.

Luego de aplicar los distintos casos de pruebas, se pudo comprobar que el flujo de trabajo de la función está correcto pues cumple con las condiciones necesarias que se habían planteado al devolver los resultados esperados inicialmente.

3.1.2 Pruebas de Caja Negra

En esta prueba se examinan las especificaciones que señalan lo que el programa debe hacer y como lo debe llevar a cabo. Para ejecutarlos se desarrollan casos de prueba reales para cada condición o combinación de condiciones y se analizan los resultados que arroja el sistema para cada uno de los casos. En esta estrategia se verifica el programa considerándolo una caja negra. La prueba no se hace en base al código, sino a la interfaz. (Yaimí Márquez Alpízar, 2007)

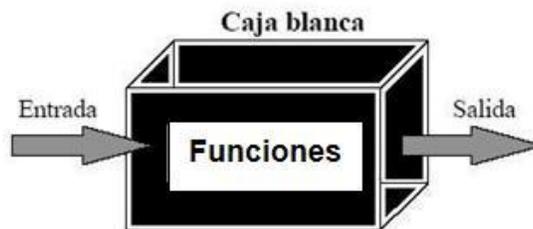


Ilustración 19. Prueba de Caja Negra.

Las pruebas de Caja Negra pretenden encontrar estos tipos de errores:

- Funciones incorrectas o ausentes.
- Errores en la interfaz.
- Errores en estructuras de datos o en accesos a BD.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Para aplicar la Prueba de Caja Negra se toma de ejemplo la descripción del caso de prueba para el caso de uso Agregar Ficha de Entrada de Información y se realiza el flujo varias veces recreando posibles escenarios. Para todos los escenarios, el sistema debe ser capaz de detectar el error y alertar apropiadamente al usuario para su posterior corrección, de lo contrario se toma como una no conformidad y se deberá rectificar en otra iteración de los casos de pruebas.

Diseño del Caso de prueba 1: Caso de Uso Agregar Ficha de Entrada de Información.(Segunda iteración)

Descripción General:

El Caso de Uso inicia cuando se necesita agregar al sistema una nueva ficha de entrada de información.

Condiciones de Ejecución:

- El usuario debe estar autenticado.

Capítulo 3: Validación de la Solución Propuesta

- El usuario debe escoger “Agregar” después que se visualice la lista de fichas de entradas de información existentes e introducir los campos requeridos.
- El usuario debe pulsar el botón “Insertar”.

En la siguiente tabla se describen los escenarios correspondientes a cada sección. Un escenario no es más que un flujo de los eventos que ocurran.

Tabla 6. Secciones a probar en el CU_ Agregar Ficha de Entrada de Información.

Nombre de la Sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1. Agregar ficha	EC 1.1 Agregar una ficha nueva.	Se agrega una ficha nueva en el sistema.
	EC 1.2 Seleccionar opción Cerrar.	Mediante este escenario no se realiza ningún cambio, se cierra la ventana y se vuelve a la página Gestionar Ficha de Entrada de Información.
	EC 1.3 Seleccionar Opción Cancelar.	Mediante este escenario se limpian los campos de la ventana.
	EC 1.4 Existen datos incompletos.	Mediante este escenario el sistema muestra un mensaje informando que los datos del formulario no son válidos.
	EC 1.5 Existen datos incorrectos.	Mediante este escenario el sistema muestra un mensaje de información indicando que los datos introducidos no son válidos.
SC 2. Buscar ficha.	EC 2.1. Busca una ficha existente.	Mediante este escenario se busca una ficha.
	EC 2.2 Seleccionar opción Cerrar.	Mediante este escenario no se realiza ningún cambio, se cierra la ventana y se vuelve a la página Gestionar Ficha de Entrada de Información.
	EC 1.3 Seleccionar Opción Cancelar.	Mediante este escenario se limpian los campos de la ventana.
SC 3. Modificar ficha.	EC 3.1. Modificar una ficha.	Mediante este escenario se modifican los datos de la ficha seleccionada.
	EC 3.2. Seleccionar opción Cerrar.	Mediante este escenario no se realiza ningún cambio, se cierra la ventana y se vuelve a la página Gestionar usuario.
	EC 3.3. Existen datos incompletos.	Mediante este escenario el sistema muestra un mensaje informando que los datos del formulario no son válidos.
	EC 3.4 Existen datos incorrectos.	Mediante este escenario el sistema muestra un mensaje de información

Capítulo 3: Validación de la Solución Propuesta

		indicando que los datos introducidos no son válidos.
--	--	--

En la siguiente tabla se listan todos los campos asociados a un escenario determinado, describiendo por cada uno de ellos las características que lo definan.

Tabla 7. Descripción de variables. SC 1. Agregar ficha.

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	No de entrada	Campo de texto	no	El campo solo permite introducir números.
2	Autor	Campo de texto	no	El campo puede contener letras, números y caracteres especiales.
3	Origen	Lista desplegable	no	El campo sólo permite letras.
4	Título	Campo de texto	no	El campo sólo permite introducir letras y números.
5	Fecha de entrada	Campo de selección	no	El campo sólo permite introducir números respetando el formato AAAA/MM/DD.
6	Cantidad de ejemplares	Campo de texto	no	El campo sólo permite introducir números.
7	Costo	Campo de texto	no	El campo sólo permite introducir números y caracteres como (,y.) para números floats.
8	Vía de adquisición	Campo de texto	no	El campo sólo permite letras y números.
9	Formato	Lista desplegable	no	El campo sólo permite los valores que están en la lista desplegable.

Para documentar los casos de prueba claramente, se utiliza un formato de matriz como en la siguiente tabla. La primera columna contiene la identificación del caso de prueba, la segunda tiene una breve descripción del mismo, incluyendo el escenario que es probado y el resto de columnas menos la última, contienen los elementos de datos que serán utilizados para poner las pruebas en ejecución. La última columna contiene una descripción de la salida prevista del caso de prueba.

Las celdas de la tabla contienen **V**, **I**, o **N/A**. El valor V indica válido, el I, inválido y N/A indica que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante.

Capítulo 3: Validación de la Solución Propuesta

Tabla 8. Matriz de Datos SC 1. Agregar ficha.

Escenario	Respuesta esperada	No de entrada	Autor	Origen	Título	Fecha de entrada	Cantidad de ejemplares	Costo	Vía de adquisición	Formato	Resultado de la prueba
EC 1.1 Agregar una ficha nueva.	Se agrega una ficha nueva en el sistema.	V	V	V	V	V	V	V	V	V	Satisfactorio
EC 1.2 Seleccionar opción Cerrar	Se cierra la ventana y se vuelve a la página Gestionar ficha de entrada de información.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Satisfactorio
EC 1.3 Seleccionar Opción Cancelar	Se limpian los campos de la ventana.	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Satisfactorio
EC 1.4 Existen datos incompletos (*)	El sistema muestra un mensaje informando que los elementos del formulario no son válidos.	I	V	V	V	V	V	V	V	V	Satisfactorio
EC 1.5. Existen datos incorrectos. (*)	El sistema muestra un mensaje informando que el descriptor debe contener sólo letras.	I	V	V	V	V	V	V	V	Satisfactorio	Satisfactorio

Nota: El símbolo de (*) significa todas las posibles variantes para esos campos.

Capítulo 3: Validación de la Solución Propuesta

La tabla 9 muestra un registro de las no conformidades encontradas durante todo el proceso.

Tabla 9. Registro de defectos y dificultades detectados.

Elemento	Número	No Conformidad	Aspecto Correspondiente	Etapas de Detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo de Desarrollo
-	-	-	-	-	-	-	-	-	-

Conclusiones del capítulo

La calidad del software está muy relacionada con las pruebas que se le realizan al mismo, de ahí la importancia que se le atribuye a todo el proceso de pruebas. En este capítulo se expone la secuencia de acciones necesarias para lograr el diseño e implantación de los casos de las pruebas de caja blanca usando la técnica del camino básico.

La construcción del modelo de prueba permitió la elaboración de los casos de prueba de caja negra consistentes en probar las funcionalidades por cada caso de uso. Una primera revisión arrojó como resultado 3 errores, 2 consistentes en errores en los mensajes de validación y uno en la validación de un campo, los cuales fueron corregidos en la segunda revisión donde todas las pruebas tuvieron resultados satisfactorios, demostrando que las funcionalidades de los casos de uso son correctas.

Conclusiones Generales

En este trabajo de diploma se realizó un análisis del marco teórico de la investigación, posibilitando el sustento científico del mismo. De igual forma se detallaron las herramientas, tecnologías y las metodologías propuestas por el proyecto SIDIJ para desarrollar la implementación del sistema. La especificación de los requisitos funcionales, en conjunto con el análisis del diseño de clases y el modelo de datos del software posibilitaron la definición de las funcionalidades de la aplicación informática para cumplir los objetivos planteados.

Con el desarrollo de este trabajo se implementó el SIDIJ para agilizar los procesos de gestión y prestación de servicios que se realizan en el CENDIJ, validado correctamente a través de pruebas de software de Caja Blanca y Caja Negra.

Recomendaciones

Las recomendaciones propuestas para la continuidad del presente trabajo son:

- Realizar una segunda versión del sistema.
- Consultar este documento como material de estudio, guía y apoyo para el posterior desarrollo de la segunda versión del sistema y lograr así su despliegue.

Bibliografía

- Alvarez, Miguel Angel. 2009.** Biblioteca Digital del Gobierno Bolivariano de Venezuela. *Manual de CodeIgniter*. [En línea] 2009. <http://www.biblioteca-digital.net.ve/wordpress/wp-content/uploads/2010/07/manual-codeigniter.pdf>.
- **2010.** desarrolloweb. *CodeIgniter*. [En línea] 2010. <http://www.desarrolloweb.com/articulos/codeigniter.html>.
- Carrero, Angel. 2011.** Programación en castellano. *Distintos Framework para PHP*. [En línea] 2011. http://www.programacion.com/articulo/distintos_framework_para_php_379.
- cibernetia. 2011.** cibernetia. *Conceptos básicos del servidor web*. [En línea] 2011. http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php.
- Cobas, Walberto Díaz. 2011.** Revista Tino. *ExtJS (Análisis)*. [En línea] 2011. http://revista.jovenclub.cu/index.php?option=com_content&task=view&id=634&Itemid=100.
- Collector. 2004.** Collector. *Sistema Gestor de base de datos SGBD*. [En línea] 2004. http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
- Corporate and Sales Headquarters, North America. 2011.** <http://www.magicdraw.com/>. *Magicdraw*. [En línea] 2011. <http://www.magicdraw.com/>.
- Docencia Universitaria. 2010.** Docencia Universitaria. *Sitios Web Dinámicos contruidos con Software Libre*. [En línea] 2010. http://docenciauniversitaria.cl/index.php?option=com_k2&view=item&id=97:sitios-web-din%C3%A1micos-contruidos-con-software-libre&Itemid=6.
- DocExpert. 2011.** LegisLaw. [En línea] 2011. <http://www.legislaw.com.ar/servipa/soft.htm>.
- Doctor, Lex. 2011.** Lex Doctor. [En línea] 2011. <http://www.lex-doctor.com/>. 5.
- Duarte, Maicol. 2008.** Seminario de Programación. *NetBeans IDE*. [En línea] 2008. <http://seminprogramacion.blogspot.com/2008/06/netbeans-ide.html>.
- EcuRed. 2011.** EcuRed. *Herramientas CASE: Visual Paradigm*. [En línea] 2011. http://www.ecured.cu/index.php/CASE#Visual_Paradigm.
- Edition, Rational Rose Enterprise. 2011.** EcuRed. *Rational Rose Enterprise Edition*. [En línea] 2011. http://www.ecured.cu/index.php/Rational_Rose_Enterprise_Edition.
- Electrónica, Laboratorios. 2010.** Geocities.ws. *Anotaciones de RUP*. [En línea] 2010. <http://www.geocities.ws/gustsucc/Archivos/AnotacionesRUP.pdf>.
- Estrada., Jason Arturo Galvez. 2011.** Scribd. *NETBEANS 6.0 PRIMERA PARTE*. [En línea] 2011. <http://es.scribd.com/doc/24576936/netbeans-6-0manual-beta1>.
- Fowler, Martin. 2010.** *Refactoring: Improving the Design of Existing Code*. 2010.
- Gonzales, Benjamin. 2010.** Introducción a Zend Framework. [En línea] 2010. <http://manual.zfdes.com/es/introduction.installation.html>.
- González, Rubén y Pérez, Sergio. 2010.** <http://es.scribd.com>. *Introducci'on al Rational Rose*. [En línea] 2010. <http://es.scribd.com/doc/55623280/LESE-2-Introduccion-a-Rational-Rose>.
- Green, Samuel Swett. 1876.** *Personal relations between librarian and readers*. 1876. 1.
- Headquarters, Company. 2011.** <http://www.visual-paradigm.com/>. *Visual Paradigm*. [En línea] 2011. <http://www.visual-paradigm.com/>.
- Hofmann Hubert F, Franz Lehner. 2001.** *Requirements Engineering as a Succes Factor in Software Projects*. 2001.
- Identi. 2011.** Identi. *¿Qué es Apache?* [En línea] 2011. <http://www.identi.es/?topic=37883>.
- Ivy F. Hooks, Kristin A. Farry. 2001.** *Costumer Centered Products: Creating Successful Products Throuth Smart Requirements Managemend*. 2001.

- James Rumbaugh, Ivar Jacobson, Grady Booch. 1998.** *EL Lenguaje Unificado de Modelado. Manual de Referencia.* 1998.
- JSON. 2011.** JSON. *Introducción a JSON.* [En línea] 2011. <http://www.json.org/json-es.html>.
- Jurídica, Sistema Documental de Informática. 2011.** Corte Suprema de la Nación. República de Argentina. *Sistema Documental de Informática Jurídica.* [En línea] 2011. <http://www.csjn.gov.ar/>. 3.
- Jurídicos, Diccionario de Términos. 1999.** *Diccionario de Términos Jurídicos.* 1999.
- Jurisoft. 2011.** Sistemas de Informática Jurídica. [En línea] 2011. <http://www.jurisoft.es/ie/empresa.aspx?menu=1#>. 4.
- Kmaleon, Level. 2010.** Level Kmaleon. [En línea] 2010. <http://www.kmaleon.com/kamaleon.html>. 7.
- La Revista Informática, HTML. 2006.** La Revista Informática. *Lenguaje de Programación HTML.* [En línea] 2006. <http://www.larevistainformatica.com/Lenguaje-programacion-HTML.htm>.
- La Revista Informática, JavaScript. 2006.** La Revista Informática.com. *Lenguaje de Programación JavaScript.* [En línea] 2006. <http://www.larevistainformatica.com/JavaScript.htm>.
- Lanzillotta, Analía. 2004.** MasterMagazine. *Definición de Lenguaje de programación.* [En línea] 2004. <http://www.mastermagazine.info/termino/5560.php>.
- Legislaw. 1999.** Legislaw. *El Banco Jurídico Argentino.* [En línea] 1999. <http://www.legislaw.com.ar>.
- Lenguajes de Programación. 2009.** Lenguajes de Programación. *Programación Estructurada.* [En línea] 2009. <http://www.lenguajes-de-programacion.com/programacion-estructurada.shtml>.
- Lenguajes de Programación, Java. 2009.** Lenguajes de Programación. *Programación Java.* [En línea] 2009. <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
- M., Jesús J. Lárez. 2010.** Scribd. *Introducción al Lenguaje de Programación C.* [En línea] octubre de 2010. <http://es.scribd.com/doc/56071453/5/Entorno-de-Programacion-Integrado>.
- Martin, Robert C. 2002.** *Continuous Care Vs. Initial Design.* 2002.
- Martínez, Alejandro Martínez y Raúl. 2010.** Escuela Politécnica Superior de Albacete – Universidad de Castilla la Mancha. *Guía a Rational Unified Process.* [En línea] 2010. <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf>.
- Mary Beth Chrissis, Mike Konrad, Sandy Shrum. 2009.** *Guía para la Integración de Procesos y la Mejora de Productos.* 2009.
- Metodologías, Scribd. 2011.** Scribd. *Metodología de desarrollo de software.* [En línea] 2011. <http://es.scribd.com/doc/12983329/Metodologia-de-Desarrollo-de-Software>.
- Mordevre, Instituto de Educación Camp de. 2008.** Introducción a los lenguajes web. Programación en PHP y Bases de datos. *Programación en PHP y Bases de datos.* [En línea] 2008. http://www.iescamp.es/fileadmin/informatica/php_y_mysql.pdf.
- Objetos, Programación Orientada a. 2009.** Lenguajes de Programación. *Programación Orientada a Objetos.* [En línea] 2009. <http://www.lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>.
- Penadés, Patricio Letelier y M^a Carmen. 2006.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] 2006. <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
- PostgreSQL-es. 2010.** PostgreSQL-es. *Sobre PostgreSQL.* [En línea] 2010. http://www.postgresql.org.es/sobre_postgresql.
- Potencier, Fabien. 2009.** *El Tutorial Jobeet.* 2009.
- Potencier, François Zaninotto y Fabien. 2007.** *Symfony. La Guía definitiva.* 2007.
- Pressman, Roger S. 2005.** *Un Enfoque Práctico.* Mexico : Mcgraw-hill Mexico, 2005.
- Sclavos, Stratton. 2011.** Sencha. *Compatibilidad en los Navegadores.* [En línea] 2011. <http://www.sencha.com/products/extjs/>.

- Scribd CASE. 2011.** Scribd Qué es una Herramienta CASE. *Qué es una Herramienta CASE*. [En línea] 2011. http://es.scribd.com/doc/57334530/Herramienta-CASE#ad_unit=Doc_Sideboard_MediumRectangle_BTF_300x250&url=http%3A//es.scribd.com/doc/57334530/Herramienta-CASE%23ad_unit%3DDoc_Sideboard_MediumRectangle_BTF_300x250%26url%3Dhttp%253A//es.scribd.com/doc/573345.
- Scribd. 2010.** Scribd. [En línea] 2010. <http://es.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
- .** 2011. Scribd. *Conceptos básicos del servidor web*. [En línea] 2011. <http://es.scribd.com/doc/7826492/Conceptos-basicos-Del-Servidor-Web>.
- Silva, Abogado Luis Raciél Rodríguez. 2010.** *Los Sistemas de Gestión de Servicios Legales. Propuesta De Modelación de un Sistema para una Oficina de Asistencia Legal en Cuba*. 2010.
- Sommerville, Ian. 2005.** *Ingeniería de Software*. Alicante : s.n., 2005.
- Sowre. 2011.** Sowre. *Herramientas de desarrollo de software*. [En línea] 2011. <http://www.sowre.es/tecnologias-y-soluciones-sowre-negocios/entornos-de-desarrollo-herramientas-software>.
- Spirate. 2011.** Spirate. *Tutorial básico de CSS*. [En línea] 2011. <http://www.spirate.net/css-2/tutorial-basico-de-css/?wap2>.
- Symfony.es. 2009.** Symfony.es. *NetBeans ya incluye soporte para Symfony*. [En línea] 2009. <http://www.symfony.es/categoria/ide/>.
- Systems, Company Sparx. 2011.** Sparx Systems. *Enterprise Architect*. [En línea] 2011. <http://www.sparxsystems.com/products/ea/index.html>.
- Taboada, Ricardo Pelegrín. 2010.** Persona en Cuba. [En línea] 2010. <http://www.revistapersona.com.ar/Persona54/54Cuba.htm>.
- Teresita Rojas, Luis E. Mendoza, María A. Pérez. 2011.** EcuRed. *Indicadores organizacionales para comparación de herramientas CASE en Venezuela*. [En línea] 2011. http://www.lisi.usb.ve/publicaciones/05%20herramientas/herramientas_06.pdf.
- tu Función. 2007.** tu Función. *Los mejores IDEs para Php*. [En línea] 2007. <http://www.tufuncion.com/ide-php>.
- UML, Star. 2011.** Star UML. *Star UML*. [En línea] 2011. <http://staruml.sourceforge.net/en/>.
- Visbal, Doctora en Ciencias de la Información Sara M. Artilles. 2009.** Biblioteca Virtual en Salud. Cuba. *La gestión documental, de información y el conocimiento en la empresa. El caso de Cuba*. [En línea] 2009. http://bvs.sld.cu/revistas/aci/vol19_5_09/aci02509.htm.
- VP, EcuRed. 2011.** EcuRed. *Visual Paradigm*. [En línea] 2011. http://www.ecured.cu/index.php/Visual_Paradigm.
- 2009.** www.gestiopolis.com. *Estudio sobre la industria del Software a nivel mundial. Caracterización en América Latina y Cuba*. [En línea] 23 de 06 de 2009. <http://www.gestiopolis.com/administracion-estrategia/estudio-sobre-la-industria-del-software-en-america-latina.htm>.
- Yaimí Márquez Alpízar, Yenni Valdés Hechavarría. 2007.** *PROCEDIMIENTO GENERAL DE PRUEBAS DECAJA BLANCA APLICANDO LA TÉCNICA DEL CAMINO BÁSICO*. 2007.

Glosario de Términos

- **Nodo**

Un nodo es un elemento físico que existe en tiempo de ejecución y que representa un recurso computacional, que en general tiene al menos una memoria y a menudo capacidad de procesamiento.

- **Layout**

Es la ordenación y colocación de todos los elementos que componen una página web, es decir textos, imágenes, tablas y gráficos. También son elementos del layout los colores y el tipo de letra. A la hora de realizar los diseños hay que tener en cuenta que un layout claro permitirá una navegación mucho más fácil.

- **Framework**

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

- **Script**

Un script o como también se lo conoce, un archivo de órdenes o archivo de procesamiento por lotes, es un programa simple, que se almacena en un archivo de texto plano y cuyo uso fundamental resulta a la hora de tener que realizar diversas tareas como ser la combinación de componentes, la interacción con el usuario o con el sistema operativo en cuestión. Facilita la automatización de tareas a través de la creación de pequeñas utilidades.

- **Acrónimo Recursivo**

Un acrónimo recurrente o acrónimo recursivo es aquel que, a diferencia de los acrónimos comunes, no parece explicar del todo su significado completo. Mantiene la estructura de los acrónimos comunes en relación a que sus nombres nacen como resultado de la particular abreviatura de una frase, de la que se seleccionan sólo las iniciales de algunas palabras claves. Sin embargo, en un acrónimo recursivo, una de sus letras representa al propio acrónimo.

- **IDE**

Es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

- **Plataforma**

En informática, una plataforma es precisamente el principio, ya sea de hardware o software, sobre el cual un programa puede ejecutarse. Ejemplos típicos incluyen: arquitectura de hardware, sistema operativo, lenguajes de programación y sus librerías de tiempo de ejecución.

- **Memoria RAM**

La memoria RAM (Random Access Memory Module o memoria de acceso aleatorio) es un tipo de memoria que utilizan los ordenadores para almacenar los datos y programas a los que necesita tener un rápido acceso.

- **Licencia BSD**

La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution).

- **Aplicación Web**

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador web. Es una aplicación que se codifica en un lenguaje soportado por los navegadores web, en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener sin distribuir e instalar software a miles de usuarios potenciales.

- **Logs**

Son los archivos que registran toda la actividad de un servidor, aplicación o software.

- **jQuery**

Es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de plug-ins

y efectos visuales para la creación de aplicaciones web.

- **XML**

Permite describir el sentido o la semántica de los datos, pues separa el contenido de la presentación. Describe el contenido a través de etiquetas o marcas.

- **Plugin**

Programa que puede anexarse a otro para aumentar sus funcionalidades.

- **Applets**

Es un programa Java diseñado para ejecutarse en una página Web a través de un navegador que soporte Java. Todas las últimas versiones de Netscape o Microsoft Internet Explorer lo incluyen por defecto.

- **Cookies**

Porción de datos que un servidor de internet entrega al navegador de un usuario que le hizo alguna consulta para que el navegador los guarde. Suele tratarse de información sobre la conexión o los datos requeridos. De esta manera, el servidor puede saber lo que hizo el usuario durante su última visita y facilitarle la próxima.

- **ORM**

Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, valiéndose de un motor de persistencia.

- **URL**

Son las siglas de Localizador de Recurso Uniforme la dirección global de documentos y de otros recursos en la World Wide Web. La primera parte de la dirección indica qué protocolo utilizar, la segunda parte especifica la dirección IP o nombre de dominio donde se localiza el recurso.