

Universidad de las Ciencias Informáticas
Facultad 9



Trabajo de diploma para optar por el título de
Ingeniero en Informática

Título: Desarrollo de un componente de software para importar reportes desde documentos Excel a Sistemas Gestores de Bases de Datos en el Centro de Investigaciones del Petróleo (CEINPET).

Autores: Jackeline Ortiz Fidalgo

Fredy Hector Rios Morales.

Tutores: Ing. Alejandro Orgelio Hernández Cebrián.

Ing. Edgar A. George de Armas.

Co-Tutor: Ing. Yuniel Eliades Proenza Arias.

Ciudad de La Habana, Cuba.
Año del 52 Aniversario del Triunfo de la Revolución.
Junio de 2010.

DECLARATORIA DE AUTORÍA.

Declaramos ser autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y al Centro de Investigaciones del Petróleo (CEINPET) a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2010.

Jackeline Ortiz Fidalgo.

Fredy Hector Rios Morales.

Firma del Autor.

Firma del Autor.

Ing. Alejandro Orgelio Hernández Cebrián.

Ing. Edgar A. George de Armas.

Firma del Tutor

Firma del Tutor.



RESUMEN

El desarrollo de software basado en componentes hace necesario contar con herramientas adecuadas. El presente trabajo de diploma está enfocado a desarrollar un componente de software que procese documentos en formato Excel e importe la información contenida en los mismos hacia bases de datos en PostgreSQL, en aras de mejorar la gestión de la información en el Laboratorio de Lodo y Cemento.

Para el desarrollo del componente propuesto se analizaron las necesidades actuales en el Centro de Investigaciones del Petróleo (CEINPET), además se tuvieron en cuenta las metodologías y tecnologías actuales; utilizando como metodología de desarrollo de software Agile Unified Process (AUP), como lenguaje de modelado se utilizó el Lenguaje Unificado de Modelado (UML) y Visual Paradigm for UML Enterprise Edition como herramienta de ingeniería asistida por computadora (CASE), como lenguaje de programación PHP, como servidor web APACHE, jQuery como librería de JavaScript y la librería PHPEXCEL para el manejo de documentos Excel, como entorno de desarrollo integrado (IDE) NetBeans, como Sistema Gestor de Bases de Datos, PostgreSQL.

Se presentan además los principales artefactos generados durante el proceso de desarrollo, siguiendo la metodología seleccionada, que le dan solución al problema existente. Finalmente, se sugieren algunas recomendaciones para próximas versiones del mismo.

Palabras Claves: componente de software, CEINPET.

INDICE

INTRODUCCIÓN	4
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.	8
1.1 INTRODUCCIÓN.....	8
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	8
1.2.1 <i>Gestión de la información</i>	9
1.3 CONCEPTOS ASOCIADOS AL COMPONENTES DE SOFTWARE.	10
1.3.1 <i>Modelos de desarrollo por componentes.</i>	11
1.3.2 <i>Beneficios del Desarrollo de Software basado en Componentes.</i>	12
1.4 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES.	13
1.5 CONCLUSIONES.....	14
CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES.	15
2.1 INTRODUCCIÓN.....	15
2.2 METODOLOGÍAS EXISTENTES.....	15
2.2.1 <i>Metodologías ágiles</i>	15
2.2.1.1 Agile Unified Process (AUP).	16
2.2.1.2 Extreme Programming (XP).	18
2.2.2 <i>Metodologías robustas.</i>	18
2.2.2.1 Rational Unified Process (RUP).	18
2.2.2.2 Microsoft Solutions Framework (MSF).	19
2.3 SELECCIÓN DE LA METODOLOGÍA A UTILIZAR.	20
2.4 LENGUAJE DE MODELADO Y HERRAMIENTAS.	21
2.4.1 <i>Lenguaje Unificado de Modelado 2.0 (UML).</i>	22
2.4.2 <i>Herramientas CASE (Computer Assisted Software Engineering).</i>	22
2.4.2.1 Visual Paradigm for UML Enterprise Edition 6.4.	23
2.4.2.2 Rational Rose Enterprise Edition.	24
2.4.3 <i>Selección de la herramienta CASE a utilizar.</i>	24
2.5 SISTEMA GESTORES DE BASES DE DATOS.	25
2.5.1 <i>PostgreSQL.</i>	25
2.5.2 <i>Oracle.</i>	26
2.5.3 <i>MYSQL.</i>	27
2.6 SELECCIÓN DEL GESTOR DE BASES DE DATOS A UTILIZAR.....	28
2.7 LENGUAJE DE PROGRAMACIÓN.....	28

2.7.1	<i>Java</i>	28
2.7.2	<i>C++</i>	29
2.7.3	<i>PHP 5.3</i>	30
2.8	SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN A UTILIZAR.....	31
2.9	LIBRERÍAS A UTILIZAR.	32
2.9.1	<i>jQuery</i>	32
2.9.2	<i>PHPExcel</i>	33
2.10	SERVIDOR WEB.....	34
2.10.1	<i>Servidor Web Apache 2.0</i>	34
2.10.2	<i>Servidor Web Internet Information Server (IIS)</i>	34
2.11	SELECCIÓN DEL SERVIDOR WEB A UTILIZAR.	35
2.12	AMBIENTE DE DESARROLLO.	36
2.12.1	<i>NetBeans 6.8</i>	36
2.12.2	<i>Eclipse</i>	37
2.13	SELECCIÓN DEL AMBIENTE DE DESARROLLO A UTILIZAR.	38
2.14	CONCLUSIONES.....	38
CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.....		39
3.1	INTRODUCCIÓN.....	39
3.2	ACTORES DEL NEGOCIO.....	39
3.3	TRABAJADORES DEL NEGOCIO.	39
3.4	DESCRIPCIÓN DEL NEGOCIO.	40
3.5	DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	41
3.6	DESCRIPCIÓN TEXTUAL DEL CASO DE USO DE NEGOCIO.....	41
3.6.1	<i>Caso de Uso: Solicitar Informe Final</i>	41
3.7	REGLAS DEL NEGOCIO.	43
3.8	REQUERIMIENTOS FUNCIONALES.	44
3.9	REQUERIMIENTOS NO FUNCIONALES.....	44
3.10	DESCRIPCIÓN DEL SISTEMA PROPUESTO.	46
3.10.1	<i>Descripción del Actor del Sistema</i>	46
3.10.2	<i>Diagrama de Casos de Uso del Sistema</i>	46
3.11	DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL SISTEMA.....	46
3.12	CONCLUSIONES.....	51
CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....		53

4.1	INTRODUCCIÓN.....	53
4.2	DESCRIPCIÓN DE LA ARQUITECTURA.....	53
4.3	PATRONES.....	55
4.3.1	<i>Patrones de diseño empleados.....</i>	<i>55</i>
4.4	VISTA LÓGICA DEL SISTEMA.....	59
4.5	DISEÑO DEL SISTEMA.....	60
4.5.1	<i>Diagrama de Clases del Diseño.....</i>	<i>60</i>
4.6	MODELO DE IMPLEMENTACIÓN.....	62
4.6.1	<i>Diagrama de Componentes.....</i>	<i>63</i>
4.7	DIAGRAMA DE DESPLIEGUE.....	63
4.8	VALIDACIÓN DE LA SOLUCIÓN A TRAVÉS DEL CRITERIO DE EXPERTOS.....	64
4.9	OPINIONES DE LOS EXPERTOS.....	66
4.10	CONCLUSIONES.....	67
	CONCLUSIONES GENERALES.....	68
	RECOMENDACIONES.....	69
	TRABAJOS CITADOS Y CONSULTADOS.....	70

INTRODUCCIÓN

El funcionamiento del mundo actual resulta prácticamente inconcebible sin el aporte del petróleo como fuente energética de primer orden, el cual se ha convertido en la fuente de energía más importante del mundo, hasta el punto de que existen varios países cuya economía depende exclusivamente de la explotación de este recurso. El petróleo representa un papel importante para el progreso económico de la humanidad, lo que conlleva a que exista un mercado donde cada día se insertan nuevas empresas que se dedican a realizar la búsqueda de petróleo.

En todo el orbe se hace necesario trazar estrategias para racionalizar la explotación y el comercio del mismo. Hoy por hoy, las empresas de petróleo manejan gran caudal de información, por lo que se hace necesario que la misma se encuentre de manera persistente en sistemas de bases de datos, la gestión de la información toma un papel importante en las operaciones rutinarias de las empresas petroleras.

En Cuba se encuentra una organización que se dedica a la exploración, explotación, refinación, transporte de distribución, compra y venta de petróleo, sus derivados y carburantes nacionales, la cual constituye la fusión de la Unión del Petróleo con la Unión del Combustible con el nombre de CUBAPETRÓLEO (CUPET), creada el 25 de marzo de 1993.

La Unión CUBAPETRÓLEO (CUPET) tiene la misión fundamental de satisfacer las necesidades del mercado nacional de hidrocarburos de forma competitiva, a partir del aumento de la producción y la optimización del uso de los combustibles nacionales, como contribución a la independencia económica del país.

Dentro del CUPET se encuentra el CEINPET en el mismo se halla situado el Laboratorio de Lodo y Cemento, el cual se encarga de controlar el trabajo de las compañías extranjeras con los fluidos de perforación y el comportamiento de los fluidos a partir de los datos de los reportes, dichos reportes son creados a través de los datos diarios de la perforación de los pozos.

Este proceso se lleva cabo en el Laboratorio de Lodo y Cemento del CEINPET, el cual se realiza de forma no optimizada mediante la confección de tablas en Excel que se elaboran introduciendo los datos de los reportes. Debido al gran cúmulo de información que poseen dichos reportes, el proceso se torna un poco tedioso, pues se realiza de manera manual y los especialistas se ven obligados a desperdiciar un tiempo valioso

que se pudiera emplear en otras actividades de mayor peso para la institución. Esto conlleva a la inexistencia de un mecanismo que agilice el almacenamiento de los datos para el uso posterior de estos.

Por lo anteriormente planteado el **problema a resolver** quedaría definido en la siguiente interrogante: ¿Cómo disminuir los retrasos que existen en la consulta, revisión y análisis de los reportes recibidos en el Centro de Investigaciones del Petróleo (CEINPET)?

Con el propósito de darle cumplimiento al problema planteado se propone a modo de **objetivo general** desarrollar un componente de software que procese documentos en formato Excel e importe la información contenida en los mismos hacia bases de datos en PostgreSQL. Para darle solución al problema planteado se toma como **objeto de estudio** el proceso de consulta y revisión de la información contenida en los reportes recibidos en el CEINPET.

El **campo de acción** de la investigación se centrará en la informatización del proceso de consulta y revisión de la información contenida en los reportes recibidos en el CEINPET. Partiendo de lo antes citado y para tener una guía a lo largo de toda la investigación, surge como **idea a defender** si se desarrolla un componente de software que procese documentos en formato Excel e importe la información contenida en los mismos hacia bases de datos en PostgreSQL, se logra agilizar el proceso de consulta y revisión de la información contenida en los reportes recibidos en el CEINPET.

Posibles resultados: Se espera aportar un componente de software que procese documentos en formato Excel e importe la información contenida en los mismos hacia bases de datos en PostgreSQL, en aras de mejorar la gestión de la información en el Laboratorio de Lodo y Cemento.

Para alcanzar el cumplimiento del objetivo de este trabajo de diploma se han propuesto desarrollar las siguientes **tareas de investigación:**

1. Analizar el Estado del Arte y elaborar el marco teórico referencial sobre el tema.
2. Identificar las tendencias de las tecnologías y metodologías actuales en el desarrollo del tipo de software en cuestión.
3. Definir las herramientas, tecnologías y metodologías a utilizar para el desarrollo.
4. Modelar los procesos de Negocio asociados al funcionamiento del sistema.
5. Realizar Diseño e Implementación del software.

Para el desarrollo de las tareas de investigación se utilizaron **métodos de investigación**, tanto teóricos como empíricos.

Entre los **Métodos Teóricos** utilizados se encuentra el **Analítico-Sintético**, el cual facilita mediante el análisis y la síntesis, la búsqueda de lo fundamental en la bibliografía consultada, los temas tratados, metodologías, herramientas de modelado, arquitectura para este tipo de software, lenguajes de programación, ambientes de desarrollo, entre otros temas.

Además, se utilizó el **Histórico-Lógico**, al inicio del trabajo, se desarrolló un estudio del estado del arte de la problemática; se analizaron las ventajas y desventajas de cada una de las herramientas y las tendencias en la resolución de esta problemática.

Al mismo tiempo, se empleó el de **Modelación**, el cual se muestra mediante los diagramas obtenidos, por medio de las herramientas de modelado y con el Lenguaje Unificado de Modelación (UML), donde se acumula la información de los artefactos generados de todas las etapas por las que pasa el software: negocio, requerimientos, diseño e implementación. Con este método se logrará un mejor entendimiento del problema y percepción de los conceptos que son necesarios analizar.

Por otra parte, entre los **Métodos Empíricos** se encuentra la **Observación participativa**, en la misma se llevó a cabo un registro visual del trabajo en el Laboratorio de Lodo y Cemento del CEINPET, con los reportes diarios que envían las Compañías de Servicios de Perforación de Pozos Petroleros.

También se empleó la **Entrevista**, las cuales fueron realizadas a partir de conversaciones planificadas con el responsable del Laboratorio de Lodo y Cemento del CEINPET, con el fin de obtener información acerca del problema existente en los reportes de lodo y de perforación cuando estos llegan al CEINPET en forma de documentos Excel o en formato PDF. Ver [Anexo 1.1](#).

El contenido del presente trabajo de diploma está estructurado de la siguiente manera:

Capítulo 1. Fundamentación teórica: se expone el estado del arte del objeto de estudio de la presente investigación y se definen los elementos teóricos que lo sustentan. Se enuncian conceptos que posibilitan un mejor entendimiento de lo planteado en la situación problemática y el marco del problema en sentido general.

Capítulo 2. Tendencias y Tecnologías actuales: se aborda acerca de las tecnologías y herramientas a utilizar en el desarrollo del componente de software, además de

fundamentar la elección del lenguaje de programación que se escoja, así como de las Metodologías de Desarrollo, el Lenguaje de Modelado y la Herramienta CASE.

Capítulo 3. Presentación de la solución propuesta: se exponen algunos de los principales artefactos generados en los Flujos de Trabajo de Modelación del Negocio y Requerimientos. Del primero es posible encontrar, entre otros, la explicación de los procesos del negocio, el Diagrama de Casos de Uso del Negocio y la descripción de cada uno de ellos; del segundo, los requerimientos funcionales y no funcionales, el Diagrama de Casos de Uso del Sistema y la descripción de los casos de uso críticos.

Capítulo 4. Construcción de la solución propuesta: Se describen los artefactos relacionados con el diseño e implementación, además de realizar la validación para comprobar que el sistema cumple con los requerimientos funcionales y que presenta una óptima calidad.

CAPÍTULO 1: Fundamentación Teórica.

1.1 Introducción.

En este capítulo se aborda en detalle todo lo relacionado con la fundamentación teórica, con el objetivo de definir los elementos teóricos para la realización de la investigación. Se especifican algunos conceptos asociados a la investigación, al igual que se profundiza en el Objeto de Estudio y la Situación Problemática, haciendo una descripción detallada de la misma. A partir de todos estos elementos se desarrolla la investigación, haciéndose más sencilla la comprensión de los temas que serán abordados en lo adelante.

1.2 Conceptos asociados al dominio del problema

Para lograr un mayor entendimiento del trabajo a continuación se reflejan conceptos asociados al dominio del problema.

✦ Fluidos de Perforación

Fluido de perforación es una mezcla de un solvente (base) con aditivos o productos, que cumplen funciones físico-químicas específicas, de acuerdo con a las necesidades operativas de una formación a perforar. (Rosabal, 2008)

Los **lodos de perforación** son los fluidos bombeados que circulan a través del pozo mientras este es perforado. Su composición se ajusta a medida que cambian las exigencias, de acuerdo con la profundidad de la perforación y los otros materiales encontrados. (Tricalcicos, 2006)

Los fluidos o lodos de perforación son muy significativos en la industria del petróleo. Estos tienen como objetivo principal el de disminuir la cantidad de petróleo descargada dentro del ambiente. Lo cual ayuda a que la perforación en el pozo sea vertiginosa y segura. Si un fluido es mal seleccionado o posee propiedades impropias, las cuales deben de ser determinadas por distintos ensayos, conduciría a la merma parcial o total del pozo y por supuesto a grandes gastos de recursos económicos.

✦ Funciones del fluido de perforación.

El Fluido de Perforación es un fluido de características químicas y físicas apropiadas, que puede ser aire o gas, agua, petróleo y combinaciones de agua y aceite con diferente contenido de sólidos.

Entre sus principales Funciones se encuentran (Martínez García, y otros, 2008):

- ✦ Evacuar los recortes de Perforación.
- ✦ Controlar las Presiones de la Formación.
- ✦ Suspender y descargar los recortes.
- ✦ Obturar las formaciones permeables.
- ✦ Mantener la estabilidad del pozo.
- ✦ Minimizar daños a la formación.
- ✦ Enfriar, lubricar y alivianar la columna de perforación.
- ✦ Asegurar una evaluación adecuada de la formación.
- ✦ Transmitir energía hidráulica a herramientas y trépano.
- ✦ Controlar la corrosión.
- ✦ Facilitar la Cementación y Terminación.
- ✦ Minimizar el Impacto sobre el Medio Ambiente.

Las funciones más importante del fluido de perforación son: enfriar y lubricar el trépano y la sarta de perforación, aquí la circulación del fluido de perforación enfría la barrena y el conjunto de perforaciones, alejando el calor de la fuente y distribuyéndolo en todo el pozo, de igual forma la circulación del fluido de perforación enfría la columna de perforación a una temperatura más baja que la temperatura del fondo.

Igualmente se debe de mantener la estabilidad del pozo, ya que constituye un equilibrio complejo de factores mecánicos (presión y esfuerzo) y químicos. La composición química y las propiedades del lodo deben combinarse para proporcionar un pozo estable hasta que se pueda introducir y cementar la tubería de revestimiento. (Martínez García, y otros, 2008)

1.2.1 Gestión de la información

La gestión de la información es una de las disciplinas que desde el siglo pasado se adentra en los estudios informacionales, Gloria Ponjuán Duarte define **gestión de la información** al proceso mediante el cual se obtienen, despliegan o utilizan recursos básicos (económicos, físicos, humanos, materiales) para manejar información dentro y para la sociedad a la que sirve. (Ponjuán Duarte, 2003)

El proceso de gestión de información debe ser valorado sistémicamente en diferentes dimensiones y el dominio de sus esencias permite su aplicación en cualquier organización. (Ponjuán Duarte, 2003)

Según Óscar González Martín **sistema de gestión de bases de datos** (SGBD o DBMS ‘Database Management System’) consiste en una colección de datos interrelacionados y un conjunto de programas que permiten a los usuarios acceder y modificar dichos datos. (Ruiz González, y otros, 1999/2000)

1.3 Conceptos asociados al Componentes de software.

Julio Casal Terreros, plantea: *“La complejidad de los sistemas computacionales actuales nos ha llevado a buscar la reutilización del software existente. El desarrollo de software basado en componentes permite reutilizar piezas de código pre elaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión.”* (Casal Terreros, Julio, 2009)

Hoy en día el desarrollo de software basado en componentes se ha transformado en el elemento más seguro, tanto para la construcción de grandiosos sistemas como para aplicaciones de software. Este se ha convertido en el pilar de la Revolución Industrial del Software y se proyecta hoy en día en diversas formas de hacer software de calidad con los costos más bajos del mercado y en tiempos que antes eran impensables.

Componentes de software.

Los componentes de software surgen como una evolución derivada de la programación orientada a objetos, estos ofrecen un desarrollo mucho más rápido y con mucho menos esfuerzo, reduciendo costos y tiempos de desarrollo, aprovechando las ventajas de la reutilización de código.

Hoy por hoy existen diversas definiciones de componentes de software complementarias entre sí.

Julio Casal Terreros plantea, que un componente es una unidad de composición de aplicaciones software, que posee un conjunto de interfaces y un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio. (Casal Terreros, Julio, 2009)

Mientras que **Clemens Szyperski**, plantea que un componente de software es una unidad de composición con una interfaz especificada contractualmente y con dependencias explícitas de contexto, un componente de software puede ser desplegado de manera independientemente y es tema de composición por terceras partes. (WordPress, 2009)

En esencia, un **componente** es una pieza de código pre elaborado que encapsula alguna funcionalidad expuesta a través de interfaces estándar. (WebCab-Components)

Como conclusión de estas definiciones se puede decir que un componente de software es una parte de software que se puede combinar con otros componentes para generar un conjunto aún mayor, tales como otro componente, subsistema o sistema. Un componente define una o más interfaces, desde las cuales se puede tener acceso a los servicios que este ofrece a los demás componentes. Un componente puede presentarse en forma de código fuente o código objeto; puede estar escrito en un lenguaje funcional, procedural o en lenguaje orientado a objetos; y puede ser tan simple como un botón GUI (Graphical User Interface) o tan complejo como un subsistema.

1.3.1 Modelos de desarrollo por componentes.

En nuestros días existen diferentes modelos de desarrollo por componentes, entre los que se encuentran COM, CORBA y Java Beans, a continuación se describe de forma breve cada uno de ellos en lo que se refiere al desarrollo basado en componentes.

COM (Component Object Model): Es la propuesta de Microsoft en materia de componentes. COM es usado por desarrolladores para formar componentes de software re-utilizables, los cuales se unen para crear aplicaciones, y aprovechar los servicios de Windows (Microsoft). COM es esencialmente un esquema de integración, de modo que los componentes que se construyen para trabajar bajo este modelo, deben describir su comportamiento bajo las consideraciones de este esquema, el cual es popularmente conocido como estándar binario. En este tipo de entornos las interfaces de los componentes publican un conjunto de apuntadores a tablas de memoria (tablas virtuales de funciones); el llamado entre componentes se hace a través de los apuntadores, las interfaces son definidas utilizando el lenguaje IDL (Interface Description Language).

CORBA (Common Object Request Broker Architecture): Es una infraestructura abierta para el manejo de objetos distribuidos que está siendo estandarizada por el

Object Management Group (OMG) (corba.org, 2009). En CORBA, un objeto se considera como una instancia de una clase que encapsula operaciones, atributos y excepciones. CORBA permite que los objetos puedan comunicarse con otros sin necesidad de preocuparse por donde están localizados o por quien han sido diseñados. CORBA utiliza el mismo principio del modelo anterior, es decir, las interfaces de los componentes son definidas a través de IDL lo que permite la interoperabilidad de componentes construidos en diferentes lenguajes de programación.

Java Beans: Actualmente uno de los modelos de desarrollo de software basado en componentes con mayor aceptación, es el propuesto por Sun Microsystems (java.sun.com), conocido como Java Beans. Java Beans es una API(Application Programming Interface) implementada para la construcción y uso de componentes escritos en Java, los cuales son comúnmente llamados Beans. Esta API es proporcionada por Sun como una herramienta visual que permite la carga, utilización, modificación, así como también interconexión de Beans, con el propósito de construir nuevos Beans, applets o bien aplicaciones completas. Java Beans se caracteriza por que los componentes (EJB) solo muestran los servicios que ofrecen a través de sus interfaces y solo pueden ser accedidos mediante ellas.

1.3.2 Beneficios del Desarrollo de Software basado en Componentes.

Los componentes de software son una opción que la industria informática brinda para hacer que la construcción de nuevos sistemas sea más fácil y rápida, es por esto por lo que cada día toman mayor importancia en el desarrollo de sistemas empresariales.

Según una investigación hecha por Julio Casal Terreros, Ingeniero de Sistemas Computacionales de la Universidad Católica de Santiago de Guayaquil, el Desarrollo de Software Basado en Componentes posee grandes ventajas en su uso, como son:

1. **Reutilización del software.** Lleva a alcanzar un mayor nivel de reutilización de software.
2. **Simplifica las pruebas.** Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
3. **Simplifica el mantenimiento del sistema.** Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.

4. **Mayor calidad.** Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

La principal ventaja que presenta el desarrollo de Software basado en Componentes, es la reutilización de componentes, de esta manera, los componentes se diseñan con el objetivo de poder reutilizarlos en otros proyectos, favoreciendo a la necesidad de realizar software complejos en cortos periodos de tiempo y a la vez con un menor esfuerzo en la realización del mismo, los costos son mucho menores que desarrollar un software desde cero, la reutilización de código previamente realizado y probado facilita el trabajo, mejorando la fiabilidad del producto final.

1.4 Análisis de otras soluciones existentes.

Como resultado de la consulta y estudio de materiales bibliográficos, ha sido posible la identificación de un software cuya funcionalidad está basada en el importe de datos, de manera rápida, desde archivos MS Excel, MS Access, DBF, TXT, CSV, XML hacia el sistema gestor de Bases de Datos PostgreSQL, dicho software se nombra EMS Data Import for PostgreSQL, el mismo es propietario, lo que implica que no es un software de código abierto, y a la vez hay que pagar por la licencia del mismo, no obstante, este se puede descargar gratis con una licencia de 30 días.

EMS Data Import for PostgreSQL

EMS Data Import for PostgreSQL es una poderosa herramienta para importar datos, hacia tablas de PostgreSQL. Provee parámetros configurables de importación, incluyendo formatos de datos de origen para todos los campos y formatos de datos de destino para los campos seleccionados, opciones de confirmación de ejecución (commit), número de registros a saltar, entre otros. Data Import para PostgreSQL incluye un asistente que te permite configurar todas las opciones de importación para diferentes archivos de manera visual y un utilitario de líneas de comandos para ejecutar la importación en un sólo "clic". (EMS-Database-Management-Solutions)

Características principales:

1. Importación desde los 10 formatos más comunes: MS Excel 97-2007, MS Access, XML, DBF, TXT, CSV, MS Word 2007, formato OpenDocument y HTML.
2. Importación de datos hacia una o varias tablas o vistas desde diferentes bases de datos.

3. Un número de modos de importación Insert All (insertar todos), Insert New (insertar nuevo), Insert (insertar) o Update (actualizar), entre otros.
4. Soporte UNICODE.
5. Parámetros configurables de importación para cada archivo de origen
6. Opción para salvar todos los parámetros de importación configurados en la sesión actual del asistente.
7. Utilitario de líneas de comandos para importar datos utilizando el archivo de configuración.
8. Poderoso módulo visual de opciones.
9. Interfaz tipo asistente amigable para el usuario. (EMS-Database-Management-Solutions)

Actualmente no existe un software que no sea autónomo, que permita el importe de datos hacia Sistemas Gestores de Bases de Datos.

En Cuba, el CEINPET, no cuenta con un software que permita disminuir los retrasos que existen en la consulta, revisión y análisis de los reportes recibidos de la Empresa de Perforación y Extracción de Petróleo de Centro (EPEP Centro), satisfaciendo los formatos de los reportes generados.

1.5 Conclusiones

Con el desarrollo de este capítulo se ha logrado definir de forma clara todos los aspectos teóricos-conceptuales, necesarios para la realización de la investigación propuesta, es decir, se ha detallado el objeto de estudio, así como la descripción general de este. También se han analizado los conceptos fundamentales referentes al desarrollo basado en componentes de software y los beneficios que estos presentan. A partir del resultado del estudio plasmado en este capítulo, se puede proceder a seleccionar las tecnologías necesarias para el desarrollo del componente de software.

CAPÍTULO 2: Tendencias y Tecnologías actuales.

2.1 Introducción.

En la actualidad existe una gran variedad tecnológica que ofrece excelentes posibilidades para que grandes y pequeñas empresas puedan dar soluciones efectivas a sus necesidades. En este capítulo se hace un análisis de las principales tendencias y tecnologías actuales a considerar para ser empleadas en el desarrollo del software que dará solución a la presente investigación. También se describen los lenguajes, las metodologías, las tecnologías y herramientas a considerar para su posterior utilización en el desarrollo, analizando sus características, y seleccionando las mejores propuestas con el objetivo de dar cumplimiento con la mayor eficiencia y calidad posible al objetivo general de la presente investigación.

2.2 Metodologías Existentes.

En el desarrollo de software la necesidad de organizar ó estructurar de forma correcta y disciplinada, es uno de los factores más importantes pues sin esto de seguro los mismos podían convertirse en verdaderas catástrofes, obteniendo como resultados grandes pérdidas de tiempo y recursos, en fin se arribaría a un total fracaso. Para evitar tales errores es preciso definir una estrategia para darle un orden a las tareas posibles a desarrollar, así como también llevar a cabo una guía de cómo efectuar las actividades, en fin llevar a cabo un conjunto de procedimientos y pasos que se deben de seguir para el desarrollo de un software constituyendo los mismos una metodología de desarrollo de software.

Existen disímiles metodologías de desarrollo de software pero todas ellas conforman las dos grandes corrientes por así decirlo, referentes a los proceso de software, las mismas son, las metodologías robustas o tradicionales y las metodologías ágiles. Entre las más utilizadas se encuentran el Proceso Unificado de Software (RUP), Microsoft Solutions Framework (MSF), la Extreme Programing (XP) y Agile Unified Process (AUP).

2.2.1 Metodologías ágiles.

Las **metodologías ágiles** son aquellas que se caracterizan cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien

documentado), y *adaptable* (permite realizar cambios de último momento). (P.Letelier, 2006)

2.2.1.1 Agile Unified Process (AUP).

Agile Unified Process (AUP), es una versión simplificada del Proceso Unificado de Rational (RUP). Descrito en una forma simple, fácil de entender y brinda un enfoque de desarrollo de software utilizando técnicas ágiles y conceptos del RUP. El proceso unificado (*Unified Process* o UP) es un marco de desarrollo de software iterativo e incremental.

La Fig.1 muestra el ciclo de vida de esta metodología. AUP abarca siete flujos de trabajos, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente. El modelado agrupa los tres primeros flujos de RUP (Modelamiento del negocio, Requerimientos y Análisis y Diseño). Dispone de cuatro fases igual que RUP: Creación, Elaboración, Construcción y Transición.

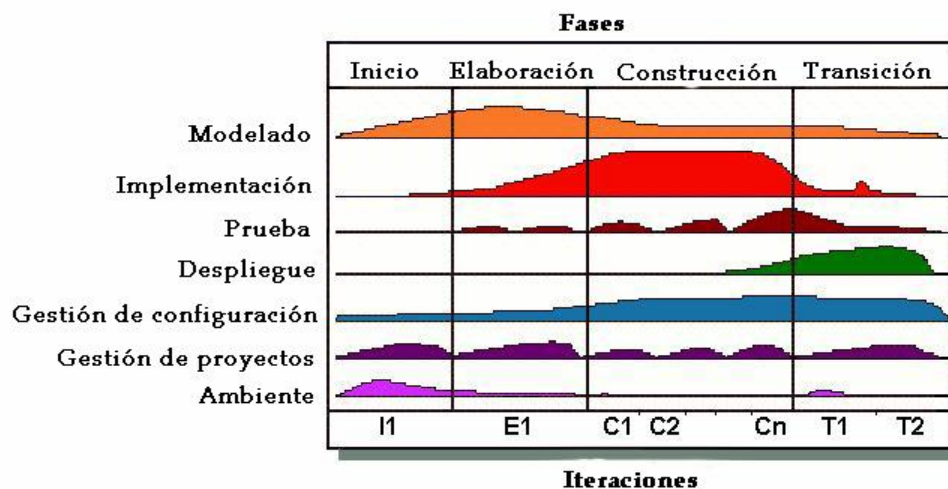


Fig. 1 Esquema que muestra los flujos de trabajo y las fases de AUP.

El ciclo de vida de AUP, se puede ver en sus cuatro fases, y su naturaleza iterativa se expresa en sus siete disciplinas que son:

- ♦ **Modelado:** Comprender el negocio de la organización, comprender el dominio del problema abordado por el proyecto, e identificar una solución al mismo que sea viable.
- ♦ **Implementación:** Transformar el modelo realizado en código ejecutable y realizar pruebas de nivel básico, en particular pruebas unitarias.

- ✦ **Prueba:** Realizar una evaluación objetiva para asegurar la calidad. Esto incluye buscar defectos, validar que el sistema funcione como debería, y verificar que se cumplen los requerimientos.
- ✦ **Despliegue:** Planificar la liberación del sistema.
- ✦ **Gestión de configuración:** Administrar el acceso a los artefactos del proyecto.
- ✦ **Gestión de proyectos:** Dirigir las actividades que forman parte del proyecto.
- ✦ **Ambiente:** Facilitar todo el entorno que permita el normal desarrollo del proyecto.

AUP divide el ciclo de desarrollo en 4 fases:

Inicio: Identificación del alcance y dimensión del proyecto, propuesta de la arquitectura y de presupuesto del cliente.

Elaboración: Confirmación de la idoneidad de la arquitectura.

Construcción: Desarrollo incremental del sistema, siguiendo las prioridades funcionales de los implicados.

Transición: Validación e implantación del sistema.

AUP es una metodología de desarrollo ágil, la cual se basa en los siguientes principios:

- ✦ **El personal sabe lo que está haciendo.** La gente no va a leer detallado el proceso de documentación, pero algunos quieren una orientación de alto nivel y/o formación de vez en cuando.
- ✦ **Simplicidad.** Todo se describe concisamente utilizando un puñado de páginas, no miles de ellos.
- ✦ **Agilidad.** El ajuste a los valores y principios de la Alianza Ágil.
- ✦ **Centrarse en actividades de alto valor.** La atención se centra en las actividades esenciales para el desarrollo, no todas las actividades que suceden forman parte del proyecto.
- ✦ **Herramienta de la independencia.** Usted puede usar cualquier conjunto de herramientas que usted deseé con el ágil UP. Lo aconsejable es utilizar las herramientas más adecuadas para el trabajo, que a menudo son herramientas simples o incluso herramientas de código abierto. (Cordero)

AUP define actividades que el equipo de desarrolladores debe realizar para construir, validar y entregar un software que satisfaga las necesidades de los involucrados.

2.2.1.2 Extreme Programming (XP).

La metodología Extreme Programming (XP) o Programación Extrema, es la primera metodología ágil y la que le dio conciencia al movimiento actual de metodologías ágiles. XP es una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (2009). XP se define especialmente para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

El objetivo de la XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional claro, desde el punto de vista del negocio. (Valverde Rebaza, y otros, 2007)

XP es un proceso muy orientado a la programación, debido al bajo número de documentación a generar, además la programación se hace en pareja que es un punto a favor de XP, pero el código pertenece al equipo completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento, para el final.

Una de las herramientas más importantes en la metodología XP es el desarrollo orientado a pruebas, que utiliza las pruebas unitarias como eje de todo desarrollo. Las interacciones suelen ser muy cortas y se promueve a los programadores a buscar soluciones y experiencia con ellas, programar sin miedo a descomponer el sistema.

2.2.2 Metodologías robustas.

Las *metodologías robustas* son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo; llamadas también metodologías tradicionales o clásicas, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema. (P.Letelier, 2006)

2.2.2.1 Rational Unified Process (RUP).

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. (Jacobson, y otros, 2004)

Está basado en componentes, lo cual quiere decir que el sistema de software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas. Este utiliza el lenguaje de modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software y se sostiene sobre tres ideas básicas, **dirigido por casos de uso**, donde los casos de uso definen lo que el usuario desea a partir de la captura de requisitos y la modelación del negocio, **centrado en la arquitectura**, característica que brinda una visión completa del sistema, se describen los procesos del negocio que son más importantes para comprenderlo, desarrollarlo y producirlo de una forma eficaz e **iterativo e incremental**, donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo, lo que hace único un proceso unificado. Para que estas ideas se logren se necesita un proceso polifacético, que tenga en cuenta ciclos, frases, flujos de trabajo, gestión de riesgos, control de la calidad, gestión de proyecto y control de la configuración.

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes. Cada ciclo consta de cuatro fases, **Inicio, Elaboración, Construcción y Transición**. Cada fase se subdivide a su vez en iteraciones donde en cada una se realiza una evaluación para determinar si los objetivos de la fase fueron cumplidos.

2.2.2.2 Microsoft Solutions Framework (MSF).

Microsoft Solutions Framework (MSF) es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. (S.A, Implementación personalizada Gattaca)

MSF es considerado más un marco de trabajo que una metodología, debido a que es flexible y abierto, puede adaptarse para ajustarse a los requerimientos y necesidades particulares de una organización.

MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Este se compone por 6 modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

El Modelo de Proceso de Desarrollo de MSF describe un ciclo de vida que puede ser usado para desarrollar software de manera exitosa, estableciendo el orden en el cual se deben realizar las actividades. (Mendoza, y otros)

Este modelo consiste en cinco fases distintas, cuyos nombres dependen del tipo del proyecto en el que se aplica. Cada fase del proceso de desarrollo culmina con un hito visible.

Fase 1: *Visión.* En esta fase el equipo y el cliente definen los requerimientos del negocio y los objetivos generales del proyecto.

Fase 2: *Planeación.* Durante la fase de planeación el equipo crea un borrador del plan maestro del proyecto, además de un cronograma del proyecto y de la especificación funcional del proyecto.

Fase 3: *Desarrollo.* Esta fase involucra una serie de liberaciones internas del producto, desarrollados por partes para medir su progreso y para asegurarse que todos sus módulos o partes están sincronizados y pueden integrarse.

Fase 4: *Estabilización.* Esta fase se centra en probar el producto. El proceso de prueba hace énfasis en el uso y el funcionamiento del producto en las condiciones del ambiente real.

Fase 5: *Implantación.* En esta fase el equipo implanta la tecnología y los componentes utilizados por la solución, estabiliza la implantación, apoya el funcionamiento y la transición del proyecto, y obtiene la aprobación final del cliente.

La Metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología.

2.3 Selección de la Metodología a utilizar.

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto, Recursos técnicos y humanos, tiempo de desarrollo y tipo de sistema, son algunas de las cuestiones a valorar a la hora de escoger una metodología. Históricamente, las

metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. (José H.Canós)

Para facilitar la selección de la metodología a emplear en la realización de la investigación, se efectuó una pequeña comparación entre las metodologías antes expuestas.

	AUP	XP	RUP	MSF
Dirigido por casos de uso.	SI	NO	SI	NO
Desarrollo iterativo e incremental.	SI	NO	SI	NO
Participación activa del usuario.	NO	SI	NO	NO
Adaptación del proceso.	SI	SI	SI	SI
Centrado en la arquitectura.	SI	NO	SI	SI

Tabla 1 Comparación entre AUP, XP, RUP, MSF.

Teniendo en cuenta las particularidades de cada una de las metodologías de desarrollo, se ha seleccionado **Agile Unified Process (AUP)**. AUP proporciona un ambiente de desarrollo de software iterativo e incremental. A menudo, es calificado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto de software, pero dado que es un marco de procesos, puede ser adaptado ya que es muy semejante a RUP, AUP propone los mismos roles, artefactos de RUP, pero en una versión simplificada; sólo se utilizan los artefactos que son imprescindibles y realmente necesarios para la realización del producto. Además, AUP adopta muchas de las técnicas ágiles de XP y otros procesos ágiles que aun mantienen la formalidad de RUP.

De igual forma AUP se preocupa especialmente de la gestión de riesgos, es decir, propone para aquellos elementos con alto riesgo, prioridad en el proceso de desarrollo y que sean abordados en etapas tempranas del mismo.

2.4 Lenguaje de modelado y Herramientas.

El progreso de software ha tenido un gran avance en los últimos tiempos, y las herramientas de modelado forman un componente muy significativo en el entorno de desarrollo, puesto que son esenciales para el análisis de sistemas. Las herramientas mejoran la forma en que ocurre el desarrollo y tiene influencia sobre la calidad del resultado final.

En la actualidad se han creado una serie de herramientas con este fin, las cuales han sido mejoradas por parte de sus desarrolladores con el fin de encontrar en ellas fiabilidad, eficiencia, entre estas se encuentra: Visual Paradigm y Rational Rose Enterprise Edition, dichas herramientas usan el Lenguaje de Modelado UML para la especificación, la visualización, la construcción y la documentación de los artefactos de los sistemas de software y también para otros tipos de sistemas.

2.4.1 Lenguaje Unificado de Modelado 2.0 (UML).

UML (por sus siglas, del inglés **Unified Modeling Language**) es el lenguaje estándar de modelado para software. Es un lenguaje gráfico que visualiza, especifica, construye y documenta los artefactos del sistema. UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas ó diagramas estandarizados, proporcionando a los desarrolladores un vocabulario que incluya tres categorías: elementos, relaciones y diagramas.

UML suministra mecanismos de extensibilidad, los cuales permiten a sus usuarios clarificar su sintaxis y su semántica. UML puede tanto ajustarse a un sistema, proyecto ó proceso de desarrollo específico si es necesario. (Jacobson, y otros, 2004)

UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es significativo destacar que UML es un "lenguaje gráfico" para visualizar, especificar, construir y documentar un sistema de software y no para describir métodos ó procesos.

2.4.2 Herramientas CASE (Computer Assisted Software Engineering).

Roger Pressman plantea que las herramientas CASE ayudan a los gestores y practicantes de la ingeniería de software en todas las actividades asociadas a los proyectos de software. Automatizan las actividades de gestión de proyectos, gestionan todos los productos de los trabajos elaborados a través del proceso y ayudan a los ingenieros en el trabajo de análisis, diseño y codificación.

La herramientas CASE brindan una gran gama de componentes que incluyen todos o la mayoría de los requisitos necesarios para el desarrollo de los sistemas, además son creadas con una gran exactitud en torno a las necesidades de los desarrolladores de

software para la automatización de procesos incluyendo el análisis, diseño e implementación. Ofrecen una gran plataforma de seguridad a sistemas que las usan.

Las herramientas CASE se han venido ampliando y desarrollando, existe una gran variedad de estas con características específicas, a continuación describiremos algunas de ellas, como es Visual Paradigm for UML Enterprise Edition 6.4 y Rational Rose Enterprise Edition.

2.4.2.1 Visual Paradigm for UML Enterprise Edition 6.4.

Visual Paradigm es una herramienta CASE que utiliza “UML”: como ***lenguaje de modelado***. Visual Paradigm soporta el ciclo de vida completo del desarrollo de software: Análisis y Diseño orientados a objetos, Construcción, Pruebas y Despliegue. En Visual Paradigm for UML, el diseño está centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. Es una herramienta diseñada para una amplia gama de usuarios, incluyendo el analista del sistema, ingenieros de software, entre otros.

Visual Paradigm for UML se caracteriza por lo siguiente:

- Soporta todos los diagramas de SysML y diagramas de entidad-relación.
- Ofrece amplias características de modelado de casos de uso incluyendo la función completa de UML, diagrama de casos de uso y editor de flujo de eventos.
- Genera Java, C#, C++, PHP y lenguaje de definición de datos (DDL) para todas las bases de datos populares. (Paradigm, 2007)
- Permite diseñar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. El mismo proporciona abundantes tutoriales del Lenguaje de Modelado, demostraciones interactivas de UML y proyectos UML.
- Visual Paradigm for UML se integra con herramientas Java, como son: Eclipse/IBM, NetBeans IDE, entre otras.
- Visual Paradigm for UML es apoyado por un conjunto de lenguajes tanto en la generación del código como en la Ingeniería Inversa, por mencionar algunos ejemplo los cuales tiene la capacidad de soporte podríamos hablar de Java, C, C++, CORBA IDL, PHP, XML, Schema, Ada, y Python.

- ✦ Proporciona soporte a varios lenguajes en generación de código e ingeniería inversa a través de plataformas java, Ruby, .NET, Python, CORBA IDL, Java, C++, XML Schema, etc.
- ✦ Visual Paradigm for UML apoya la generación de código del C#, .Net, Object Definition Language (ODL), Flash ActionScript, Delphi, Perl y Ruby.

2.4.2.2 Rational Rose Enterprise Edition.

Rational Rose Enterprise Edition es una potente herramienta de modelado visual para ayudar en el análisis y diseño de software orientado a objetos. Es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++ y Visual Basic. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Rational Rose Enterprise posee características adicionales incluidas:

- ✦ Rational Rose Enterprise está basada en modelos que se integra con las bases de datos y los IDE de las principales plataformas del sector.
- ✦ Rational Rose Enterprise soporta el análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
- ✦ Rational Rose Enterprise posee características de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- ✦ Rational Rose Enterprise tiene soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- ✦ Rational Rose Enterprise tiene integración con cualquier sistema de control de versiones compatible con SCC, como IBM Rational ClearCase.
- ✦ Rational Rose Enterprise es capaz de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.

2.4.3 Selección de la herramienta CASE a utilizar.

Después de realizar un análisis entre las herramientas CASE, se escogió **Visual Paradigm For UML Enterprise Edition 6.4**, la cual soporta las últimas versiones del mismo, (Lenguaje de Modelado Unificado), y la Notación y Modelado de Procesos de

Negocio. Además, esta disponibles en varias ediciones, cada una destinada a unas necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal.

Visual Paradigm For UML Enterprise Edition, es una herramienta muy completa y fácil de emplear. Permite que varios desarrolladores trabajen en el mismo diagrama y vean los cambios que se hacen en tiempo real, además de ayudar a los desarrolladores a mejorar la construcción del modelo del software, lo que posibilita acelerar la producción y mejorar la calidad del trabajo.

En cambio Rational Rose Enterprise Edition es una herramienta muy utilizada pero es software propietario y no es multiplataforma, al contrario de Visual Paradigm For UML Enterprise Edition que es una herramienta multiplataforma y algunas de sus versiones son gratuitas.

2.5 Sistema Gestores de Bases de Datos.

Los Sistemas Gestores de Bases Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Son programas que admiten crear y conservar las bases de datos asegurando su integridad, confidencialidad y seguridad.

Un **Sistema Gestor de Bases de Datos (SGBD)** permite: definir, crear y mantener las bases de datos, además permite la autonomía entre los datos y los programas de aplicación, minimizar las redundancias, garantiza la integridad, la seguridad y protección de los datos, proporciona la manipulación de la información, permite un control centralizado, también proporciona un acceso controlado a estas y un lenguaje de manejo de datos para la inserción, actualización, eliminación y consulta de información en estas bases de datos.

A continuación se expone las características fundamentales de Oracle, PostgreSQL y MySQL, lo que permitirá la selección del que se va a utilizar en la implementación del componente de software.

2.5.1 PostgreSQL.

PostgreSQL es uno de los sistemas gestores de bases de datos de código abierto más avanzada del mundo. PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en Postgres, este posee diversas características dentro de las cuales están:

- ✦ PostgreSQL es un servidor de base de datos objeto relacional libre, el cual está liberado bajo la licencia BSD, es decir, que puede ser utilizado, modificado y distribuido por todo el mundo, libre de cargo para cualquier propósito, sea comercial, privado, o académico.
- ✦ PostgreSQL corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, entre otros.
- ✦ PostgreSQL posee un soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, entre otros. **(PostgreSQL-Global-Development-Group)**
- ✦ PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor.
- ✦ PostgreSQL acerca los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.
- ✦ PostgreSQL soporta casi toda la sintaxis SQL (incluyendo sub-consultas, transacciones, y tipos y funciones definidas por el usuario). **(PostgreSQL-Global-Development-Group)**

2.5.2 Oracle.

Oracle es un sistema de administración de base de datos (o RDBMS Relational Data Base Management System por las siglas en inglés), fabricado por Oracle Corporation, básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. (Villagomez, y otros, 2009)

Oracle, es una base de datos Orientadas a Objetos (BDOO) con beneficios OODBMS (Object Oriented DataBase Management System), ya que permite almacenar y manipular información que puede ser digitalizada (representada) por Objetos, proporciona una estructura flexible con acceso ágil, rápido con gran capacidad de modificación. (Villagomez, y otros, 2009)

Oracle, posee una amplia gama de funcionalidades, las más importantes se definen a continuación:

Los sofisticados **mecanismos de seguridad** de Oracle controlan el acceso a los datos sensibles utilizando un conjunto de privilegios.

Los sofisticados procedimientos de **realización de copias de seguridad y recuperación**, es decir, las copias de seguridad que permiten crear copias secundarias

de los datos de Oracle, y los procedimientos de restaurar los datos a partir de la copias de seguridad.

Ofrece una **gestión flexible del espacio**. Se puede asignar un cierto espacio de disco para el almacenamiento de los datos, y controlar las subsiguientes asignaciones instruyendo a Oracle sobre cuánto espacio reservar para los requerimientos futuros.

Oracle **proporciona conectividad hacia y desde paquetes software de otros fabricantes**. Utilizando las extensiones a la base de datos Oracle, se puede trabajar con información almacenada con otros sistemas de bases de datos

El servidor Oracle también se encarga de la **integridad de los datos**. Si se produce cualquier tipo de fallo mientras un usuario está cambiando los datos en una base de datos, ésta tiene la capacidad de deshacer o cancelar cualquier transacción sospechosa.

La estrategia de **copias de seguridad y recuperación** de Oracle permite minimizar la pérdida de datos y el tiempo de para cuando se produce un problema.

2.5.3 MYSQL.

MySQL es un sistema para la administración de bases de datos relacional (RDBMS) rápido y sólido. MySQL es la base de datos de código abierto más popular del mundo, compite con sistemas RDBMS (Relational Database Management System) propietarios conocidos, como Oracle, SQL Server y DB2.

MySQL posee disímiles características dentro de las cuales están:

- ✦ MySQL utiliza el lenguaje de consulta estructurado (SQL).
- ✦ MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL (General Public License) de la GNU.
- ✦ MySQL es un servidor multiusuario y de subprocesamiento múltiple.
- ✦ MySQL utiliza SQL como lenguaje de consulta estructurado, el lenguaje estándar para la consulta de bases de datos utilizado en todo el mundo.
- ✦ MySQL se distribuye bajo una licencia de código abierto en la actualidad, pero también existen licencias comerciales.
- ✦ MySQL posee gran versatilidad, es decir, trabaja tanto con sistemas operativos basados en Unix como con el sistema operativo Windows, de Microsoft.
- ✦ MySQL dispone de una arquitectura que lo hace extremadamente rápido y fácil de personalizar.

- ✦ MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- ✦ MySQL ss. Open Source, es decir, el código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales. (MySQL-Hispano.org.)

2.6 Selección del Gestor de Bases de Datos a Utilizar.

Teniendo en cuenta las particulares de cada uno de los gestores de bases de datos, se llegó a la conclusión, que el más idóneo a utilizar en este trabajo de diploma es **PostgreSQL** pues es un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. PostgreSQL es más completo que MySQL ya que permite métodos almacenados, restricciones de integridad, vistas, etc. aunque las últimas versiones de MySQL se han hecho grandes avances en ese sentido. MySQL posee diversas ventajas dentro de las cuales está su alto rendimiento, su bajo coste, su facilidad de configuración y aprendizaje, su accesibilidad a código fuente, entre otras. Mientras que PostgreSQL intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase. También PostgreSQL ofrece una garantía de integridad en los datos mucho más fuerte que MySQL.

Otra característica es que aporta potencia y flexibilidad adicional como son las restricciones (constraints), disparadores (triggers), reglas (rules) e integridad transaccional.

2.7 Lenguaje de programación.

Lenguaje de programación no es más que un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas. (Mitecnologico)

Existen muchos lenguajes de programación, para desarrollar aplicaciones tanto de escritorio como web, sin embargo, para este trabajo de diploma fueron tomados en consideración solo tres de ellos: Java, C++ y PHP.

2.7.1 Java.

Java es un lenguaje de programación orientado a objetos, el cual ha sido concebido, desde sus orígenes, Es un lenguaje capaz de producir código totalmente transportable, puede ser ejecutado en múltiples plataformas, es uno de los escasos lenguajes cuyos programas pueden ser transportados de sistema operativo, computadora o entorno, sin necesidad de cambiar el código, es un lenguaje sencillo, fácil de aprender. Java es dinámico, se adapta con suma facilidad a los cambios en los componentes fundamentales del lenguaje

Posee un conjunto grande de clases, ordenadas en paquetes, que se pueden usar en los programas. La programación de Java permite el desarrollo de programas de rendimiento seguro y alto en las plataformas múltiples, esto hace que sea robusto, puesto que no permite el manejo directo de memoria. Java resuelve los problemas de seguridad y limitantes tecnológicas al proponer una nueva forma de hacer las cosas. Java tiene una característica adicional: es multihilos, lo que le permite atender eficientemente varios procesos a la vez.

Los programas escritos en Java pueden ejecutarse en cualquier arquitectura, ya que no se genera un programa ejecutable para cada una de estas, sino un archivo de arquitectura neutral con códigos que no dependen de algún tipo de procesador específico.

2.7.2 C++.

El C++ es un lenguaje híbrido, que permite programar tanto en estilo procedimental (como si fuese C), como en estilo orientado a objetos, como en ambos a la vez. Este lenguaje se puede emplear mediante programación basada en eventos para crear programas que usen interfaz gráfica de usuario. (Mora, 2006)

C++ es a la vez un lenguaje procedural (orientado a algoritmos) y orientado a objetos. Como lenguaje procedural se asemeja al C y es compatible con él. Como lenguaje orientado a objeto se basa en la filosofía completamente diferente, que exige del programador un completo cambio de mentalidad. (Jalón, y otros)

Entre las características que C++ posee se encuentran:

- ♦ C++ es un lenguaje versátil, potente y general.
- ♦ C++ presenta una gran versatilidad es decir, es un lenguaje de propósito general, por lo que se puede emplear para resolver cualquier tipo de problema.
- ♦ Es un lenguaje que está estandarizado y un mismo código fuente se puede compilar en diversas plataformas.

- ✦ El C++ es uno de los lenguajes más rápidos en cuanto ejecución.

El C++ una de las desventajas que presenta es que obliga a hacer todo el trabajo manualmente al igual que el lenguaje de programación C, lo que provoca a los desarrolladores dificultades en su aprendizaje.

El C++ mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original. (Jalón, y otros)

2.7.3 PHP 5.3.

PHP (acrónimo de Hypertext Preprocessor) es un lenguaje "del lado del servidor" (esto significa que PHP funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas Web dinámicas. Este puede ser incluido con facilidad dentro del código HTML, y permite una serie de funcionalidades extraordinarias.

PHP está disponible para una gran cantidad de sistemas operativos diferentes. Puede escribir código PHP en todos los sistemas operativos gratuitos del tipo Unix, como Linux y FreeBSD, versiones comerciales de Unix, como Solaris e IRIX o en las diferentes versiones de Microsoft Windows. (Welling, y otros, 2003)

Entre sus características fundamentales están:

- ✦ PHP posee gran número de funciones predefinidas. A diferencia de otros lenguajes de programación.
- ✦ PHP está dotado de un gran número de funciones que simplifican enormemente tareas habituales, como descargar documentos, enviar correos, trabajar con cookies y sesiones, entre otras funciones.
- ✦ PHP dispone de una amplia gama de librerías, esto permite que PHP pueda ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras, entre las que se encuentra la poderosa librería PHPEXCEL, la misma brinda gran cantidad de funcionalidades sobre el manejo con documentos Excel.
- ✦ PHP es un lenguaje de secuencia de comandos de servidor, diseñado específicamente para la Web. Dentro de una página Web puede incrustar código PHP que se ejecutará cada vez que se visite una página. El código PHP es interpretado en el servidor Web y genera código HTML y otro contenido que el visitante verá.

- ✦ PHP es un producto de código abierto, lo que quiere decir que se puede acceder a su código. Puede ser utilizado, modificado y redistribuido sin coste alguno.
- ✦ PHP dispone de una conexión propia a todos los sistemas de base de datos.

2.8 Selección del lenguaje de programación a utilizar.

Java es un lenguaje multiplataforma, lo que significa que no está enlazado a un sistema operativo en concreto y los programas desarrollados con el mismo funcionan correctamente tanto en Windows como en Linux, este solo necesita de la máquina virtual.

Además, el entorno de desarrollo Java, posee un conjunto de bibliotecas, como es **JExcelAPI**, la cual permite a los desarrolladores de Java leer, escribir y modificar las hojas de cálculo Excel de forma dinámica. Pero esta presenta la desventaja que solo lee los libros creados en Excel 95,97 y 2003 y generará los libros que pueden ser leídos por Excel 97 y versiones posteriores, lo que no facilitará el trabajo con los documentos Excel, ya que son varios los formatos que se reciben en el Laboratorio de Lodo y Cemento.

De igual forma, en C++ existe una biblioteca que permite la lectura y escritura de archivos xls, sin necesidad de Microsoft Excel, además es fácil de utilizar. **LibXL** es una potente librería que posee disímiles características, la misma se puede utilizar para extraer datos de una hoja de cálculo existente, editar una hoja de cálculo y generar una nueva hoja de cálculo desde cero. Esta librería está protegida por leyes de derecho de copia y derechos de autor internacionales tratados, así como otras leyes de propiedad intelectual, por lo que no es recomendable hacer uso de la misma pues se tendría que pagar por el uso de la licencia.

PHP al igual que Java es un lenguaje orientado a objetos, posee abundante documentación, por lo que soporta las características esenciales del paradigma de la programación orientada a objeto: encapsulación, herencia y polimorfismo, además PHP cuenta con todas las funciones suficientes y necesarias para brindar una solución al problema planteado. PHP es código abierto, lo cual significa que el usuario no depende de una compañía específica para mejorar algunas funcionalidades. PHP se puede obtener en la Web y su código está disponible bajo la licencia GPL. El uso de PHP está muy extendido entre las empresas de desarrollo, además de tener compatibilidad con el Visual Paradigm y PostgreSQL que son la herramienta CASE y el gestor de BD que se seleccionó para el diseño e implementación de este software.

En PHP existe una biblioteca que facilita el trabajo con documentos Excel, conocida como **PHPExcel**, dicha librería es software libre, la misma puede ser redistribuida y/o modificada bajo los términos de la GNU Lesser General Public (GNU LGPL), Licencia publicada por la Free Software Foundation. La LGPL está destinada más a bibliotecas de software que a ejecutables. Esta librería permite leer y escribir distintos tipos de formatos, por ejemplo Excel 2007, PDF, HTML, entre otros.

Por todo lo planteado anteriormente se decide seleccionar **PHP** para el desarrollo del componente, debido a las ventajas que el mismo proporciona.

2.9 Librerías a utilizar.

2.9.1 jQuery.

jQuery es una librería de JavaScript, rápida y concisa que facilita la navegación de un documento HTML, manipulación de eventos, animación e interacción Ajax para desarrollos web rápidos, es una librería pensada para interactuar con los elementos de una web por medio del DOM (Document Object Model). Lo que la hace tan especial es su sencillez y su reducido tamaño. jQuery fue creada por John Resig, es un proyecto Open-Source el cual fue liberado en enero del año 2006.

jQuery es una librería liviana, la misma posee diferentes características, dentro de las que se encuentran:

- ♦ jQuery es compatible con los navegadores, Firefox 2.0+, Safari 2.0.2+, Opera 9+, Internet Explorer 6+.
- ♦ jQuery usa las licencias MIT y GPL permitiendo su uso en proyectos autónomos y privativos.
- ♦ jQuery accede al documento html.
- ♦ Maneja eventos de los elementos de la página.
- ♦ Manipula estilos CSS.
- ♦ Posee componentes GUI como cuadro de dialogo, tablas, paneles colapsables, calendarios, entre otros.
- ♦ Permite manipular cualquier elemento de un documento HTML.
- ♦ Permite agregar efectos visuales a la mayoría de los elementos de un documento html.
- ♦ Permite trabajar en modalidad AJAX y para esto nos provee algunas funciones para manipular HTML o JSON (JavaScript Object Notation).

- ✦ jQuery entrega una serie de utilidades para programar de forma más limpia y libre de errores.
- ✦ jQuery permite codificar de forma simple, separada y en el paradigma de programación invasiva.
- ✦ Permite extender la librería de una forma muy simple, esto es conocido como plugins jQuery.
- ✦ jQuery está diseñado para cambiar la forma que escribes JavaScript.

Resumiendo jQuery provee muchas características nuevas de JavaScript, sintaxis simple de aprender, una vigorosa compatibilidad de plataforma cruzada y todo esto es un pequeño archivo jquery.js.

2.9.2 PHPEXcel

PHPEXcel es una biblioteca que provee un conjunto de clases para el lenguaje de programación PHP. Esta librería se basa en el estándar de OpenXML de Microsoft y PHP.

La misma es una potente librería que posee diferentes características como son: la creación de hojas de cálculo con metadatos (autor, título, descripción,...), configurar los tipos de fuentes, bordes de celdas, rellenos, gradientes, añadir imágenes, definir los estilos de las imágenes, combinar celdas, añadir hojas de trabajo a hoja de cálculo, agregar datos y fórmulas a celdas individuales, proteger rangos de celdas con una contraseña, soporta formato para celdas, diferentes tipos de letra, estilos de fuente y tipos de datos para las celdas individuales, admite cadenas de texto enriquecido, grupos de filas, columnas, celdas de validación de datos, insertar, eliminar filas, columnas; calcular los valores de la fórmula, añadir comentarios a una celda, entre otras funcionalidades. Al igual que el software de hoja de cálculo de escritorio, PHPEXcel representa una hoja de cálculo que contiene una o más hojas de cálculo, que contienen celdas con datos, fórmulas, entre otras propiedades. (Balliauw, 2010)

PHPEXcel, cuenta con una clase de este mismo nombre, que es el núcleo de sí misma, también contiene referencias a las hojas de cálculo que genera, la configuración de la seguridad y los datos del documento en cuestión.

El paquete PHPEXcel proporciona algunos lectores y escritores de datos, entre ellos uno para el formato de hoja de cálculo Open XML (también conocido como formato de Excel 2007 de archivos). (Balliauw, 2010)

2.10 Servidor Web.

Un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI, seguridad SSL y páginas activas del servidor (ASP) entre otros. (i-Concepts)

2.10.1 Servidor Web Apache 2.0.

Apache, es un servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es un servidor que posee disímiles de características como:

- ♦ Multiplataforma.
- ♦ Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.
- ♦ Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache.
- ♦ Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Igualmente trabaja con Java y páginas jsp, teniendo todo el soporte que se necesita para tener páginas dinámicas.
- ♦ Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Apache se puede configurar, para que ejecute un determinado script cuando ocurra un error en concreto. (Ciberaula)

2.10.2 Servidor Web Internet Information Server (IIS).

Internet Information Server (IIS) es el servidor de páginas web avanzado de la plataforma Windows. IIS incluye una serie de servicios para los ordenadores que funcionan con Windows, como son: FTP, SMTP, NNTP y HTTP/HTTPS, convirtiendo un ordenador en un servidor de Internet, es decir, que las computadoras que tiene este servicio instalado pueden publicar páginas web, tanto local como remotamente (servidor web).

El servidor web se basa en varios módulos que dan la capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y

ASP.NET. Del mismo modo,, pueden ser incluidos los de otros fabricantes, como PHP o Perl.

IIS es compatible con un gran número de interfaces que pueden utilizarse para desarrollar otras características un sitio Web.

Internet Information Server incluye los siguientes componentes:

- ✦ Servicios de Internet: WWW, FTP y gopher.
- ✦ Administrador de servicios de Internet, la herramienta para administrar los servicios de Internet.
- ✦ Conector de bases de datos de Internet, el componente para enviar consultas a bases de datos.
- ✦ Administrador de claves, la herramienta para la instalación de las claves de Secure Sockets Layer (SSL).

Internet Information Server posee la autenticación implícita, que permite a los administradores autenticar a los usuarios de forma segura a través de servidores de seguridad y proxy.

Microsoft ha mejorado fundamentalmente su software en el campo de los servicios Web. Los avances vienen motivados sobre todo por la seguridad y el rendimiento pero todavía padece de algunas brechas de seguridad, IIS solo puede operar en plataformas Windows.

2.11 Selección del servidor web a utilizar.

A partir de un estudio realizado, se escogió como Servidor Web, **Apache 2.0** primeramente por su aceptación universal y gran potencialidad. Al mismo tiempo puede ser adaptado a diferentes entornos y necesidades, este goza de diferentes módulos de apoyo y con la API de programación de módulos. Apache es un servidor fácil de configurar y de instalar, posee una arquitectura muy modular, es decir, consta de muchas de las funcionalidades básicas de un servidor web son provista por módulos.

Mientras que Internet Information Server (IIS) es propietario, ideal para aplicaciones sobre Windows lo que no constituye un aspecto a considerar según los objetivos propuestos, por otra parte, Apache posee una licencia que es descendiente de la licencia BSD (Berkeley Software Distribution), no es GPL, lo que permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.

2.12 Ambiente de desarrollo.

Un **entorno de desarrollo integrado** (en inglés Integrated Development Environment o IDE) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. (Petra.eutio)

En la actualidad los entorno de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python, PHP y Visual Basic, entre otros). Los entornos de desarrollo (IDEs) han incorporado a medida que evolucionan características de integración multi-código y multi-plataforma.

Existen muchos entornos de desarrollo, sin embargo, para este trabajo de diploma fueron tomados en consideración solo dos de ellos: NetBeans 6.8 y Eclipse, que gozan de reconocimiento en el mundo.

2.12.1 NetBeans 6.8.

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

El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos, así como una selección diversa de complementos de terceros. (NetBeans)

NetBeans es un entorno multiplataforma para crear aplicaciones Java. Es la plataforma que más módulos incluyen tales como un editor, herramientas para trabajar con el código fuente (Java, C++ y otros), control de versiones. NetBeans puede sincronizar su modelo en Visual Paradigm y el código en NetBeans. (Visual-Paradigm)

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java

que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest) que lo identifica como módulo.

La plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están la administración de las interfaces de usuario, administración de las configuraciones del usuario, administración del almacenamiento, administración de ventanas, frameworks basado en asistentes, entre otros. (Wiccsi)

2.12.2 Eclipse.

Eclipse es un entorno de desarrollo integrado (IDE, Integrated Development Environment), que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. (Chápela Martínez, Jairo, 2007)

Eclipse es un IDE abierto y multiplataforma para desarrollar lo que el proyecto llama Aplicaciones de Cliente Enriquecido (RCA) opuesto a las Aplicaciones Cliente Liviano (LCA) basada en navegadores.

El IDE de Eclipse emplea módulos para proporcionar toda su funcionalidad, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, se necesiten o no.

Eclipse puede extenderse incorporando otros lenguajes de programación como son C/C++, generando desarrollos como el C-C++ Development Toolkit (CDT), también brinda soportes para Python, PHP, entre otros lenguajes de programación. (Wiccsi)

Eclipse representa una tecnología probada, fiable y escalable en la que diseñar, desarrollar e implementar productos comerciales (Cla, 2009). Eclipse sirve como IDE para Java y cuenta con numerosas herramientas de desarrollo de software. También da soporte a otros lenguajes de programación, como son C/C++, Cobol, Fortran, PHP o Python. A la plataforma base de Eclipse se le pueden añadir extensiones (plugins) para extender la funcionalidad.

Eclipse se distribuye bajo licencia EPL (Eclipse Public License). Esta licencia es considerada como libre por la Free Software Foundation (FSF) y por la Open Source Initiative (OSI). La licencia EPL permite usar, modificar, copiar y distribuir nuevas versiones del producto licenciado.

2.13 Selección del ambiente de desarrollo a utilizar.

Como ambiente de desarrollo a utilizar se seleccionó el **NetBeans 6.8** el cual esta basado en código abierto y en una plataforma de aplicación de escritorio genérica. NetBeans está escrito en Java pero puede servir para cualquier otro lenguaje de programación, también posee numerosas características que hacen que el IDE sea atractivo para cualquier desarrollador, incluyendo la amplia integración de las características específicas de la tecnología Java que no se encuentran disponibles en otros conjuntos de herramientas de aplicaciones multiplataforma.

NetBeans 6.8 permite generar informes en HTML o en Map y los informes de uso sencillo incluyen diagramas y pueden colocarse fácilmente en intranets o pueden imprimirse para su revisión, además está integrado con la herramienta CASE Visual Paradigm for UML Enterprise Edition.

2.14 Conclusiones.

En este capítulo se profundizó en el análisis de las características fundamentales de las tecnologías, lenguajes de programación y sistemas gestores de bases de datos candidatos para la implementación de la propuesta de este trabajo. Además, se tuvo como principal elemento a considerar en cada selección la compatibilidad con el sistema operativo Linux, además de que sean herramientas y tecnologías libres, o de código abierto.

Finalmente, se llegó a la conclusión de que para la realización del componente de software se empleará Agile Unified Process (AUP) como Metodología a seguir, UML 2.0 como lenguaje de modelado para especificar, construir y documentar los artefactos del sistema y Visual Paradigm for UML Enterprise Edition 6.4 como herramienta de modelado. El lenguaje a utilizar es PHP, el cual es un lenguaje multiplataforma y a la vez potente, el servidor será APACHE 2.0 por las características que este posee, además es un servidor que puede ser adaptado a diferentes entornos y necesidades. También se empleará jQuery como librería de JavaScript, esta permite una interfaz más amigable, para aplicaciones de escritorio, y es software libre y de código abierto, y PHPEXCEL para el manejo de documentos Excel. El entorno de desarrollo más adecuada para la implementación es NetBeans 6.8, pues permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software, es un proyecto de código abierto, es decir, un producto libre y gratuito sin restricciones de uso, como gestor de bases de datos PostgreSQL por las grandes ventajas que este tiene.

CAPÍTULO 3: Presentación de la solución propuesta.

3.1 Introducción.

Para crear un sistema con calidad se hace necesario percibir la estructura y dinámica de la empresa u organización en cuestión y los procesos que en ella tienen lugar, con el objetivo de alcanzar una visión de las dificultades existentes, además de obtener un entendimiento común entre clientes y desarrolladores.

En este capítulo se efectúa la modelación del negocio, el cual posibilita obtener una vista más clara de los procesos que tienen lugar en la actualidad. Se hace una descripción de actores y trabajadores del negocio y se presenta el modelo de objetos. También se realiza una descripción de la solución propuesta, planteándose los requisitos funcionales y no funcionales del sistema a desarrollar, el diagrama de casos de uso del sistema y la descripción de los casos de uso críticos.

3.2 Actores del negocio.

Un Actor del Negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa. Un actor es el rol que juega cuando interactúa con el negocio, beneficiándose de sus resultados. (Jacobson, y otros, 2004)

Actor del negocio.	Justificación
DIP de Perforación.	Este es el actor del negocio encargado de solicitar al Laboratorio de Lodo y Cemento del CEINPET, un informe final de los reportes ya procesados.

Tabla 2 Actor del negocio.

3.3 Trabajadores del negocio.

Un Trabajador del Negocio es una abstracción de una persona, o grupo de personas, una máquina o un sistema automatizado que actúa con el negocio realizando una o varias actividades, colaborando con otros trabajadores del negocio y manipulando entidades del negocio. (Jacobson, y otros, 2004)

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

Trabajadores del negocio.	Justificación
Supervisor de Perforación de las Compañías de Servicios.	Este es el trabajador del negocio encargado de enviar al Laboratorio de Lodo y Cemento del CEINPET, los reportes diarios del trabajo de las Compañías de Servicio con los fluidos de perforación.
Especialista Principal de Lodo y Cemento.	Este es el trabajador del negocio encargado dentro del Laboratorio de Lodo y Cemento del CEINPET, de recibir diariamente los reportes procedentes de las diferentes compañías que se dedican a la perforación de los pozos y de procesar los datos de los mismos.

Tabla 3 Trabajadores del negocio.

3.4 Descripción del Negocio.

La DIP de perforación le solicita al CEINPET la evaluación y control del trabajo de las Compañías de Servicio en los pozos petroleros en cuanto a la utilización de los fluidos de perforación, donde se determinó que el Laboratorio de Lodo y Cemento del CEINPET sería el encargado de llevar a cabo este proceso.

Inicialmente los Supervisores de Perforación de las Compañías de Servicios, envían los reportes de cada pozo petrolero en perforación, en los cuales se almacenan las propiedades de los fluidos, los datos del volumen que se encuentran tanto en el proceso de circulación como en los tanques de la superficie y de reserva, los datos del consumo de aditivos, las características de las herramientas de perforación, la profundidad inicial y final en la que se encuentra el pozo, entre otros.

Estos reportes son recibidos en el Laboratorio de Lodo y Cemento por el Especialista Principal de Lodo y Cemento, el cual procede a verificar si los datos están correctos, además de obtener los datos relativos a los fluidos, los cuales se organizan en tablas según correspondan las propiedades reológicas, de filtrado, las físico-químicas entre otras propiedades. Con todos estos aspectos se evalúa el comportamiento de estos parámetros realizando gráficos y ubicando todos los parámetros mencionados dentro de intervalos requeridos, especificados, óptimos o indeseables y así dar un dictamen del

cumplimiento de los objetivos del servicio, o de la perforación en sí con vista a optimizar la actividad.

Después que se tiene el veredicto final el Especialista Principal de Lodo y Cemento emite un informe final de los reportes ya analizados el cual será enviado a la DIP de Perforación.

3.5 Diagrama de Casos de Uso del Negocio.

El diagrama de Caso de Uso del Negocio figura gráficamente los procesos del negocio que tienen lugar en una empresa y la interacción entre los actores del negocio y dichos procesos. (Jacobson, y otros, 2004)

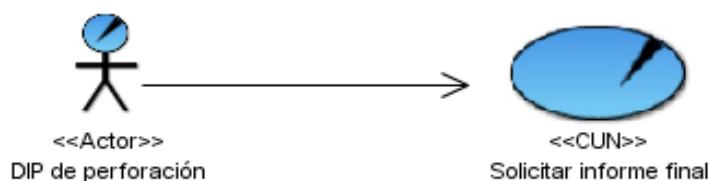


Fig. 2 Diagrama de Casos de Uso del Negocio.

3.6 Descripción textual del Caso de Uso de Negocio.

Los Casos de Uso del Negocio consisten en la descripción de las secuencias de actividades que, en conjunto, producen algo para el actor del negocio. El proceso consiste en un flujo básico de una o más alternativas de flujos.

3.6.1 Caso de Uso: Solicitar Informe Final.

Caso de Uso:	Solicitar Informe Final
Actores:	DIP de Perforación.
Resumen:	<p>El caso de uso se inicia cuando la DIP de Perforación le solicita al Laboratorio de Lodo y Cemento del CEINPET, un informe final de los reportes de los fluidos de perforación.</p> <p>El Supervisor de Perforación de las Compañías de Servicios envía los reportes diarios de los fluidos de perforación al Laboratorio de Lodo y Cemento, donde el Especialista Principal de Lodo y Cemento es el encargado de recibir estos reportes y de</p>

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

	<p>procesarlos. Este obtiene los datos relativos a los fluidos, los cuales se organizan en tablas según sus propiedades reológicas, de filtrado, propiedades físico-químicas y así sucesivamente, con esta información se elabora el informe final para enviárselo a la DIP de Perforación.</p>
Flujo Normal de Eventos	
Acción del Actor	Respuesta
<p>1. La DIP de Perforación le solicita al Laboratorio de Lodo y Cemento del CEINPET, el Informe Final del procesamiento de los reportes de fluidos de perforación diariamente.</p>	<p>2. El Supervisor de Perforación de las Compañías de Servicios envía los reportes diarios de los fluidos de perforación al Especialista Principal de Lodo y Cemento el cual recibe los reportes.</p> <p>3. El Especialista Principal de Lodo y Cemento, verifica que los datos de los reportes estén correctos. Ver Flujos Alternos 3.1. <i>Datos Incorrectos</i>.</p> <p>4. El Especialista Principal de Lodo y Cemento obtiene los datos relativos a los fluidos.</p> <p>5. El Especialista Principal de Lodo y Cemento evalúa el comportamiento de los fluidos de perforación.</p> <p>6. El Especialista Principal de Lodo y Cemento, organiza en tablas los datos según corresponda, por sus propiedades reológicas (tensiones de corte a diferentes velocidades, viscosidad plástica, punto de cedencia, geles, etc.), las de filtrado (filtrado API, filtrado HPHT, revoque), las propiedades físico-químicas (concentración de cationes, de aniones, etc.) y así sucesivamente.</p> <p>7. El Especialista Principal de Lodo y Cemento realiza gráficas para evaluar el comportamiento de los parámetros.</p> <p>8. El Especialista Principal de Lodo y Cemento ubica todos los parámetros mencionados dentro de</p>

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

	<p>los intervalos requeridos (especificados, óptimos o indeseables).</p> <p>9. El Especialista Principal de Lodo y Cemento concluye con la evaluación final del cumplimiento de los objetivos del servicio, o de la perforación en vista a optimizar la actividad.</p> <p>10. El Especialista Principal de Lodo y Cemento, consulta los resultados anteriores, para elaborar y enviar el informe.</p>
11. La DIP de Perforación recibe el Informe Final.	
Flujos Alterno 3.1. Datos Incorrectos	
Acción del Actor	Respuesta
	<p>3.1. El Especialista Principal de Lodo y Cemento informa al Supervisor de Perforación de las Compañías de Servicios que existen errores en el reporte enviado.</p> <p>3.2 El Supervisor de Perforación de las Compañías de Servicios corrige los errores del reporte.</p>

Tabla 4 Descripción textual del Caso de Uso de Negocio "Solicitar Informe Final".

3.7 Reglas del negocio.

En un proceso de negocio existen reglas de negocio las mismas detallan políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que sistematizan algún aspecto del negocio.

En el proceso de negocio actual se definieron varias reglas, las cuales se muestran a continuación:

- ✦ Entrega de un informe diario de los reportes de fluidos de perforación por parte del Laboratorio de Lodo y Cemento del CEINPET a la DIP de Perforación.
- ✦ Envío de los reportes de perforación por el Supervisor de Perforación de las Compañías de Servicios al Especialista Principal de Lodo y Cemento.

- ✦ Envío de los reportes diarios al Especialista Principal de Lodo y Cemento del CEINPET.
- ✦ La información que se almacena en los reportes no puede presentar errores.

3.8 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, para que las peticiones del cliente queden satisfechas.

A continuación se muestran los requerimientos funcionales del sistema:

RF1. El sistema debe permitir cargar documentos Excel.

RF2. El sistema debe permitir importar documentos Excel.

RF3. El sistema debe permitir cargar plantillas de información.

RF4. El sistema debe permitir eliminar plantillas de información.

RF5. El sistema debe permitir guardar plantillas de información.

3.9 Requerimientos No Funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, además son aspectos importantes que el producto debe cumplir para lograr un producto atractivo, usable, rápido o confiable (Pressman, 2005). A continuación se enuncian, separados en categorías, los diferentes requisitos no funcionales que el componente debe satisfacer.

A continuación se muestran los requerimientos no funcionales del sistema:

- ✦ **RNF1. Usabilidad.**

RNF1.1. El sistema solo será utilizado por aquellas personas que tengan relación con la información que se maneja y que tengan los conocimientos básicos para el manejo de aplicaciones Web, además contara con un manual de usuario, con instrucciones de tipo paso a paso, para entender el trabajo del sistema.

- ✦ **RNF2. Restricciones de diseño e implementación.**

RNF2.1. Para la implementación del sistema se utilizará el lenguaje de programación PHP 5.3.

RNF2.2. El IDE a usar será el NetBeans 6.8.

RNF2.3. Se utilizará el Visual Paradigm 6.4 como herramienta CASE para el modelado del sistema.

♦ **RNF 3. Apariencia o Interfaz Externa.**

RNF3.1. La interfaz principal del sistema tendrá las dimensiones mínimas de 1024x768 píxeles para una buena visualización y adaptación a diferentes resoluciones de pantalla.

RNF3.2. La interfaz principal contará con el logotipo del sistema.

RNF3.3. La interfaz debe ser agradable, que combine correctamente los colores, tipo de letra y tamaño y que los iconos estén en correspondencia con lo que representan.

♦ **RNF4. Portabilidad.**

RNF4.1. El sistema no solo correrá sobre Windows sino también sobre Linux.

♦ **RNF 5. Software.**

RNF 5.1. Para el servidor Web/BD se requiere de una computadora que tenga sistema operativo Microsoft Windows XP o Linux, Servidor Apache, Gestor de Base de Datos PostgreSQL, PHP 5.3.

RNF 5.2. Para las maquinas clientes se requiere de una computadora que tenga, instalado un navegador ya sea Mozilla Firefox, Opera, Safari o Google Chrome y como sistema operativo Microsoft Windows (en cualquiera de sus versiones) o GNU/Linux (en cualquiera de sus distribuciones).

♦ **RNF 6. Hardware.**

RNF 6.1. El sistema estará instalado en un Servidor, donde se encuentra el Servidor Web y el Gestor de Base de Datos.

Requisitos Mínimo:

- ♦ Procesador: 800 MHz, Memoria: 256 MB, Disco Duro: 2.5 GB, Tarjeta de Red: Ethernet 10/100 MB.

Requisitos Recomendados:

- ♦ Procesador: 1.8 GHz, Memoria: 512 MB, Disco Duro: 5 GB, Tarjeta de Red: Ethernet 10/100 MB.

RNF 6.2. El sistema deberá ser operado en máquinas clientes con las siguientes características:

Requisitos Mínimo:

- ♦ Procesador: 300 MHz, Memoria: 128 MB, Disco Duro: 2.5 GB, Tarjeta de Red: Ethernet 10/100 MB.

Requisitos Recomendados:

- Procesador: 1.8 GHz, Memoria: 512 MB, Disco Duro: 5 GB, Tarjeta de Red: Ethernet 10/100 MB.

3.10 Descripción del Sistema Propuesto.

3.10.1 Descripción del Actor del Sistema.

Cada trabajador del negocio que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema. (Pressman, 2005)

Actor del sistema.	Descripción
Especialista Principal de Lodo y Cemento.	Este es el actor del sistema encargado de importar los reportes de perforación de diferentes compañías de servicio.

Tabla 5 Descripción del Actor del Sistema.

3.10.2 Diagrama de Casos de Uso del Sistema.

Un diagrama de Casos de Uso del Sistema describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores con una asociación entre cada par actor/caso de uso que interactúan. (Jacobson, y otros, 2004)

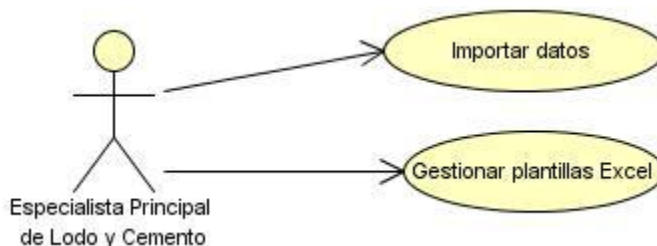


Fig. 3 Diagrama de Casos de Uso del Sistema.

3.11 Descripción textual de los Casos de Uso del Sistema

Un caso de uso es una unidad coherente de funcionalidad, externamente visible, proporcionada por una unidad del sistema y expresada por secuencias de mensajes intercambiados por la unidad del sistema y uno o más actores. El propósito de un caso de uso es definir una pieza de comportamiento coherente, sin revelar la estructura interna del sistema. Desde el punto de vista de los usuarios, éstas pueden ser situaciones anormales. Desde el punto de vista de los sistemas, son las variaciones adicionales que deben ser descritas y manejadas. (Rumbaugh, y otros, 2005)

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

Caso de Uso:	Importar datos.
Actores:	Especialista Principal de Lodo y Cemento.
Resumen:	El caso de uso se inicia cuando el Especialista Principal de Lodo y Cemento desea importar los datos de un nuevo reporte hacia el gestor de Base de Datos PostgreSQL.
Precondiciones:	-
Referencias:	RF1, RF2
Prioridad:	Crítica
Flujo Normal de Eventos.	
Acción del Actor	Respuesta del Sistema
1. El Especialista Principal de Lodo y Cemento selecciona la opción de Conectar.	2. El sistema visualiza la interfaz para conectarse a una base de datos.
3. El Especialista Principal de Lodo y Cemento procede a introducir los datos para conectarse a la bases de datos, en caso contrario el sistema muestra la interfaz principal.	4. El sistema se conecta a la bases de datos y muestra la interfaz principal.
5. El Especialista Principal de Lodo y Cemento selecciona la opción de cargar el documento Excel.	6. El sistema visualiza la interfaz para buscar el archivo a cargar.
7. El Especialista Principal de Lodo y Cemento selecciona el archivo a cargar.	8. El sistema verifica que el archivo sea un documento Excel, en caso contrario: ver flujo alternativo 1 Error de compatibilidad de formato. 9. El sistema carga el documento y lo visualiza en una interfaz amigable.
10. El Especialista Principal de Lodo y Cemento procede a insertar los datos en la bases de datos. 11. El Especialista Principal de Lodo y	13. El sistema muestra un mensaje confirmando la terminación del importe satisfactorio, en caso contrario ver flujo alternativo 2 Error de importe.

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

Cemento selecciona la opción Gestionar Plantilla: ver descripción de caso de uso: Gestionar plantillas Excel.	
12. El Especialista Principal de Lodo y Cemento selecciona la opción Importar.	
Flujo Alternativo 1 Error de compatibilidad de formato.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje reportando el error en el documento a cargar.
Flujo Alternativo 2 Error de importe.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje reportando posibles errores ocurridos durante el proceso de importe.
Poscondiciones	El documento Excel quedó almacenado en la Base de Datos.

Caso de Uso:	Gestionar plantillas Excel.
Actores:	
Resumen:	El caso de uso se inicia cuando el Especialista Principal de Lodo y Cemento desea cargar plantilla, guardarla o eliminarla para realizar el importe de datos.
Precondiciones:	
Referencias:	RF3, RF4, RF5
Prioridad:	Crítica
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1. El sistema le muestra al Especialista Principal de Lodo y Cemento las opciones de: Cargar Plantilla, Guardar Plantilla y Eliminar Plantilla
2. El Especialista Principal de Lodo y Cemento	2. En caso de que el Especialista Principal

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

selecciona la opción deseada.	de Lodo y Cemento selecciona la opción: a) “Cargar Plantilla”: ver sección “Cargar Plantilla” b) “Guardar Plantilla”: ver sección “Guardar Plantilla”. c) “Eliminar Plantilla”: ver sección “Eliminar Plantilla”.
Sección “Cargar Plantilla”.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra una interfaz con todas las plantillas guardadas.
2. El Especialista Principal de Lodo y Cemento selecciona la plantilla a cargar, en caso contrario ver flujo alternativo 1 Anulación de cargado de plantilla.	3. El sistema carga la plantilla y muestra un mensaje de información con la opción de “Aceptar” o “Cancelar”.
4. El Especialista Principal de Lodo y Cemento selecciona la opción de “Aceptar”, en caso contrario ir a la “ Flujo Normal de Eventos ”	5. El sistema carga la plantilla y muestra un mensaje satisfactorio, en caso contrario: ver flujo alternativo 2 Error de cargado de plantilla.
Flujo Alternativo 1 Anulación de cargado de plantilla.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje alertando que no se seleccionó ninguna plantilla.
Flujo Alternativo 2 Error de cargado de plantilla.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje reportando posibles errores ocurridos durante el proceso de carga.
Sección “Guardar Plantilla”.	
Acción del Actor	Respuesta del Sistema
2. El Especialista Principal de Lodo y Cemento accede a escribir el nombre de la nueva	1. El sistema muestra una interfaz para nombrar la nueva plantilla a guardar.

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

plantilla a guardar, en caso contrario ver flujo alternativo 1 Anulación de guardado de plantilla.	
	<p>3. El sistema verifica que el campo de texto no esté vacío, en caso contrario ver flujo alternativo 2 Error de guardado de plantilla.</p> <p>4. El sistema verifica de que no exista una plantilla con el mismo nombre, en caso contrario ver flujo alternativo 3 Error de existencia de guardado de plantilla.</p>
5. El Especialista Principal de Lodo y Cemento selecciona la opción “Aceptar” en caso contrario: ver flujo alternativo 1 Anulación de guardado de plantilla.	6. El sistema muestra un mensaje de información confirmando la realización satisfactoria del proceso de guardado.
Flujo Alternativo 1 Anulación de de guardado de plantilla.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje de información que el guardado de plantilla ha sido cancelado.
Flujo Alternativo 2 Error de guardado de plantilla.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje de información que el nombre de la plantilla es obligatorio para guardarla.
Flujo Alternativo 3 Error de existencia de guardado de plantilla.	
Acción del Actor	Respuesta del Sistema
	<p>2. El sistema muestra un mensaje reportando que la plantilla ya existe.</p> <p>3. El sistema muestra un mensaje de información con la opción de “Aceptar” o “Cancelar”.</p>
4. El Especialista Principal de Lodo y Cemento selecciona la opción “Aceptar”	5. El sistema muestra un mensaje reportando que la plantilla se ha

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

en caso contrario ver flujo alternativo 1 Anulación de guardado de plantilla.	guardado correctamente.
Sección “Eliminar Plantilla”.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra una interfaz para seleccionar la plantilla a eliminar.
2. El Especialista Principal de Lodo y Cemento selecciona la plantilla a eliminar. 3. El Especialista Principal de Lodo y Cemento selecciona la opción eliminar.	4. El sistema muestra un mensaje de información con la opción de “Aceptar” o “Cancelar”.
5. El Especialista Principal de Lodo y Cemento selecciona la opción de Aceptar, en caso contrario ver flujo alternativo 1 Anulación de la eliminación.	6. El sistema elimina la plantilla y muestra un mensaje confirmando la realización satisfactoria de dicho proceso en caso contrario: ver flujo alternativo 2 Error de eliminación de plantilla.
Flujo Alternativo 1 Anulación de la eliminación.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje reportando que la eliminación de la plantilla ha sido cancelada.
Flujo Alternativo 2 Error de eliminación de plantilla.	
Acción del Actor	Respuesta del Sistema
	1. El sistema muestra un mensaje reportando posibles errores ocurridos durante el proceso de eliminación.
Poscondiciones	Se cargo la plantilla, se guardo o elimino correctamente.

3.12 Conclusiones.

Siguiendo la metodología de desarrollo AUP, en este capítulo se hizo una descripción detallada de los actores, trabajadores del negocio, de los casos de uso del negocio, y de las reglas del negocio. También se presentó el diagrama de actividades correspondiente al caso de uso del negocio. Se sintetizaron los principales

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.

requerimientos funcionales y los no funcionales, con el objetivo de hacer fácil la construcción del sistema.

Con todos los elementos expuestos en este capítulo, se puede proceder a desarrollar el diseño y la construcción del sistema, vigilando que se cumplan todas las funcionalidades anteriormente analizadas.

Capítulo 4. Construcción de la solución propuesta.

4.1 Introducción.

Luego de haber definido las herramientas a utilizar y las funcionalidades del componente, se debe proseguir al Diseño e Implementación del mismo. En el siguiente capítulo se definen importantes parámetros para la construcción de la solución propuesta, a partir de los aspectos que se trataron en el capítulo anterior. Además se explica la arquitectura empleada y el conjunto de patrones de diseño usados para garantizar la viabilidad y calidad requeridas del componente a construir. De igual forma se analiza la fase de implementación a partir de los resultados del diseño, describiendo el estado actual del sistema en término de componente.

4.2 Descripción de la Arquitectura.

Según plantea **Barry Boehm**, en el libro *“Engineering Context (for Software Architecture)”*, en 1995: *“Si un proyecto no ha logrado una arquitectura del sistema, incluyendo su justificación, el proyecto no debe empezar el desarrollo en gran escala. Si se especifica la arquitectura como un elemento a entregar, se puede usar a lo largo de los procesos de desarrollo y mantenimiento.”*

La arquitectura del software alude a la estructura global del software y a las formas en que esta proporciona la integridad conceptual de un sistema. En su forma más simple, la arquitectura es la estructura jerárquica de los componentes del programa (módulos), la manera en que los componentes interactúan y la estructura de datos que van a utilizar los componentes. (Pressman, 2005)

Para la realización de este componente se definió utilizar el estilo de Llamada y Retorno, que permite construir una estructura de programa relativamente fácil de modificar y ajustar a escala, dentro de este, el patrón de diseño Modelo Vista Controlador (MVC), donde todo el proceso está dividido en 3 partes, el Modelo, la Vista y el Controlador. Este patrón se centra en la separación del modelo y la vista, mientras que el controlador es el encargado de relacionarlos. Su principal característica es aislar la vista del modelo.

El patrón MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones, es decir, se tratan

como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.

La Vista será responsable de mostrar, mediante el uso de interfaces gráficas, la información almacenada en el modelo, además será la encargada de recibir información del usuario mediante acciones efectuadas por el mismo a través de dichas interfaces.

El Controlador especifica todas las reglas de negocio que permitirán ejecutar la totalidad de las operaciones que son ordenadas desde los objetos gráficos que se han definido previamente en el componente interfaz. Para ello se debe procesar la información que toma del paquete Vista para devolverle finalmente una respuesta, de modo que el usuario sea capaz de visualizar el resultado de la operación que ha ordenado. Además, se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones.

El Modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes, también realiza el encapsulamiento de los datos almacenados en objetos accesibles por los otros dos paquetes del sistema: Controlador y Vista.

Entre las ventajas y desventajas que este estilo presenta están las siguientes:

Ventajas.

- ✦ Permite tener un completo control sobre el comportamiento de una aplicación.
- ✦ Soporte de vistas múltiples, es decir, dado que la vista se encuentra aislada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente.
- ✦ Las interfaces tienden a cambiar más rápido que las reglas de negocios. Agregar nuevos tipos de vistas no afectan al modelo.
- ✦ Permite un diseño modular, posibilitando que los diseñadores y los desarrolladores trabajen conjuntamente.

Desventajas.

- ✦ Puede aumentar un poco la complejidad de la solución, porque está guiada por eventos, puede ser algo más difícil de depurar.
- ✦ El tiempo de desarrollo de una aplicación que implementa el patrón MVC es mayor, por el alto nivel de complejidad que este posee, al introducir nuevos

niveles de indirección y por lo tanto aumenta ligeramente la complejidad de la solución.

- ✦ MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.
- ✦ Si el modelo experimenta cambios frecuentes, la vista podría desbordarse con un diluvio de requerimientos de actualizaciones.

4.3 Patrones.

En el libro **“The Timeless Way of Building”**, el Arquitecto **Christopher Alexander** define el concepto de patrón de la siguiente manera: *“Cada patrón es una regla de 3 partes, que expresa una relación entre un contexto, un problema y una solución. Como un elemento en el mundo, cada patrón es una relación entre un contexto, un sistema de fuerzas que ocurren repetidamente en ese contexto y una configuración espacial que permite que esas fuerzas se resuelvan entre sí.”*

Del mismo modo expone que: *“Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”.*

A modo de complemento un patrón es una representación o estructura determinable e identificable en diferentes dominios. Los patrones ofrecen soluciones a un problema recurrente, identificando los subsistemas, componentes y las unidades de colaboración entre ellos, al mismo tiempo, ayudan a diseñar un sistema en un breve tiempo, a resolver problemas proponiendo una solución, y facilitando la documentación, comunicación entre los desarrolladores y a aliviar la complejidad del software en varias fases en el ciclo de vida.

4.3.1 Patrones de diseño empleados.

Un patrón de diseño es una abstracción de una solución en un nivel alto. Los patrones solucionan problemas que existen en muchos niveles de abstracción. (Ciberaula, 2006)

Los patrones de diseño proporcionan un esquema para refinar los subsistemas y las relaciones entre los componentes de un sistema. Conjuntamente componen soluciones exitosas a problemas frecuentes que se presentan durante el diseño de una aplicación. Impiden cometer errores, son soluciones concretas a problemas frecuentes que se

presentan, están basados en la experiencia acumulada de expertos de desarrollo de software.

La utilización de patrones de diseño es una práctica recomendada en la ingeniería de software, pues ayuda a la estandarización, flexibilidad y robustez en la arquitectura. Cada patrón resuelve un problema, pero es importante saber combinarlos de manera adecuada cuando es necesario.

A continuación se describen los patrones que serán utilizados en el desarrollo del componente.

Patrones Creacionales ofrecen soporte a una de las tareas más comunes dentro de la programación orientada a objetos: la instanciación. Estos ayudan a crear un sistema independiente de cómo se crean, se componen y se representan sus objetos.

♦ Singleton

Garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a esta instancia.

Beneficios.

- › Acceso controlado a una única instancia, es decir, encapsula su única instancia.
- › Permite el perfeccionamiento de las operaciones y de la representación.
- › Facilita cambiar, atender y permitir más de una instancia de la clase Singleton.
- › Usar este patrón es más flexible que las clases operacionales.

♦ Registry

Permite mantener de forma global objetos que han sido registrados.

Beneficios.

- › La construcción del patrón Registry define una forma común y reutilizable para almacenar y recuperar configuraciones de datos globales y posiblemente persistentes.
- › La estructura del contenido del Registry se puede ampliar fácilmente para acomodar nuevos nombres de entradas y además nuevos tipos de entradas.
- › Permite almacenar datos de configuración en un registro global, de tal forma que sea fácil de compartir entre los diferentes módulos de la aplicación y los controladores.

Patrones de diseño **GRASP** (**G**eneral **R**esponsibility **A**signment **S**oftware **P**arterns), describen los principios esenciales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

♦ Patrón Experto.

Permite asignar una responsabilidad al experto en información.

Beneficios.

- › Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide.
- › El comportamiento se distribuye entre las clases que cuentan con la información requerida alentando con ello definiciones de clases sencillas y más cohesivas que son fáciles de comprender y mantener.

♦ Patrón Creador.

Permite asignar quien debería ser el responsable de la creación de una nueva instancia de alguna clase. Guía la asignación de responsabilidades relacionadas con la creación de objetos.

El propósito general de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento.

Beneficios.

- › Brinda soporte de bajo acoplamiento, lo cual admite menos dependencias con respecto al mantenimiento y mejores oportunidades de reutilización. Es probable que el acoplamiento no aumente, pues la clase creada tiende a ser visible a la clase creador, debido a las asociaciones actuales que llevaron a elegirla como el parámetro adecuado.

♦ Patrón Controlador

El propósito de este patrón es asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión.

Beneficios.

- › Mayor potencial de los componentes reutilizables.
- › Razonamiento sobre el estado de los casos de uso:

- › Asegurar que la secuencia es válida.
- › Razonar sobre el estado actual.

♦ **Patrón Bajo Acoplamiento.**

Este patrón especifica cómo dar soporte a una dependencia escasa y a un aumento de la reutilización, enfocándose en asignar una responsabilidad para mantener bajo acoplamiento.

El bajo acoplamiento estimula la asignación de responsabilidades de forma tal que la inclusión de éstas no incremente el acoplamiento, creando clases más independientes y con mayor resistencia al impacto de los campos, que aumentan la productividad y la posibilidad de reutilización.

El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios. Es válido precisar que este patrón no puede verse de forma independiente a los patrones Experto y Alta Cohesión, sino más bien incluirse como otro de los principios del diseño que influyen de forma determinante a la hora de la asignación de responsabilidades.

Beneficios.

- › No afectan los cambios en otros componentes.
- › Conveniente para reutilizar.
- › Fácil de entender de manera aislada.

♦ **Patrón Alta Cohesión.**

El objetivo de este patrón es asignar responsabilidades de tal forma que la cohesión siga siendo alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Beneficios.

- › Incrementa la claridad y facilidad de comprensión del diseño.
- › Simplifica el mantenimiento.
- › Soporta bajo acoplamiento. Una mala cohesión lleva a mal acoplamiento y viceversa.
- › El grano fino de funcionalidad altamente relacionada incrementa la reutilización.

4.4 Vista Lógica del Sistema.

La vista lógica describe las clases más importantes que formarán parte del ciclo de desarrollo. Representa los paquetes más abstractos del sistema y las relaciones que entre ellos existen ya sea de dependencia o de uso.

La siguiente imagen muestra la distribución de los paquetes más significativos dentro de cada una de las partes definidas para la aplicación y la dependencia entre ellos.

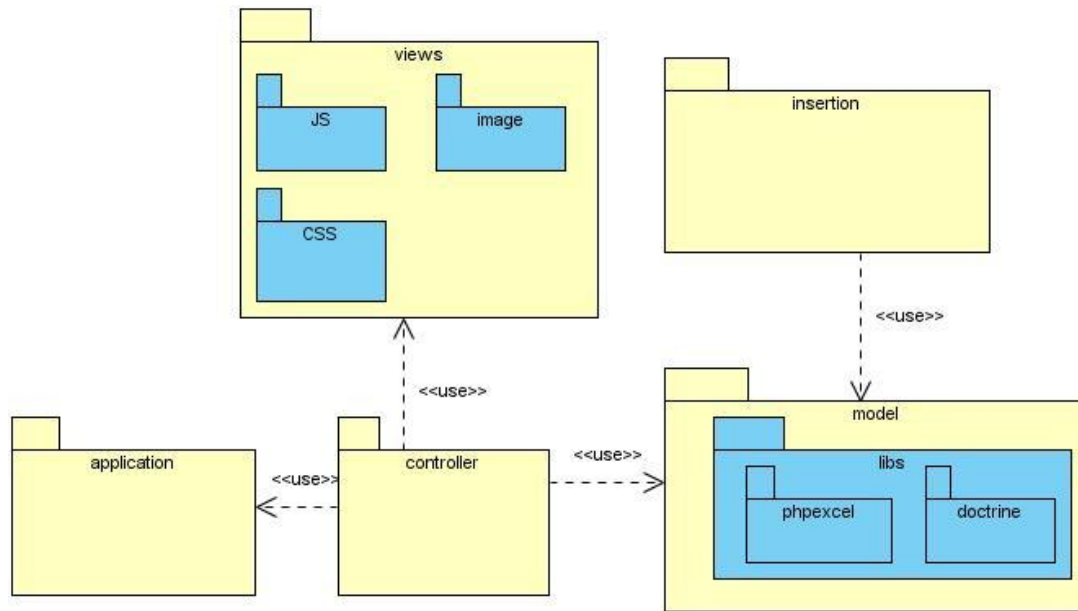


Fig. 4 Vista Lógica.

La vista está compuesta por cinco paquetes principales:

✦ **Capa de Presentación o Vista:**

La **Capa de Presentación o Vista** es la capa que interactúa con el usuario, es la que se encarga de mostrar y manejar cualquier evento del mismo, para posteriormente enviar la petición al Controlador Frontal y mostrar una respuesta en dependencia de su petición.

Views: El paquete contiene los archivos de apoyo necesarios para el diseño gráfico de la interfaz del sistema.

CSS: El paquete CSS contiene el conjunto de hojas de estilo de la aplicación.

Imágenes: El paquete contiene las imágenes del sistema con formato .jpg, .png o .gif)

JS: El paquete contiene los archivos de Javascript con extensión .js

✦ **Capa Controlador:**

Contiene las funcionalidades del negocio que dan respuesta a los requerimientos. La misma recibe peticiones de la capa de Presentación o Vista y procesa la lógica de negocio basada en las peticiones de los usuarios. Es la capa que media entre la Vista y el Modelo.

Controller: Este paquete contiene un conjunto de clases controladoras que encapsulan la lógica del negocio. Estas clases se encargan de mediar entre las peticiones del usuario y las clases modelo.

♦ **Capa Modelo:**

Contiene las clases persistentes, de las que cada una tiene correspondencia con una tabla de la base de datos. La capa Modelo es la contenedora de información, la cual es manejada por la capa Controladora para mostrarla en la capa de Presentación o Vista.

Model: El paquete contiene un conjunto de clases que controlan el tratamiento de los datos.

Libs: El paquete contiene las clases y librerías externas. Además, contiene el código común que el sistema utiliza.

♦ **Capa Application:**

Application: Este paquete contiene un conjunto de clases bases que permiten crear instancias de las mismas y utilizar sus funcionalidades.

♦ **Capa Insertion**

Contiene una clase que permite transformar un XML a partir de un formato definido en consultas SQL, y ejecutarlas en el gestor de base de datos PostgreSQL.

4.5 Diseño del Sistema.

El diseño de software se encuentra en el núcleo técnico de la ingeniería del software y se aplica independientemente del modelo de diseño de software que se utilice. Una vez que se analizan y especifican los requisitos del software, el diseño del software es la primera de las tres actividades técnicas; diseño, generación de código y prueba; que se requieren para construir y verificar el software. (Pressman, 2005)

4.5.1 Diagrama de Clases del Diseño.

En una aplicación Web, modelar las páginas, sus enlaces y todo el contenido dinámico tanto del lado del servidor, como del cliente, es muy importante, pues éstos son los

CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

artefactos que necesitan conocer los desarrolladores para implementar un producto con calidad.

Los diagramas de clases figuran las relaciones que poseen las clases que conservan la información manipulada por el sistema, todo el código que irán creando las páginas, así como el contenido dinámico de las misma, una vez que estén en el navegador del cliente.

Una vez precisada la base de la arquitectura, se muestra a continuación los Diagramas de Clases del Diseño de los Casos de Uso del Sistema.

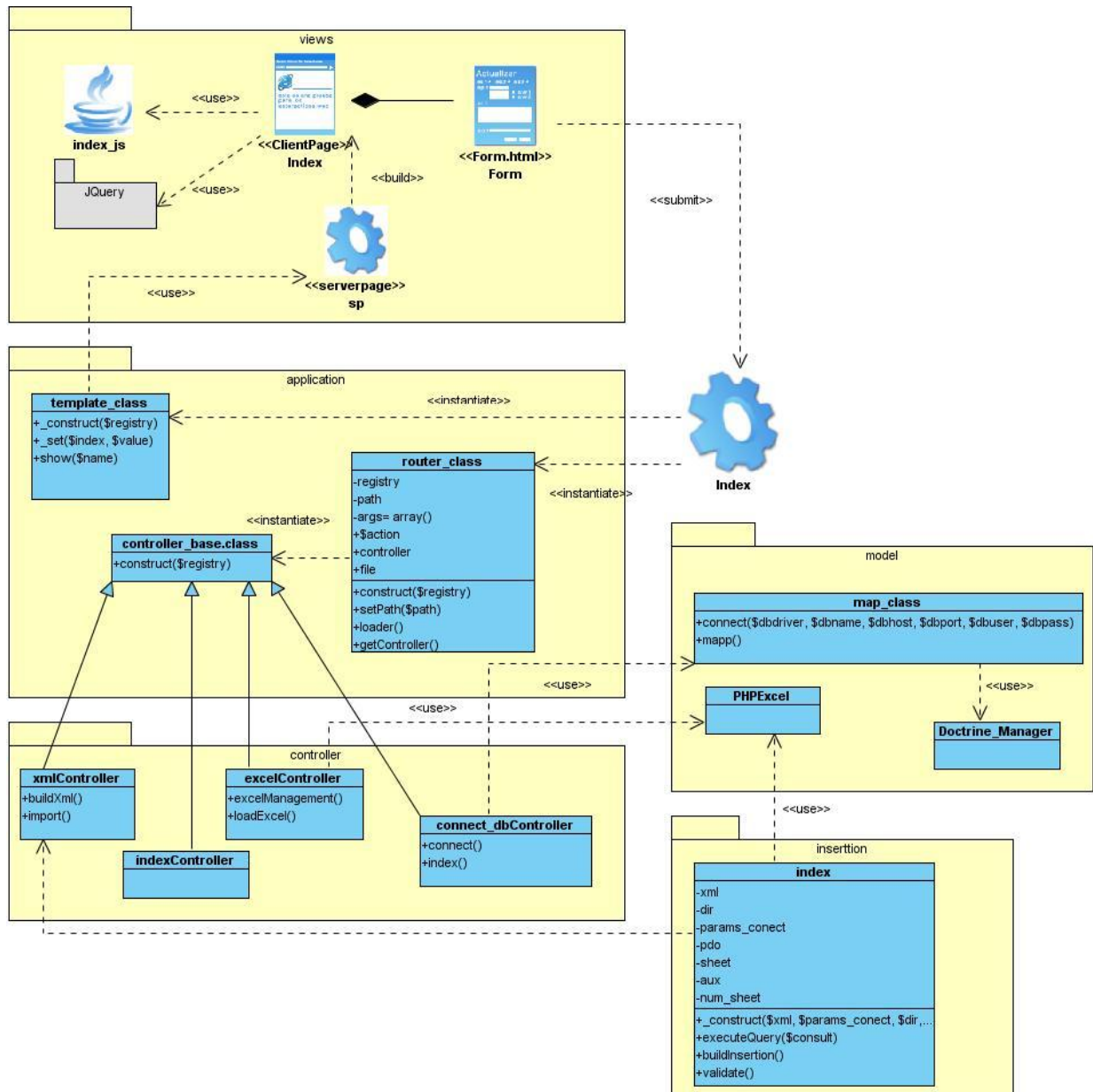


Fig. 5 Diagrama de Clases del Diseño Importar Datos.

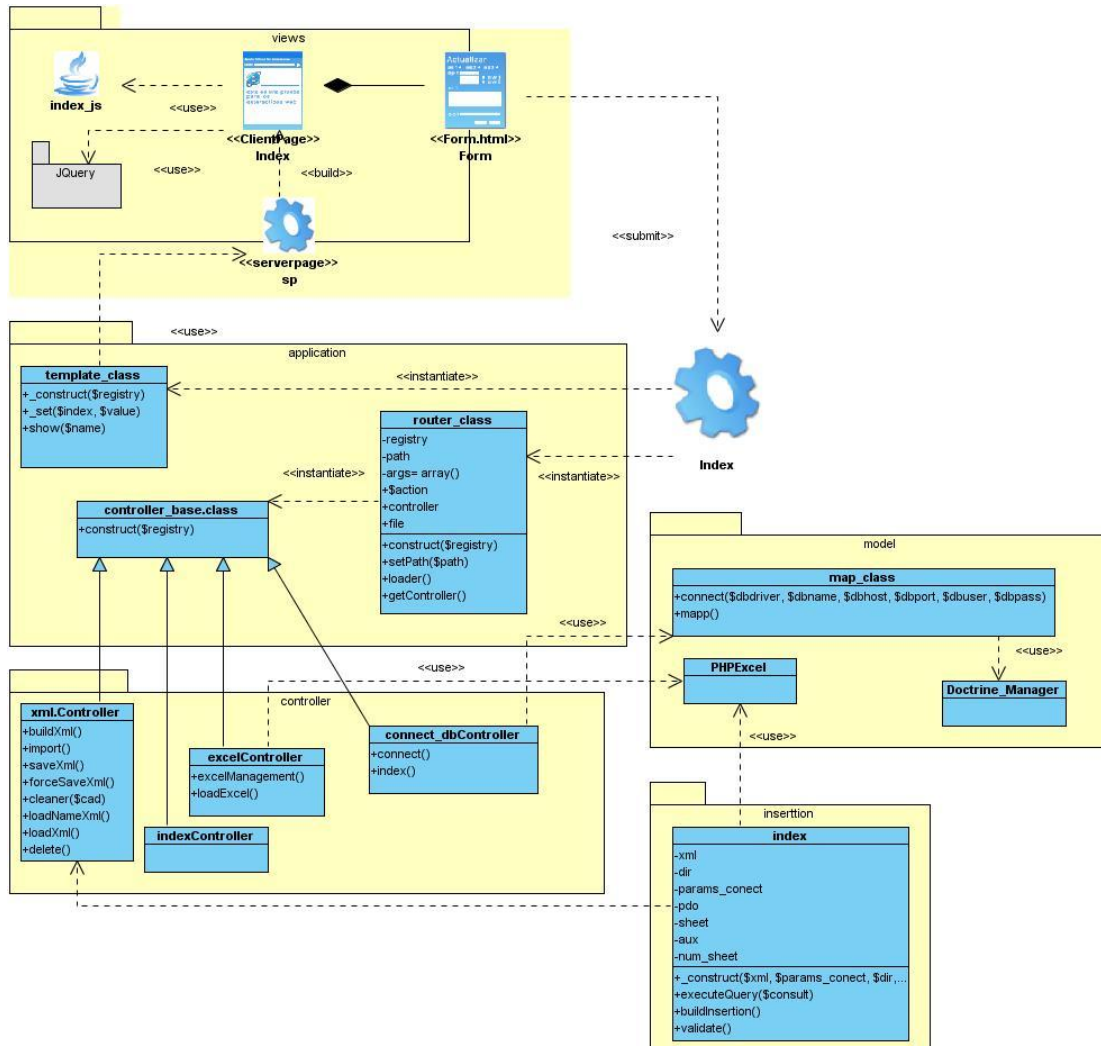


Fig. 6 Diagrama de Clases del Diseño Gestionar Plantilla Excel.

4.6 Modelo de Implementación.

La implementación empieza con el resultado del diseño y se implementa en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El modelo de implementación está compuesto por un conjunto de subsistemas y componentes que constituyen la composición física de la implementación del sistema.

El objetivo esencial del modelo de implementación, es representar cómo las clases del modelo de diseño se implementan en términos de componentes, tales como ejecutables, ficheros de código fuentes o tablas de una base de datos.

4.6.1 Diagrama de Componentes.

Los diagramas de componentes detallan los elementos físicos del sistema y sus relaciones. Los mismos son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Estos cubren la vista de implementación estática y se relacionan con los diagramas de clases ya que un componente suele tener una o más clases e interfaces.

El componente cuenta con tres subsistemas de implementación: Vistas, Modelo y Controladoras, que están estructurados según el patrón arquitectónico MVC. En el subsistema Vistas se encuentran encapsulados los componentes que admiten la interacción directa con el usuario del sistema. El subsistema Modelo, aglomera las clases que interactúan con la base de datos y velan por el cumplimiento de las reglas del negocio. El subsistema Controladoras contiene todas las clases que manipulan los eventos del usuario y realizan peticiones al modelo para mostrarlas en las vistas.

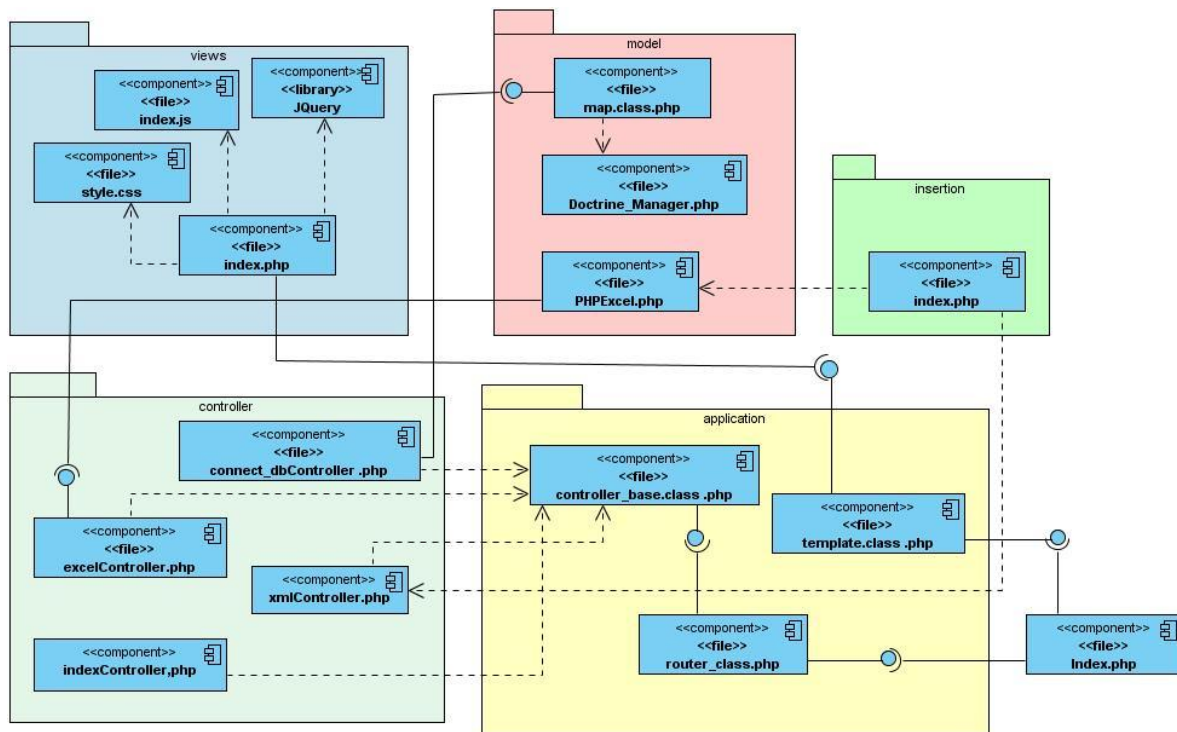


Fig. 7 Vista General del Diagrama de Componentes.

En el diagrama de componentes expuesto anteriormente, se muestra la relación entre los diferentes ficheros usados en la implementación del sistema.

4.7 Diagrama de Despliegue.

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. (SparxSystems, 2007)

La intención del modelo de despliegue no es describir la infraestructura, sino el camino en el cual los componentes específicos deben corresponder a una aplicación que despliega a través de él.

El diagrama de despliegue de la aplicación está representado por dos nodos. Uno de ellos es una PC Cliente, está a su vez se comunica con el servidor Web a través del protocolo HTTP, el cual, haciendo uso del protocolo ADO, establece comunicación con el servidor de Base de Datos en el cual corre la Base de Datos. En la siguiente figura se muestra el Diagrama de Despliegue.

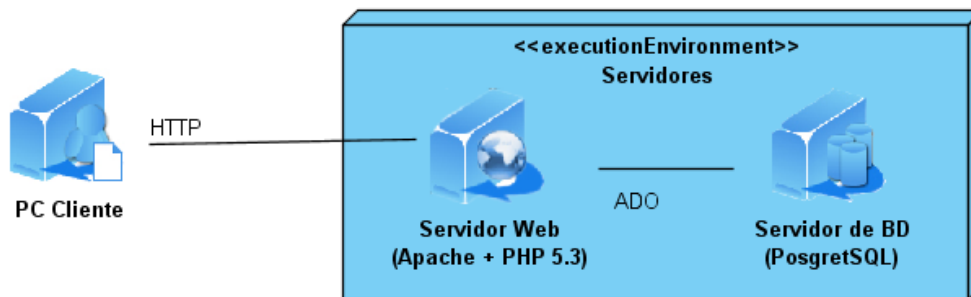


Fig. 8 Diagrama de Despliegue.

4.8 Validación de la solución a través del criterio de expertos.

La validación de todo software es necesaria ya que proporciona un alto grado de confianza y seguridad en el software y en los resultados que se obtienen al aplicarlo.

Un experto no es más que aquella persona que está capacitada grandemente en un campo, conjuntamente de tener los conocimientos esenciales y tener una amplia experiencia en el tema. Un experto es capaz de dar interpretaciones correctas en dicho campo, es capaz de valorar problemas y dar su recomendación al respecto.

Para la selección de los expertos se tuvo en cuenta varios aspectos, como son: experiencia profesional que poseen, su reconocido prestigio profesional, los años vinculados a la UCI, disposición a participar en la encuesta, para de este modo tener la posibilidad de obtener criterios de diversos especialistas y realizar un análisis integral

CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

del problema de estudio. El método de evaluación de expertos se emplea para comprobar la calidad y efectividad de los resultados de las investigaciones.

Elegir los expertos atendiendo a las particulares mencionadas anteriormente, ayuda a obtener resultados con calidad, junto a otras cualidades propias de éstos como pueden ser: la seriedad, la honestidad, la sinceridad, la responsabilidad y otras en este sentido, que hacen que las opiniones brindadas sean confiables y válidas para el objetivo propuesto.

Para realizar la validación se seleccionaron 6 expertos siguiendo los criterios expuestos inicialmente, quienes respondieron a la encuesta de autovaloración de los expertos, (Ver [Anexo 1.2](#)), la cual se aplico de forma anónima, con el objetivo que la opinión de un experto no repercuta en otro experto, de esta forma cada experto defenderá sus ideas.

Para la validación se tuvo en cuenta los siguientes parámetros que se especifican a continuación, dándole una puntuación del 1 al 5, donde 5 es la máxima evaluación. Los criterios establecidos son:

1. Calidad de la aplicación.
2. Novedad científica.
3. Aporte científico.
4. Facilidad de uso.
5. Diseño
6. Usabilidad
7. Adaptabilidad a diferentes entornos de producción de software.
8. Impacto en el área para la cual está destinada.

Expertos Criterios	1	2	3	4	5	6	Total x Indicadores	% que representa del total.
Calidad de la aplicación.	5	5	5	5	5	5	30	100%
Novedad científica.	4	5	5	5	4	5	28	93.3%
Aporte científico.	3	5	5	5	4	5	27	90%
Facilidades de uso.	4	4	5	5	5	3	26	86.6%
Diseño	5	5	5	5	5	4	29	96.6%
Usabilidad	5	4	4	5	5	4	27	90%

CAPÍTULO 4. CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.

Adaptabilidad a diferentes entornos de producción de software.	5	5	4	5	4	5	28	93.3%
Impacto en el área para la cual está destinada.	5	5	5	5	5	5	30	100%

Tabla 6 Puntuación de Expertos.

Conociendo los criterios que obtuvieron mayor puntuación luego de concluida la evaluación, se podrán predecir las fortalezas de la propuesta realizada. En la Fig.16 se representan los criterios evaluados así como el valor obtenido por cada uno de ellos expresado en porciento.

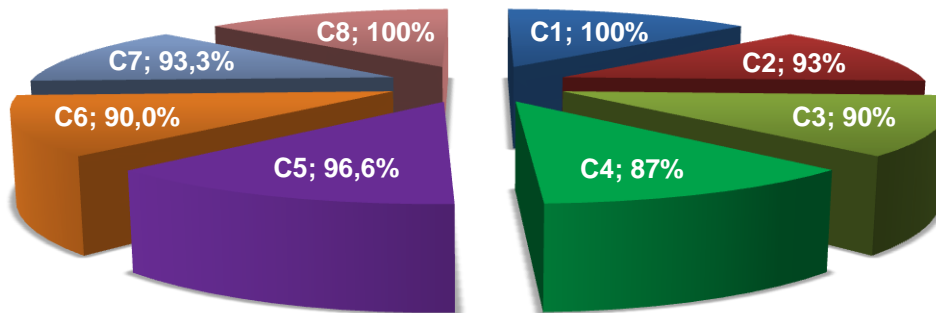


Fig. 9 Representación del valor de los criterios en porciento.

4.9 Opiniones de los expertos

El componente realizado posee gran importancia, el mismo permite importar documentos Excel hacia Bases de Datos en PostgreSQL, mejorando así la gestión de la información en el Laboratorio de Lodo y Cemento. La creación de este componente constituye un aporte fundamental en la informatización del proceso de consulta y revisión de la información contenida en los reportes recibidos en el CEINPET, al mismo tiempo disminuye en gran medida los errores que se puedan cometer cuando esta tarea se realiza de forma manual.

Luego de analizar las validaciones realizada por los expertos se puede decir que:

- ✦ El 100% de los expertos consideran que el componente realizado posee calidad desde su punto de vista, y que para el área que se desarrolla va a provocar un gran impacto, ya que hoy en día no se cuenta con un sistema que importe los datos que contienen los reportes recibidos, hacia un gestor de base de datos.
- ✦ El 66.6 % de los expertos considera que el componente implementado posee adaptabilidad en diferentes entornos de producción de software.
- ✦ El 83.3% de los expertos evaluó de 5 el diseño, y el resto propuso la calificación de 4.
- ✦ El 50% de los expertos evaluó de 5 la usabilidad del sistema, y el resto propuso la calificación de 4.
- ✦ El 66.6% de los expertos calificó de 5 el aporte científico de la investigación, el 16.6% otorgó la calificación de 4 y el resto emitió una evaluación de 3.

4.10 Conclusiones.

La ingeniería de software desempeña un papel fundamental a la hora de realizar un sistema. En este capítulo han quedado expuesto los principales artefactos de Diseño e Implementación, se realizó una descripción de la arquitectura utilizada y de los patrones aplicados en el diseño del componente. Del mismo modo, se definieron las clases del diseño y se representaron sus relaciones en el diagrama de clases del diseño.

Además se mostró el modelo de implementación y el modelo de despliegue que permite un mejor entendimiento de la distribución física y lógica del sistema. Finalmente, se realizó la validación del sistema a través del criterio de expertos para demostrar que se cumplieron con los objetivos de forma satisfactoria.

CONCLUSIONES GENERALES

Una vez finalizado el desarrollo del presente trabajo, se logró desarrollar un componente de software, con la finalidad de procesar documentos en formato Excel e importar la información contenida en los mismos hacia bases de datos en PostgreSQL.

Por otra parte, se arribaron a las siguientes conclusiones:

- ♦ Se capturaron los requisitos funcionales y no funcionales que debía tener el sistema. Esto conjuntamente con el modelo de sistema sentó las bases para desarrollar todo el trabajo de manera organizada y poder dar cumplimiento a las expectativas del cliente.
- ♦ Se obtuvo un componente web, desarrollado en entornos libres y que da respuesta a la necesidad que lo originó. Además, de ser un sistema capaz de informar al usuario los errores, inmediatamente que son producidos.
- ♦ Se realizó un diseño y una construcción del sistema teniendo en cuenta el patrón arquitectónico Modelo Vista Controlador, combinado con patrones de diseño.
- ♦ Con el desarrollo del sistema se logró una mayor reutilización del código y organización de las funcionalidades del sistema, garantizando un mayor tiempo de vida útil del sistema.

Por todo lo antes expuesto se concluye que el objetivo y las tareas propuestos para el presente trabajo de diploma, se han alcanzado satisfactoriamente. Además, se incluyen una serie de recomendaciones que pueden ser útiles en el desarrollo de nuevas versiones.

RECOMENDACIONES

Tras haber cumplido con el objetivo trazado en esta investigación se recomienda:

- ✦ Desarrollar una segunda versión para incorporar otros gestores de base de datos, debido a que el sistema solo soporta el Gestor de Bases de Datos PostgreSQL.
- ✦ Extender el sistema de manera que pueda ser utilizado no solo en el CEINPET, sino en cualquier empresa que requiera de este tipo de servicios.
- ✦ Seguir optimizando el sistema para mejorar el rendimiento de la aplicación en cuanto a la velocidad y los formatos de visualización de Microsoft Excel.
- ✦ Desarrollar futuras versiones, no solo mejorando las funcionalidades sino haciendo extensibles a aplicaciones de escritorio.

TRABAJOS CITADOS Y CONSULTADOS

Sqlmanager. [En línea] [Citado el: 10 de diciembre de 2009.] <http://www.sqlmanager.net/en/products/postgresql/dataimport>.

Visual-Paradigm. [En línea] [Citado el: 3 de Noviembre de 2009.] <http://www.visual-paradigm.com/product/vpuml/index.jsp>.

SUN "Java Beans". [En línea] [Citado el: 10 de Febrero de 2010.] <http://java.sun.com>.

Sitio oficial de Visual Paradigm. [En línea] [Citado el: 10 de diciembre de 2009.] <http://www.visual-paradigm.com/>.

jQuery. [En línea] [Citado el: 15 de Febrero de 2010.] <http://www.jquery.cl/>.

Chápela Martínez, Jairo. 2007. Introducción al entorno de desarrollo Eclipse. [En línea] 26 de Septiembre de 2007. [Citado el: 21 de Diciembre de 2009.] http://www.gris.det.uvigo.es/wiki/pub/Main/MiscResources/Manual_Eclipse.pdf.

XP Programación Extrema. *Club Developers*. [En línea] [Citado el: 12 de diciembre de 2009.] <http://www.clubdevelopers.com/index.php?p=38>.

Alexander, Christopher. 1979. *The Timeless Way of Building*. 1979. 0-19-502402-8.

Archer, Tom. 2001. *A fondo C#*. [ed.] Carmelo Sinchez Gonzitlez. ESPAÑA : s.n., 2001. Scott Wiltamuth, Miembro del equipo de diseño de C#, Microsoft Corporation. 84-481-3246-7.

Astigarraga, Eneko. *El Método Delphi*. [En línea] [Citado el: 24 de Mayo de 2010.] http://www.unalmed.edu.co/~poboyca/documentos/documentos1/documentos-Juan%20Diego/Plnaifi_Cuencas_Pregado/Sept_29/Metodo_delphi.pdf.

Balliau, Maarten. 2010. PHPEXcel Developer Documentation . *PHPEXcel*. [En línea] 11 de Enero de 2010. [Citado el: 2 de Abril de 2010.] <http://phpexcel.codeplex.com/documentation.1.7.2>.

Becerril C, Francisco. 1998. *Java a sus alcances*. Mexico : s.n., 1998. 970-10-1774-9.

Berzal Galiano, Fernando y Cortijo Bon, Francisco. 2006. *El lenguaje de programación C#*. s.l. : Alfaomega, 2006.

2008. Blog de Qualitrain. Tipos de Metodologías Ágiles. *Qualitrain*. [En línea] 2008. [Citado el: 12 de diciembre de 2009.] <http://www.qualitrain.com.mx/index.php/Procesos-de-software/Metodologias-agiles-de-desarrollo-segunda-parte.html>.

Boda, Pedro, Ninahuanca, Kattia y Martinez, Javier y Gonzales, Benjamín. 2009-11-15. *Zend Framework Manual en Español*. s.l. : Zend Technologies Inc., 2009-11-15. 471.

Casal Terreros, Julio. 2009. MSDN. *Desarrollo de Software basado en Componentes*. [En línea] 2009. [Citado el: 12 de diciembre de 2009.] <http://msdn.microsoft.com/es-es/library/bb972268.aspx>.

César F. Acebal, Juan M. Cueva Lovelle. 2002. *XP*. 2002. Depto. de Informática, Área de Lenguajes y Sistemas Informáticos, Universidad de Oviedo; Socios de ATI.

Ciberaula. Ciberaula Linux. *Una Introducción a APACHE*. [En línea] Ciberaula. [Citado el: 15 de Febrero de 2010.] http://linux.ciberaula.com/articulo/linux_apache_intro/.

—. **2006.** Patrones de Diseño en aplicaciones Web con Java J2EE. *Ciberaula-Java*. [En línea] 2006. [Citado el: 10 de Abril de 2010.] http://java.ciberaula.com/articulo/disenio_patrones_j2ee/.

Cla, Eric. 2009. Eclipse Plug-ins (Tapa Blanda). *LibreriaBabilonia*. [En línea] 01 de Octubre de 2009. [Citado el: 21 de Diciembre de 2009.] <http://www.libreriababilonia.com/978-84-415-2623-5/eclipse-plug-ins/resumen/#resumen>.

2009. CORBA. *CORBA ® Basics*. [En línea] 18 de Junio de 2009. [Citado el: 10 de Febrero de 2010.] <http://www.corba.org/>.

corba.org. 2009. WELCOME to the CORBA Directory. *CORBA ® Basics*. [En línea] 18 de Junio de 2009. [Citado el: 10 de Febrero de 2010.] <http://corba-directory.omg.org/>.

Cordero, Jorge Luis. *Metodologías Ágiles. Proceso Unificado Ágil (AUP)*. Bolivia : Universidad Unión Bolivariana.

Definicion De Lenguaje De Programacion. *Mitecnologico*. [En línea] [Citado el: 15 de Enero de 2010.] <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion>.

Elizondo, Perla Inés Velasco. Abril/2001. *Prueba de Componentes de Software Basadas en el modelo de JavaBeans*. s.l. : Departamento de Ingeniería y Tecnología unidad de estudios de posgrado.Universidad Autónoma de Tlaxcala, Abril/2001.

EMS-Database-Management-Solutions. *Sqlmanager*. [En línea] [Citado el: 10 de diciembre de 2009.] <http://www.sqlmanager.net/en/products/postgresql/dataimport>.

Entornos de Desarrollo Integrado. *Sitio Web de la E.U de Ingeniería Técnica Informática de Oviedo*. [En línea] [Citado el: 6 de Enero de 2010.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.

Fabien Potencier, François Zaninotto. *Symfony 1.2, la guía definitiva*. www.librosweb.es .

Ferguson, Jeff, y otros. 2003. *La biblia de C#*. Madrid : EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2003. 84-415-1484-4.

Flanagan, David y Hill, Mc Graw. 1999. *Java en pocas palabras*. [trad.] Roberto Becerril Flores. 2. Mexico : s.n., 1999. 970-10-2070-7.

free download manager. [En línea] [Citado el: 9 de diciembre de 2009.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.

free download manager. [En línea] [Citado el: 11 de diciembre de 2009.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28Iglesia_Anglicana%29_%5BMac_OS_X_cuenta_14717_p/.

Gallego Vázquez, José Antonio en: Guía Práctica para usuarios. 2003 . *Desarrollo Web con PHP y MySQL*. Madrid : Anaya, 2003 . 84-415-1525-5.

Gama, Erich, y otros. 1998. *Design Patterns CD.Elements of Reusable Object-Oriented Software*. 1998.

Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. *Aprende.com*. [En línea] [Citado el: 21 de Diciembre de 2009.] <http://eaprende.com/gestor-de-basededatos-mysql-postgresql-sqlite.html>.

Gilfillan, Ian. 2003. *La biblia MYSQL*. Madrid : Anaya Multimedia, 2003. 8441515581.

González, Antonio Adán Máximo. Diciembre / 2005. *Aplicación de la Tecnología de Componentes Java-Beans para realizar el control de un Robot Trepador*. s.l. : Escuela Técnica Superior de Ingeniería de Telecomunicación Universidad Politécnica de Cartagena, Diciembre / 2005.

Hernández Orallo, Enrique. *El Lenguaje Unificado de Modelado (UML)*.

i-Concepts. *i-Concepts. Preguntas Frecuentes*. [En línea] [Citado el: 2 de Febrero de 2010.] <http://www.intersap.net/faqs.php#10>.

2009. *Ingeniería del Software: Metodologías y Ciclos de Vida*. s.l. : Laboratorio Nacional de Calidad del Software de INTECO, 2009.

Introducción a Internet Information Server. *ADR formación.com*. [En línea] [Citado el: 20 de febrero de 2010.] <http://www.adrformacion.com/curso/intranet/leccion1/InternetInformationServer.htm>.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2004. *El Proceso Unificado de Desarrollo de Software*. s.l. : Félix Varela, 2004.

Jalón, Javier García de, y otros. *Aprenda C++ como si estuviera en primero*. s.l. : Escuela Superior de Ingenieros Industriales de San Sebastián.Universidad de Navarra.

java.sun.com. *SUN "Java Beans"*. [En línea] [Citado el: 10 de Febrero de 2010.] <http://java.sun.com>.

José H.Canós, Patricio Letelier y M.^a Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software. [En línea] [Citado el: 21 de Enero de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.

Joskowicz, José, Ing. iie. [En línea] [Citado el: 10 de diciembre de 2009.] Universidad de Vigo, España . <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>.

Larman, Craig. 1999. *UML y Patrones.Introducción al análisis y diseño orientado a objetos*. Mexico : s.n., 1999. 970-17-0267-1.

López, Angel. 1997. *Java la programación del futuro*. Argentina : s.n., 1997. 987-9131-38-X.

Martínez García, Alegna y Cedeño González, Johnny. 2008. *Sistema para la Evaluación y Control del trabajo con los Fluidos de perforación."MUDMAN"*. Ciudad Habana : UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS, 2008.

Masip, David. 2002. *DesarrolloWeb.com*. [En línea] 19 de Julio de 2002. [Citado el: 21 de Diciembre de 2009.] <http://www.desarrolloweb.com/articulos/840.php>.

Mastermagazine. [En línea] [Citado el: 10 de diciembre de 2009.] <http://www.mastermagazine.info/termino/4182.php>.

Mendoza Sánchez, María A. Metodologías de Desarrollo del Software. *Informatízate*. [En línea] [Citado el: 13 de diciembre de 2009.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

Mendoza, Luis E, Grimán, Anna y Pérez, María. Especialización de MSF para el Desarrollo basado en Componentes de un Sistema Colaborativo. [En línea] [Citado el: 15 de Diciembre de 2009.] http://www.lisi.usb.ve/publicaciones/07%20integracion%20de%20sistemas/integracion_23.pdf.

Microsoft. *Component Object Model Technologies*. [En línea] [Citado el: 10 de Febrero de 2010.] <http://www.microsoft.com/com>.

Microsoft. Component Object Model Technologies. *Microsoft*. [En línea] [Citado el: 10 de Febrero de 2010.] <http://www.microsoft.com/com>.

Mitecnologico. Definición de Lenguaje de programación. *Mitecnologico*. [En línea] [Citado el: 15 de Enero de 2010.] <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion>.

Mora, Sergio Luján. 2006. *C++ Paso a Paso*. Alicante : Publicaciones de la Universidad de Alicante, 2006. 84-7908-888-5.

Moreno Navarro, Juan José. *Introducción al Software basado en Componentes*. [En línea] [Citado el: 29 de noviembre de 2009.] Curso de Doctorado LSIS (junto con M. Collado).

http://74.125.93.132/search?q=cache:Yx38vFaS_sEJ:ls.fi.upm.es/~jjmoreno/sbc/sbc.ppt+Unidad+de+composici%C3%B3n+de+aplicaci.

MySQL-Hispano.org. Introducción a MySQL. *WebEstilo*. [En línea] [Citado el: 21 de Diciembre de 2009.] <http://www.webestilo.com/mysql/intro.phtml>.

NetBeans. Información de la versión de NetBeans IDE 6.8. *NetBeans*. [En línea] [Citado el: 12 de Enero de 2010.] http://netbeans.org/community/releases/68/index_es.html.

NetBeans Integration. *Visual-Paradigm*. [En línea] [Citado el: 10 de Enero de 2010.] <http://www.visual-paradigm.com/tutorial/netbeansintegrationtutorial.jsp>.

Nuevos Entornos y Plataformas para el Desarrollo de Software. *Wiccsi*. [En línea] [Citado el: 12 de Enero de 2010.] <http://www.wiccsi.com.ar/proyectos/14.1.pdf>.

P.Letelier. 2006. Introducción Proceso de Desarrollo de SW. *Portal Desarrollo de Software*. [En línea] 15 de Febrero de 2006. [Citado el: 21 de Enero de 2010.] <https://pid.dsic.upv.es/C1/Material/default.aspx>.

Paradigm, Visual. 2007. Why Visual Parading for UML. *Visual-Paradigm*. [En línea] 2007. [Citado el: 3 de Noviembre de 2009.] <http://www.visual-paradigm.com>.

Petra.euitio. Entornos de Desarrollo Integrado. *Sitio Web de la E.U de Ingeniería Técnica Informática de Oviedo*. [En línea] [Citado el: 6 de Enero de 2010.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.

2008. Petróleo y biotecnología: análisis del estado del arte y tendencias. *bvs*. [En línea] 26 de diciembre de 2008. [Citado el: 9 de diciembre de 2009.] http://bvs.sld.cu/revistas/aci/vol19_1_09/aci03109.htm.

Ponjuán Duarte, Gloria. 2003. *Gestión documental, de información y del conocimiento...puntos de contactosy diferencias*. La Habana: Ciencias de la Información, 2003. Vol. 34.

1996-2005. PostgreSQL 8.1.0 Documentation. The PostgreSQL Global Development Group. 1996-2005.

PostgreSQL-Global-Development-Group. PostgreSQL 8.1.20 Documentation. The PostgreSQL Global Development Group. [En línea] [Citado el: 21 de Diciembre de 2009.] <http://www.postgresql.org/docs/8.1/interactive/index.html>.

Presentación de ,etodología MSF (Microsoft Solution Framework). **S.A, Implementacion personalizada Gattaca.**

Pressman, Roger S. 2005. *Ingenieria del Software un enfoque practico*. La Habana : Felix Varela, 2005. págs. 159-187.

2006. QuimiNet.com. [En línea] 4 de abril de 2006. [Citado el: 12 de diciembre de 2009.] http://www.quiminet.com.mx/ar5/ar_advvcvdAAss-usos-y-aplicaciones-de-las-bentonitas.htm.

Rational Rose Enterprise. *IBM*. [En línea] [Citado el: 18 de Enero de 2010.] <http://www-142.ibm.com/software/products/es/es/enterprise/>.

Rosabal, Sheyla Bonell. 2008. Petróleo y biotecnología: análisis del estado del arte y tendencias. *bvs*. [En línea] 26 de diciembre de 2008. [Citado el: 9 de diciembre de 2009.] http://bvs.sld.cu/revistas/aci/vol19_1_09/aci03109.htm.

Ruiz González, Francisco y González Martín, Óscar. 1999/2000. *Arquitecturas de Sistemas de Bases de Datos*. 1999/2000. UNIVERSIDAD DE CASTILLA LA MANCHA, ESCUELA SUPERIOR DE INFORMÁTICA BASES DE DATOS.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2005. *El Lenguaje Unificado de Modelado.Manual de Referencia*. Madrid : Pearson Educación., 2005.

S.A, Implementación personalizada Gattaca. *Presentación de Metodología MSF(Microsoft Solution Framework)*. s.l. : GATTACA S.A.

Schmuller, Joseph. 2000. *Aprendiendo UML en 24 Horas*. s.l. : Bradley L. Jones, 2000. PEARSON EDUCACION.

SparxSystems. 2007. Diagrama de Despliegue UML 2.0. *SparxSystems*. [En línea] Sparx Systems Pty Ltd, 2007. [Citado el: 4 de Mayo de 2010.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

2009. Tecnología Educativa. *Desarrollo de un sistema evaluador*. [En línea] WordPress, 13 de Agosto de 2009. [Citado el: 10 de Febrero de 2010.] <http://www.tecnologiaeducativa.net/web2/2009/08/4-desarrollo-de-un-sistema-evaluador-basado-en-componentes/>.

Tricalcicos, Fosfatos. 2006. QuimiNet.com. *Usos y aplicaciones de las Bentonitas* . [En línea] 4 de abril de 2006. [Citado el: 12 de diciembre de 2009.] http://www.quiminet.com.mx/ar5/ar_advvcvdAAss-usos-y-aplicaciones-de-las-bentonitas.htm.

Valverde Rebaza, Carlos Jorge y Amaro Calderón, Sarah Dámaris. 2007. *Metodologías Ágiles*. Trujillo : s.n., 2007. Universidad Nacional de Trujillo.Facultad de Ciencias Físicas y Matemáticas.

Villagomez, Gerardo, Álvarez, Harold y Vivanco, Danny y Echeverría, Fabricio. 2009 . Implementación de la Migración de la Base de Datos del Sistema Canopus de Informix 9.4C a Oracle 10g. *Repositorio de la Escuela Superior Politécnica del Litoral*. [En línea] 20 de Febrero de 2009 . [Citado el: 21 de Enero de 2010.] <https://www.dspace.espol.edu.ec/bitstream/123456789/678/1/1191.pdf>.

Visual-Paradigm. NetBeans Integration. *Visual-Paradigm*. [En línea] [Citado el: 10 de Enero de 2010.] <http://www.visual-paradigm.com/tutorial/netbeansintegrationtutorial.jsp>.

WebCab Components. *About Component Based Development*. [En línea] [Citado el: 12 de diciembre de 2009.] <http://webcabcomponents.com/componentization.shtml>.

WebCab-Components. WebCab Components. *About Component Based Development*. [En línea] [Citado el: 12 de diciembre de 2009.] <http://webcabcomponents.com/componentization.shtml>.

Welling, Luke y Thomson, Laura. 2003. *Desarrollo Web con PHP y MySQL*. Madrid : Anaya, 2003.

Wiccsi. Nuevos Entornos y Plataformas para el Desarrollo de Software. *Wiccsi*. [En línea] [Citado el: 12 de Enero de 2010.] <http://www.wiccsi.com.ar/proyectos/14.1.pdf>.

WordPress. 2009. Tecnología Educativa. *Desarrollo de un sistema evaluador*. [En línea] WordPress, 13 de Agosto de 2009. [Citado el: 10 de Febrero de 2010.] <http://www.tecnologiaeducativa.net/web2/2009/08/4-desarrollo-de-un-sistema-evaluador-basado-en-componentes/>.