

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



Implementación de los Reportes del Módulo de Presentación del Proyecto “Convenio Integral de Cooperación Cuba-Venezuela” (CICCV).

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

Autor: Yojaimer Sánchez Villa.

Tutor: Ing. Dalgis Rogelio López Góngora.

Ciudad de la Habana, 2008-2009.

DECLARACION DE AUTORIA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ___ días del mes de _____ del año 2009.

Yojaimer Sánchez Villa

Dalgis R. López Góngora

Firma del autor

Firma del tutor

AGRADECIMIENTOS.

AGRADECIMIENTOS

A mis abuelos y padres por siempre apoyarme, aconsejarme y darme toda la confianza del mundo.

A toda mi familia por siempre confiar en mí.

A nuestra Revolución y en especial al Comandante en Jefe Fidel Castro Ruz por crear la Universidad de Ciencias Informáticas, por ser nuestro guía y enseñarnos a ser cada día mejores revolucionarios.

A la Universidad de Ciencias Informáticas por darme la oportunidad de formarme como Ingeniero.

A mi novia por quererme tanto y estar siempre pendiente de mí.

A los que han sido mis hermanos dentro y fuera de la universidad.

A todos aquellos profesores que supieron guiarme por el mejor camino.

A todos mis compañeros del 3503 y a los del antiguo grupo 3109.

A todas aquellas personas que de una forma u otra ayudaron a la realización de este trabajo de diploma.

DEDICATORIA.

DEDICATORIA

A mis abuelos Violeta y Medardo que han estado conmigo siempre en todo momento y se han esforzado tanto en criarme y enseñarme todo lo que se. Gracias por todo, los quiero mucho y éste trabajo se los dedico a ustedes.

A mi abuelo Ismael que siempre me apoyo y creyó en mí.

A mi abuela Aurora que siempre me quiso mucho, me dió su apoyo y muchos consejos, espero que donde te encuentres estés orgullosa de mí.

A mis padres por su confianza, amor y sabios consejos. Por prepararme para la vida y apoyarme siempre en mis decisiones.

A todas mis tías, tíos, primas, primos y mi hermana ya que siempre me brindaron su apoyo.

A mi tío Hermes que siempre me brindó su ayuda, su confianza y en éste momento estoy logrando lo que él siempre espero de mí.

A mi novia que me quiere muchísimo y a mis suegros que son mis segundos padres.

A todas mis amigas y amigos tanto dentro como fuera de la universidad que si los menciono a todos nunca terminaría. Gracias por hacerme pasar momentos inolvidables y estar siempre a mi lado en los buenos y malos momentos.

Gracias a todos ustedes hoy les puedo dedicar éste trabajo.

RESUMEN.

RESUMEN

La República de Cuba y la República Bolivariana de Venezuela, en aras de intensificar sus relaciones diplomáticas en los marcos del ALBA, mantienen en conjunto un acuerdo de colaboración, los cuales comprenden la concepción y ejecución de varios proyectos.

El control de los proyectos se efectúa de forma semiautomática, mediante reuniones de coordinación, vía telefónica y por correo electrónico, para darle soporte a la gestión de estos, se crea el proyecto “Convenio Integral de Cooperación Cuba-Venezuela” (CICCV), el cual consta de 7 módulos entre los cuales se encuentra el módulo Presentación.

En el módulo de Presentación es donde se elaboran los proyectos, se trazan las metas, los objetivos y se destinan los recursos que se invertirán para la ejecución del proyecto. Éste módulo consta de un conjunto de reportes necesarios para llevar a cabo un mayor nivel de gestión, control y seguimiento de los proyectos que se ejecutan.

En el presente trabajo se realiza la implementación de los reportes del Módulo de Presentación con el objetivo de obtener una buena implementación del mismo y con la calidad que éste requiere. Por lo que se realizó un estudio del arte de las metodologías de desarrollo, el entorno de desarrollo y las métricas para la implementación. Se utilizaron estándares de codificación y se generaron los artefactos correspondientes a la implementación, como son: el diagrama de despliegue y el diagrama de componente. Finalmente se validan los artefactos obtenidos para que la implementación termine con la calidad que se requiere.

INDICE DE FIGURAS.

INDICE DE FIGURAS

Figura 1 Fases de la metodología XP.....	7
Figura 2 Fases e iteraciones que propone RUP.	9
Figura 3 Transcurso del flujo de implementación según RUP.....	12
Figura 4 Diagrama de clase CU InversionHistorica.	46
Figura 5 Diagrama de clase CU InversionSectores.....	47
Figura 6 Cuadro de diálogo de crear nueva plantilla de informe.	48
Figura 7 Bandas del informe.....	49
Figura 8 Crear parámetros en Ireport.	51
Figura 9 Crear campos en Ireport.	52
Figura 10 Crear variables en Ireport.	53
Figura 11 Plantilla del CU FichaProyecto.....	55
Figura 12 Plantilla del CU InversionEntesMinisterios.	56
Figura 13 Plantilla del CU InversionHistorica.....	57
Figura 14 Diagrama de despliegue.....	58
Figura 15 Diagrama de componentes.	60
Figura 16 Estructura del aplicación.....	62
Figura 17 Estructura de la carpeta de las clases controladoras.	63
Figura 18 Estructura de la carpeta entidades.....	64
Figura 19 Estructura del paquete WebContent.....	65
Figura 20 Estructura de la carpeta reportes y dentro de estas la de conf y presentacion.	66
Figura 21 Estructura de la carpeta presentacion dentro de la carpeta conf.....	67
Figura 22 Diagrama de clase del CU FichaProyecto.....	69
Figura 23 Implementación de la clase controladora del CU InversionHistorica.....	70
Figura 24 Implementación de la clase entidad del CU InversionHistorica.....	71
Figura 25 Configuración del xml presentacion-applicationContext-reporte.....	73
Figura 26 Configuración del xml web.....	74
Figura 27 Vista del filtro de búsqueda del reporte de proyectos.	78
Figura 28 Vista del filtro de búsqueda del reporte fichas de proyectos.	79
Figura 29 Errores al escribir el mensaje “solo puede ingresar datos numéricos “90	90
Figura 30 No existe concordancia entre la fecha de inicio y fin.	90
Figura 31 Los datos: tiempo, cantidad, precio permite signo al inicio del número.....	91
Figura 32 Acta de Aceptación del Módulo de Presentación.	100
Figura 33 Acta de finiquito del sistema CCV.....	102
Figura 34 Implementación de la clase InversionHistoricaController.....	105

INDICE.

INDICE

AGRADECIMIENTOS.....	II
DEDICATORIA	III
RESUMEN.....	IV
INDICE DE FIGURAS.....	V
INDICE	VI
INTRODUCCION.....	1
CAPÍTULO 1: FUNDAMENTACION TEORICA.....	4
1.1- INTRODUCCION.	4
1.2- SISTEMAS DE REPORTES EN EL MUNDO.	4
- Obtención de reportes en Cuba.....	4
1.3- METODOLOGÍAS DE DESARROLLO DE SOFTWARE.	5
1.4- ENTORNO DE DESARROLLO.....	12
1.4.1- PLATAFORMA DE DESARROLLO.	12
1.4.2- SISTEMAS GESTORES DE BASE DE DATOS (SGDB).	19
1.4.3- LENGUAJES DE PROGRAMACION.	24
1.4.4- HERRAMIENTAS A UTILIZAR.	27
1.4.5- ENTORNO DE DESARROLLO INTEGRADO (IDE).....	33
1.4.6- SERVIDOR DE APLICACIONES.....	35
1.4.7- FRAMEWORKS DE DESARROLLO.....	37
1.5- METRICAS PARA LA IMPLEMENTACION.....	42
1.6- CONCLUSIONES.....	43
CAPITULO 2: PROPUESTA DE SOLUCION AL PROBLEMA.....	44
2.1- INTRODUCCION.	44
2.2- ELEMENTOS DE DISEÑO UTILIZADOS EN LA IMPLEMENTACION.....	44
2.3- OBTENCION DE LAS PLANTILLAS PARA GENERAR LOS REPORTES.....	47
Creación de las plantillas	48

INDICE.

Plantillas de los reportes del módulo de Presentación.	54
2.4- FLUJO DE TRABAJO DE IMPLEMENTACION.	58
Diagrama de despliegue.	58
Diagrama de componente.	59
2.5- ESTANDARES DE CODIFICACION.	60
2.6- ESTRUCTURACION DE LOS PAQUETES PARA LA IMPLEMENTACION.	62
2.7- IMPLEMENTACION Y CONFIGURACION.	68
Implementación de las clases controladoras y las clases entidades.	68
2.8 - CONCLUSIONES.	75
CAPITULO 3: VALIDACION DE LA SOLUCION PROPUESTA.	76
3.1- INTRODUCCION.	76
3.2- PRUEBAS DE CAJA NEGRA.	76
-Validaciones de los campos de entrada.	77
-Diseño de casos de prueba.	80
-No Conformidades Detectadas.	88
3.4- CONCLUSIONES.	92
CONCLUSIONES GENERALES.	93
RECOMENDACIONES.	95
BIBLIOGRAFIA.	96
GLOSARIO DE TERMINOS.	99
ANEXOS.	100

INTRODUCCION.

INTRODUCCION

El proceso de desarrollo de software desde sus inicios surgió con la finalidad de crear productos para agilizar el trabajo y simplificar el esfuerzo de las personas. Con el perfeccionamiento en las tecnologías de la informática y las comunicaciones (TIC), el desarrollo de software ha tomado gran importancia en la sociedad, cada día se hace más necesario el desarrollo de sistemas automatizados más potentes, pensando en las expectativas de los clientes al menor costo posible.

La Facultad 3 de la Universidad de las Ciencias Informáticas (UCI), cuenta con un grupo de proyectos productivos que aportan al país una considerable suma de dinero desde hace varios años con la exportación de software principalmente hacia la República Bolivariana de Venezuela, la facultad se ha ido ganando el prestigio paulatinamente con los productos que en ella se desarrollan. Por éste prestigio ganado en la Facultad 3 se le asigna la responsabilidad de automatizar un sistema para la gestión de la información de los proyectos entre ambos países.

Cuando empieza el proceso revolucionario en la República Bolivariana de Venezuela el presidente Hugo Chávez realiza una visita oficial a Cuba para estrechar las relaciones entre ambos países y firmar una serie de convenios para el desarrollo de ambos países en varias ramas de la economía. A partir de este momento se consolidaron las relaciones y aumenta el número de proyectos a ejecutar, éste aumento vienen dado a la creación de la Alternativa Bolivariana para las Américas (ALBA).

En el año 2008 se decide crear un proyecto para automatizar la información existente de los proyectos que se están realizando y los que fueron realizados, a éste proyecto se le dio el nombre de "Convenio Integral de Cooperación Cuba – Venezuela" (CICCV).

Para el control de la información existente de los proyectos que han sido firmados y los que no se han firmados existen varias deficiencias, una de las que tiene un gran peso es que la información generada es controlada en formato duro y existe un uso excesivo de papales.

INTRODUCCION.

Cada día aumenta el número de estos proyectos y llevar a cabo ésta tarea teniendo la información en formato duro es muy difícil llevar un control absoluto de los datos. Es muy engorroso realizar una revisión completa de la información y se corre el riesgo de que estos documentos puedan duplicarse, perderse o deteriorarse lo que sería una pérdida irreversible.

Otra deficiencia que existe actualmente es el tiempo que demora obtener una información específica de varios proyectos, es muy difícil realizar un reporte que contenga un gran conjunto de datos pues para lograr esto habría que realizar un arduo trabajo y se debe tener en cuenta todos los documentos involucrados en la actividad. El tiempo es un factor fundamental por el cual se decidió realizar el proyecto CICCIV pues una vez creados los proyectos tienen que ser revisados y aceptados por los ministros de ambos países. Al ser aceptados estos tienen que ser aprobados y firmados por los presidentes de los dos países, los cuales se reúnen todos los años para realizar éste proceso, el cual es llamado Mixta. La duración de una Mixta era de 4 a 7 días debido al tiempo necesitado para obtener informaciones específicas de los proyectos.

Por eso se le dió la tarea al proyecto CICCIV de automatizar todo éste proceso, el cual se venía realizando de forma manual y se le encomienda la tarea de automatizar los reportes para facilitar la evaluación y la selección de los proyectos a ejecutar, no exista pérdida de información y se pueda realizar una revisión completa de los proyectos en un breve espacio de tiempo.

Problema Científico a resolver

¿Cómo obtener los reportes para la informatización del Módulo de Presentación del Proyecto “Convenio Integral de Cooperación Cuba –Venezuela” que faciliten la evaluación y selección de los proyectos a ejecutar?

Objeto de estudio

El proceso de desarrollo de software.

Campo de acción

Implementación de los reportes del Módulo de Presentación del Proyecto “Convenio Integral de Cooperación Cuba-Venezuela”.

INTRODUCCION.

Objetivo general

Desarrollar el sistema de reportes del Módulo de Presentación del proyecto de informatización del Convenio Integral de Cooperación Cuba-Venezuela que faciliten la evaluación y selección de los proyectos a ejecutar.

Objetivos específicos:

- ❖ Realizar un estado del arte de las herramientas de diseño de reportes, de los sistemas de gestión, de los métodos para desarrollar reportes e integrarlos con el lenguaje de programación Java y un estudio del IDE de desarrollo Eclipse.
- ❖ Analizar los frameworks Hibernate y Spring para la obtención de los datos para generar los reportes.
- ❖ Diseñar las plantillas de los reportes.
- ❖ Implementar las clases necesarias para desarrollar los reportes en el Módulo de Presentación.

Posibles resultados:

- ❖ Obtener las plantillas de los reportes.
- ❖ Obtener las clases controladoras de los reportes.
- ❖ Obtener las clases entidades para los reportes.

Hipótesis:

Si se desarrollan adecuadamente los reportes para el Módulo de Presentación del proyecto “Convenio Integral de Cooperación Cuba – Venezuela” se facilitará una buena evaluación y selección de los proyectos a ejecutar.

CAPÍTULO 1: FUNDAMENTACION TEORICA

1.1- INTRODUCCION.

En este capítulo se realizará un estudio del arte donde se abordará acerca de las tendencias y tecnologías actuales de la información. Se justificarán las herramientas, tecnologías y metodología, así como los frameworks, el lenguaje, la plataforma y el servidor de aplicación que se utilizará en el desarrollo de éste trabajo. También se hará referencia al estado de la obtención de reportes en Cuba y se analizarán los sistemas de reportes en el mundo.

En éste trabajo se realizará un estudio acerca de las diferentes herramientas para la generación de reportes, así como también se abordará sobre cómo se deben realizar los sistemas de reportes para el desarrollo de los productos de software.

1.2- SISTEMAS DE REPORTES EN EL MUNDO.

Actualmente en el mundo existen numerosas herramientas para la generación de reportes, mientras éste campo se desarrolla surgen nuevas herramientas las cuales facilitan más el trabajo a los desarrolladores. Uno de los mayores problemas que existen para la generación de los reportes es el de obtener reportes dinámicos, ya que la mayoría de las herramientas existente hoy en día están hechas para realizar un determinado reporte. Éstas herramientas son elaboradas con el objetivo de poder realizar un reporte determinado, si el usuario desea obtener más datos en el mismo reporte es necesario modificar el reporte que se realizó.

- Obtención de reportes en Cuba.

En Cuba se ha ido desarrollando la industria del software, un ejemplo de este desarrollo lo constituye la creación de la Universidad de las Ciencias Informáticas, la cual ha sido la principal impulsora y desarrolladora de la automatización de todas las empresas del país. En cada uno de los procesos que han sido automatizados en Cuba actualmente se observa la obtención de reportes. Lo que brinda una consolidación de la información y apoyan a la hora de tomar de decisiones.

1.3- METODOLOGÍAS DE DESARROLLO DE SOFTWARE.

Una metodología es un conjunto de componentes que especifican:

- ❖ Cómo se debe dividir un proyecto en etapas.
- ❖ Qué tareas se llevan a cabo en cada etapa.
- ❖ Qué salidas se producen y cuándo se deben producir.
- ❖ Qué restricciones se aplican.
- ❖ Qué herramientas se van a utilizar.
- ❖ Cómo se gestiona y controla un proyecto.

Las principales ventajas facilitadas por una metodología para el desarrollo de software son: (Mendoza Sánchez, 2004)

- ❖ Mejora de los procesos de desarrollo: esto trae consigo que las personas del proyecto trabajen bajo un marco común, uso de herramientas de ingeniería software y recopilación de mejores prácticas para proyectos futuros.
- ❖ Mejora de los productos: asegura que los productos cumplen con los objetivos de calidad propuestos, detección temprana de errores y se garantiza la trazabilidad de los productos a lo largo del desarrollo.
- ❖ Mejora de las relaciones con el cliente: El cliente percibe el orden en nuestros procesos, le facilita al cliente el seguimiento de la evolución del proyecto y se establecen mecanismos para asegurar que los productos desarrollados cumplen con las expectativas del cliente.

Para desarrollar un software se tiene que tomar como guía una metodología, la cual es la encargada de elaborar un plano y guiar al equipo de desarrollo. Para guiar el desarrollo del software en la actualidad existen varias metodologías como: Extreme Programming (XP), Rational Unified Process (RUP), SCRUM, Feature - Driven Development (FDD) y Adaptive Software Development (ASD). Entre las mencionadas anteriormente se va a argumentar algunas de las más usadas en estos momentos. (Canós, 2007)

-Extreme Programming (XP).

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

Ésta metodología fue desarrollada por **Kent Beck**, el cual plantea lo siguiente:

«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.»

XP surge como respuesta y posible solución a los problemas derivados del cambio en los requerimientos, se plantea como una metodología a emplear en proyectos de riesgo y aumenta la productividad. Tiene como objetivo: la satisfacción del cliente. Ésta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. (Calero Solís, 2003) (González Barbone, 2007)

Otro objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software. XP define cuatro variables para proyectos de software: coste, tiempo, calidad y ámbito.

Esta metodología está compuesta por varias fases:

Fases de la Metodología XP

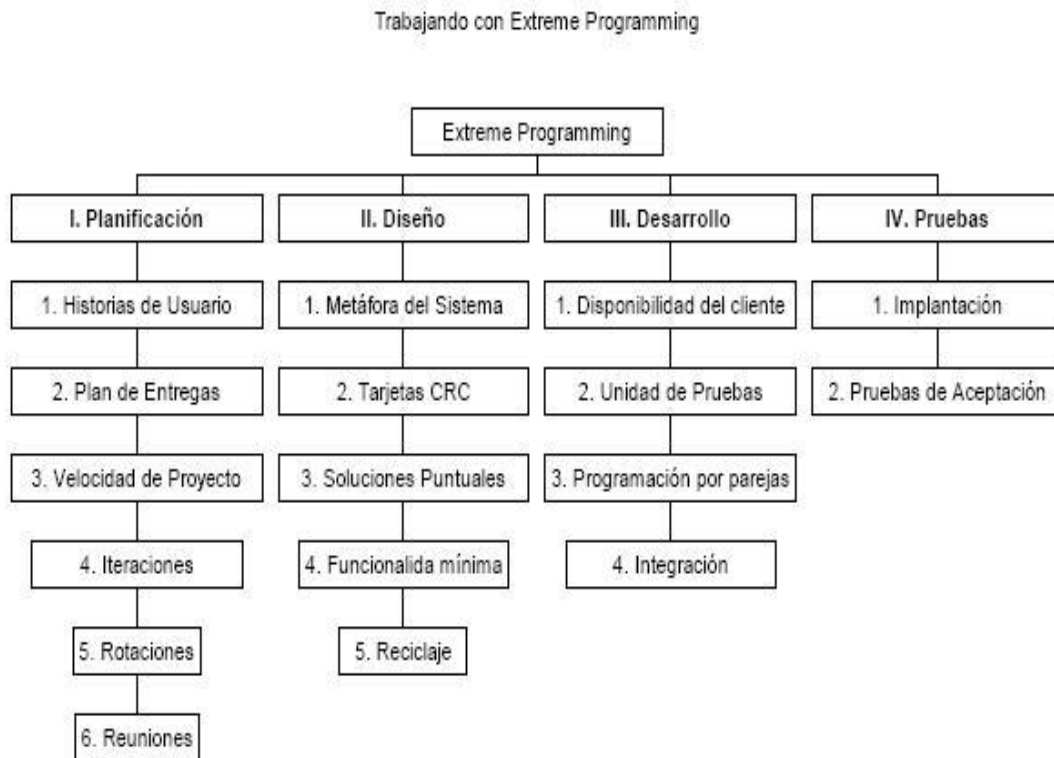


Figura 1 Fases de la metodología XP.

Acerca de esta metodología existen varias críticas como las que se pueden observar a continuación:

- ❖ Se critica el mito de las 40 horas semanales, y que es un lujo para las exigencias del mercado.
- ❖ Se dicen que solo puede funcionar con programadores muy buenos, como Kent Beck, que son capaces de hacer un buen diseño, sencillo y fácilmente extensible.
- ❖ XP es más una filosofía de trabajo que una metodología. Por otro lado ninguna de las prácticas defendidas por XP son invención de este método, XP lo que hace es ponerlas todos juntas.

- ❖ XP está diseñado para grupos de pequeños programadores más de 10 ya sería muy complicado, y para que estén en el mismo centro de trabajo.

Las metodologías tradicionales imponen un proceso disciplinado para tratar de hacer el trabajo predecible, eficiente y planificado. Estos métodos están orientados a documentos y se vuelven demasiado Burocráticos e ineficaces. XP es más liviana y ágil y están orientadas más a las personas que a los procesos. (Fernández Escribano, 2002)

-Rational Unified Process (RUP).

Es una metodología orientada a procesos, define un ciclo de vida iterativo incremental priorizando el uso de lenguajes de modelado, casos de uso y centrado en la arquitectura. Es un proceso de Ingeniería del Software que proporciona una visión disciplinada para la asignación de tareas y responsabilidades en las organizaciones de desarrollo de software. (Jacobson, et al., 2000)

El modelo RUP está compuesto por:

- ❖ Roles.
- ❖ Actividades.
- ❖ Artefactos.
- ❖ Disciplina.
- ❖ Flujos de trabajo.

Cuatro son las fases que dividen el ciclo de vida de un proyecto RUP, las cuales concluyen con un hito bien definido donde deben tomarse ciertas decisiones:

1.- Inicio: Es la fase de la idea, de la visión inicial de producto, su alcance. El esbozo de una arquitectura posible y las primeras estimaciones. Concluye con el “hito de objetivo”.

2.- Elaboración: Comprende la planificación de las actividades y del equipo necesario. La especificación de las necesidades y el diseño de la arquitectura. Termina con el “hito de Arquitectura”.

3.- Construcción: Desarrollo del producto hasta que se encuentra disponible para su entrega a los usuarios. Termina con el “hito del inicio de la capacidad operativa”.

4.- Transición: Traspaso del producto a los usuarios. Incluye: manufactura, envío, formación, asistencia y el mantenimiento hasta lograr la satisfacción de los usuarios.

Las iteraciones que propone RUP para el desarrollo de software son.

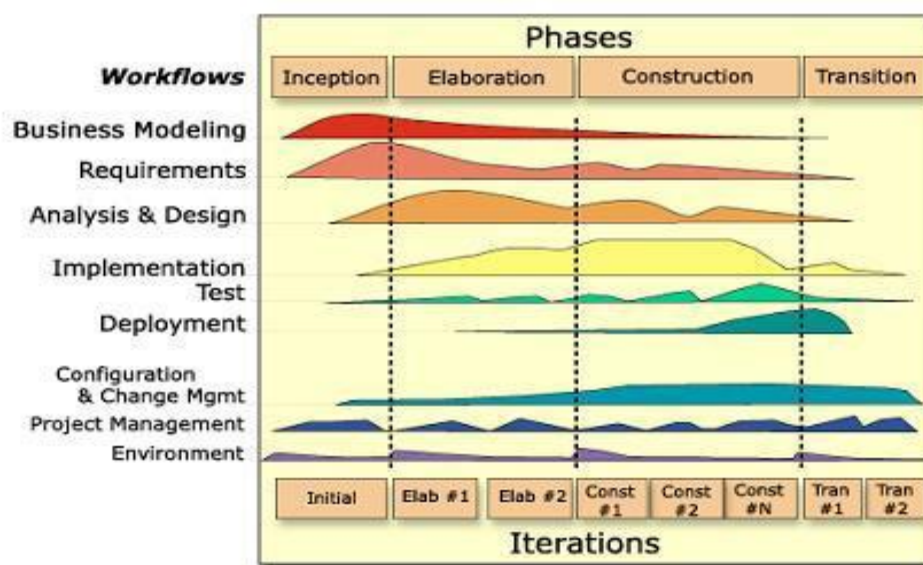


Figura 2 Fases e iteraciones que propone RUP.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. (Jacobson, et al., 2000)

RUP se propone varias metas como:

- ❖ Reducir en gran medida el riesgo que representa la construcción de sistemas complejos, porque evoluciona de forma incremental partiendo de sistemas más pequeños en los que ya se tiene confianza.

FUNDAMENTACION TEORICA.

- ❖ Proveer un camino metódico, sistemático para desarrollar, diseñar y validar una arquitectura.
- ❖ Proveer un enfoque disciplinado para asignar tareas y responsabilidades dentro del desarrollo del sistema.
- ❖ Asegurar la producción de un software de alta calidad que reúna las necesidades de los usuarios finales dentro de un plan y un presupuesto predecible.

Por las características y ventajas que tiene esta metodología se decidió utilizarla para el desarrollo del software, es utilizada como una política en la universidad pues la mayoría de los proyectos realizados han sido guiados por esta metodología, también hay que resaltar que se ha ganado un gran prestigio dentro de las grandes industrias de software de todo el mundo, además es una metodología que se basa mucho en la calidad final del producto, que es el principal objetivo de todo software y es lo que realmente el usuario desea.

-Flujo de implementación.

La fase de implementación empieza con el resultado del diseño. La mayor parte de la arquitectura del sistema es capturada durante el diseño. El propósito de la implementación no es más que desarrollar la arquitectura y el sistema como un todo.

Los propósitos de la implementación son:

- ❖ Planificar las integraciones del sistema necesarias en cada iteración.
- ❖ Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- ❖ Implementar las clases y subsistemas encontrados durante el diseño.
- ❖ Probar los componentes individualmente y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones del sistema.

La implementación es el centro durante las iteraciones de construcción, aunque también se lleva a cabo trabajo de implementación durante la fase de elaboración, para crear la línea base ejecutable de la arquitectura, y durante la fase de transición,

para tratar defectos y errores tardíos como los encontrados con distribuciones beta del sistema. (Jacobson, et al., 2000)

En el modelo de implementación se denota la implementación actual del sistema en términos de componentes y subsistemas de implementación, es natural mantener el modelo de implementación a lo largo de todo el ciclo de vida del software.

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. Algunos estereotipos estándar de componentes son: ejecutables, librerías, tablas, documentos, entre otras.

En éste flujo de trabajo se encuentran varios artefactos como son:

1. Modelos de implementación.
2. Componentes.
3. Subsistemas de implementación.
4. Interfaz
5. Descripción de arquitectura.
6. Plan de integración de construcciones.

Los **diagramas de despliegue** y **componentes** conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación. Éste flujo está fuertemente determinado por el lenguaje de programación.

Éste es un flujo de trabajo donde el mayor peso lo tiene en la fase de construcción, pues es en éste flujo es donde se realiza la implementación de componentes y subsistemas, para que una vez realizada ésta hacer la integración, realizarles las pruebas necesarias y finalmente obtener el producto final con la calidad que se requiere.

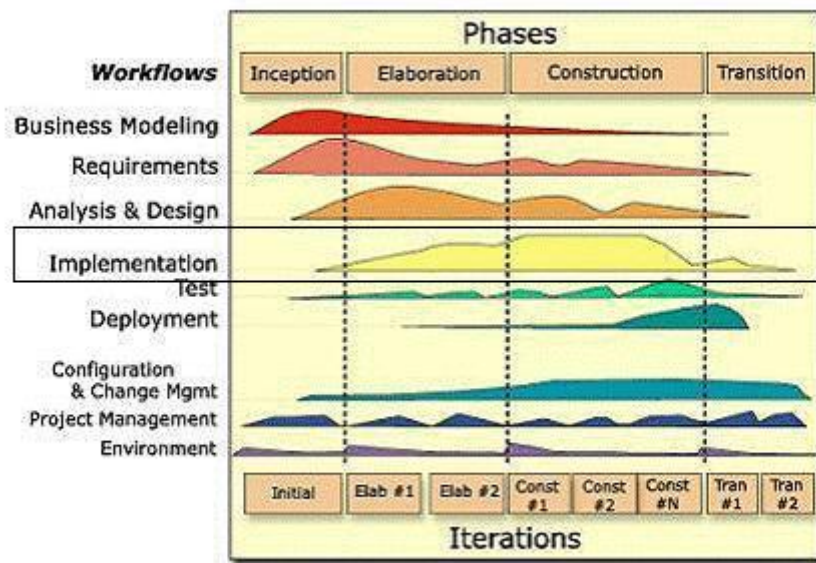


Figura 3 Transcurso del flujo de implementación según RUP.

1.4- ENTORNO DE DESARROLLO.

El entorno de desarrollo viene dado por la selección de la plataforma en la cual se va a desarrollar una aplicación o producto de software. En la actualidad existen numerosas plataformas para desarrollar, algunas de ellas privatizadas y otras no. Una de las plataformas que más auge ha tomado en los últimos tiempo es la Java Enterprise Edition 2 (J2EE).

1.4.1- PLATAFORMA DE DESARROLLO.

Una plataforma de desarrollo empresarial ha de ofrecer una serie de servicios a los arquitectos y desarrolladores encaminados a facilitar el desarrollo de aplicaciones empresariales, al tiempo que ofrece la mayor cantidad posible de funcionalidades a los usuarios.

Normalmente una plataforma de desarrollo empresarial tiene los siguientes requisitos:

-Escalabilidad: ofrece una buena escalabilidad tanto horizontal como vertical de modo que si aumenta la carga del sistema podamos añadir servidores o ampliar los existentes sin que sea necesario realizar modificaciones.

FUNDAMENTACION TEORICA.

-Mantenibilidad: permite añadir modificar los componentes existentes sin que se modifique el comportamiento del sistema.

-Disponibilidad: debe tener soporte de arquitecturas tolerantes a fallos y sistemas de redundancia que asegure que el sistema estará siempre disponible.

-Extensibilidad: debe brindar la posibilidad de añadir nuevos componentes y capacidades al sistema sin que se vean afectados el resto de componentes.

-Manejabilidad: deben ser fácilmente manejables y configurables.

-Seguridad: debe tener buenos sistemas de seguridad tanto a nivel de autenticación, como de autorización y transporte.

-Rendimiento: debe ofrecer automáticamente balanceo de carga y mecanismos que permitan aumentar el rendimiento de manera transparente al usuario.

La importancia de una plataforma empresarial es que todos estos componentes se nos ofrecen de manera automática de modo que los desarrolladores son mucho más productivos. Muchos arquitectos e ingenieros están comenzando a ver un nuevo requisito indispensable para cualquier plataforma de desarrollo empresarial: **La disponibilidad de soluciones libres.**

El que una plataforma disponga de soluciones libres aporta unos beneficios extra muy importantes a los arquitectos ya que obtienen un gran abanico de soluciones de bajo coste que pueden evaluar, y que además suelen tener detrás una amplia comunidad de desarrolladores para poder resolver los problemas que vayan apareciendo. En la actualidad las plataformas más importantes son aquellas que disponen del apoyo de gran cantidad de empresas, entidades o asociaciones y que disponen de grupos de estandarización que aseguren el futuro de las mismas.

-NetBeans.

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

FUNDAMENTACION TEORICA.

asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

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:

- ❖ Administración de las interfaces de usuario (ej. menús y barras de herramientas).
- ❖ Administración de las configuraciones del usuario.
- ❖ Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
- ❖ Administración de ventanas.
- ❖ Framework basado en asistentes (diálogo paso a paso).

Ésta plataforma 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 file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en ésta plataforma pueden ser extendidas fácilmente por otros desarrolladores de software. (NetBeans)

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó éste proyecto en junio de 2000 y continúa siendo el patrocinador principal.

-CORBA.

Ésta es una de las plataformas de desarrollo empresarial más importante. Sin duda, el modelo a seguir por cualquier plataforma empresarial.

FUNDAMENTACION TEORICA.

Las ventajas que ofrece CORBA son muy importantes, de las cuales las que más sobresalen son: (Molpeceres Touris, y otros, 2002)

- ❖ Soporte de múltiples sistemas operativos.
- ❖ Soporte de múltiples lenguajes.
- ❖ Gran cantidad de servicios: mensajería, transacciones y persistencia.
- ❖ Controlada por un organismo serio, OMG.

Las ventajas son realmente muy importantes, aún así, CORBA arrastra unos problemas que suponen un verdadero lastre como son:

-Complejidad: es una plataforma de desarrollo muy compleja, aunque existen capas de abstracción que facilitan el desarrollo de aplicaciones, lo cierto es que desarrollar un simple programa no es una labor trivial.

-Burocracia: La evolución de las especificaciones está sujeta a demasiados pasos de burocracia, lo que origina en que para ver novedades en la plataforma sea necesario esperar grandes cantidades de tiempo.

-Pocas soluciones libres: Como consecuencia de su complejidad y de la lentitud de su evolución se deriva que existen pocas soluciones libres.

-NET.

La plataforma de desarrollo empresarial de Microsoft ofrece a los desarrolladores algunas ventajas interesantes: (Molpeceres Touris, y otros, 2002)

-Soporte de múltiples sistemas lenguajes: aunque no soporte todas sus características, lo cierto es que con .NET es posible desarrollar aplicaciones utilizando simultáneamente varios lenguajes de programación.

-Ideal para entornos Microsoft: si una empresa dispone de gran cantidad de software y hardware dependiente de Microsoft, la mejor opción para continuar desarrollando es ésta plataforma, ya que su integración con los productos es perfecta.

FUNDAMENTACION TEORICA.

-Visual Studio .NET: la plataforma .NET dispone de ésta gran herramienta, que además de su potencia ofrece a un entorno homogéneo de desarrollo.

-Un gran departamento de marketing: .NET será una plataforma que se utilizará ampliamente a nivel empresarial gracias al departamento de marketing de Microsoft que destaca por su tremenda eficacia.

-Requiere desarrolladores poco experimentados: bajo la plataforma de desarrollo de Microsoft es posible utilizar lenguajes como VB .NET que hacen muy sencilla la creación de aplicaciones empresariales.

Aún así, si la lista de ventajas es bastante grande, la lista de desventajas no le tiene nada que envidiar, ésta plataforma contiene muchas desventajas, como son:

-No soporta múltiples sistemas operativos: el mundo de .NET gira en torno al sistema operativo Windows y aunque se están intentando trasladar partes importantes de la plataforma, como la CLR o C#, a otros sistemas operativos, lo cierto es que estas partes forman una parte ínfima de la totalidad de la plataforma de Microsoft.

-Un único dueño: está dominada única y exclusivamente por Microsoft. Esto supone un grave problema ya que es una única empresa la que puede añadir y quitar características según sea necesario. Además esto hace que la competencia sea nula y no se estimula la evolución de la plataforma.

-Pocas soluciones libres: no existe una correspondencia exacta entre las partes de la plataforma .NET y soluciones libres. Existen proyectos como Mono o dotGNU que están portando algunas de sus partes, aún así, no se puede crear una arquitectura completa utilizando solamente productos basados en software libre.

-Java Enterprise Edition 2 (J2EE).

La plataforma J2EE es un conjunto de herramientas que crean un escenario ideal para el desarrollo y despliegue de aplicaciones escalables en la Web, y que cuentan con las siguientes características: (Molpeceres Touris, y otros, 2002)

Portable: se puede realizar una aplicación en una máquina Windows o Linux y cuando se haya terminado puede utilizarse en cualquier plataforma para la que haya disponible una Máquina Virtual Java (JVM).

Escalable: si una empresa ve incrementado su número de clientes, nada más sencillo que añadir nuevos componentes J2EE a una aplicación Web para soportar al aumento de clientes, sin tener que reescribir todo el código de nuevo.

Altamente Soportada: prácticamente cualquier empresa de software tiene un contenedor de componentes (o servidor de aplicaciones) Web compatibles con J2EE, entre ellas IBM (Websphere), BEA (WebLogic), Apache (Tomcat), la propia Sun con su nuevo servidor de aplicaciones iPlanet, MacroMedia con (JRun).

A continuación se podrán observar algunas de las ventajas fundamentales que tiene esta plataforma:

-Soporte de múltiples sistemas operativos: al ser una plataforma basada en el lenguaje Java, es posible desarrollar arquitecturas basadas en J2EE utilizando cualquier sistema operativo donde se pueda ejecutar una máquina virtual Java.

-Organismo de control: está controlada por un organismo formado por más de 500 empresas. Entre las empresas que lo forman están algunas de las más importantes del mundo informático (SUN, IBM, Oracle, SAP, HP, AOL, entre otras) lo que garantiza la evolución de la misma.

-Competitividad: muchas empresas crean soluciones basadas en J2EE y que ofrecen características como rendimiento y precio. De este modo el cliente tiene una gran cantidad de opciones a elegir.

-Madurez: creada en el año 1997 como respuesta a la tecnología MTS de Microsoft, J2EE tiene ya varios años de vida y una gran cantidad de proyectos importantes a sus espaldas.

-Soluciones libres: es posible crear arquitecturas completas basadas exclusivamente en productos de software libre. No sólo eso, sino que los arquitectos normalmente disponen de varias soluciones libres para cada una de las partes de su arquitectura.

Aún así, la plataforma de J2EE también tiene desventajas, algunas importantes como:

-Depende de un único lenguaje: depende exclusivamente del lenguaje Java.

-Complejidad: aunque no es una plataforma tan compleja como CORBA, no existe un VB .NET en Java. La creación de aplicaciones bajo J2EE requiere normalmente desarrolladores más experimentados que los necesarios para desarrollar bajo .NET.

-Heterogeneidad: existe una gran heterogeneidad en las soluciones de desarrollo. No existe en J2EE un símil a Visual Studio .NET.

Después de ver todas las características, las ventajas y desventajas que posee ésta plataforma hay que resaltar que es una plataforma segura, mientras otros modelos de aplicaciones empresariales requieren medidas de seguridad específicas en cada aplicación. El entorno de seguridad de la plataforma J2EE permite que se definan unas restricciones de seguridad en el momento de despliegue de la aplicación, aislando así las aplicaciones de la complejidad de las implementaciones de seguridad, la plataforma J2EE hace portables una gran complejidad de implementaciones de seguridad. (Allamaraju, y otros, 2001)

Reduce el coste y la complejidad del desarrollo multicapa y los servicios empresariales. Las aplicaciones J2EE pueden desplegarse rápidamente y pueden mejorarse fácilmente.

Por el estudio realizado anteriormente y partiendo de las ventajas y desventajas que presentan las plataformas de desarrollo estudiadas, las cuales son las más utilizadas en el mundo se decidió tomar la plataforma J2EE como plataforma para el desarrollo del software, pues las desventajas que posee ésta plataforma no son comparables con las desventajas vistas en las demás plataformas estudiadas.

1.4.2- SISTEMAS GESTORES DE BASE DE DATOS (SGDB).

¿Qué es una base de datos?

Una base de datos es un método de almacenamiento de información, mediante el uso de tablas, que a su vez contienen campos, columnas y registros. En ellas se puede guardar todo tipo de información, de manera ilimitada, para ser consultada o modificada posteriormente. Presenta grandes ventajas en cuanto a tiempo y eficacia, puesto que elimina todo tipo de información innecesaria o redundante. (Silberschatz, y otros, 2002)

¿Qué son los Sistemas Gestores de Bases de Datos?

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la Base de datos y el usuario. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan éste tema, o temas relacionados, se mencionan el término SGBD lo que es equivalente a Sistema Gestor de Bases de Datos. (Korth, y otros, 1993)

Hoy en día, son muchas las aplicaciones que requieren acceder a datos. Estos datos se deben almacenar en algún soporte permanente, y las aplicaciones deben disponer de un medio para acceder a ellos. Normalmente, la forma en que un programa accede a un fichero es a través del Sistema Operativo (SO). Este SO ayuda a proveer de funciones como abrir archivo, leer información del archivo, guardar información, entre otras funciones, pero no indica que el SGBD sea parte del SO, solamente es una aplicación. No obstante, este procedimiento de acceso a ficheros es altamente ineficaz cuando se trata con un volumen elevado de información. Es aquí donde aparecen los Sistemas Gestores de Bases de Datos los que proporcionan un interfaz entre aplicaciones y sistema operativo, consiguiendo, entre otras cosas, que el acceso a los datos se realice de una forma más eficiente, más fácil de implementar y, sobre todo, más segura. (Plattini, 1997)

Existen distintos objetivos que deben cumplir los SGBD:

- ❖ Abstracción de la información.
- ❖ Independencia.
- ❖ Redundancia mínima.

FUNDAMENTACION TEORICA.

- ❖ Consistencia.
- ❖ Seguridad.
- ❖ Integridad.
- ❖ Respaldo y recuperación.
- ❖ Control de la concurrencia.
- ❖ Tiempo de respuesta.

Ventajas:

- ❖ Facilidad de manejo de grandes volúmenes de información.
- ❖ Gran velocidad en muy poco tiempo.
- ❖ Independencia del tratamiento de información.
- ❖ Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- ❖ No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- ❖ Integridad referencial al terminar los registros.

Software bajo licenciamiento comercial.

-Oracle.

Sin duda alguna la actual unión entre Dell y Oracle constituye uno de los principales encuentros tecnológicos al servicio de las necesidades empresariales actuales, tras alcanzar más de 22.000 instalaciones de software Oracle en equipo Dell, las empresas han demostrado un sólido éxito en la tarea de entregar mayor beneficio empresarial a una amplia gama de clientes. (SouKup)

Es uno de los manejadores de bases de datos más robustos que existe actualmente. Brinda todos los servicios, tiene un manejador básico de base de datos, tal como MySQL, pero sin olvidar opciones avanzadas como las subconsultas. A pesar de contar con un gran número de posibilidades para realizar las consultas complejas, no es lento al momento de ejecutar las consultas. (Gelais)

El problema máximo al que se enfrenta un usuario cuando decide utilizar Oracle, es el precio. Es un manejador de bases de datos muy poderoso, su precio es un punto

importante a considerar; ya que sólo está al alcance de empresas medianas o grandes o para instituciones bien consolidadas.

Software bajo licenciamiento libre.

-MySQL Server.

Es la base de datos de código fuente abierto más usada del mundo desarrollado y proporcionado por MySQL AB, la cual es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL. (Saenz Russi, 2005)

El servidor MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad y seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet. (Gim Gray)

Entre las principales características tiene:

- ❖ Escrito en C y C++.
- ❖ Trabaja bajo diferentes plataformas.
- ❖ Procesos MultiHilo, capacidad de trabajar servidores con varios procesadores.
- ❖ Velocidad en la utilización de joins y procesos de optimización.
- ❖ Manejo de la memoria a través de manejo del buffer y cache.

Desventajas:

- ❖ Al ser un sistema fácil de manejar, nos impide el aprendizaje del desarrollo de páginas web.
- ❖ Los índices son una desventaja en aquellas tablas las que se utiliza frecuentemente operaciones de escritura (Insert, Delete, Update), esto es porque los índices se actualizan cada vez que se modifica una columna.
- ❖ Los índices también suponen una desventaja en tablas demasiado pequeñas puesto que no necesitaremos ganar tiempo en las consultas.

- ❖ Tampoco son muy aconsejables cuando pretendemos que la tabla sobre la que se aplica devuelva una gran cantidad de datos en cada consulta.
- ❖ Por último hay que tener en cuenta que ocupan espacio y en determinadas ocasiones incluso más espacio que los propios datos.

- PostgreSQL.

Se diseñó como una base de datos orientada a objetos. Esto significa, que las tablas no son tablas, sino objetos, y las tuplas son instancias de ese objeto. Puedes crear nuevos tipos de datos, hacer herencias entre objetos, tiene transacciones, integridad referencial, vistas y multitud de funcionalidades, pero es lento y pesado. Han incorporado la llamada MVCC (Multi-Version Concurrency Control) con lo que los bloqueos de escritura actúan sólo en la sesión del cliente, no en las de los demás clientes.

El sistema de gestión de base de datos PostgreSQL nace en la universidad de Berkeley - California, en los años 80, como un proyecto académico y actualmente se encuentra en la versión 8.3, siendo permanentemente mantenido por la comunidad Open Source.

La coordinación del desarrollo del PostgreSQL es realizada por el Global Development Group, que está formada por un amplio grupo de desarrolladores alrededor del mundo, que permite evolucionar constantemente en cuanto a la corrección de errores y la implementación de nuevas funcionalidades. (Saenz Russi, 2005)

Es una base de datos relacional y con su código fuente disponible libremente. Es el motor de bases de datos de código abierto más potente del momento y en sus últimas versiones empieza a no tener que envidiarle nada a otras bases de datos comerciales. Sus características técnicas la hacen una de las bases de datos más robusta del mercado. Durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. Un punto importante y una de las ventajas más importante que tiene es la poca necesidad de recursos de hardware que requiere.

Algunas características de este gestor de base de datos son: (Worsley, et al., 2002)

DBMS Objeto-Relacional: aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas.

Altamente Extensible: soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

Soporte SQL Comprensivo: soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins).

API Flexible: la flexibilidad del API ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL.

Cliente/Servidor: usa una arquitectura proceso por usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Ventajas:

Instalación ilimitada: es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Esto tiene varias ventajas adicionales:

- ❖ Modelos de negocios más rentables con instalaciones a gran escala.
- ❖ No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- ❖ Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.

Mejor soporte que los proveedores comerciales: además de las ofertas de soporte, tiene una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios.

Estabilidad y confiabilidad legendarias: es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

FUNDAMENTACION TEORICA.

Extensible: el código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales.

Multiplataforma: está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.

Herramientas gráficas de diseño y administración de bases de datos: existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect).

Desventajas:

- ❖ Es un sistema robusto, por lo que consume gran cantidad de memoria.

Partiendo de las ventajas y desventajas mostradas anteriormente se decidió tomar como gestor de base de datos a PostgreSQL, debido a que este trae muy buenas ventajas las cuales harán posible que se pueda obtener el producto final con la calidad requerida. Aunque es un sistema más lento que MySQL, es un sistema mucho más viable para uso de grandes cantidades de base de datos a las que se puede acceder simultáneamente por múltiples usuarios, cosa que MySQL no logra.

1.4.3- LENGUAJES DE PROGRAMACION.

Para la construcción del sistema, se necesita realizar un estudio de los posibles lenguajes de programación que permita crear con mayor facilidad y rapidez dicho sistema. A continuación la descripción de algunos lenguajes para ver características, usos y ventajas para tomar la decisión del que será utilizado para la implementación del sistema.

-C#.

C# es un lenguaje de programación eficaz, diseñado para realizar aplicaciones empresariales. Éste toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. (Ferguson, 2003) (Gonzalez)

FUNDAMENTACION TEORICA.

Es el único lenguaje que ha sido diseñado específicamente para ser utilizado en la Plataforma .NET, éste presenta varias características muy importantes como son:

- **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose, son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes hay que simular.

- **Orientado a objetos:** como todo lenguaje de programación de propósito general, es un lenguaje orientado a objetos.

-**Orientado a componentes:** su propia sintaxis incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.

-**Seguridad:** incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, esto evita que se produzcan errores difíciles de detectar.

-PHP.

PHP (Hypertext Preprocessor) es un lenguaje multiplataforma y multiparadigma para el desarrollo de páginas Web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto y a que es de código abierto, es uno de los más populares y extendidos en el desarrollo de aplicaciones Web. Cada día aumenta el número de programadores que utilizan éste lenguaje. (Olson, et al., 2006)

Actualmente la red de Internet sirve de soporte a una gran cantidad de sistemas de información y comunicaciones que engloban áreas tan importantes como la investigación, el comercio electrónico, la visualización de información y el correo electrónico. (Achour)

Dentro de las principales ventajas que ofrece el lenguaje PHP se encuentran:

- ❖ Es multiplataforma, multiparadigma, libre y gratuito.

FUNDAMENTACION TEORICA.

- ❖ Cuenta con una buena capacidad de conexión con la mayoría de los manejadores de base de datos.
- ❖ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ❖ Presenta una buena integración con Apache y MySQL.

- Java.

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de aplicación. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades de los desarrolladores y de las tecnológicas más punteras. (Agulló, 2004)

Una de las principales características que favoreció la difusión y el crecimiento del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto quiere decir que si se realiza un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues anteriormente tenían que realizar un programa para cada sistema operativo. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java, lo que posibilita que este último se entienda perfectamente.

Otra característica es que éste lenguaje de programación es Orientado a Objetos (OO) se refiere a un método de programación y al diseño del lenguaje. Una primera idea es diseñar el software de forma que los distintos tipos de datos que use estén unidos a sus operaciones. Así, los datos y el código se combinan en entidades llamadas objetos.

Un objeto puede verse como un paquete que contiene el código y los datos. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando su calidad y reduciendo el número de

proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software.

La tecnología Java ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma J2EE, está compuesta básicamente por 2 elementos: el lenguaje Java y su plataforma. La plataforma se refiere a la máquina virtual de Java. El programador es el que determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java es el responsable de gestionar el ciclo de vida de los objetos.

Se decidió utilizar el lenguaje de programación Java, ya que éste brinda una solución para cada necesidad que se pueda presentar, además de ser el lenguaje de programación que utiliza la plataforma J2EE. El cliente opinó acerca de su satisfacción con éste lenguaje.

1.4.4- HERRAMIENTAS A UTILIZAR.

Para lograr una mejor visión y conocimiento del estado del arte en el tema tratado, se realizó un análisis de algunas herramientas de generación de reportes como es el caso de Crystal Report y Jasper Report.

En cuanto a las herramientas para diseñar los reportes, se estudiaron las herramientas más utilizadas en estos momentos para realizar el diseño, las cuales son: Agata Report, Dynamic Jasper, Ireport, Crystal Report, Reporting Service y Actuate Enterprise Reporting.

Herramientas de generación de reportes.

Entre las herramientas para generar reportes se encuentran:

-Crystal Report.

Crystal Report es una herramienta que ha formado parte de Visual Studio desde 1993, y ahora es el estándar de creación de informes de Visual Studio 2005, se incluye en todas sus copias y se integra directamente esta plataforma.

Brinda al usuario, entre otras ventajas, la posibilidad de elegir el tipo de base de datos en que desea trabajar y facilita los cambios en el diseño, es decir, a la hora de editar el informe, se pueden cambiar los formatos, insertar o eliminar objetos y moverlos. (De Nova, 2006)

Esta herramienta utiliza una jerarquía para agrupar los datos, es decir, clasifica los datos según la jerarquía otorgada y así define la estructura de todo el informe. Como consecuencia, si se desea generar reportes con estructura compleja, el usuario está obligado a crear varios subreportes. Elabora los reportes en tiempo de diseño ya que está dirigida más bien para el programador y no para el usuario.

- JasperReport.

Es una herramienta para la generación de informes. Está implementada en Java, es de código abierto y fue desarrollada por Teodor Danciu para facilitar el agregar capacidades de reporte a las aplicaciones Java. No es una herramienta por sí sola ya que no se puede instalar. Para utilizarla es necesario añadirla a las aplicaciones Java por medio de la inclusión de su librería al classpath de la aplicación, la cual no tiene ningún tipo de dependencia con las librerías de Java, lo que posibilita utilizar JasperReport también para aplicaciones Java de escritorios. (Curto Díaz, 2007)

Permite incluir datos de texto, imágenes, gráficos, subreportes, tablas, entre otras características básicas en los reportes, para que estos tengan un aspecto profesional. Las principales características que provee son:

1- Permite una diagramación flexible de los reportes: los reportes se pueden dividir en secciones opcionales que son: título, encabezado de página, una sección para los detalles del reporte, pie de página y una sección de resumen que aparece al final del reporte.

2- Permite que los desarrolladores le surtan datos en varias formas: los desarrolladores pueden pasar datos a los reportes por medio de parámetros. Estos parámetros pueden ser instancia de cualquier clase de Java.

3- Pueden generar sub-reportes: permite la creación de subreportes dentro de los reportes lo que facilita el diseño.

4- Los reportes son capaces de presentar los datos de manera textual o a través de gráficos: son capaces de mostrar los datos que le son pasados y pueden generar o calcular con esos datos otros datos de forma dinámica y mostrarlos.

5- Pueden generar marcas de agua: permite generar textos o imágenes de fondo para utilizarlo como marcas de agua con el propósito de identificar el reporte o simplemente por motivos de seguridad.

6- Se pueden exportar los reportes a una multitud de formatos: los reportes generados pueden ser exportados a una multitud de formatos como PDF, XLS, RTF, HTML, XML, CVS y texto plano.

Requerimientos de JasperReport:

- ❖ Se requiere tener instalado en el equipo el JDK 1.4 o posterior. No basta con tener instalado el J2RE (Run Time Environment).

Se integra perfectamente con JFreeChart que es una librería libre para la generación de todo tipo de gráficas. Para generar el XML es recomendable el uso de la herramienta visual Ireport que es un editor gráfico que también está implementado en Java y se integra perfectamente con el JasperReport.

Herramientas para diseñar los reportes.

Para optar por una herramienta se debe realizar un estudio de las principales herramientas existente para la generación de reportes. Tradicionalmente los reportes se entregaban impresos en papeles, pero cada día los empleados calificados necesitaban datos en otros formatos, que fueran más fáciles de usar y que les ofrecieran un mayor nivel de detalles y flexibilidad. Los consumidores de información necesitan estar en capacidad de interactuar con los resultados obtenidos, para poder tomar sus propias decisiones, basándose en los datos. Aquí radica precisamente la importancia de los sistemas generadores de reportes.

FUNDAMENTACION TEORICA.

Para conocer las principales características de los sistemas generadores de reportes así como sus desventajas se realizó el siguiente estudio basándose en varios aspectos.

Herramientas	Plataforma	Características	Desventajas
Agata Report	Libre	<ul style="list-style-type: none">-Es fácil de instalar y ejecutar.-Es extremadamente flexible.-Multiplataforma (Windows y Linux)-Soporta gráficos, imágenes y sub-reportes-Soporta varios orígenes de datos-Exporta salidas en diferentes formatos (PDF, XML y HTML)	<ul style="list-style-type: none">-Depende de la plataforma PHP por tanto:-Diseñado solo para la Web usando PHP
Dynamic Jasper	Libre	<ul style="list-style-type: none">-Mantiene las principales características del Jasper Report ya que trabaja directamente con él.-Añade cierto grado de dinamismo a los reportes en tiempo de ejecución.-Reduce la complejidad del Jasper Report para reportes simples y de mediana complejidad.	<ul style="list-style-type: none">-Deficiente robustez y eficacia para reportes complejos.-Dependiente de la plataforma Java.
Crystal Report	Propietaria	<ul style="list-style-type: none">-Soporta gráficos, imágenes y sub-reportes.-Da soporte para Web y aplicaciones de escritorio.-Viene integrado a algunas	<ul style="list-style-type: none">-Para modificar el reporte es preciso recompilar el proyecto donde se encuentra.

FUNDAMENTACION TEORICA.

		<p>herramientas de desarrollo como Visual Studio.NET.</p> <ul style="list-style-type: none"> -Cuenta con años de experiencia y desarrollo. -Cuenta con un lenguaje de programación propio y soporta sentencias de Visual Basic. -Exporta salidas en diferentes formatos (PDF, XML y HTML) 	-Es propietario.
Reporting Service	Propietaria	<ul style="list-style-type: none"> -Utiliza el formato estándar XML. -Es una potente herramienta para generar reportes a partir de servidores de Microsoft SQL Server. -Exporta salidas en diferentes formatos (PDF, XML y HTML) 	<ul style="list-style-type: none"> -Fuente de datos restringida a servidores de Microsoft SQL Server. -Dependiente de la plataforma .NET. -Es propietario.
Ireport	Libre	<ul style="list-style-type: none"> - Herramienta visual de diseños de informes para JasperReport potente, intuitiva y fácil de usar. - Requiere el JSDK 1.4 o superior. - Consume bastante RAM, así que es recomendable disponer al menos, 256 MB de memoria y unos 20 MB de espacio libre en el disco duro. -Permite editar informes complejos con graficas, imágenes y subinformes. -Permite visualizar y exportar en varios formatos (PDF,XLS, RTF, HTML y EXCEL) - 100% escrito en Java y además 	-No es suficientemente robusto y flexible como los muy usados Crystal Report y Active Report.

FUNDAMENTACION TEORICA.

		OpenSource y gratuito. - Tiene asistentes para las plantillas. - Soporta internacionalización. - Soporta JDBC.	
Actuate Enterprise Reporting	Propietario	-Flexible en cuanto al formato y diseño. -De fácil manejo y mantenimiento -Potente para reportes internos.	-Es propietario. -Diseñado solo para la Web

Se puede concluir que existen buenas herramientas para la generación de reportes, las cuales soportan gráficos, imágenes y sub-reportes, permiten exportar las salidas en diferentes formatos, como PDF, HTML, XML, XLS, RTF y EXCEL.

- Ireport.

Ireport es un diseñador visual para JasperReport es de código libre y fue escrito en Java. Ayuda a los usuarios y desarrolladores que usan JasperReport a diseñar reportes visualmente a través de una interfaz rica y simple de usar. Provee las funciones más importantes para crear reportes en poco tiempo, pues permite construir reportes, generar archivos "jasper" y "print" de prueba. Ésta herramienta nació como una herramienta de desarrollo, pero puede utilizarse como una herramienta de oficina para adquirir datos almacenados en una base de datos, sin pasar a través de alguna otra aplicación, ya que ella misma permite realizar conexiones a bases de datos, obtener de ellas los datos necesarios y finalmente mostrarlos. (Ireport, 2007)

Requerimientos de instalación (Windows 2000, NT, XP):

- ❖ Sun Java Development Kit (JDK) 1.4 Standard Development Kit (SDK) o superior.
- ❖ Si se desea conectar con una base de datos, se debe proporcionar el Driver Java Database Connectivity (JDBC) correspondiente.

Mediante el estudio de estas herramientas y analizando sus principales características y desventajas se decidió utilizar para diseñar las plantillas de los reportes del software la herramienta Ireport. Ya que es una herramienta bajo licenciamiento libre, es muy usado actualmente en el desarrollo de los reportes en grandes industrias de software, además, las desventajas que presentan no son considerables con respecto a las demás herramientas estudiadas. Es un diseñador visual para JasperReport, el cual fue el que se seleccionó para la generación de reportes.

1.4.5- ENTORNO DE DESARROLLO INTEGRADO (IDE).

-NetBeans.

El entorno de desarrollo integrado (IDE) NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extender el IDE NetBeans, es un producto libre y gratuito sin restricciones de uso. (NetBeans)

El NetBeans es un IDE de código abierto escrito completamente en Java y utiliza la plataforma NetBeans. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en control de versiones y refactorización.

La versión actual de es la 6.5 que fue lanzada el 19 de Noviembre de 2008, la cual extiende las características existentes del Java Enterprise Ediction (JEE). NetBeans Enterprise Pack soporta el desarrollo de aplicaciones empresariales con JEE 5, incluyendo herramientas de desarrollo visuales de SOA, orientación a servicios web, herramientas de esquemas XML y modelado UML.

Sun Studio, Sun Java Studio Enterprise y Sun Java Studio Creator de Sun Microsystems han sido todos basados en el IDE NetBeans. Desde Julio de 2006 éste IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL).

Es utilizado para el desarrollo de aplicaciones de escritorio usando como lenguaje de programación Java y la plataforma NetBeans.

-Eclipse.

El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un entorno de desarrollo integrado (IDE) con un compilador de Java interno y un modelo completo de los archivos fuente, lo que permite técnicas avanzadas de refactorización y análisis de código. (Storkel, 2002)

La versión actual de Eclipse dispone de las siguientes características:

- ❖ Editor de texto.
- ❖ Resaltado de sintaxis.
- ❖ Compilación en tiempo real.
- ❖ Pruebas unitarias con JUnit.
- ❖ Control de versiones con CVS.
- ❖ Integración con Ant.
- ❖ Asistentes (wizards): para creación de proyectos, clases y tests.
- ❖ Refactorización.
- ❖ A través de "plugins" libremente disponibles es posible añadir: Control de versiones con Subversion.
- ❖ Integración con Hibernate.

Lenguajes de programación utilizados en Eclipse

Lenguaje	Líneas de código	%
Java	1,911,693	92.66%
ANSI C	133,263	6.46%
C++	10,082	0.49%
JSP	3,613	0.18%

FUNDAMENTACION TEORICA.

SH	2,066	0.10%
PERL	1,468	0.07%
PHP	896	0.04%
SED	2	0.00%

Se decidió utilizar el IDE Eclipse, ya que éste como se puede observar en la tabla anterior, el lenguaje de programación java es el más utilizado en él, además de que se utilizará la plataforma J2EE, la cual se integra perfectamente con éste IDE y es el más utilizado en las aplicaciones que se han desarrollado en la universidad.

Para la implementación del software se observa que se está utilizando tanto las herramientas como otros productos basados en software libre, lo cual trae como ventaja fundamental que no esté presente un gasto por cuestiones de licenciamientos sin descuidar su calidad.

1.4.6- SERVIDOR DE APLICACIONES.

Un servidor de aplicaciones es un software que proporciona aplicaciones a los equipos o dispositivos cliente, por lo general a través de Internet y utilizando el protocolo http. Los servidores de aplicación se distinguen de los servidores Web por el uso extensivo del contenido dinámico y por su frecuente integración con bases de datos.

Un servidor de aplicación maneja la mayoría de las transacciones relacionadas con la lógica y el acceso a los datos de una aplicación. La ventaja principal de un servidor de aplicaciones es la facilidad para desarrollarlas, puesto que éstas no necesitan ser programadas y en cambio, se arman a partir de módulos previstos por el servidor de aplicaciones. Algunos de los servidores de aplicaciones que se existen para el desarrollo de aplicaciones son:

-Enhydra.

Enhydra es un proyecto maduro. Creado originalmente por la empresa Lutris, fue abandonado cuando ésta decidió dedicarse al desarrollo de proyectos comerciales. Por fortuna, ha vuelto a la carga de manos del consorcio ObjectWeb que ha recogido el relevo de todos sus desarrollos. Se integra a la perfección con productos como JOnAS y su licencia es una mezcla de varios tipos (BSD, Apache y LGPL.). (Enhydra, 2007)

Quizás una de las cosas más interesantes es su proyecto Enhydra MicroEdition. Éste proyecto pretende ofrecer una serie de productos, que tradicionalmente se tendría en un servidor, destinados hacia pequeños dispositivos. Así se puede utilizar sistemas como kXML o kSOAP para utilizar tanto SOAP como XML o instalar un servidor web como kHTTP.

-Jetty.

Jetty es el otro gran contenedor web libre del que dispone el mercado. En marcha desde 1995, donde actuaba ya como servidor web tradicional. Éste contenedor creado originalmente en Australia es la mayor competencia de Tomcat, aunque el menor tamaño de su comunidad y de la publicidad que recibe lo hacen pasar más desapercibido. Al igual que Tomcat, también cumple con las especificaciones más actuales de Servlets y JSP. (Brown, 2008)

Jetty incluye un servidor HTTP bastante aceptable, y aunque puede integrarse con Apache, su rendimiento está mejor valorado que el de Tomcat. Desde la última versión de JBoss, es el contenedor web que viene por defecto con este popular servidor de aplicaciones por lo que es de esperar que su uso se vaya incrementando. Jetty además se integra con gran cantidad de productos libres y está siendo ampliamente utilizado en proyectos como Tapestry, Avalon o Maven. Su licencia es compatible con la Artistic License.

-Apache (TOMCAT).

FUNDAMENTACION TEORICA.

Apache Tomcat es sin lugar a dudas el proyecto de software libre más famoso escrito en Java. Creado a partir de código donado por SUN Microsystems a la Apache Software Foundation, ha crecido hasta convertirse en la implementación oficial de referencia de Servlets y JSP sustituyendo a la implementación de SUN original. Esto lo convierte, en el contenedor sobre el que todos los demás prueban su adhesión a las especificaciones.

Aunque siempre se le ha acusado de un rendimiento pobre, el caso es que a partir de las nuevas versiones ha mejorado considerablemente su rendimiento. Quizás siga sin ser la opción más indicada para una aplicación con miles de clientes simultáneos, pero ningún servidor es adecuado para una carga así. Puede integrarse sin demasiados problemas con el servidor web Apache y para aplicaciones empresariales grandes con servidores de aplicaciones como JBoss o JOnAS. (Foundation)

Se puede integrar con muchos otros productos con licencias propietarias. En estos momentos es el contenedor web que más se está utilizando en servidores de aplicaciones comerciales, donde gigantes como IBM con su todopoderoso WebSphere han elegido a Tomcat como contenedor web.

Es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software Licence.

Una vez estudiado algunos servidores de aplicación se decidió trabajar con el servidor de aplicaciones Apache Tomcat, ya que éste posee muy buenas ventajas y facilidades para los desarrolladores, además de ser un servidor que interactúa muy bien con la plataforma de desarrollo J2EE y el lenguaje de programación Java. Con esta selección se puede ver que el software será desarrollado bajo una arquitectura basada en software libre.

1.4.7- FRAMEWORKS DE DESARROLLO.

Los frameworks son arquitecturas de software bien definidas creados tanto para organizar, como para integrar diferentes componentes de un proyecto con la idea de facilitar su desarrollo. El desarrollo de frameworks en la actualidad posee gran

aceptación entre los desarrolladores, pues éste brinda la posibilidad de reutilizar el código del diseño y el código fuente.

-Symfony.

Symfony fue diseñado para optimizar el desarrollo de las aplicaciones web, posee varias características las cuales brindan numerosos beneficios para desarrollar aplicaciones web fundamentalmente. (Symfony.es)

A continuación se verán reflejadas algunas de las características que posee éste:

- ❖ Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.
- ❖ Es compatible con la mayoría de gestores de bases de datos.
- ❖ Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web.
- ❖ Es multiplataforma.

Ha sido probado en varios proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel, lo que demuestra la confiabilidad de éste y fue desarrollado completamente con PHP 5. (License, et al., 2009)

Automatiza la mayoría de los elementos comunes de las aplicaciones web, como por ejemplo:

- ❖ La capa de presentación utiliza plantillas que pueden ser creadas por diseñadores HTML sin ningún tipo de conocimiento del framework.
- ❖ Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.
- ❖ La gestión de la caché es formidable ya que reduce el ancho de banda utilizado y la carga del servidor.
- ❖ Los formularios incluyen validación automatizada y relleno automático de datos, lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.

Symfony posee una curva de aprendizaje elevada, está basado en el estilo arquitectónico Modelo-Vista-Controlador, el cual está formado por 3 niveles.

-Zend Framework.

Es utilizado para desarrollar aplicaciones y servicios Web con PHP, brinda soluciones para construir sitios web, es orientado a objeto y posee una estructura de componentes bien diseñada con pocas dependencias entre ellos. (Framework)

Presenta varias características:

- ❖ Robustas clases para autenticación y filtrado de entrada.
- ❖ Incluye objetos de las diferentes bases de datos.
- ❖ Trabaja con el patrón Modelo Vista Controlador y PHP5.
- ❖ Posee completa documentación de alta calidad.
- ❖ Es Open Source.

En la última versión se encuentra liberado y trae muchas novedades como son:

- ❖ Posee soporte para documentos Open.
- ❖ Mejoras de ficheros en el envío y en la internacionalización.

Todavía le falta madurar más para lograr convertirse en unos de los frameworks más usados en el mundo, su curva de aprendizaje es elevada y se está constantemente trabajando por parte de los desarrolladores de Zend, para que ocupe una posición importante en el desarrollo de las diferentes aplicaciones.

-Spring.

Facilita a los desarrolladores la implementación del software, promoviendo buenas prácticas de programación. Maneja objetos del negocio y además se integra fácilmente con otros frameworks. El framework Acegi está íntimamente ligado a Spring, el cual se encarga de manejar la seguridad de la aplicación. Éste framework se hizo open source en febrero de 2003. (Walls, 2005)

Tiene el objetivo de facilitar la construcción de aplicaciones en Java, se puede utilizar en cualquier tipo de aplicaciones, no solamente es utilizado en aplicaciones web y es

FUNDAMENTACION TEORICA.

ligero debido al poco impacto que tiene en las aplicaciones. Lo que trae varios beneficios arquitectónicos como son:

- ❖ Puede organizar eficazmente la capa intermedia de cualquier aplicación, tanto de escritorio como web.
- ❖ Las aplicaciones con Spring son fáciles de testear.
- ❖ Facilita el desarrollo de buenas prácticas, reduce el costo de programación en una aplicación.

Spring está basado en el principio de inyección de dependencias, esta técnica hace externo el manejo y la creación de las dependencias de los componentes, logrando así obtener una mayor claridad y limpieza en el código. (Harrop, 2005)

Como ventajas fundamentales que trae la utilización de Spring están:

- ❖ Reduce potencialmente el código de enlace entre los diferentes componentes de la aplicación.
- ❖ Permite la reconfiguración de las dependencias sin necesidad de compilar todo el código.
- ❖ Disminuye el margen de errores.
- ❖ Fomenta un buen diseño de la aplicación.

La principal característica que posee Spring es el soporte a la programación orientada a aspectos (AOP), la cual es una tecnología del momento en el mundo de Java, esto posibilita el manejo de trazas, de la seguridad y de las transacciones. Sin lugar a duda es una buena opción para el desarrollo de aplicaciones, pues como se puede observar facilita un bajo acoplamiento y permite obtener un código limpio y claro. (Johnson, 2005)

-Hibernate.

Realiza el mapeo entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. Se encarga de gestionar la capa de Acceso a Datos de un software y realiza la transición de los datos de un modelo relacional a un modelo orientado a objetos. Está basado en la implementación del patrón DAO, pues crea una capa separada que se ocupa del acceso a datos con

total independencia del gestor y la base de datos, lo que brinda la posibilidad de trabajar con varios gestores y bases de datos dentro de la misma aplicación. (Bauer, 2005)

Soporta todos los sistemas gestores de bases de datos SQL y se integra sin restricciones con los contenedores web más conocidos. Constituye un motor de persistencias que implementa múltiples funcionalidades y fue liberado bajo la Lesser GNU Public License.

Tiene como principales características las siguientes:

- ❖ Soporta el paradigma de orientación a objetos de una manera natural, como es caso de utilizar herencia, composición, polimorfismo y las librerías de colecciones de Java.
- ❖ Posee lenguaje de consultas, lo que proporciona una independencia del SQL de cada base de datos ya sea para la recuperación como para el almacenamiento de objetos.
- ❖ Permite una gran variedad de mapeos para colecciones y objetos dependientes.
- ❖ Posee una cache de dos niveles, un alto rendimiento y puede ser utilizado en un clúster.

Presenta un mecanismo de persistencia transparente lo que trae consigo que los objetos desconocen su capacidad de persistencia, no necesita un contenedor de aplicaciones, son de fáciles de manejar y pueden ser usados para transportar los datos a cualquier capa de la aplicación. Estas son algunas de las razones por las que ha ganado en preferencia y popularidad entre los grandes desarrolladores de aplicaciones. (Heudecker, 2003)

Por lo planteado anteriormente se propuso que se apliquen los siguientes frameworks, los cuales brindan ciertas facilidades y servicios para gestionar las funcionalidades de cada una de las capas en las que está organizada la aplicación.

Para la implementación de la capa de Negocio se decidió utilizar Spring por todas las ventajas que éste posee. Como actualmente en las aplicaciones informáticas, el acceso a datos representa un serio problema, Hibernate fue el seleccionado para

implementar la capa de Acceso a Datos pues proporciona una solución factible a dicho inconveniente.

1.5- METRICAS PARA LA IMPLEMENTACION.

Los objetivos primarios para las métricas orientadas a objetos son:

- ❖ Entender mejor la calidad del producto.
- ❖ Evaluar la efectividad del proceso.
- ❖ Mejorar la calidad del trabajo llevado a cabo a nivel del proyecto.

Cada uno de estos objetivos son importantes, ya que para el ingeniero de software la calidad del producto debe ser lo más importante. El software orientado a objetos (OO) es fundamentalmente diferente del software desarrollado con el uso de métodos convencionales, por este motivo las métricas para sistemas OO deben ser afinadas a las características que distinguen al software OO del software convencional. Berard define cinco características que regulan las métricas especializadas, estas características son: (Olsina, et al., 2003) (Olsina, et al., 2002)

- ❖ Localización.
- ❖ Encapsulación.
- ❖ Ocultamiento de información.
- ❖ Herencia.
- ❖ Técnicas de abstracción de objetos.

Localización: la localización es una característica del software que indica la manera en que la información se concentra en un programa. En el contexto OO, la información se concentra por la encapsulación de datos y procesos dentro de los límites de una clase u objeto. Las métricas que reflejan la manera en la que las clases colaboran deben ser capaces de acomodarse a relaciones uno a muchos y muchos a uno.

Encapsulación: Berard la define como el empaquetamiento de una colección de elementos. La encapsulación engloba las responsabilidades de una clase, incluyendo sus atributos y operaciones. Eleva la medición a un nivel de abstracción más alto.

Ocultamiento de información: oculta los detalles operacionales de un componente, proporciona la información necesaria para acceder al componente.

1.6- CONCLUSIONES.

Éste capítulo recoge elementos importantes relacionados a los reportes, partiendo de un análisis de los antecedentes del tema, así como un estudio de las metodologías y herramientas candidatas a usar en el desarrollo de la solución propuesta. Se argumentan las principales características que poseen y que hicieron posible la selección para el desarrollar nuestro trabajo.

Con la selección de la metodología de desarrollo y del entorno de desarrollo nuestra solución no tiene grandes costes de licencias por lo que el margen de beneficios que se puede ofrecer es muy grande. Queda claramente demostrado que las arquitecturas basadas en software libre son las que ofrecen la mejor relación calidad/precio.

CAPITULO 2: PROPUESTA DE SOLUCION AL PROBLEMA.

2.1- INTRODUCCION.

En este capítulo se abordará la descripción de la propuesta para la solución del problema de la investigación. Se va a partir de los diagramas de Casos de Usos (CU) y de los diagramas de clases que fueron diseñados en el flujo de trabajo análisis y diseño. Se abordará acerca de la implementación del diseño de las plantillas de cada uno de los reportes.

2.2- ELEMENTOS DE DISEÑO UTILIZADOS EN LA IMPLEMENTACION.

A continuación se muestra el diagrama de CU del sistema donde se verán identificados aquellos CU que pertenecen a la implementación de los reportes de dicho módulo.

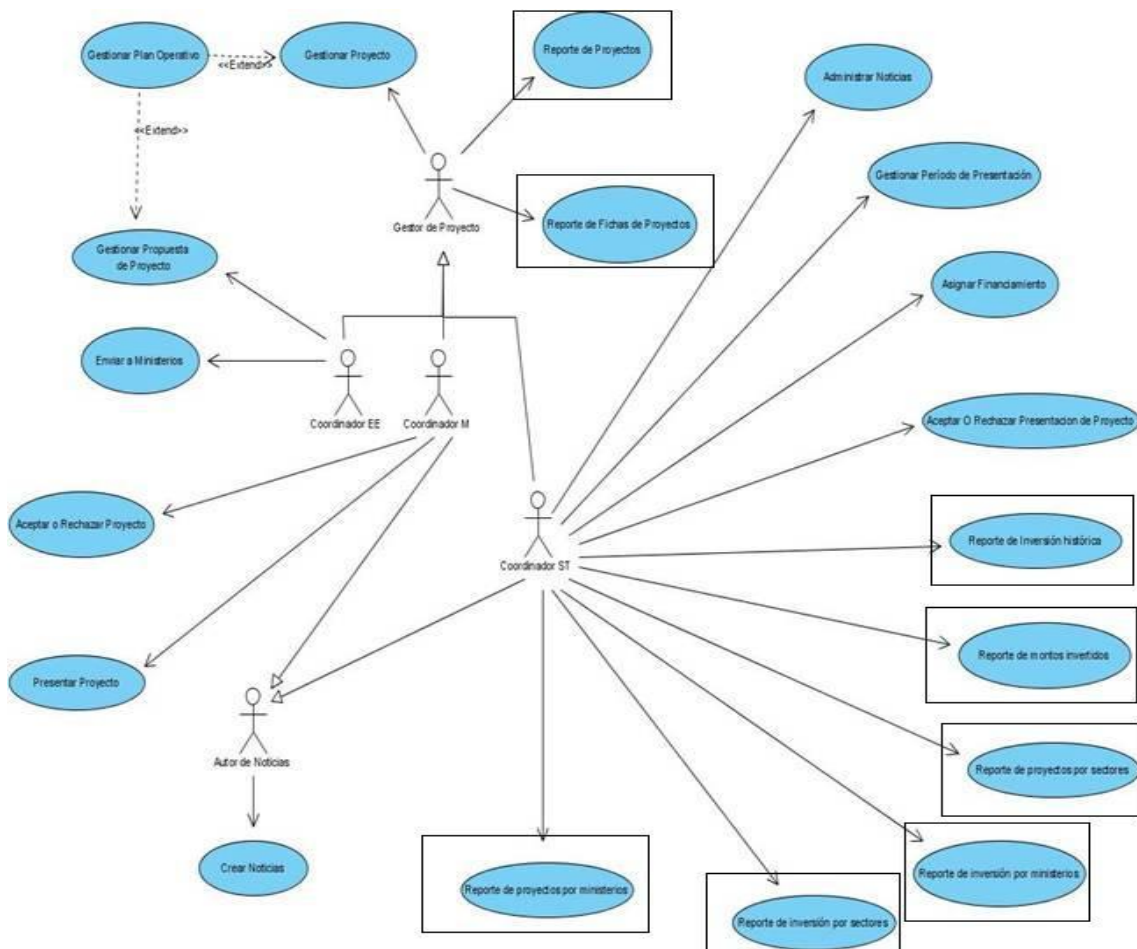


Figura 1 Diagrama de CU del sistema.

PROPUESTA DE SOLUCION AL PROBLEMA.

Para comenzar la implementación de los reportes son necesarios algunos artefactos del flujo análisis y diseño como es el caso de los diagramas de clases, estos diagramas mostrarán las clases que se utilizarán, donde en cada clase se verán sus atributos y los métodos que se implementarán. Es muy importante contar con los prototipos de interfaz de los reportes además de poder contar también con el modelo de negocio, pues sería muy útil ya que para implementar los reportes se tienen que diseñar las plantillas, las cuales son las que mostrará el reporte. Será muy importante realizar el modelo de componente para guiar la implementación del producto.

Se realizaron los diagramas de clases de todos los CU del sistema los que se encuentran en el documento: Diagramas de clases de los Casos de Usos de los reportes de presentación (Sánchez Villa, 2009). A continuación se podrá observar el diagrama de clase de los CU más importantes y de mayor complejidad de los reportes del módulo de presentación del proyecto CICCIV.

Estos diagramas son:

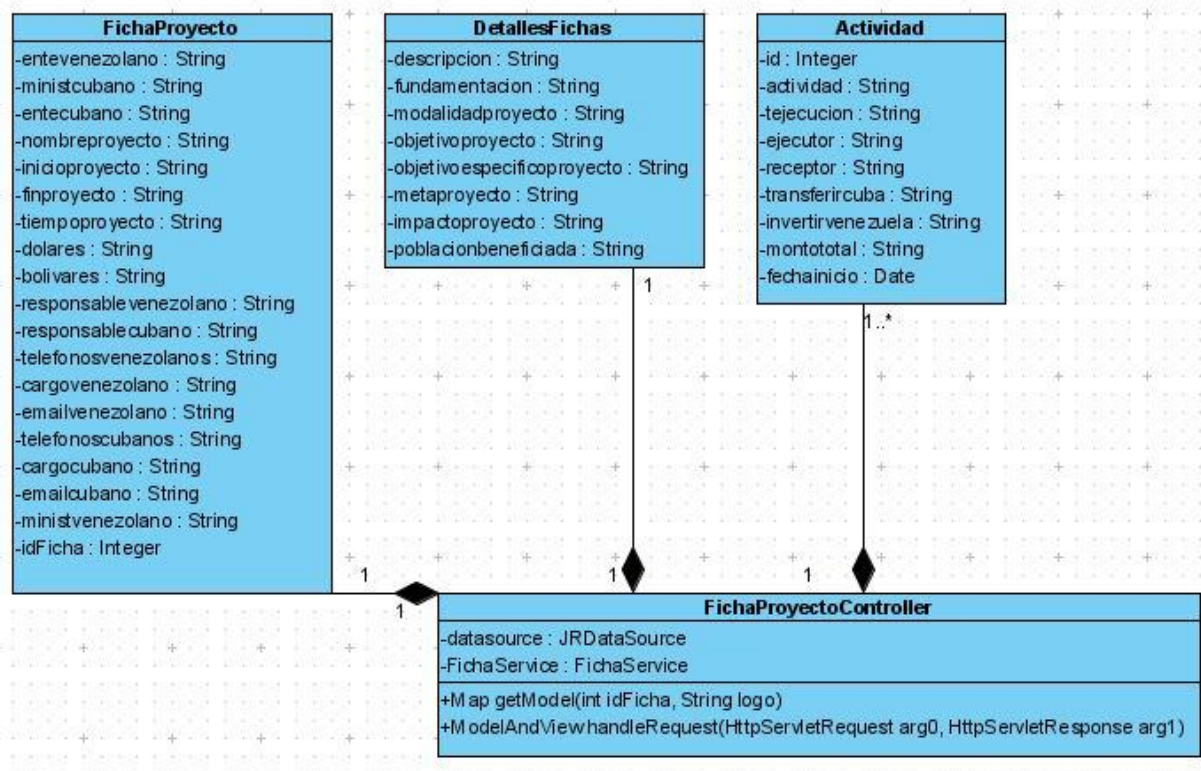


Figura 2 Diagrama de clase CU Ficha Proyecto.

PROPUESTA DE SOLUCION AL PROBLEMA.

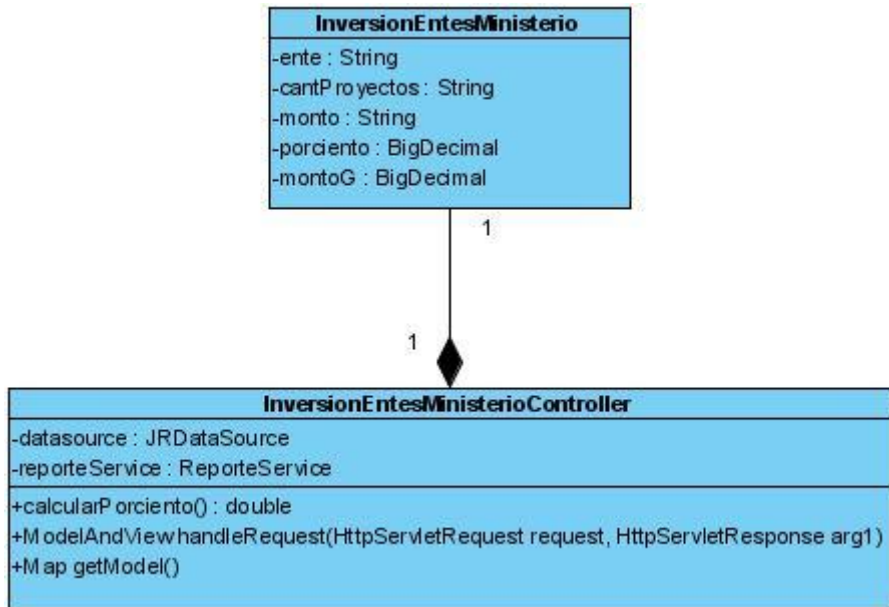


Figura 3 Diagrama de clase CU InversionEntesMinisterios.

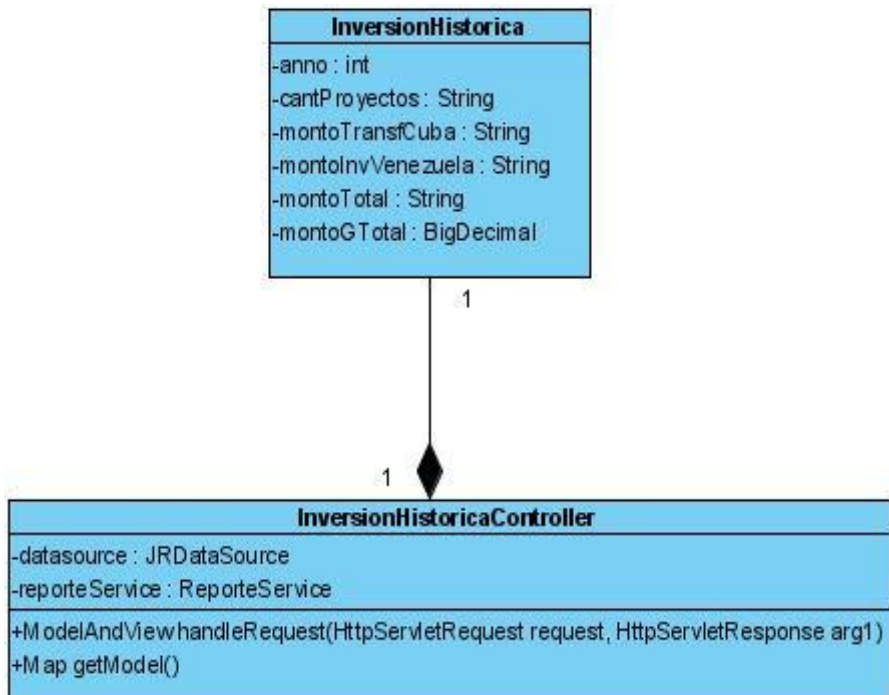


Figura 4 Diagrama de clase CU InversionHistorica.

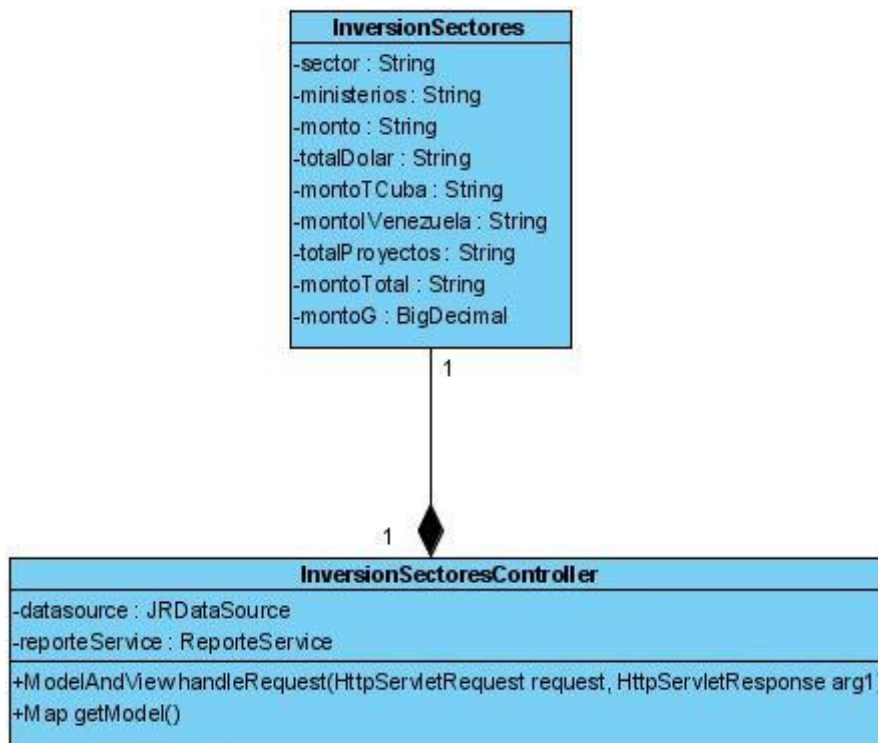


Figura 5 Diagrama de clase CU InversionSectores.

2.3- OBTENCION DE LAS PLANTILLAS PARA GENERAR LOS REPORTES.

Para confeccionar y generar las plantillas se utilizará la herramienta Ireport, seguidamente se podrá observar cómo fueron diseñadas estas plantillas partiendo del modelo de negocio y el prototipo de interfaz.

La plantilla deberá ser rellena con los datos que conformarán el informe. Su diseño puede incluir todo tipo de elementos gráficos (cuadrículas, gráficas, imágenes y recuadros). Todos estos parámetros constituyen el fichero JRXML, es decir, un diseño de informe que ha de ser compilado por el propio Ireport utilizando las clases de JasperReport. Una vez compilado el informe se obtiene un fichero .jasper, el cual será utilizado desde la aplicación para generar el reporte.

En todo informe, los datos a mostrar pueden ser estáticos o dinámicos, los cuales procederán de una base de datos. Se debe configurar el DataSource para que Ireport pueda acceder a la misma. Ireport admite un gran número de bases de datos, entre las que se encuentran MySQL, HSQL, Oracle y Postgress.

PROPUESTA DE SOLUCION AL PROBLEMA.

Sin embargo, sólo se usará la conexión a base de datos durante la fase de desarrollo de las plantillas, para verificar que el correcto funcionamiento. Ireport permite crear cualquier tipo de informe de una manera sencilla y rápida a través de una interfaz gráfica intuitiva, diseñada tanto para las personas que no están familiarizadas con ésta tecnología como para usuarios expertos o desarrolladores que ya conocen ésta herramienta, ahorrándoles tiempo durante el desarrollo de informes.

Dicha herramienta brinda la posibilidad de crear las plantillas con las características que el cliente desee, estas plantillas están formadas por 8 bandas, las cuales tienen un motivo específico.

Creación de las plantillas

Para obtener una nueva plantilla se debe ir a Menú Informes -> Nuevo, donde aparece el siguiente cuadro de diálogo:

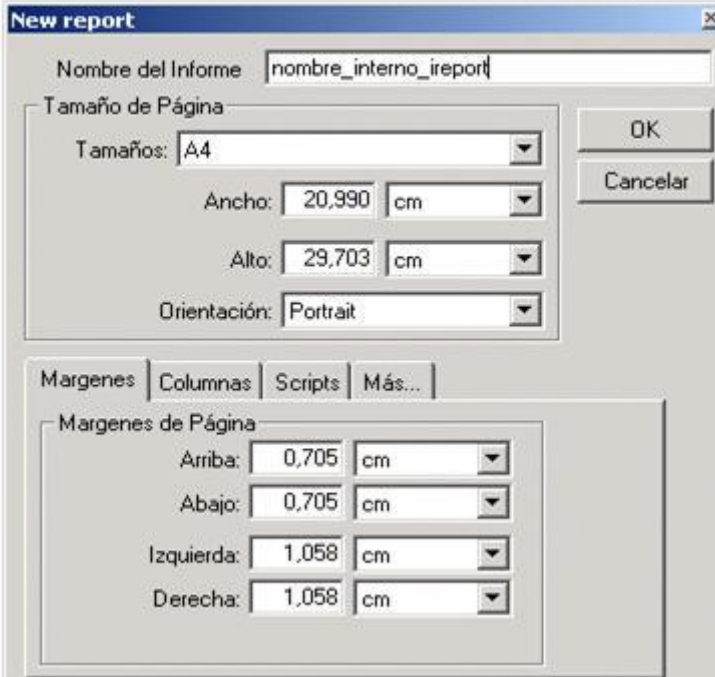


Figura 6 Cuadro de diálogo de crear nueva plantilla de informe.

En el cual tenemos “**Nombre del informe**”, éste es el nombre interno que va a tener el informe en Ireport, el cual coincide con el nombre del fichero XML y el fichero .jasper que se genera una vez que se compila la plantilla, éste fichero .jasper es el que se utilizará para generar el reporte desde la aplicación.

PROPUESTA DE SOLUCION AL PROBLEMA.

También aparece “**Tamaño de la página**”, normalmente el estándar que se utiliza es la medida A4.

Hay que entrar en la pestaña “**Más...**”, allí aparece “**Codificación XML**”, donde se debe poner **ISO-8859-1** para que se puedan escribir la “ñ” y no de error al compilar la plantilla.

Una vez que se pulsa “**OK**”, se crea la plantilla y se recomienda que ésta sea guardada. En la siguiente figura se muestra como queda ésta plantilla inicialmente.

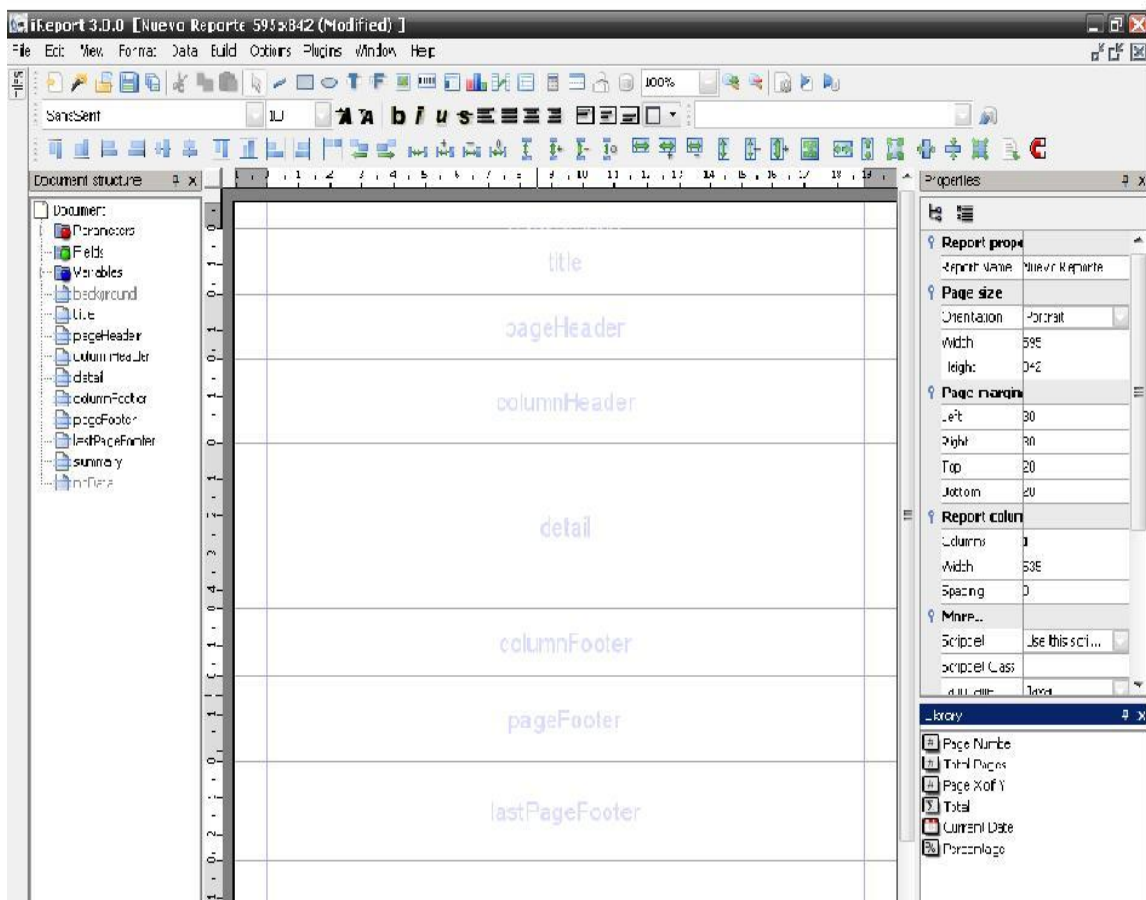


Figura 7 Bandas del informe.

En ella aparecen marcadas diferentes áreas las que se denominan “**bandas**”.

Un informe en iReport está pensado a nivel de página. Lo que se define es el aspecto general de una página y no se pueden definir elementos fuera del tamaño de la misma.

PROPUESTA DE SOLUCION AL PROBLEMA.

El significado de cada banda es el siguiente:

- ❖ **Title:** se mostrará sólo una vez y es al principio del informe.
- ❖ **PageHeader:** es la cabecera de la página, se repite cada vez que se muestre una página nueva.
- ❖ **ColumnHeader:** es la cabecera de las columnas y su comportamiento es análogo a **PageHeader**.
- ❖ **Detail:** es la encargada de mostrar los elementos que tienen alguna repetición, estos elementos se mostrarán en los subinformes.
- ❖ **ColumnFooter:** Pie de la columna. Su comportamiento es análogo a **ColumnHeader**.
- ❖ **PageFooter:** Pie de página, se repite una vez por página. Su comportamiento es análogo a **PageHeader**.
- ❖ **Summary:** Sólo se repite una vez por informe en la última página del mismo. Su comportamiento es análogo a **Title**.

Parámetros (\$P):

Los parámetros se utilizan para definir la aparición o no de textos o para mostrar algún valor concreto que no se pasa como campo (\$F). Cuando los parámetros se usan para mostrar o no un texto, tiene un valor "**true**" o "**false**", porque tanto los parámetros como los campos se pasan como textos.

Los parámetros se utilizan cuando es un dato que no se conoce o no siempre es el mismo, es decir, va a cambiar algunas veces. Éste se utilizará por ejemplo para mostrar el intervalo de años que se toman la información que se muestra en el reporte.

Como se crean los parámetros en Ireport:

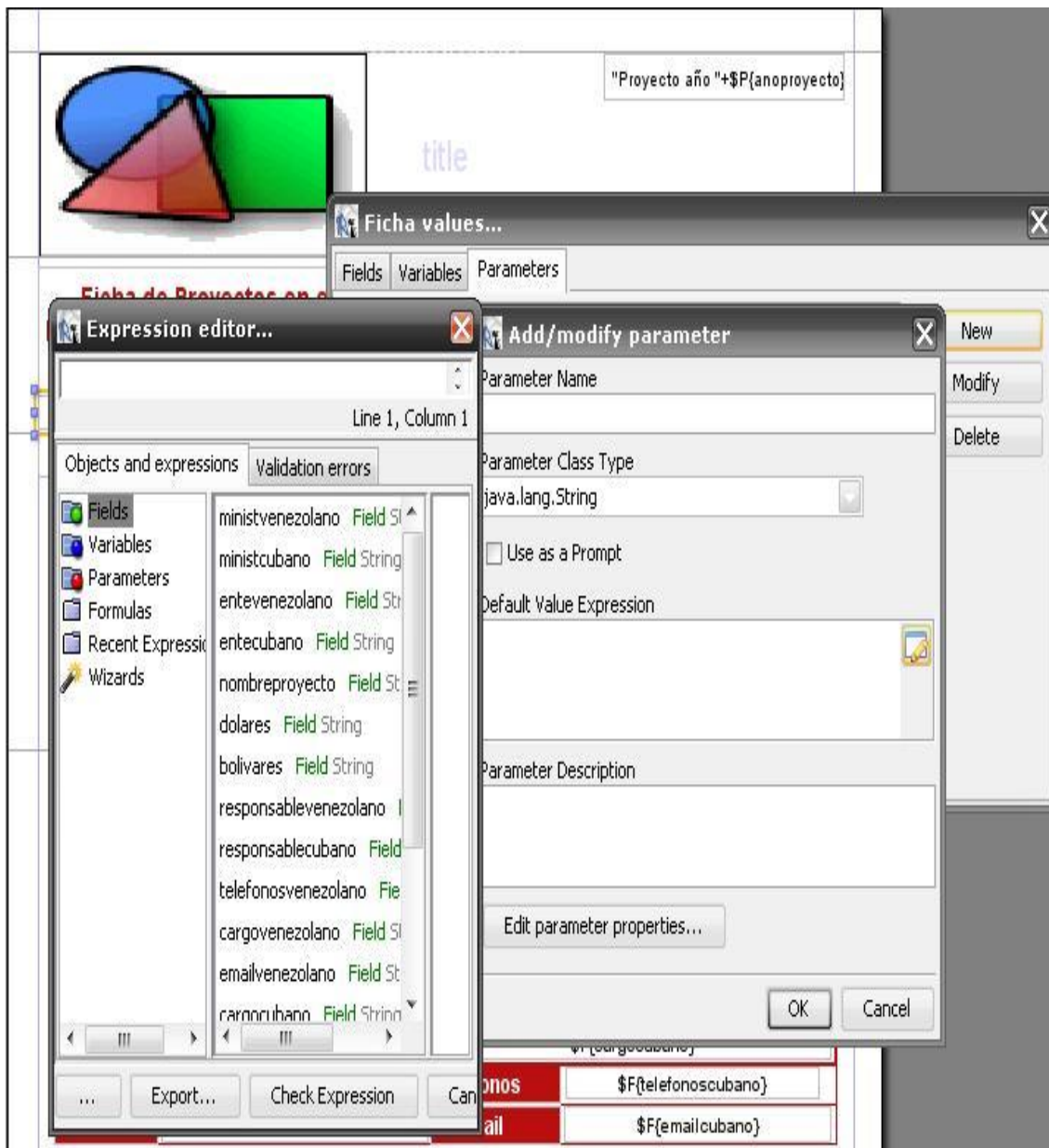


Figura 8 Crear parámetros en Ireport.

Campos (\$F):

Siempre se van a definir como datos de tipo texto, serán los campos que se recuperen de la base de datos y se pasen en un objeto DataSource. Los campos son los que más se utilizan, debido a que es donde se muestra la información que sea repetitiva, es donde se mostrará toda la información que se quiere sin importar cuantas veces se repita dicha información.

PROPUESTA DE SOLUCION AL PROBLEMA.

Como se crean los campos en Ireport:

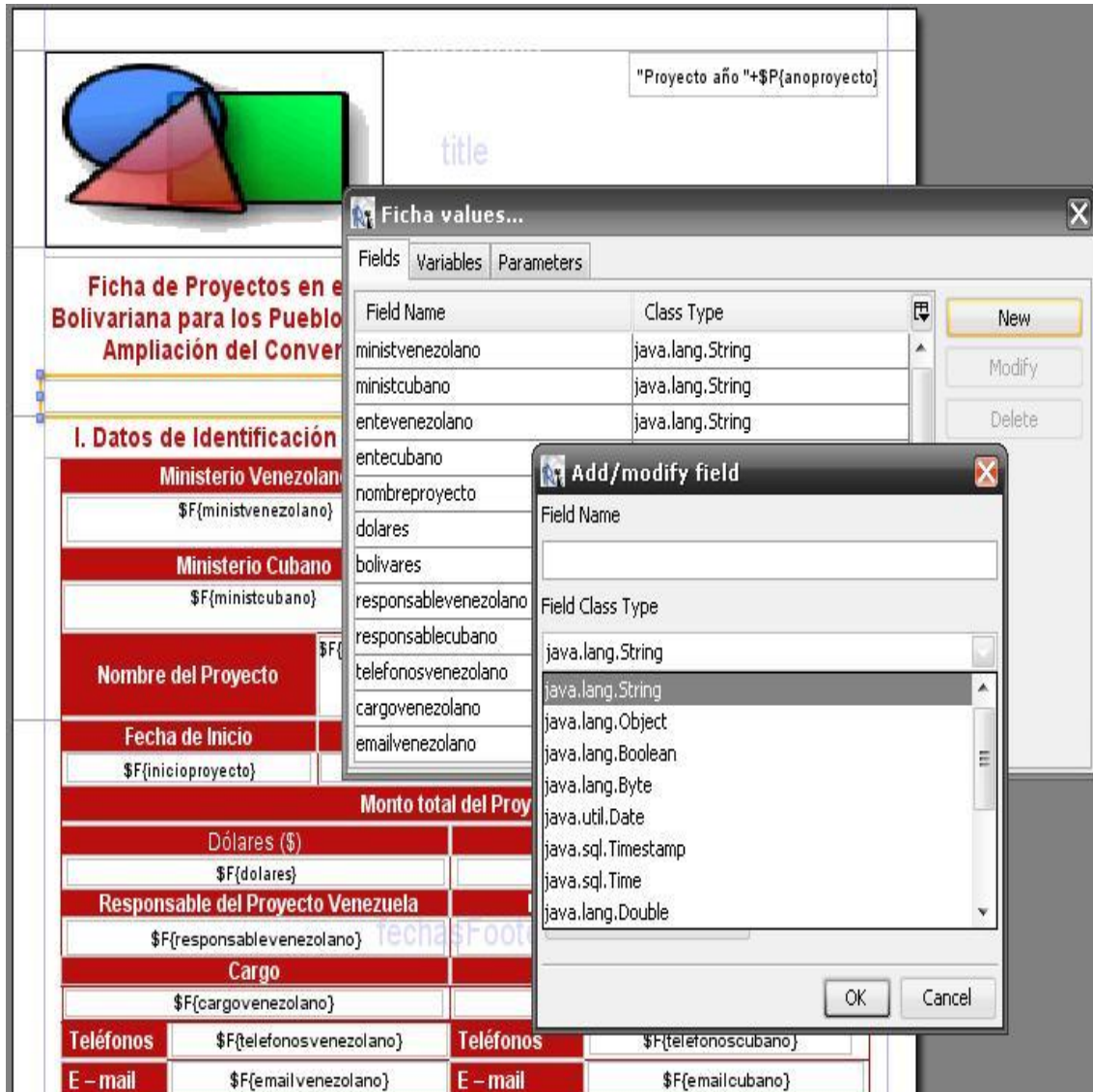


Figura 9 Crear campos en Ireport.

Variables (\$V):

Se utilizan para hacer cálculos dentro del informe, como por ejemplo calcular una suma de campos o un paginado. Ésta se utilizará para mostrar algunos cálculos que son necesarios mostrar en el informe, la utilización de variables es fundamental pues nos ahorra el tiempo de realizar cálculos ya que ellas mismas realizan estas operaciones.

PROPUESTA DE SOLUCION AL PROBLEMA.

Se observará como se crean las variables en Ireport:

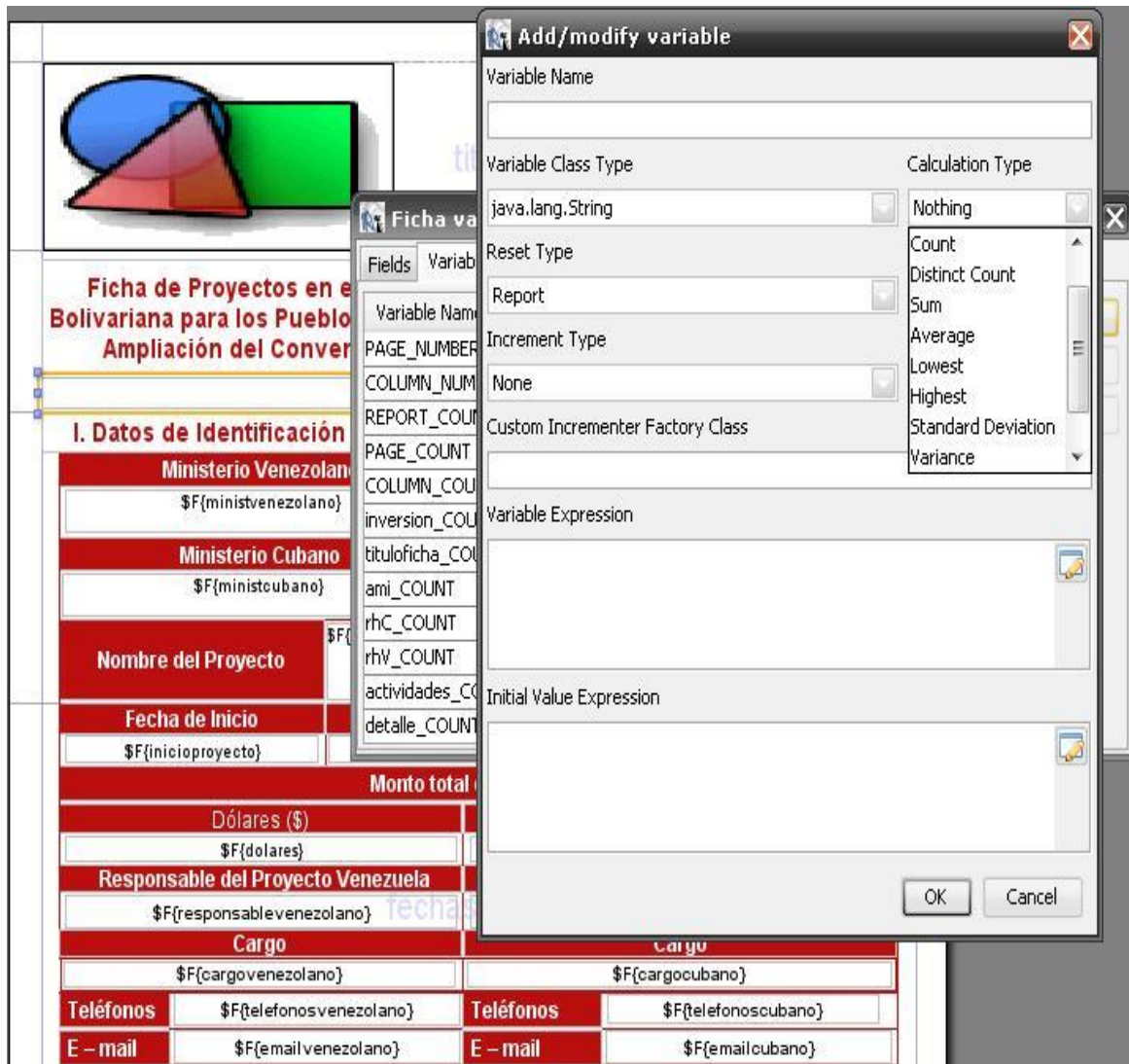


Figura 10 Crear variables en Ireport.

Subreportes:

Se usarán sólo cuando se quieran mostrar numerosas ocurrencias de varios campos agrupados de alguna forma determinada. Si un informe es lo suficientemente complejo para tener varios grupos de repetición hay que usar subinformes o subreporte.

Un subinforme no es más que otro fichero XML asociado al fichero XML principal y que comparte todos o parte de sus datos y es llamado desde él. Tanto el informe como el/los subinformes se deben definir de un modo determinado para que cumplan unos requisitos predeterminados.

PROPUESTA DE SOLUCION AL PROBLEMA.

Jasper sólo puede tener asociado un fichero fuente (XML) a la hora de la generación del informe, por lo que todos los ficheros de subinformes deberán estar pre-compilados (.jasper). El fichero XML en el que definamos el subinforme debe llamarse igual que el fichero ".jasper" que hemos definido en la llamada al subinforme desde el informe principal.

Para realizar el subreporte es necesario tener en cuenta:

- ❖ Cuando definimos el tamaño del informe, el ancho del mismo debe ser el mismo ancho del elemento "subreporte" que se pintado en la pantalla del informe principal.
- ❖ Hay que ponerle todos los márgenes a 0, ya que los márgenes ya los tiene el informe principal.
- ❖ No se debe pagarinar. De eso se tiene que encargar el informe principal.
- ❖ Los elementos que se repitan tienen que ir en la banda de detalle.

Plantillas de los reportes del módulo de Presentación.

Dado los prototipos de interfaz tomados de la fase de inicio del proyecto en el documento: " Plantilla DCS - Modelo de Casos de uso del sistema v1.0" (Montané Izaguirre, et al., 2008) se generaron las plantillas de los reportes, aquí se verán algunas plantillas de los CU más importantes, se realizaron 53 plantillas tomando en cuenta las plantillas de los subreportes, las cuales se diseñan al igual que la de los reportes, el resto de las plantillas se pueden encontrar en la aplicación.

Lo que se encuentra en color gris son subreportes que tiene dicho reporte ya que la información que se va a manejar es muy grande y requiere un mayor espacio y complejidad. En éste caso en particular existen cuatro subreportes, uno para el plan operativo, dos para los recursos humanos en ambos países y otro para la adquisición de materiales, pues en un proyecto pueden existir numerosos recursos humanos, planes operativos y adquisiciones de materiales, esto trae consigo una repetición de muchos datos, por lo que sería imposible realizar este reporte sin la utilización de subreportes.

PROPUESTA DE SOLUCION AL PROBLEMA.

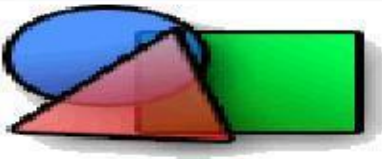
	"Proyecto año "+\$P{anoproyecto}						
title							
Ficha de Proyectos en el marco de la Instrumentación de la Alternativa Bolivariana para los Pueblos de Nuestra América, a partir de la Modificación y Ampliación del Convenio Integral de Cooperación Cuba – Venezuela							
\$P{nomixta}							
I. Datos de Identificación del Proyecto							
Ministerio Venezolano \$F{ministvenezolano}	Ente Ejecutor Venezolano \$F{entevenezolano}						
Ministerio Cubano \$F{ministcubano}	Ente Ejecutor Cubano \$F{entecubano}						
Nombre del Proyecto	\$F{nombrepryecto}						
Fecha de Inicio \$F{inicioproyecto}	Fecha de Culminación \$F{finproyecto}	Tiempo de Ejecución \$F{tiempoproyecto}					
Monto total del Proyecto							
Dólares (\$) \$F{dolares}	Bolívares (Bs.F) \$F{bolivares}						
Responsable del Proyecto Venezuela \$F{responsablevenezolano}	Responsable del Proyecto Cuba \$F{responsablecubano}						
Cargo \$F{cargovenezolano}	Cargo \$F{cargocubano}						
Teléfonos \$F{telefonosvenezolano}	Teléfonos \$F{telefonoscubano}						
E – mail \$F{email venezolano}	E – mail \$F{email cubano}						
II. Detalle de los recursos necesarios para el logro del objetivo del Proyecto							
Plan Operativo							
No	Actividad	Tiempo de Ejecución	Ejecutor	Receptor	Transferir a Cuba	Invertir en Venezuela	Monto Total
Recursos Humanos por Venezuela							
Categoría	Cantidad	Honorario	Tiempo				
Recursos Humanos por Cuba							
Categoría	Cantidad	Honorario	Tiempo				
Adquisición de Materiales e Insumos							
Materiales e Insumos	Cantidad	Monto	País				
pageFooter							
"Página " + \$V "" + \$V							

Figura 11 Plantilla del CU FichaProyecto.

PROPUESTA DE SOLUCION AL PROBLEMA.

Inversión por Entes Ejecutores de un Ministerio			
del año \$P{anioInicio}		al año \$P{anioFin}	
\$P{fecha}			
Nombre del Ministerio: \$P{Ministerio}			
Entes Ejecutores	Números de proyectos	Monto (USD)	Porcentaje (%)
\$F{ente}	\$F{cantProyectos}	\$F{monto}	\$F{porcentaje}

Monto total : \$P{montoTotal}
Total de proyectos: \$P{totalProyectos}

Figura 12 Plantilla del CU InversionEntesMinisterios.

Aquí se hace uso de una de las gráficas que facilita la herramienta, en éste caso se ve una gráfica de pastel, también existen otros tipos de gráficas como la de pastel en tercera dimensión (3D), brinda la posibilidad de utilizar las gráficas de barras, de puntos y de líneas entre otras.

PROPUESTA DE SOLUCION AL PROBLEMA.

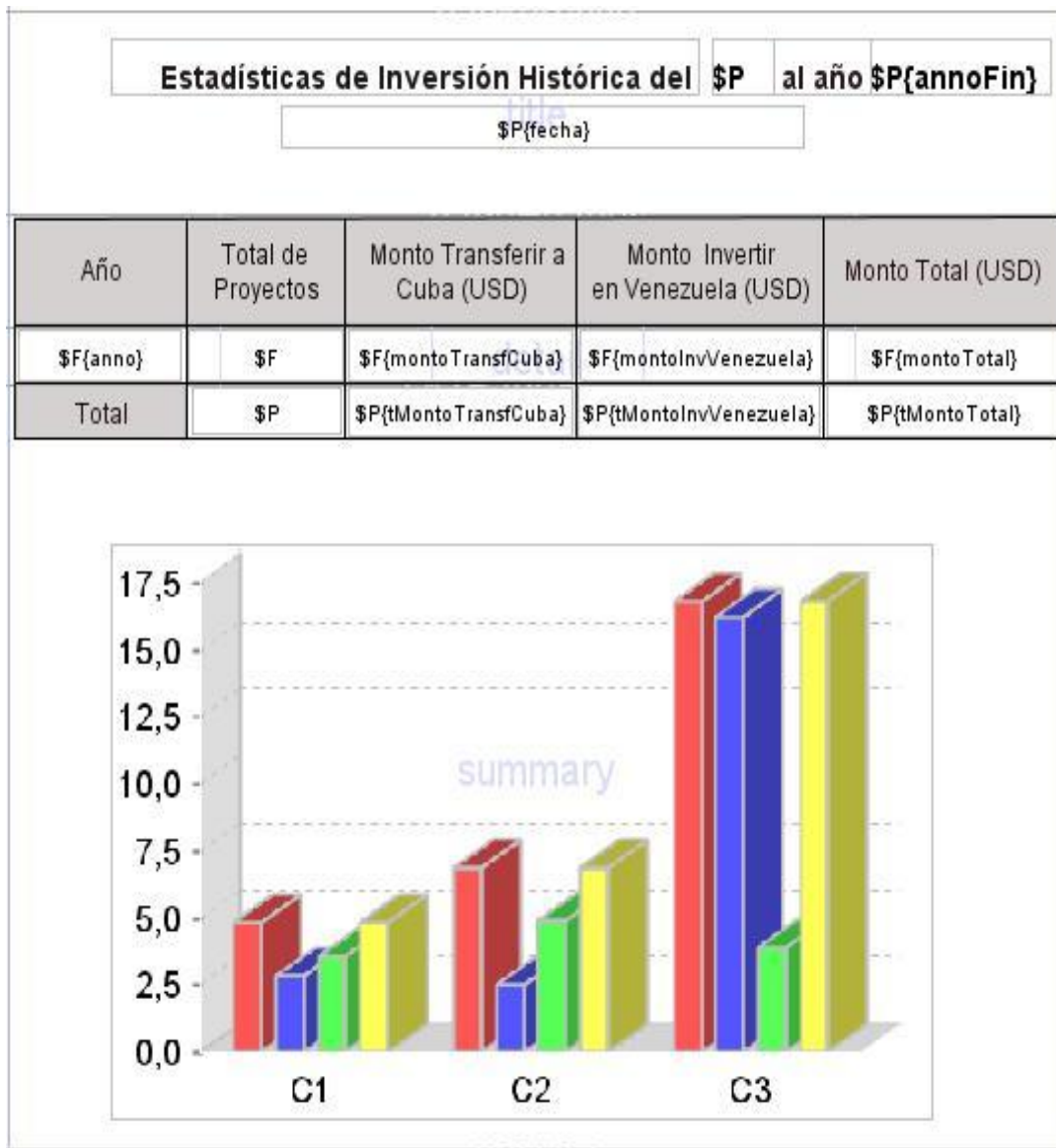


Figura 13 Plantilla del CU InversionHistorica.

Una vez obtenidas todas las plantillas se pasa a la implementación de las clases necesarias, tanto las clases entidades como las clases controladoras, las cuales son las encargadas de obtener toda la información necesaria desde la capa de negocio y acceso a datos y mostrar el reporte en el producto final. Se realizará una clase entidad por cada CU y por cada CU se implementará una clase controladora, en el caso de los subreportes, por cada uno se implementará una clase entidad, estas clases serán utilizadas por la clase controladora del reporte principal, lo que significa que cada clase entidad no necesariamente tendrá una clase controladora.

2.4- FLUJO DE TRABAJO DE IMPLEMENTACION.

Dentro del flujo de trabajo Implementación, el cual tiene su mayor peso en la fase de construcción, es importante la realización de los artefactos más importantes.

Se observará a continuación cuales fueron estos artefactos y como quedaron desarrollados para lograr obtener una buena y eficiente implementación de los reportes del Módulo de Presentación del proyecto CICCIV.

Ya que el modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación, es natural mantener el modelo de implementación a lo largo de todo el ciclo de vida del software.

Diagrama de despliegue.

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

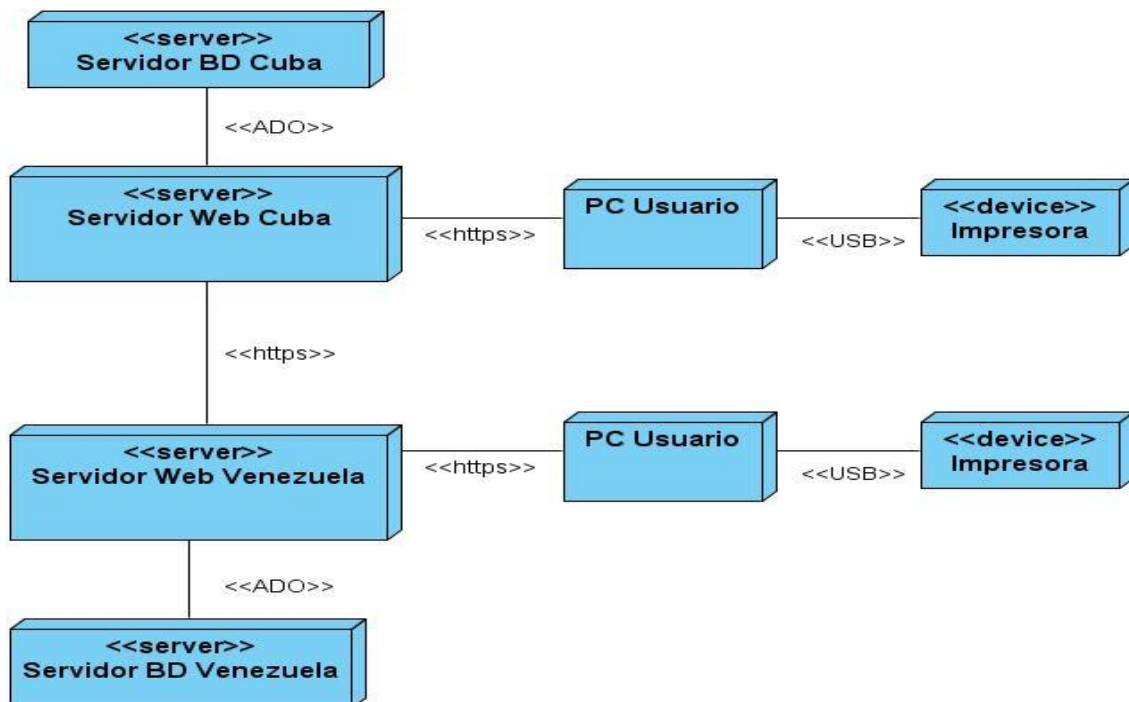


Figura 14 Diagrama de despliegue.

Diagrama de componente.

Un subsistema de implementación es una colección de componentes y otros subsistemas de implementación usados para estructurar el modelo de implementación y dividirlos en pequeñas partes que pueden ser integradas y probadas de forma independiente. Los subsistemas de implementación incluyen dependencias y otras informaciones. Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando:

- ❖ Los subsistemas de implementación, sus dependencias a la hora de importar código y organizar los subsistemas de implementación en capas.

Estos se utilizan para mostrar las dependencias de compilación de los ficheros de código, relaciones de derivación entre ficheros de código fuente y ficheros.

PROPUESTA DE SOLUCION AL PROBLEMA.

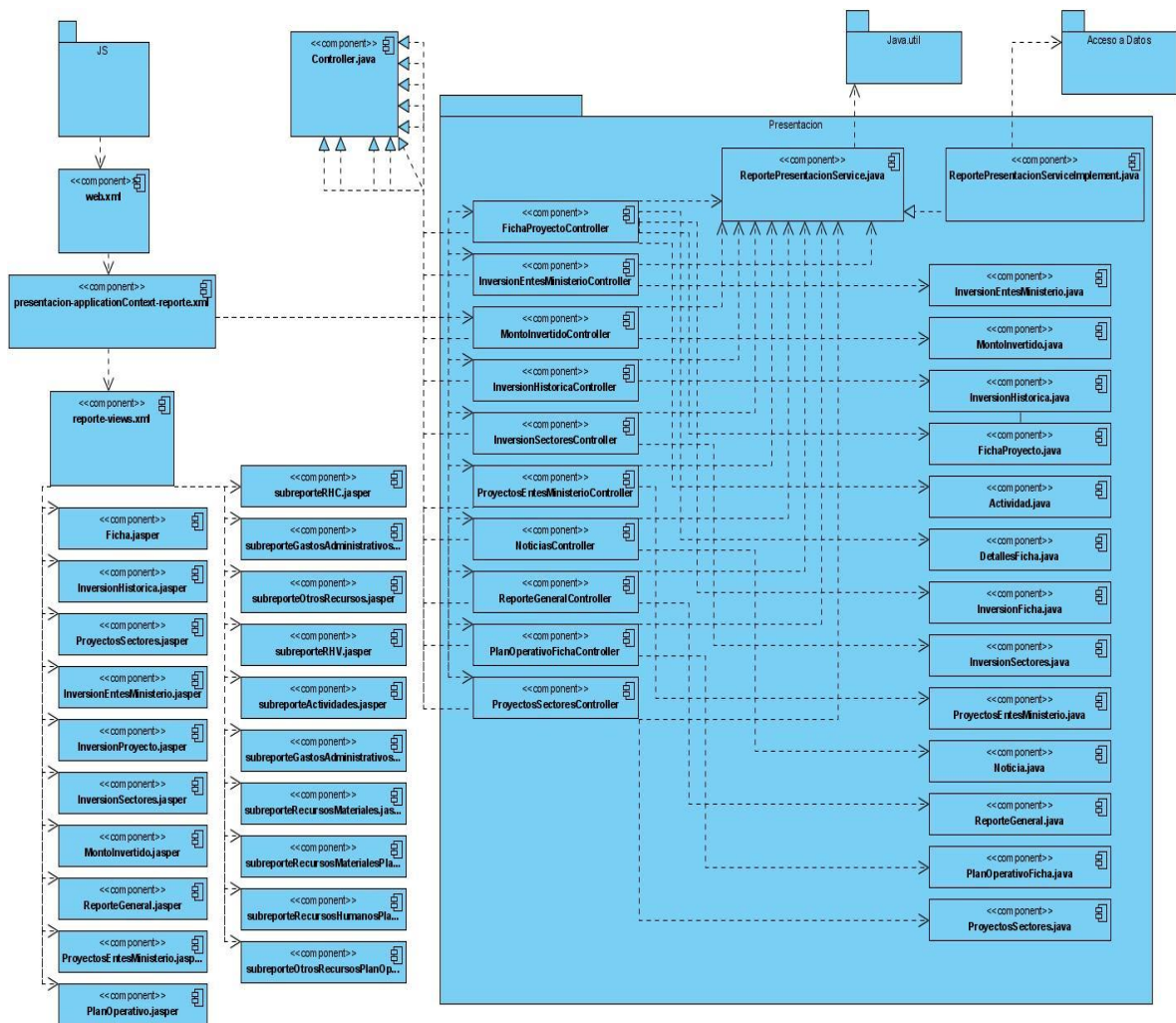


Figura 15 Diagrama de componentes.

2.5- ESTANDARES DE CODIFICACION.

Los estándares de codificación son reglas específicas a una lengua que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores, lo que evita la ocurrencia de errores. Durante la implementación estos estándares ayudan a la utilización o reutilización de códigos de los compañeros de trabajo del equipo de desarrollo, además otra de las ventajas que tiene es que ayuda a entender el código y a obtener un producto final organizado y con una muy buena calidad. Por lo visto anteriormente se utilizaron estándares de codificación en la solución, esto trae consigo la facilidad de que cuando se le quiera realizar mantenimiento al producto el código sea entendido con claridad.

PROPUESTA DE SOLUCION AL PROBLEMA.

Las reglas que fueron utilizadas para la implementación del software, fueron:

Paquetes:

Regla: Los nombres de los paquetes siempre comenzarán con minúscula.

Justificación: Los nombres los paquetes deben de ir en minúscula para que esto no tienda a confundir a los desarrolladores. Este es un formato clásico en cuanto al desarrollo de software.

Clases:

Regla: Los nombres de clases comienzan con mayúsculas y las palabras que la forman en minúsculas en caso de que sea un nombre compuesto la inicial siempre comenzarán con mayúscula.

Justificación: Los nombres de clase abundan en el código por lo que es práctico utilizar nombres intuitivos para ellas. La marca distintiva en este caso es la mayúscula inicial y las intermedias si es un nombre compuesto. Este formato es clásico y compatible con estándares de notación para lenguajes de programación.

Variables de las clases:

Regla: Las variables miembros de clase se escriben con minúsculas, las variables con nombres compuestos, comienzan con la primer palabra enteramente en minúscula y el resto comenzando con mayúscula.

Justificación: Las variables miembros de las clases suelen ser las más importantes y más usadas en casi todos los programas, por lo tanto es importante que su lectura sea fácil y se entienda.

Métodos de las clases:

Regla: Los métodos miembros de clases siguen la notación idéntica que para las variables de las clases, es decir, comienzan con minúscula y separando las palabras con mayúsculas.

Justificación: Estos nombres son muy utilizados y es clave una lectura rápida y clara de su contenido.

Comentarios:

PROPUESTA DE SOLUCION AL PROBLEMA.

Regla: Se usan los comentarios en línea para facilitar la comprensión del código, sobre todo en procedimientos complejos. Se comentarán los métodos al principio para facilitar la comprensión y entendimiento del código.

2.6- ESTRUCTURACION DE LOS PAQUETES PARA LA IMPLEMENTACION.

Como se observó en el capítulo 1 se va a utilizar una arquitectura basada en capas, esto se debe a la utilización del patrón de diseño MVC, para desarrollar éste producto se utilizarán 3 capas, la capa de presentación, la capa de negocio y la capa de acceso a datos. El módulo quedó estructurado de la siguiente forma:

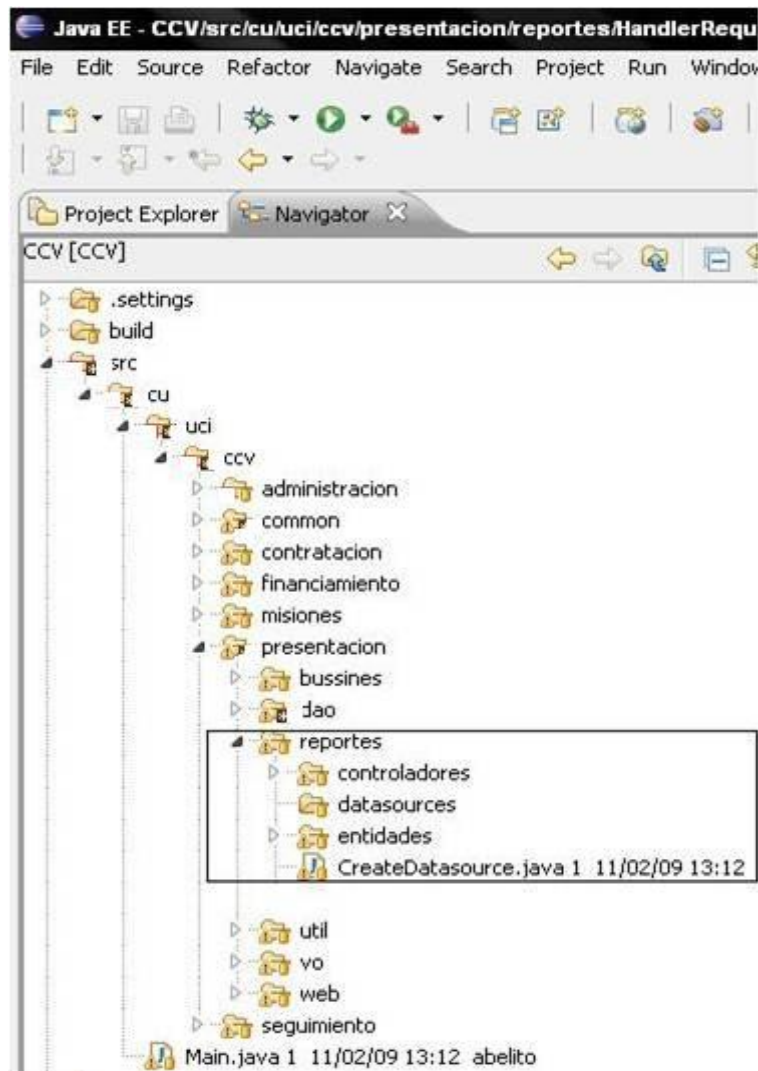


Figura 16 Estructura del aplicación.

PROPUESTA DE SOLUCION AL PROBLEMA.

En las carpetas que están señaladas es donde estarán ubicadas todas las clases necesarias para el desarrollo de los reportes. Dentro de la carpeta reportes existen tres carpetas, estas son: la carpeta controladores, la carpeta datasources, la carpeta entidades y la clase CreateDatasource.java. Como lo dice el nombre en la carpeta entidades estarán guardadas todas aquellas clases entidades que se necesiten, dentro de la carpeta controladores estarán todas las clases controladoras que tienen los reportes y en la carpeta datasources se guardarán las clases datasource que se crearon o se modificaron para poder obtener la información necesaria para generar el reporte. La clase CreateDatasource.java es la encargada de crear el datasource que es llamado en la clase controladora para la obtención de datos.

Se observará ahora algunas figuras de cómo quedó finalmente organizado lo planteado anteriormente.



Figura 17 Estructura de la carpeta de las clases controladoras.

PROPUESTA DE SOLUCION AL PROBLEMA.

La carpeta entidades se guardarán todas las clases entidades que se implementaron para la automatización de los reportes del módulo de Presentación del proyecto CICCv y quedó estructurada de la siguiente forma:



Figura 18 Estructura de la carpeta entidades.

Éstas no serán las únicas carpetas que tendrán guardado todo lo relacionado con los reportes pues dentro de la aplicación se guardan las plantillas de los reportes (.jasper), las cuales son las encargadas de darle el formato al reporte, sin ésta es imposible mostrarle el reporte al cliente, pues también les brinda al cliente varias ventajas.

PROPUESTA DE SOLUCION AL PROBLEMA.

En el contenido de la carpeta WEB-INF se puede encontrar las carpetas **conf**, **lib** y **reportes**, además del XML llamado web.xml. Dentro de la carpeta reportes se encuentran las carpetas **conf** y **presentacion**. En la carpeta **conf** se encuentra el xml reporte-views el cual es configurado para generar los reportes. Dentro de la carpeta **presentacion** se encuentran las plantillas de todos los reportes.

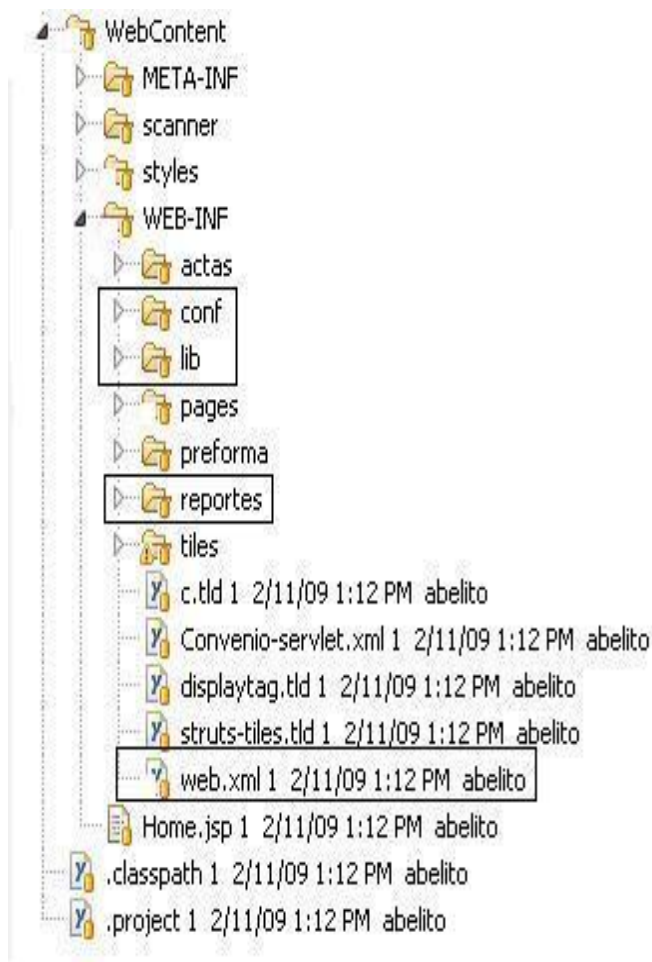


Figura 19 Estructura del paquete WebContent.

En la carpeta **lib** estarán guardadas todas las librerías tanto de Ireport como de JasperReport necesarias para el buen funcionamiento y desarrollo de los reportes, estas librerías son las encargadas de que se puedan observar las gráficas de las plantillas, de que el datasource pueda obtener los datos e insertarlo dentro de la plantilla y ésta se le muestre al usuario.

PROPUESTA DE SOLUCION AL PROBLEMA.

En la carpeta **presentacion** que se encuentra dentro de la carpeta **reportes** se guardarán todos los ficheros .jasper, es decir, las plantillas de los reportes que se van a generar en la aplicación. Las cuáles son las encargadas de mostrarle al cliente el reporte con el prototipo que se estableció cuando se hizo el levantamiento de los requisitos para el desarrollo de éste proyecto. También se observa dentro de la carpeta **reportes** la carpeta **conf** que es la tiene el xml reporte-views.

Se mostrará en la siguiente figura como quedaron estructuradas estas carpetas una vez realizada la implementación completa del módulo de Presentación del proyecto CICCIV.

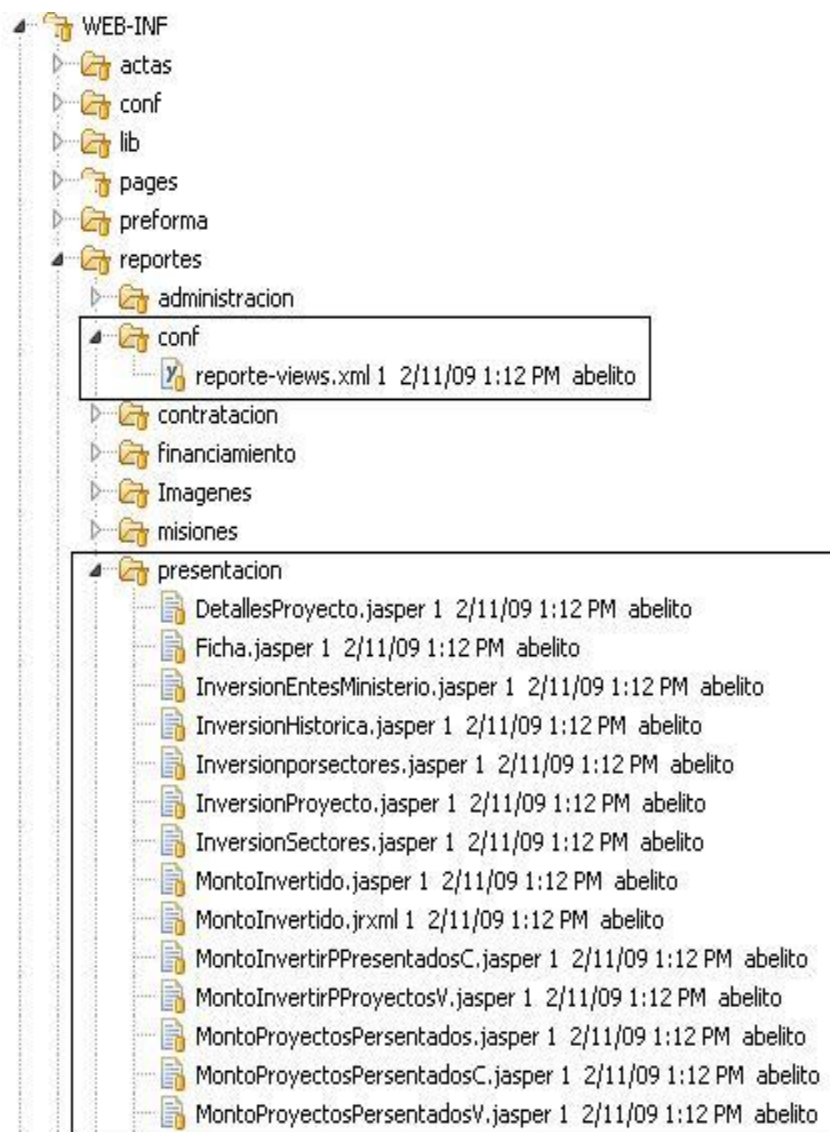


Figura 20 Estructura de la carpeta reportes y dentro de estas la de conf y presentacion.

PROPUESTA DE SOLUCION AL PROBLEMA.

Para mostrar los reportes en la aplicación es imprescindible realizar la configuración de los xml. No solamente habría que configurar los XML que pertenecen a los reportes pues también se tienen que configurar los XML de la aplicación para tener el permiso y la visibilidad en la aplicación. Se podrá ver ahora donde se guardarán estos ficheros.

La carpeta **conf** será la encargada de guardar todos los XML de configuración de todo el módulo de Presentación, dentro de esta carpeta se encuentran los siguientes XML:

- presentacion-applicationContext-bussines.xml
- presentacion-applicationContext-dao.xml
- presentacion-applicationContext-remoteClient.xml
- presentacion-applicationContext-remoteServer.xml
- presentacion-applicationContext-reporte.xml
- presentacion-applicationContext-web.xml

Los cuales se verán en la siguiente figura:

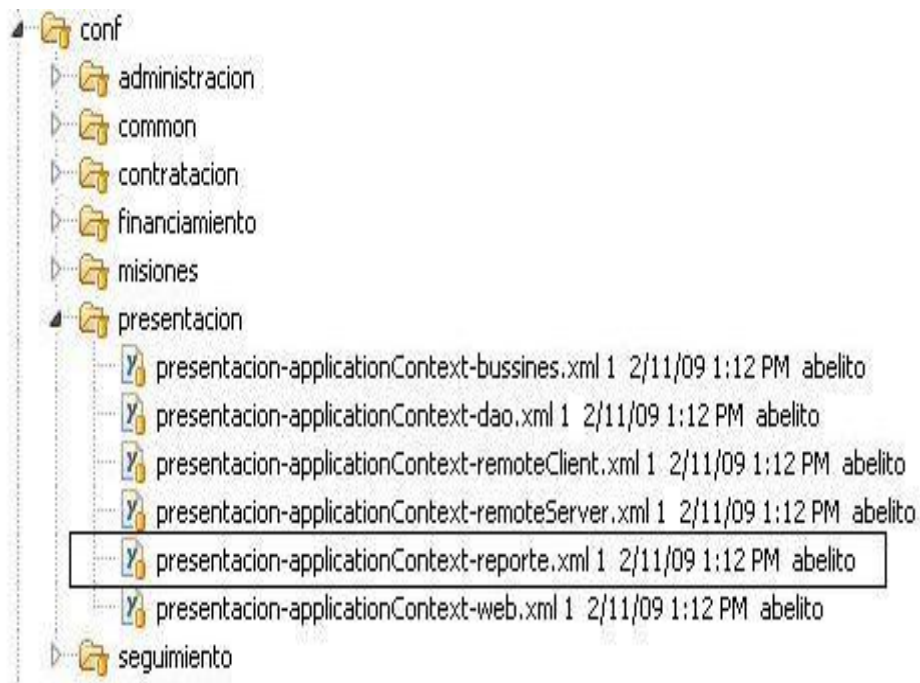


Figura 21 Estructura de la carpeta presentacion dentro de la carpeta conf.

Estos xml son configurados para el buen funcionamiento de cada una de las capas. De estos xml que se observan se hará un enfoque en el que esta señalado, ya que éste tiene que ver con la configuración que se tiene que realizar para el desarrollo y buen funcionamiento de los reportes.

2.7- IMPLEMENTACION Y CONFIGURACION.

En éste epígrafe se estará hablando acerca de como quedó la implementación de todas las clases y la configuración de todos los xml que se utilizaron para la implementación de todos los reportes del módulo de Presentación del proyecto CICCIV. Se abordará también acerca de cómo se realizaron estas dos importantes actividades.

Implementación de las clases controladoras y las clases entidades.

Las clases controladoras son las encargadas de obtener toda la información que se necesita, es aquí donde se realiza el llamado a métodos de clases de la capa de negocio, las cuales son las encargadas de obtener, modificar y organizar toda la información necesaria obtenida desde la capa de acceso a datos.

Dichas clases son las que trabajan con las clases entidades, como se está trabajando con la programación orientada a objeto (POO), estas clases entidades permiten trabajar con objetos desde las clases controladoras, otra ventaja es que se puede trabajar desde la misma controladora con varias clases entidades.

Para lograr la implementación se realizó el artefacto diagrama de clase, este artefacto no pertenece al flujo de trabajo de implementación pues éste se realiza en el flujo de trabajo de Análisis y Diseño, pero es imprescindible en el flujo de implementación para poder realizar la implementación de las clases con las cuales va a contar para la solución al problema científico planteado en éste trabajo. Éstos diagramas fueron realizados partiendo de los requisitos (Montané Izaguirre, et al., 2008) y del Modelo de Casos de Uso del sistema del proyecto (Montané Izaguirre, y otros, 2008).

Aquí se mostrará el diagrama de clases del CU FichaProyecto, el resto de los diagramas de los CU se pueden encontrar en el documento: Diagramas de clases de los Casos de Usos de los reportes de presentación. (Sánchez Villa, 2009)

PROPUESTA DE SOLUCION AL PROBLEMA.

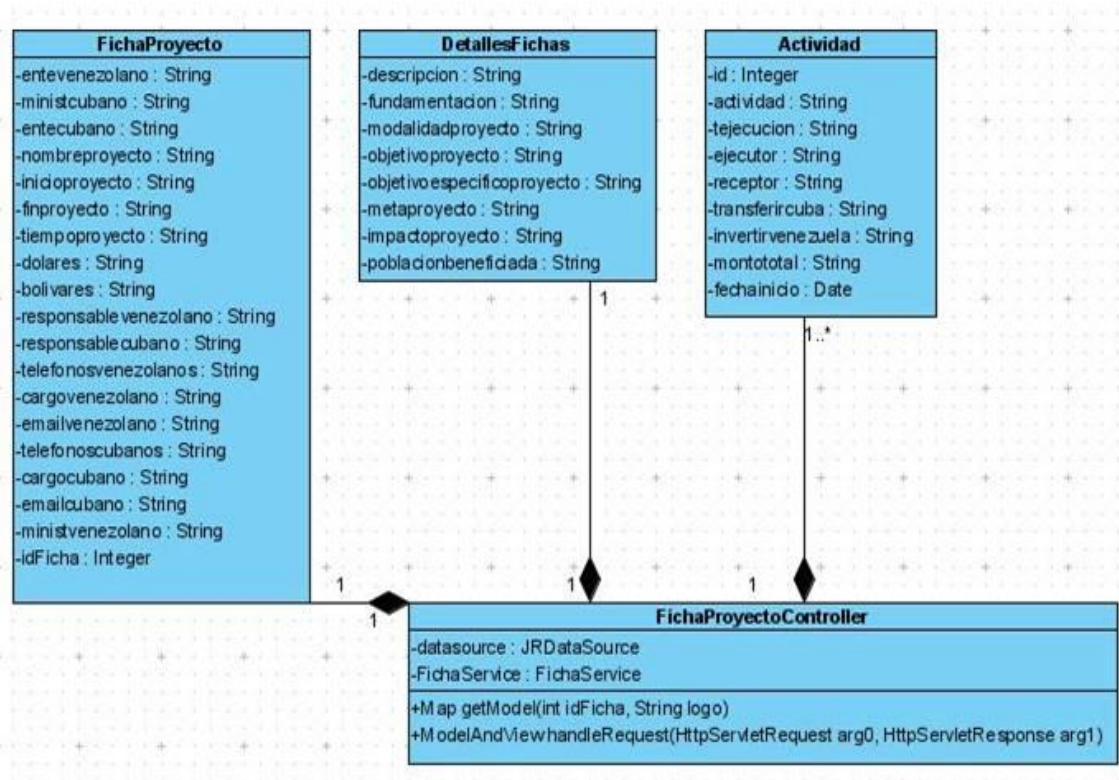


Figura 22 Diagrama de clase del CU Ficha Proyecto.

En las controladoras se van a utilizar dos métodos los cuales son:

1- ModelAndView handleRequest: el cual es el encargado de obtener de la capa de presentación los parámetros que se necesitan pasar a los métodos del negocio y los de acceso a datos para la obtención de los datos que se desean mostrar. Aquí es donde una vez que se obtienen los datos, estos se les asignan al o los objetos que se están utilizando, los cuales son los que se mostrarán finalmente en el reporte.

2- Map getModel: este es el encargado de crear un nuevo datasource y pasarle el o los objetos con toda la información que se necesita y los parámetros, estos últimos se realizan si el reporte posee parámetros.

En la siguiente figura se puede observar la clase controladora del CU InversionHistorica ya implementada, para lograr el desarrollo de los reportes se implementaron 11 clases controladoras, la implementación de estas se puede observar en la aplicación desarrollada y las clases se pueden observar en el diagrama de componentes.

PROPUESTA DE SOLUCION AL PROBLEMA.

```
package cu.uci.ccv.presentacion.reportes.controladores;

import java.math.BigDecimal;

public class InversionHistoricaController implements Controller {

    private JRDataSource datasource;
    private ReporteService reporteService;
    public InversionHistoricaController() {}
    public ReporteService getReporteService() {}
    public void setReporteService(ReporteService reporteService) {}
    public JRDataSource getDatasource() {}
    public void setDatasource(JRDataSource datasource) {}

    public ModelAndView handleRequest(HttpServletRequest request,

    public Map getModel(Collection<Object> lista,String logo,String totalProyectos,
    String totalInvertir,String totalTransferir,String totalMonto,int annoInicio,int annoFin)
    {
        Map model = new HashMap();
        datasource = CreateDatasource.createDatasource(lista);
        model.put("jrdatasource", datasource);
        model.put("tCantProyectos", totalProyectos);
        model.put("tMontoInvVenezuela", totalInvertir);
        model.put("tMontoTransfCuba", totalTransferir);
        model.put("tMontoTotal", totalMonto);
        model.put("annoInicio", annoInicio);
        model.put("annoFin", annoFin);
        model.put("DIR_LOG", logo);
        model.put("fecha", DateFormater.getDateInSpanish());
        return model;
    }
}
```

Figura 23 Implementación de la clase controladora del CU InversionHistorica.

El código del método ModelAndView handleRequest y la implementación de la clase InversionHistoricaController se puede encontrar en el (Anexo 3).

En la siguiente figura se podrá observar la implementación de la clase entidad perteneciente al CU InversionHistorica, se implementaron 14 clases entidades, la implementación de las restantes se pueden encontrar en la aplicación desarrollada.

El objetivo fundamental de estas clases es poder trabajar desde las clases controladoras con objetos de este tipo, pues desde la capa de negocio obtenemos los datos para el reporte, varias veces se obtienen más datos de los necesarios, es por eso que mediante estas clases organizamos los datos de la forma en que tenemos diseñadas las plantillas y se les pasa al DataSource para que éste sea el responsable de insertar los datos en dichas plantillas.

```
package cu.uci.ccv.presentacion.reportes.entidades;

import java.math.BigDecimal;

public class InversionHistorica {

    private int anno;
    private String cantProyectos;
    private BigDecimal montoGTotal;
    private String montoTransfCuba;
    private String montoInvVenezuela;
    private String montoTotal;

    + public int getAnno() {
    + public void setAnno(int anno) {
    + public String getCantProyectos() {
    + public void setCantProyectos(String cantProyectos) {
    + public BigDecimal getMontoGTotal() {
    + public void setMontoGTotal(BigDecimal montoGTotal) {
    + public String getMontoTransfCuba() {
    + public void setMontoTransfCuba(String montoTransfCuba) {
    + public String getMontoInvVenezuela() {
    + public void setMontoInvVenezuela(String montoInvVenezuela) {
    + public String getMontoTotal() {
    + public void setMontoTotal(String montoTotal) {
    + public InversionHistorica(int anno, String cantProyectos,
    + public InversionHistorica() {

}
```

Figura 24 Implementación de la clase entidad del CU InversionHistorica.

Configuración de los XML para el desarrollo de los reportes.

En el xml **reporte-views** es donde se implementan los beans, en los cuales se define el formato en que se mostrará el reporte, además también es el encargado de mostrarle la dirección física, es decir la "url" donde se encuentra el .jasper que no es más que la plantilla que se diseño para el reporte.

En cada beans se especifica el tipo de valor que se utilizará, en este caso para la implementación de los reportes del producto final, se utilizará el jrdatasource, éste es el encargado de manejar e insertar los datos en el o los reportes con los que está trabajando, además se define el identificador (id) del beans, esto es para poder llamarlo desde otro beans u otro xml.

PROPUESTA DE SOLUCION AL PROBLEMA.

Se tiene que implementar un beans por cada clase controladora ya que cada una de estas clases corresponde con un reporte, cada reporte tiene que configurarse en dicho xml ya que se tiene que tener tanto la url del .jasper, como el tipo de datos y la vista en la cual se mostrará el reporte, es decir de la forma en que será exportado dicho reporte. Los reportes generados con JasperReport pueden ser exportados a una multitud de formatos como PDF, XLS, RTF, HTML, XML, CVS (valores separados por coma) y texto plano. Para la realización de este proyecto se decidió exportar todos los reportes a PDF, debido a que este formato fue el que eligió el cliente.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/s
<!-- Configuraciones del modulo presentacion -->
  <bean id="inversionhistorica" class="org.springframework.web.servlet.view.jasperreports.JasperReportsPdfView">
    <property name="url">
      <value>/WEB-INF/reportes/presentacion/InversionHistorica.jasper</value>
    </property>
    <property name="reportDataKey">
      <value>jrdatasource</value>
    </property>
  </bean>
  <bean id="fichaproyecto" class="org.springframework.web.servlet.view.jasperreports.JasperReportsPdfView">
    <property name="url">
      <value>/WEB-INF/reportes/presentacion/Ficha.jasper</value>
    </property>
    <property name="reportDataKey">
      <value>jrdatasource</value>
    </property>
    <property name="subReportUrls">
      <map>
        <entry key="RHC_DIR" value="/WEB-INF/reportes/presentacion/subreporteRHC.jasper"></entry>
        <entry key="RHV_DIR" value="/WEB-INF/reportes/presentacion/subreporteRHV.jasper"></entry>
        <entry key="ACT_DIR" value="/WEB-INF/reportes/presentacion/subreporteActividades.jasper"></entry>
        <entry key="AMI_DIR" value="/WEB-INF/reportes/presentacion/subreporteAMI.jasper"></entry>
        <entry key="DETALLE_DIR" value="/WEB-INF/reportes/presentacion/DetallesProyecto.jasper"></entry>
        <entry key="INVERSION_DIR" value="/WEB-INF/reportes/presentacion/InversionProyecto.jasper"></entry>
      </map>
    </property>
  </bean>
```

Figura 4 Implementación del xml para CU InversionHistorica y el CU FichaProyecto.

PROPUESTA DE SOLUCION AL PROBLEMA.

Enfocando el tema ya en el xml **presentacion-applicationContext-reporte** se podrá observar que en éste lo que se tiene que implementar es el mapeo, donde se relacionará el xml **reporte-views** con la clase controladora a la cual pertenece el reporte que se está implementando. Se especifica el identificador (id) que se definió en el xml **reporte-views** y a la clase controladora a la que pertenece, lo otro que se tiene que hacer es darle la "url" a esta clase controladora además de poner el nombre y la referencia de la o las clases que se utilizaron de la capa de negocio.

A continuación se mostrará cómo queda implementado éste xml una vez realizada la implementación de todos los reportes. Se observará el mapeo de las clases controladoras y la implementación de los beans del caso del CU InversionHistorica y el CU FichaProyecto, los restantes se pueden observar en la aplicación.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xmlns:flow="http://www.springframework.org/schema/webflow-config"
  xmlns:jee="http://www.springframework.org/schema/jee"
  xmlns:lang="http://www.springframework.org/schema/lang"
  xmlns:osgi="http://www.springframework.org/schema/osgi"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:util="http://www.springframework.org/schema/util"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/s
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-2.1.xsd
http://www.springframework.org/schema/webflow-config http://www.springframework.org/schema/webflow-config/spr
http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-2.0.xsd
http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang-2.0.xsd
http://www.springframework.org/schema/osgi http://www.springframework.org/schema/osgi/spring-osgi.xsd
http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.1.xsd
http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-2.0.xsd">
<bean id="publicUrlMappingPresentacion" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMappin
  <property name="mappings">
    <props>
      <prop key="fichaproyecto">FichaProyectoController</prop>
      <prop key="inversionhistorica">InversionHistoricaController</prop>
    </props>
  </property>
</bean>
<bean id="FichaProyectoController" class="cu.uci.ccv.presentation.reportes.controladores.FichaProyectoControl
  <property name="fichaService" ref="FichaService"></property>
</bean>
<bean id="InversionHistoricaController" class="cu.uci.ccv.presentation.reportes.controladores.InversionHistor
  <property name="reporteService" ref="ReporteService"></property>
</bean>
```

Figura 25 Configuración del xml presentacion-applicationContext-reporte.

PROPUESTA DE SOLUCION AL PROBLEMA.

Solamente queda por ver la configuración del **web.xml**, éste se tiene que configurar para que finalmente se puedan observar los reportes. En éste es donde se le da permiso a los xml de todos los módulos del proyecto y en el caso de los reportes se le dará el permiso de visibilidad el cual es el que permite que se puedan observar estos reportes desde la aplicación.

A continuación se muestra como queda configurado finalmente éste xml una vez que se realizada la implementación del proyecto.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>Convenio</display-name>

  <servlet>
    <servlet-name>Convenio</servlet-name>
    <servlet-class>
      org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Convenio</servlet-name>
    <url-pattern>/reportes/*</url-pattern>
  </servlet-mapping>

  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
      /WEB-INF/conf/presentacion/presentacion-applicationContext-dao.xml
      /WEB-INF/conf/presentacion/presentacion-applicationContext-bussines.xml
      /WEB-INF/conf/presentacion/presentacion-applicationContext-reporte.xml
      /WEB-INF/conf/presentacion/presentacion-applicationContext-remoteServer.xml

      /WEB-INF/conf/common/common-applicationContext-properties.xml
      /WEB-INF/conf/common/common-applicationContext-dataSource.xml
      /WEB-INF/conf/common/common-applicationContext-dao.xml
      /WEB-INF/conf/common/common-applicationContext-security.xml
      /WEB-INF/conf/common/common-applicationContext-bussines.xml
    </param-value>
  </context-param>
</web-app>
```

Figura 26 Configuración del xml web.

2.8 - CONCLUSIONES.

En este capítulo obtuvimos una representación del flujo de implementación a través del diagrama de despliegue del sistema que muestra la situación física de la aplicación. También se obtuvo el diagrama de componentes que representan cada parte modular del sistema y las relaciones entre ellas, así como los diagramas de clases necesarios para lograr una buena implementación.

Se definieron y pusieron en práctica estándares de codificación en la implementación de la solución propuesta, lo que ayudó a tener un código más uniforme, claro y fácil de mantener. El diseño gráfico establecido para la aplicación cumple con las expectativas del cliente y con el objetivo de desarrollar un entorno amigable para la extracción de los reportes.

Con la realización de los artefactos, la configuración de los xml y la implementación de las clases se cumplieron los objetivos trazados en el capítulo.

CAPITULO 3: VALIDACION DE LA SOLUCION PROPUESTA.

3.1- INTRODUCCION.

En el presente capítulo se comienza la validación de la solución propuesta, a través de diferentes test de unidades, dentro de los cuales se encuentran las pruebas de caja blanca y pruebas de caja negra. Se abordará las pruebas de caja negra que fueron las que se le aplicaron al módulo de Presentación del proyecto, de las que se va a definir el objetivo, así como su alcance y los detalles. El principal beneficio de realizar esta prueba es que permite aislar segmentos del programa que pueden ser probados para verificar que funcionan correctamente.

3.2- PRUEBAS DE CAJA NEGRA.

Éste tipo de prueba es muy usada por las personas encargadas de la calidad del software ya que tiene como prioridad verificar el completo y buen funcionamiento del software. A continuación se verán algunas de las principales características de este tipo de prueba así como la realización de la misma.

Objetivo: El objetivo de realizar este tipo de prueba al sistema es para detectar el incorrecto o incompleto funcionamiento de este, así como los errores de interfaces, rendimiento y errores de inicialización y terminación.

Alcance: El proceso de pruebas de caja negra se va a centrar principalmente en los requisitos funcionales del software para verificar el comportamiento de la unidad observable externamente y la calidad funcional.

Descripción: Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales del programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

La prueba de Caja Negra no es una alternativa a las técnicas de pruebas de Caja Blanca, es más bien un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca. Dentro de las técnicas de Prueba de Caja Negra, se utilizó la prueba de unidad.

Este tipo de pruebas fue la que se le aplicó tanto a los reportes como al Módulo de Presentación del proyecto Convenio Integral de Cooperación Cuba – Venezuela, estas

VALIDACION DE LA SOLUCION PROPUESTA.

pruebas fueron realizadas por un grupo de calidad interno del proyecto, dicho grupo estaba integrado por:

Nombre y Apellidos	Rol desempeñado
Yeilin Martínez Torres	Jefa del equipo de calidad del proyecto CICC
Maylin Villanueva Santana	Miembro del equipo de calidad del proyecto CICC
Odaimy Caballero Padrón	Miembro del equipo de calidad del proyecto CICC
Yudelkys Carrasco Ramírez	Miembro del equipo de calidad del proyecto CICC

Para realizar estas pruebas fue necesario diseñar los casos de pruebas para los reportes y para el Módulo de Presentación. A continuación se mostrará el diseño de pruebas para los casos de uso más significativos (en este caso solamente se verán los casos de usos Generar Reporte de Proyectos y Generar Reporte de Fichas de Proyectos) por la extensión que presentan estos.

-Validaciones de los campos de entrada.

Antes de que se realice el diseño de pruebas es sumamente importante contar con las validaciones de los campos de entrada de cada caso de uso, para lograr esto fue necesario realizar un documento donde viene especificado las validaciones de los campos de entrada. A continuación observarán las validaciones de los casos de uso Generar Reporte de Proyectos y Generar Reporte de Fichas de Proyectos.

VALIDACION DE LA SOLUCION PROPUESTA.

Reportes → Submenú Proyectos.

The screenshot shows a web application interface for project management. At the top, there is a header with the logo and name 'Convenio Integral de Cooperación Cuba-Venezuela' and a date/time stamp: 'Bienvenido, hoy es sábado, 05 de julio de 2008, 6:44:52 AM'. Below the header, there are navigation tabs: 'Presentación', 'Seguimiento', 'Misiones', and 'Financiamiento'. The main content area is titled 'Reportes' and contains a sub-menu 'Proyectos'. The 'Busqueda de proyectos' form is the central focus, featuring several search criteria: 'País', 'Ministerio', 'Ente ejecutor', 'Nombre del proyecto', 'Marco de aprobación', 'Estado del Proyecto', 'Fuente de financiamiento', 'Monto(USD) entre:', 'Modalidad', 'Nivel de agrupación', and 'Duración en meses'. Each criterion has a corresponding dropdown menu or input field. At the bottom right of the form are 'Imprimir' and 'Generar' buttons. The footer of the page contains the copyright notice: 'COPYRIGHT 2008, Convenio Integral de Cooperación Cuba Venezuela. Todos los derechos reservados.'

Figura 27 Vista del filtro de búsqueda del reporte de proyectos.

Los campos que este posee son los siguientes:

- **Nombre de proyecto:** Se puede introducir letras, números y guiones.
- **Tiempo de duración:** Se puede seleccionar un número entero.
- **Monto entre:** Se puede introducir formato de dinero, con dos lugares después de la coma, ejemplo: xxxxxxxx.xx.
- **Nivel de Agrupación:** Obligatorio, se debe seleccionar.
- Los demás se seleccionan según lo listado.

VALIDACION DE LA SOLUCION PROPUESTA.

Reportes → Submenú Fichas de Proyectos.

Convenio Integral de Cooperación Cuba-Venezuela

Bienvenido, hoy es sábado, 05 de julio de 2008, 9:01:39 AM

Presentación | Seguimiento | Misiones | Financiamiento

Usuario: preec

Inicio | Presentación | Reportes | Ficha de Proyectos

Procesos

Documentos

Reportes

Proyectos

Ficha de Proyectos

Busqueda de proyectos

País: Todos

Ministerio: Todos

Ente Ejecutor: Todos

Nombre del proyecto:

Estado del proyecto: Todos

Marco de aprobación: Todos

Fuente de Financiamiento: Todos

Monto(USD) entre: y

Modalidad: Todos

Duración en meses: -mes-

Buscar

COPYRIGHT 2008, Convenio Integral de Cooperación Cuba Venezuela
Todos los derechos reservados

Figura 28 Vista del filtro de búsqueda del reporte fichas de proyectos.

Los campos que este posee son los siguientes:

- **Nombre de proyecto:** Se puede introducir letras, números y guiones.
- **Tiempo de duración:** Se puede seleccionar un número entero.
- **Monto entre:** Se puede introducir formato de dinero, con dos lugares después de la coma, ejemplo: xxxxxxxx.xx.
- **Nivel de Agrupación:** Obligatorio, se debe seleccionar.
- Los demás se seleccionan según lo listado.

VALIDACION DE LA SOLUCION PROPUESTA.

Las validaciones de los campos de entrada de cada caso de uso de los reportes se pueden encontrar en el documento: (Sánchez Villa, 2008)

-Diseño de casos de prueba.

Partiendo de las validaciones de los campos de entrada de cada CU se puede pasar al diseño de casos de prueba, ahora seguidamente observarán el diseño de casos de prueba que se le realizó al caso de uso Generar Reporte de Proyectos y Generar Reporte de Fichas de Proyectos.

Caso de Uso: Generar Reporte de Proyectos

Descripción General

El caso de uso consiste en hacer un reporte de los proyectos a los que el usuario tenga acceso, según el nivel de acceso que tenga.

Condiciones de Ejecución:

El sistema debe estar instalado y ejecutado correctamente.

El actor debe estar autenticado con los permisos necesarios.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC1: Reporte de fichas de proyectos	EC1: Reporte de fichas de proyectos	El sistema muestra el menú para el módulo según el rol que tenga el usuario.	
		Se muestran los campos que son: Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses.	

VALIDACION DE LA SOLUCION PROPUESTA.

		Se muestra un botón generar	
	EC2: Si el usuario es un Min accede al submenú Reportes-> Fichas de proyectos	Se muestran los campos que son: País, Ministerio, Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar	
	EC3: Si el usuario es una ST accede al submenú Reportes-> Fichas de proyectos	Se muestra la pantalla con los datos anteriores de la ficha del proyecto o la ficha del proyecto anterior.	
	EC4: presiona "Atrás"	Se ejecuta la acción de impresión de la ficha de proyecto, dando el formato requerido para una buena impresión.	
	EC5:Se presiona "Imprimir"		

VALIDACION DE LA SOLUCION PROPUESTA.

SC 1: Reporte de fichas de proyectos

Id del escenario	Escenario	Variable 1	Variable 2	Variable N	Respuesta del Sistema	Resultado de la Prueba
EC 1	Reporte de fichas de proyectos				El sistema muestra el menú para el módulo según el rol	El sistema muestra el menú para el módulo según el rol que tenga el usuario.
EC 2	Si el usuario es un Min accede al submenú Reportes- > Fichas de proyectos				Se muestran los campos que son: Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar	Se muestran los campos que son: Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar

VALIDACION DE LA SOLUCION PROPUESTA.

EC 3	Si el usuario es una ST accede al submenú Reportes- > Fichas de proyectos				Se muestran los campos que son: País, Ministerio, Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar	Se muestran los campos que son: País, Ministerio, Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar
EC 4	Presiona "Atrás"				Se muestra la pantalla con los datos anteriores de la ficha del proyecto o la ficha del proyecto anterior.	Se muestra la pantalla con los datos anteriores de la ficha del proyecto o la ficha del proyecto anterior.
EC 5	Se presiona "Imprimir"				Se ejecuta la acción de impresión de la ficha de proyecto, dando el formato requerido para una buena	Se ejecuta la acción de impresión de la ficha de proyecto, dando el formato requerido para una buena impresión.

VALIDACION DE LA SOLUCION PROPUESTA.

					impresión.	
--	--	--	--	--	------------	--

Caso de Uso: Generar Reporte de Fichas de Proyectos

Descripción General

El caso de uso consiste en hacer un reporte de las fichas de los proyectos a los que el usuario tenga acceso, según su nivel.

Condiciones de Ejecución:

El sistema debe estar instalado y ejecutado correctamente.

El actor debe estar autenticado con los permisos necesarios.

Secciones a probar en el Caso de Uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC1: Reporte de fichas de proyectos	EC1: Reporte de fichas de proyectos	El sistema muestra el menú para el módulo según el rol que tenga el usuario.	
	EC2: Si el usuario es un Min accede al submenú Reportes-> Fichas de proyectos	Se muestran los campos que son: Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar	

VALIDACION DE LA SOLUCION PROPUESTA.

	<p>EC3: Si el usuario es una ST accede al submenú Reportes-> Fichas de proyectos</p>	<p>Se muestran los campos que son:</p> <p>País, Ministerio, Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses.</p> <p>Se muestra un botón generar</p>	
	<p>EC4: presiona "Atrás"</p>	<p>Se muestra la pantalla con los datos anteriores de la ficha del proyecto o la ficha del proyecto anterior.</p>	
	<p>EC5: Se presiona "Imprimir"</p>	<p>Se ejecuta la acción de impresión de la ficha de proyecto, dando el formato requerido para una buena impresión.</p>	

SC 1: Reporte de fichas de proyectos

Id del escenario	Escenario	Variable 1	Variable 2	Variable N	Respuesta del Sistema	Resultado de la Prueba

VALIDACION DE LA SOLUCION PROPUESTA.

EC 1	Reporte de fichas de proyectos				El sistema muestra el menú para el módulo según el rol que tenga el usuario.	El sistema muestra el menú para el módulo según el rol que tenga el usuario.
EC 2	Si el usuario es un Min accede al submenú Reportes- > Fichas de proyectos				Se muestran los campos que son: Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar	Se muestran los campos que son: Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar
EC 3	Si el usuario es una ST accede al submenú				Se muestran los campos que son: País,	Se muestran los campos que son: País,

VALIDACION DE LA SOLUCION PROPUESTA.

	Reportes- > Fichas de de proyectos				Ministerio, Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar	Ministerio, Ente Ejecutor, Nombre del proyecto, Estado del proyecto, Marco de aprobación, Modalidad, Fuente de financiamiento, Nivel de agrupación, Monto, Duración en meses. Se muestra un botón generar
EC 4	presiona "Atrás"				Se muestra la pantalla con los datos anteriores de la ficha del proyecto o la ficha del proyecto anterior.	Se muestra la pantalla con los datos anteriores de la ficha del proyecto o la ficha del proyecto anterior.

VALIDACION DE LA SOLUCION PROPUESTA.

EC 5	Se presiona "Imprimir"				Se ejecuta la acción de impresión de la ficha de proyecto, dando el formato requerido para una buena impresión.	Se ejecuta la acción de impresión de la ficha de proyecto, dando el formato requerido para una buena impresión.
------	------------------------	--	--	--	-----------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------

El diseño de pruebas que se realizó, tanto para los reportes como para el Módulo completo de Presentación se puede encontrar en el documento: (Carrasco Ramírez, 2008)

-No Conformidades Detectadas.

Una vez que se obtiene el diseño de pruebas y se realizan las pruebas correspondientes a los casos de uso, se conforma un listado de No Conformidades Detectadas, las cuales se podrán apreciar a continuación, donde se podrá observar cuales fueron las No Conformidades que se detectaron una vez culminada la implementación del módulo y con él las de los reportes que se implementaron.

VALIDACION DE LA SOLUCION PROPUESTA.

Tabla de No Conformidades Detectadas.

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Importancia	Recomendación
aplicación	1	Errores al escribir el mensaje "solo puede ingresar datos numéricos"	Gestionar Plan Operativo	1ra etapa		
aplicación	2	Acepta cualquier símbolo en nombre del recurso.	Agregar recursos materiales, gastos administrativos, otros recursos.	1ra etapa		
aplicación	3	No existe concordancia entre la fecha de inicio y fin.	Gestionar Plan Operativo, en crear plan operativo.	1ra etapa		
aplicación	4	Los datos: tiempo, cantidad, precio permite poner un signo de (+) al inicio del número.	Gestionar Plan Operativo, en Asignar Recursos Materiales, en Asignar Recursos Humanos.	1ra etapa		
aplicación	5	Cuando quiero eliminar un proyecto, que le doy aceptar el mensaje me sale un mensaje de error.	Ficha de proyecto, en eliminar proyecto.	1ra etapa		

Aquí se muestran varias de las No Conformidades detectadas en el Módulo de Presentación, aquí solamente aparecerán algunas de ellas, el listado completo de ellas se pueden encontrar en el documento: (Carrasco Ramírez, 2008).

Seguidamente se encuentran algunas de estas No Conformidades que se detectaron en la aplicación cuando se le realizaron las pruebas de caja negra.

VALIDACION DE LA SOLUCION PROPUESTA.

Asignar gastos administrativos

Nombre del recurso:
ghfn

Monto:
5465hffgjhj

Descripción:
Solo puede ingresar datos numéricos

Destino:
 Transferir a Cuba
 Invertir en Venezuela

Aceptar Cancelar

Figura 29 Errores al escribir el mensaje “solo puede ingresar datos numéricos “

Datos de la actividad

Nombre de la actividad:
hgdjnfjk #@\$#%\$^\$|

Fecha de inicio:
11/06/2008

Fecha de fin:
03/06/2008

Ejecutor:
Cuba

Receptor:
Venezuela

Resonsable:
Ente Cubano

Siguiente Cancelar

Figura 30 No existe concordancia entre la fecha de inicio y fin.

VALIDACION DE LA SOLUCION PROPUESTA.

Recursos Humanos						
+ Agregar						
Categoría	Cantidad	Tiempo	Pasaje	Viáticos	Hospedaje	Honorario
tecnico	564	+65-Días	645	6879	45	65

Recursos Materiales o Suministros		
+ Agregar		
Nombre	Cantidad	Total
hbfghfghj	+523	285558

Figura 31 Los datos: tiempo, cantidad, precio permite signo al inicio del número.

Estas no conformidades detectadas cuando se realizaron las pruebas al Módulo fueron vistas, analizadas y corregidas, una vez que fueron corregidos los errores que se encontraron se les realizó nuevamente el plan de prueba donde no se encontraron errores quedando así liberado el software por el grupo de calidad del proyecto. Después de ser liberado por el grupo de calidad del proyecto paso a ser revisado por el grupo de calidad de la Universidad, el cual no encontró errores y el producto fue finalmente liberado por el grupo de calidad de la Universidad (UCI).

Como constancia de haber realizado una adecuada implementación de los reportes del Módulo de Presentación, que a la vez condujo a la adecuada implementación del módulo completo en generar se muestra el acta de aceptación del módulo Presentación (**ver Anexo 1**) por parte del cliente y el acta de finalización oficial del proyecto (**ver Anexo 2**), lo cual plasma la satisfacción del cliente con el sistema CICC.V.

3.4- CONCLUSIONES.

En este capítulo se describe de forma detallada como se llevó a cabo el flujo de trabajo Prueba descrito por RUP mediante la utilización de los casos de prueba. Se realizaron pruebas de unidad que permitieron validar la solución, y encontrar errores ocultos, así como la complejidad del software.

Después de leer éste capítulo el lector puede tener una idea de la importancia de las pruebas, que son una de las herramientas para asegurar la calidad de un producto y que ofrecen un gran valor al mismo descubriendo los errores que tienen los programas por mucho cuidado que se tenga a la hora de desarrollar la aplicación.

CONCLUSIONES GENERALES.

CONCLUSIONES GENERALES.

Con el desarrollo de este trabajo se ha demostrado la necesidad de disponer de un sistema para la gestión de toda la información de los proyectos que se llevan a cabo entre Cuba y Venezuela, logrando finalmente desarrollar un sistema informático para automatizar toda la información.

Se espera que se halle comprendido el problema planteado y la solución, así como el desarrollo de las diferentes etapas de la aplicación usando la metodología RUP. A causa del problema planteado se realizó un estudio de todas las herramientas y las tecnologías actuales, para finalmente determinar cuáles serían las herramientas necesarias para implementar el sistema, el cual tiene que ser confiable, robusto, eficiente, que cumpla con todas las expectativas del cliente y solucione los problemas existentes. Se decidió desarrollar el sistema en software libre, se utilizó como base el patrón de arquitectura MVC. Se determinó emplear como herramienta de modelado el Visual Paradigm y como metodología de desarrollo la RUP.

Dado a que éste proyecto será desarrollado en Web es necesario un servidor Web por lo que se decidió utilizar el Apache Tomcat. Para desarrollar el sistema es necesario una plataforma de desarrollo por lo que se decidió utilizar J2EE la cual tiene muy buena integración con el servidor Web. Es necesario emplear un lenguaje de programación del lado del servidor, para este caso se decidió utilizar Java, ya que este lenguaje es el utilizado por la plataforma de desarrollo.

Logrando tener ya una arquitectura para el desarrollo del sistema se puede pasar a su desarrollo, para esto fue necesario utilizar varias herramientas, las cuales se integran muy bien con la plataforma y el lenguaje seleccionado para el desarrollo, como herramienta para el diseño de los reportes se utilizó Ireport, para la generación de los reportes fue utilizada JasperReport, como IDE se decidió utilizar Eclipse y se utilizó PostgreSQL como SGBD.

Una vez terminado todo el proceso de trabajo se pudo llegar a la conclusión de que una buena implementación del Módulo de Presentación del proyecto CICCIV es de vital importancia para todo el proyecto. Por todo lo expuesto anteriormente se puede finalizar con la afirmación de que éste sistema da solución a la situación problemática que lo originó y se logra el cumplimiento del objetivo general trazado inicialmente así como los objetivos específicos. Además de que el costo por parte del software sería

CONCLUSIONES GENERALES.

mínimo ya que se fue construido con tecnologías gratuitas sin descuidar la calidad del producto.

RECOMENDACIONES.

RECOMENDACIONES.

- ❖ Mejorar el diseño gráfico de la interfaz de los reportes ya que esta primera versión fue enfocada a las funcionalidades.
- ❖ Darle mantenimiento a los reportes para que estos se mantengan actualizados según la necesidad de los clientes.
- ❖ Realizar un estudio profundo acerca de los sistemas de reportes dinámicos para que en futuras versiones se puedan implementar.

BIBLIOGRAFIA.

BIBLIOGRAFIA.

- Achour, Mehdi.** Manual de PHP. [En línea] [Citado el: 25 de febrero de 2006.] <http://www.php.net/manual/es/introduction.php>.
- Agulló, P. 2004.** *operaciones de Entrada y Salida en Java – Parte I y Parte II.* 2004.
- Allamaraju, S. y Beust, C. 2001.** *Professional Java Server Programming J2EE.* 2001.
- Bauer, C. 2005.** *Hibernate in Action, M.P. Co.* 2005.
- Brown, Mike. 2008.** *Jetty: An Efficient, Easy to Manage Alternative to WebSphere.* 2008.
- Calero Solís, Manuel. 2003.** *Una explicación de la programación extrema (XP).* 2003.
- Canós, J. H., Letelier, P. y Penadés, M. C. 2007.** *Metodologías Ágiles en el Desarrollo de Software. Universidad Politécnica de Valencia .* 2007.
- Carrasco Ramírez, Yudelkys. 2008.** *Diseño casos de prueba presentación v1.0.* 2008.
- . **2008.** *No Conformidades Presentación (ampliada) v1 0.* 2008.
- Curto Díaz, Josep. 2007.** *Comparativa herramientas reporting open source.* 2007.
- De Nova, Albert. 2006.** *Creación de informes con imágenes dinámicas en Crystal Reports con C#.* 2006.
- 2005.** *En Introducción a la Tecnología .NET. 2004-2005 . Elementos básicos del lenguaje C#. 2004-2005.* 2005.
- Enhydra, Comunidad. 2007.** Open Source Java solutions. [En línea] 2007. [Citado el: 29 de enero de 2009.] <http://www.enhydra.org/index.php>.
- Ferguson, Jeff y otros. 2003.** *La Biblia de C#.* 2003.
- Fernández Escribano, Gerardo. 2002.** *Introducción a Extreme Programming.* 2002.
- Foundation, Apache Software.** The Apache Software Foundation. [En línea] [Citado el: 9 de febrero de 2009.] <http://www.apache.org/>.
- Foundation, The Eclipse.** Eclipse. [En línea] [Citado el: 1 de febrero de 2009.] <http://www.eclipse.org/>.
- Framework, Home.Zend.** Home.Zend Framework. [En línea] [Citado el: 23 de febrero de 2009.] <http://www.framework.zend.com/docs/overview>.
- Gelais, Marie St.** *Administración de la Base de Datos Oracle 9i.*
- Gim Gray, Kalen Delaney.** *A fondo Sql Server 2000.* s.l. : Mcgraw-Hill.
- González Barbone, Víctor A. 2007.** *XP: Extreme Programming.* 2007.

BIBLIOGRAFIA.

Gonzalez, José Antonio. *El lenguaje de programación C#.*

Harrop, R. 2005. *Pro Spring, Apress.* 2005.

Heudecker, N. 2003. TheServerSide.COM. [En línea] 2003. [Citado el: 24 de febrero de 2009.] <http://www.theserverside.com/>.

Ireport, Comunidad. 2007. *Tutorial Herramienta de diseño iReport v-2.1.* 2007.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El Proceso Unificado de Desarrollo de Software.* 2000.

Johnson, R. 2005. TheServerSide.COM. [En línea] 2005. [Citado el: 24 de febrero de 2009.] <http://www.theserverside.com>.

Korth, Henry F. y Silberschatz, Abraham. 1993. *Fundamentos de bases de datos.* Madrid: McGraw-Hill : s.n., 1993.

León, Junta de Castilla y. 1999. *Guía de Iniciación al Lenguaje Java.* 1999.

License, GFDL y Version, book-1.2-en-2009-05-16. 2009. *The Definitive Guide to symfony.* 2009.

Mendoza Sánchez, María A. 2004. Informatizate. [En línea] 2004. [Citado el: 13 de enero de 2009.] <http://www.informatizate.net>.

Molpeceres Touris, Alberto y Pérez Mariñán, Martín. 2002. *Arquitectura empresarial y software libre, J2EE.* 2002.

Montané Izaguirre, Juan Carlos y Villalón Cruzata, Martín. 2008. *Plantilla DCS - Especificacion de Requisitos.* 2008.

Montané Izaguirre, Juan Carlos y Villalón Cruzata, Martín. 2008. *Plantilla DCS - Modelo de Casos de uso del sistema v 1.0.* 2008.

NetBeans. NetBeans. [En línea] [Citado el: 29 de enero de 2009.] <http://www.netbeans.org/kb/trails/platform.html>.

NetBeans, Comunidad de. NetBeans. [En línea] [Citado el: 26 de enero de 2009.] http://www.netbeans.org/index_es.html.

Olsina, L. A., y otros. 2003. *Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad.* 2003.

Olsina, L., Lafuente, G.J. y Pastor, O. 2002. *Designing a Catalogue for Metrics.* Sta Fe, Argentina : s.n., 2002. 108-122, ISSN 1666-6526.

Olson, Philip, Richter, Georg y Dovgal, Antony. 2006. *Manual de PHP, Smarty y PEAR.* 2006.

BIBLIOGRAFIA.

Plattini, Mario G. 1997. *Fundamentos y modelos de bases de datos.* Madrid: RA-MA : s.n., 1997.

Saenz Russi, Luis Humberto. 2005. *Comparación entre sistemas de gestión de bases de datos (SGBD) bajo licenciamiento libre y comercial.* Bogota : s.n., 2005.

Sánchez Villa, Yojaimer. 2009. *Diagramas de clases de los Casos de Usos de los reportes de presentación.* 2009.

—. **2008.** *Validaciones de los campos de entrada de los reportes de Presentación.* 2008.

Silberschatz, A., Korth, H. y Sudarshan, S. 2002. *Fundamentos de bases de datos.* Madrid: McGraw Hill. : s.n., 2002.

SouKup, Ron. *Microsoft SQL Server a fondo.* .

Storkel, Scott. 2002. *An Introduction to the Eclipse IDE.* 2002.

Symfony.es, 2009. *Symfony.es.* [En línea] [Citado el: 23 de febrero de 2009.] <http://www.symfony.es/documentacion/>.

Walls, C. 2005. *Spring in Action, M.P. Co.* 2005.

Worsley, John C. y Drake, Joshua D. 2002. *Practical PostgreSQL.* . 2002.

GLOSARIO DE TERMINOS.

GLOSARIO DE TERMINOS.

Componente: Parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces, y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos.

Caso de uso (CU): Representación de la agrupación de funcionalidades comunes. Representan un conjunto de iteraciones entre el sistema y sus actores.

Hibernate: Herramienta de Mapeo objeto-relacional.

Spring: Framework de código abierto de desarrollo de aplicaciones para la plataforma Java, el cual nos facilitará la creación de aplicaciones.

ANEXOS.

ANEXOS.

Anexo 1: Acta de Aceptación del Módulo de Presentación.

Acta de Aceptación

 **albet**
INGENIERÍA Y SISTEMAS

Producto: Sistema de Gestión para el seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba Venezuela (CCV)

Categoría de las pruebas: Revisión de Módulos Presentación, Administración y Configuración del CCV versión 1.0.

Fecha de conciliación: 20 de agosto de 2008

Involucrados en el proceso:


- **Por la parte del Cliente (CCV):** Sandra Cortés
- **Por la parte desarrolladora (ALBET):** Ing. Nahuel Massón Padilla
- **Observador independiente (CALISOFT):** Ing. Dayami Rodríguez Brito
- **Observador independiente (CALISOFT):** Ing. Roig Calzadilla Díaz

Observaciones del proceso:

Durante el proceso de pruebas de aceptación se identificaron un conjunto de cambios y mejoras necesarias que quedaron registrados adecuadamente en el correspondiente documento de Respuestas a las No Conformidades detectadas, con sus respectivas observaciones. Teniendo en cuenta que las No Conformidades han sido debidamente respondidas, ejecutadas por el Equipo de Desarrollo y validada la eficacia de los mismos por los clientes, se ha tomado el acuerdo de aceptar el "Sistema de Gestión para el seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba Venezuela" para la versión 1.0, con fecha 20 de agosto de 2008, el cual se anexa a esta Acta, estableciendo de esta forma la condición necesaria y suficiente para su despliegue.

Para que conste la aceptación de los resultados de las pruebas y por tanto la aceptación del "Sistema de Gestión para el seguimiento de los Proyectos del Convenio Integral de Cooperación Cuba Venezuela", dando fe del acuerdo firman la presente los principales representantes de las partes.


Sandra Cortés
(Por la parte del Cliente)


Ing. Nahuel Massón Padilla
(Por la parte Cubana - ALBET)


Ing. Roig Calzadilla Díaz
(Observador Independiente CALISOFT)



Figura 32 Acta de Aceptación del Módulo de Presentación.

Anexo 2: Acta de finiquito del sistema CCV.

ACTA DE FINIQUITO

Entre la República Bolivariana de Venezuela, actuando por órgano del Ministerio del Poder Popular para la Energía y Petróleo de la República Bolivariana de Venezuela, representado en este acto por el ciudadano **Ammar Jabour**, venezolano, mayor de edad, de este domicilio, titular de la Cédula de Identidad No. 9962729, quien actúa en su condición de Coordinador General del Convenio Integral de Cooperación Cuba-Venezuela, suficientemente facultado para este acto y debidamente designado para ejercer tal condición conforme a lo establecido en **CONTRATO E08-001-000 SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA**, que en lo sucesivo se denominará la "Parte Venezolana", por una parte; y por la otra, la sociedad mercantil **ALBET, Ingeniería y Sistemas, S.A.**, sociedad mercantil cubana constituida mediante Escritura 271 de fecha 7 de Noviembre de 2005, autorizada por la Notario Lic. Isabel Cristina Martínez Alfonso con sede en Notaría Especial del Ministerio de Justicia de Ciudad de la Habana, inscrita en el Registro Mercantil de esta ciudad con fecha 14 de Noviembre del año 2005, al Tomo XVIII, Folio 120, Hoja 11, Sección SM, con N° de inscripción 1 con domicilio social en Centro de Negocios de Miramar, Edificio Barcelona, Oficina 322, Avenida 5ta entre 76 y 78, Miramar, Municipio Playa, Ciudad de La Habana, República de Cuba, representada en este acto por el ciudadano cubano **Ibrahim Nápoles Albanés**, mayor de edad, portador de carné de identidad N° 62032504808 en su condición de Coordinador General, suficientemente facultado para este acto según lo dispuesto en **CONTRATO E08-001-000 SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA**, que en lo sucesivo se denominará "Parte Cubana", al tenor de las siguientes declaraciones y cláusulas:

CONSIDERANDO

Primero: Consta de documento privado suscrito en fecha 18 de marzo de 2008, que ambas partes celebraron un **CONTRATO DE SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA**, el cual fue celebrado al Amparo del Convenio Integral de Cooperación, suscrito el 30 de octubre del 2000 y su Addendum; de la Declaración Conjunta y el Acuerdo suscrito entre ambas Repúblicas, para la aplicación de la Alternativa Bolivariana para las Américas, firmados en diciembre del 2004 y de los Acuerdos y las Condiciones Generales firmados en el Acta de la Reunión de la Comisión Mixta Cuba-Venezuela, celebrada en La Habana, Cuba, cuyos contenidos se dan aquí enteramente por reproducidos.

Segundo: Consta que la **Parte Cubana** ha cumplido a entera satisfacción de la **Parte Venezolana** con el objeto, alcance y actividades previstas en el **Contrato** ya mencionado, así como con sus Anexos y Suplementos, cumpliendo por tanto con todos sus deberes y obligaciones por lo que se considera que la Solución ha sido implementada en las condiciones previstas y bajos los requisitos y especificaciones técnicas pactadas entre **Las Partes**.

Tercero: Consta que la **Parte Cubana** ha hecho entrega del **Informe Técnico Final**, de conjunto a toda la documentación que avala y soporta lo descrito en el Segundo de los **CONSIDERANDOS**, debidamente firmada y aceptada por los especialistas de la **Parte Venezolana**.

ANEXOS.

Cuarto: Consta que la **Parte Venezolana** ha pagado la totalidad de **UN MILLÓN CUATROCIENTOS DIECISIETE MIL OCHOCIENTOS VEINTIOCHO DOLARES DE LOS ESTADOS UNIDOS DE AMERICA CON VEINTIUN CENTAVOS** mediante la forma de pago prevista en el Contrato ya mencionado, a entera satisfacción de la Parte Cubana, quedando a la firma de la presente **Acta**.

LAS PARTES CONVIENEN:


Primero: Dar por terminada la relación contractual derivada del **CONTRATO DE SOLUCIÓN TECNOLÓGICA INTEGRAL PARA EL DESARROLLO DEL SISTEMA DE GESTIÓN PARA SEGUIMIENTO DE LOS PROYECTOS DEL CONVENIO INTEGRAL DE COOPERACIÓN CUBA VENEZUELA** de fecha 16 de marzo de 2008.

Segundo: Las Partes se otorgan el finiquito más amplio que en derecho proceda, no reservándose acción o derecho que ejercitar con posterioridad.

Leída que les fue la presente Acta, las partes se ratificaron en su contenido, para constancia firman en cuatro (4) ejemplares de igual tenor en la Ciudad de Caracas el día 17 del mes de diciembre del 2008.

Por la PARTE VENEZOLANA

Por la PARTE CUBANA



Ammar Jabour
Coordinador General



Ibrahim Nápoles Albanés
Coordinador General

Figura 33 Acta de finiquito del sistema CCV.

Anexo 3: Implementación de la clase InversionHistoricaController.

```
public ModelAndView handleRequest(HttpServletRequest request,
    HttpServletResponse arg1) throws Exception {
    int annoI= 0;
    int annoF= 0;
    BigDecimal totalProyect=new BigDecimal(0);
    BigDecimal totalMonto = new BigDecimal(0);
    BigDecimal totalInvertir = new BigDecimal(0);
    BigDecimal totalTransferir = new BigDecimal(0);
    annoI = Integer.parseInt(request.getParameter("annoInicio"));
    annoF = Integer.parseInt(request.getParameter("annoFin"));
    String logo = request.getRealPath("/WEB-INF/reportes/Imagenes/logo.png");

    Date fechaInicio=new Date();
    fechaInicio.setYear(annoI-1900);
    fechaInicio.setMonth(0);
    fechaInicio.setDate(1);

    Date fechaFin=new Date();
    fechaFin.setYear(annoF-1900);
    fechaFin.setMonth(11);
    fechaFin.setDate(31);

    List<Object[]> datos=reporteService.inversionAnual(fechaInicio, fechaFin);
    ArrayList lista = new ArrayList();
    RoundingMode redondeo = RoundingMode.HALF_EVEN;
    try {
        if (fechaFin.after(fechaInicio)) {
            if (datos.isEmpty()) {
                System.out.println("No existen datos en la Base de Datos");
                return new ModelAndView("no-datos");
            } else {
                boolean flag = true;
                for (int i = 0; i < datos.size(); i++) {
```

```
if (datos.get(i)[0] != null && datos.get(i)[1] != null
    && datos.get(i)[2] != null
    && datos.get(i)[3] != null) {
    InversionHistorica inversion = new InversionHistorica();
    BigDecimal Monto = new BigDecimal("0.0");
    BigDecimal Invertir = new BigDecimal("0.0");
    BigDecimal Transferir = new BigDecimal("0.0");
    inversion.setAnno(Integer.parseInt(datos.get(i)[0].toString()));
    inversion.setCantProyectos(datos.get(i)[1].toString());
    Invertir = Invertir.add(new BigDecimal(datos.get(i)[2].toString()).setScale(2, redondeo));
    inversion.setMontoInvVenezuela(BigDecimalFormat.convertBigDecimalToString(Invertir, 2));
    Transferir = Transferir.add(new BigDecimal(datos.get(i)[3].toString()).setScale(2, redondeo));
    inversion.setMontoTransfCuba(BigDecimalFormat.convertBigDecimalToString(Transferir, 2));
    Monto = Monto.add(Invertir.add(Transferir)).setScale(2, redondeo);
    inversion.setMontoTotal(BigDecimalFormat.convertBigDecimalToString(Monto, 2));
    inversion.setMontoGTotal(Monto);
    lista.add(inversion);

    totalProyect = totalProyect.add(new BigDecimal(datos.get(i)[1].toString()));
    totalMonto = totalMonto.add(Monto.setScale(2, redondeo));
    totalInvertir = totalInvertir.add(Invertir.setScale(2, redondeo));
    totalTransferir = totalTransferir.add(Transferir.setScale(2, redondeo));
}

else
    flag = false;
}
if (!flag) {
    System.out.println("Existen datos nulos");
    return new ModelAndView("error");
}
}
} else {
```



```
        System.out.println("La fecha inicial es mayor que la fecha final");
        return new ModelAndView("error");
    }
    String transfCuba = BigDecimalFormat.convertBigDecimalToString(totalTransferir, 2);
    String invVnz = BigDecimalFormat.convertBigDecimalToString(totalInvertir, 2);
    String totalDolares = BigDecimalFormat.convertBigDecimalToString(totalMonto, 2);
    String totalProyectos = totalProyect.toString();
    // TODO Auto-generated method stub
    return new ModelAndView("inversionhistorica", getModel(lista, logo,
        totalProyectos, invVnz, transfCuba, totalDolares, annoI,
        annoF));
} catch (Exception e) {
    return new ModelAndView("error");
}
}
```

Figura 34 Implementación de la clase InversionHistoricaController.