

**Universidad de las Ciencias Informáticas**  
**Facultad 6**



**Título:** Ambiente integrado para el modelado y ejecución de procesos de gestión de la información. Herramienta web para el diseño de reportes.

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

**Autores:** Keyler Rodriguez Velázquez  
Oscar Reyes Pérez

**Tutores:** Ing. Aida Portelles Valdes  
Lic. Yoandry Pacheco Aguila

Ciudad de la Habana, Junio de 2010



*“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio conscientes de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.”*

***Ché***

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 \_\_\_\_\_.

Keyler Rodriguez Velázquez

\_\_\_\_\_  
Firma del Autor

Ing. Aida Portelles Valdes

\_\_\_\_\_  
Firma del Tutor

Oscar Reyes Pérez

\_\_\_\_\_  
Firma del Autor

Lic. Yoandry Pacheco Aguila

\_\_\_\_\_  
Firma del Tutor

## DATOS DE CONTACTO

Tutor: Ing. Aida Portelles Valdes  
Especialidad de graduación: Ingeniería en Ciencias Informáticas  
Categoría docente: Instructor  
Categoría Científica: no  
Años de experiencia en el tema: 0  
Años de graduado: 1

Tutor: Lic. Yoandry Pacheco Aguila  
Especialidad de graduación: Licenciado en Ciencias de la Computación  
Categoría docente: Instructor  
Categoría Científica: no  
Años de experiencia en el tema: 0  
Años de graduado: 4

## **AGRADECIMIENTOS**

*A la UCI por darme la oportunidad de superarme. A todos los profesores que han contribuido con mi formación profesional.*

*A mis tutores Pacheco y Aida por su compromiso y dedicación. A Omar, Daniel y Glennis por la complicidad. A Yanet por su paciencia, atención, preocupación y apoyo.*

*A mi mamá Marianela por creer siempre en mí, por su apoyo, dedicación y amor incondicional, por quererme sin buscar una razón, por quererme porque sí, por existir.*

*A mis abuelos Olga y Chay por pensar siempre en los demás, por sus principios, por formar nuestra familia, por permitirme ser parte de sus vidas, por malcriarme.*

*A tía María Esther por su fuerza, por ser una inalcanzable luchadora por el bienestar y unión de la familia. A tío Enrique por soportarme durante 3 largos años de mi adolescencia. A ambos por la humildad y desprendimiento que los caracteriza.*

*A mi papá, Ariel, Maykel y Karel por existir.*

*A Ramón y tía Marisol por su sencillez, desinterés y altruismo.*

*A tío Miguel por ser tan querido por los demás, por ser luchador y comprometido.*

*A tía Maribel por ser como es: de alegría contagiosa, extrovertida y desenfrenada.*

*A Diannis, Dayi, Ismael, Duniesky y Mariam por ser mis víctimas y cómplices durante nuestra inolvidable infancia.*

*A Luis Miguel, Yasmani y Dayana por acabar con la tranquilidad cuando llegan a casa.*

*A tío Misael, Sergio, Rosayda y Normelvis por ser parte de la familia.*

*A Oscar, Madays, Yoel, Elieny, Yulita, Imías, Hermes, Ana María, Rachel, Yanet, Enrique, Guillermo, Ángel, Carlos y Juan Carlos por darme la oportunidad de conocerlos.*

*A Oscar nuevamente por ser mi compañero de tesis, amigo y por hacer todos los diagramas de secuencia.*

*A los que ya no están.*

**Keyler**

## **AGRADECIMIENTOS**

*A mi familia por su constante preocupación y deseos de que siguiera adelante.*

*A Paquín por toda su ayuda.*

*A mis compañeros de aula que venimos juntos desde primer año y a los que he conocido en el transcurso de la carrera.*

*Al piquete de siempre Yulita, Imias, Gladys, Danoy, Alejandro, Rayko, Alex, Dayana, Yaneisy, Alicia y Sulay.*

*A mi novia Madays por ser mi otra mitad.*

*A Keyler por ser amigo y compañero de trabajo.*

*A Omar, Daniel, Yanet y Glennis por sus colaboraciones.*

*A mis tutores Pacheco y Aida por su ayuda incondicional.*

*A mi tía Julia por haber estado ahí todo el tiempo.*

*A mis amigos del pre que nunca se olvidan.*

*A Reynaldo, Mena, Alexis, Edgar y Yoiner por ser como hermanos.*

*Agradezco a todas las personas que de una forma u otra han contribuido en mi desarrollo profesional.*

*A nuestro Comandante, a la Revolución y a la UCI por haberme dado la oportunidad de formarme como ingeniero.*

**Oscar**

## DEDICATORIA

*A todas las personas que han sido junto a mí protagonistas de este sueño de ser ingeniero. A mi mamá, familiares y amigos por su apoyo incondicional. A todos los que en algún momento preguntaron por mi tesis. A todos los que me quieren y quiero.*

*A todos, eternamente agradecido.*

***Keyler***

*A mis padres por ser la razón de mi existir, por todo su cariño y por la confianza que depositaron en mí.*

*A mi hermano por todo su apoyo y comprensión.*

*A mi abuela por ayudarme tanto y darme ánimos para seguir.*

***Oscar***

## RESUMEN

El riguroso sistema de calidad implantado en los centros biotecnológicos demanda la realización de complejos procesos de gestión de la información que generan grandes volúmenes de datos. Esta información necesita ser consultada para la obtención de diversos informes, propios de labores de control y supervisión, lo que resulta muy engorroso debido al gran cúmulo de información.

En la investigación se desarrolla una herramienta web para el diseño personalizado de los reportes que contribuye al análisis de la información almacenada y facilita el proceso de toma de decisiones en estos centros. Este generador de reportes es una herramienta complementaria del Ambiente integrado para el modelado y ejecución de procesos, desarrollado en la Facultad 6 de la Universidad de las Ciencias Informáticas.

Se emplea la metodología RUP para guiar el proceso de desarrollo de software y UML como lenguaje de modelado. La herramienta se desarrolla en el IDE NetBeans, el framework de desarrollo utilizado es Symfony y se implementa en los lenguajes PHP5 y JavaScript.

La aplicación es una herramienta web flexible y orientada al usuario que permite el diseño de los reportes a partir de los modelos de recogida de datos y brinda la posibilidad de exportar los informes obtenidos en formato PDF y XLS.

**PALABRAS CLAVE:** Generador de reportes, informes, reportes.

# Tabla de Contenidos

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>4</b>
1.1 Reportes .....	4
1.2 Sistemas generadores de reportes.....	4
1.3 Herramientas para la creación de reportes .....	6
1.4 Procesos de desarrollo de software .....	9
1.4.1 Proceso Unificado de Desarrollo (Rational Unified Process RUP).....	9
1.5 Herramientas y tecnologías de desarrollo .....	12
1.5.1 Herramientas CASE.....	12
1.5.2 Framework de desarrollo .....	13
1.5.3 Sistema gestor de base de datos .....	14
1.5.4 Lenguajes de programación .....	14
1.5.5 AJAX .....	15
1.5.6 Dojo Toolkit .....	16
1.5.7 Entorno de desarrollo.....	16
1.5.8 Servidor web: Apache 2.0.....	16
1.6 Conclusiones.....	17
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>18</b>
2.1 Objeto de estudio .....	18
2.1.1 Situación problemática.....	18
2.1.2 Objeto de automatización .....	18
2.1.3 Propuesta del sistema .....	19
2.2 Descripción del modelo de dominio .....	19
2.2.1 Definición de las clases del modelo de dominio .....	20
2.2.2 Reglas del negocio .....	21
2.3 Requerimientos del sistema.....	21
2.3.1 Requisitos funcionales .....	21
2.3.2 Requisitos no funcionales .....	22

# Tabla de Contenidos

2.4	Modelo de casos de uso del sistema.....	24
2.4.1	Diagrama de casos de uso del sistema .....	25
2.4.2	Patrones de casos de uso utilizados.....	25
2.4.3	Descripciones textuales de los casos de uso del sistema.....	26
2.5	Conclusiones.....	40
<b>CAPÍTULO 3: DISEÑO DEL SISTEMA.....</b>		<b>41</b>
3.1	Propósito del diseño.....	41
3.2	Arquitectura del sistema.....	41
3.2.1	Estilo arquitectónico utilizado .....	42
3.2.2	Descripción del patrón MVC .....	42
3.2.3	Descripción de los patrones de diseño .....	43
3.3	Modelo de diseño .....	44
3.3.1	Diagramas de clases del diseño.....	45
3.4	Diseño de la base de datos .....	50
3.4.1	Diagrama de clases persistentes .....	50
3.4.2	Modelo de datos.....	51
3.5	Diagrama de despliegue .....	52
3.6	Conclusiones.....	52
<b>CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....</b>		<b>53</b>
4.1	Modelo de implementación.....	53
4.1.1	Diagramas de componentes.....	53
4.2	Código fuente .....	56
4.3	Pantallas principales de la aplicación .....	57
4.4	Prueba.....	59
4.5	Conclusiones.....	62
<b>CONCLUSIONES.....</b>		<b>63</b>
<b>RECOMENDACIONES.....</b>		<b>64</b>

# Tabla de Contenidos

---

REFERENCIAS BIBLIOGRÁFICAS.....	65
BIBLIOGRAFÍA.....	67
GLOSARIO.....	69

## INTRODUCCIÓN

El acelerado desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs) ha tenido gran impacto en la rama de la biotecnología, ampliando las herramientas que ofrecen la posibilidad de encontrar soluciones novedosas ante los desafíos biotecnológicos de hoy. Actualmente las principales compañías biotecnológicas se concentran en Europa, Norteamérica y Asia-Pacífico, dedicadas fundamentalmente a la producción de fármacos y vacunas para la prevención y tratamiento de enfermedades. Cuba, a pesar de ser un país del tercer mundo, ocupa un lugar destacado entre los países de mayores logros en esta esfera.

En la década de los ´90 el estado cubano se planteaba una estrategia de desarrollo hacia una biotecnología moderna. Por ello, llevó a cabo un intenso proceso inversionista y de formación de personal para la creación de varios centros biotecnológicos en Ciudad de La Habana y otras provincias del país. Con el propósito de facilitar y promover acciones coordinadas entre estas instituciones, es creado el Polo Científico del Oeste de La Habana, sistema de investigación-producción que integran el Centro Nacional de Biopreparados (BIOCEN), Centro Nacional de Sanidad Agropecuaria (CENSA), Centro Nacional de Investigaciones Científicas (CNIC), el Instituto Finlay, Instituto Pedro Kourí (IPK), el Centro de Inmunología Molecular (CIM), Centro Internacional de Restauración Neurológica (CIREN) y el Centro de Ingeniería Genética y Biotecnología(CIGB), entre otros.

Estos centros asumen la responsabilidad de concentrar esfuerzos para el desarrollo tecnológico e innovación en el ámbito de la biotecnología y contribuir directamente en el desarrollo económico y social del país. Su desempeño se proyecta en las investigaciones, generando conocimientos para el desarrollo de nuevos productos, servicios y la actividad comercial, basados en un sistema de calidad que garantiza la satisfacción de los clientes.

El sistema de calidad implantado les permite sostener e incrementar su presencia en los mercados locales e internacionales de hoy y demanda la realización de complejos procesos de gestión de la información que generan grandes volúmenes de datos. Para esto se han definido modelos estándares para la recogida de datos en formato duro cuya estructura cambia constantemente, producto de nuevas investigaciones y exigencias del mercado. Los frecuentes cambios de formato provocan que se generen varias versiones sobre un modelo específico, por lo que se dificulta el almacenamiento organizado, confiable y duradero de los datos. El riguroso sistema exige que la información generada sea constantemente consultada para la obtención de diversos informes, propios de labores de control y supervisión, lo que resulta muy engorroso debido al gran cúmulo de información y cambios en el formato de los modelos. Estos informes son

requeridos de manera periódica, necesitan ser claros, precisos y contar con la cantidad de detalles suficientes para ser analizados y comprendidos con facilidad. Todo lo anteriormente explicado provoca pérdida de tiempo y obstáculos ante la toma inmediata de decisiones.

Se identifica como **problema científico**: ¿Cómo contribuir al análisis de la información almacenada para facilitar el proceso de toma de decisiones en los centros biotecnológicos?

El problema planteado se enmarca en el **objeto de estudio**: Proceso de gestión de la información de los centros biotecnológicos.

**Campo de acción**: Proceso de generación de reportes en los centros biotecnológicos.

Se define como **objetivo general**: Desarrollar una herramienta web para el diseño de reportes que contribuya al análisis de la información almacenada para facilitar el proceso de toma de decisiones en los centros biotecnológicos.

A partir del objetivo general se derivaron los siguientes **objetivos específicos**:

- ✓ Identificar las funcionalidades de la herramienta para el diseño de reportes.
- ✓ Realizar el diseño de la herramienta para el diseño de reportes.
- ✓ Realizar la implementación de la herramienta para el diseño de reportes.

Para lograr los objetivos propuestos, se trazaron las siguientes **tareas de la investigación**:

- ✓ Estudio de los diferentes Sistemas generadores de reportes existentes en el mundo.
- ✓ Investigación de las tendencias y tecnologías para desarrollar una herramienta para la generación de reportes.
- ✓ Confección del modelo de dominio.
- ✓ Identificación de requerimientos.
- ✓ Identificación de las clases del diseño.
- ✓ Elaboración del modelo de diseño.
- ✓ Diseño de la base de datos.
- ✓ Elaboración de los diagramas de componentes.
- ✓ Implementación de la herramienta para el diseño de reportes.
- ✓ Realización de pruebas funcionales.

## Resumen Capítulo 1: Fundamentación Teórica

En este capítulo se realiza un estudio de las diferentes herramientas que existen en el mundo para la generación de reportes, describiendo las principales características, funcionalidades y conceptos, así

como un análisis crítico de las mismas. Se caracterizan las tecnologías, lenguajes y herramientas empleadas en el desarrollo de la aplicación.

## **Resumen Capítulo 2: Características del Sistema**

En este capítulo se propone la solución al problema planteado e incluye una detallada descripción de las características del sistema. Se confecciona el Modelo de dominio para lograr una mejor comprensión del contexto del sistema. Se definen los requisitos funcionales y no funcionales así como los actores y casos de usos que interactúan con el sistema.

## **Resumen Capítulo 3: Diseño del Sistema**

Este capítulo contiene el diseño del sistema como parte de la solución propuesta. Describe el estilo arquitectónico y patrones del diseño seleccionados, así como los diagramas de clases e interacción que satisfacen a cada uno de los requisitos funcionales y no funcionales.

## **Resumen Capítulo 4: Implementación y Pruebas**

En este capítulo se describe cómo los elementos del modelo de diseño se implementan en términos de componentes y contiene los diagramas de componentes para cada caso de uso. También incluye las pruebas funcionales aplicadas al sistema implementado para examinar la estructura externa del software y garantizar la calidad del mismo.

# Capítulo 1: Fundamentación Teórica

---

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se realiza un estudio de las diferentes herramientas que existen en el mundo para la generación de reportes, describiendo las principales características, funcionalidades y conceptos, así como un análisis crítico de las mismas. Se caracterizan las tecnologías, lenguajes y herramientas empleadas en el desarrollo de la aplicación.

### 1.1 Reportes

Los centros biotecnológicos están atados a un conjunto de normas que guían la calidad del desempeño realizado, en la búsqueda del desarrollo de productos y servicios de calidad. Para lograrlo se realizan actividades de control y supervisión, que requieren la evaluación de los procesos que se ejecutan en estos centros. Estas actividades necesitan la obtención de diversos informes a partir de plantillas o reportes previamente definidos que garanticen mostrar los resultados para su posterior análisis.

Un reporte es una plantilla que sirve como estándar o patrón para la generación de informes, define los datos y la manera de presentación de estos. La confección de un reporte requiere distinguir claramente algunos aspectos:

- ✓ Definición del reporte: momento en que el autor del reporte define los datos y la manera de presentación de estos.
- ✓ Administración del reporte: está referido al hecho de que en las organizaciones actuales existen distintas categorías de usuarios. Por lo tanto, es importante definir quiénes serán los usuarios del reporte.
- ✓ Presentación del informe: conjunto de datos o resultado detallado de determinado reporte. Es muy común en las organizaciones que muchos informes sean requeridos de manera periódica, estos necesitan ser claros y precisos; además, deben contar con la cantidad de detalles suficientes como para que cualquier persona pueda comprenderlo. (1)

### 1.2 Sistemas generadores de reportes

Actualmente la inmensa mayoría de las grandes empresas controlan sus actividades fundamentales por medio de sistemas informáticos de diversos grados de complejidad. Los sistemas informáticos son capaces de controlar y gestionar la información de toda clase de procesos económicos y financieros, procesos de producción y cualquier otra actividad que el cliente desee informatizar. En todos los casos, un

# Capítulo 1: Fundamentación Teórica

requisito indispensable de los sistemas informáticos es brindar la posibilidad de mostrar los resultados de la gestión de la información con diferentes grados de complejidad, exactitud y detalle, según sea requerido por los clientes. Con este fin, existen en el mundo varias herramientas, tanto libres como propietarias, que permiten obtener y analizar información de repositorios de datos heterogéneos y bases de datos; precisamente son estas las herramientas para la generación de reportes, las cuales posibilitan la materialización de la gestión de la información de una manera relativamente sencilla.

Los Sistemas generadores de reportes son herramientas complementarias de los sistemas de información. Utilizan una especie de lenguaje transparente para el usuario por medio del cual éste realiza consultas a la base de datos y obtiene información de ella en forma de informe.

Un generador de reportes se compone de dos elementos básicos, un diseñador de reportes y un motor de generación como se muestra en la figura:

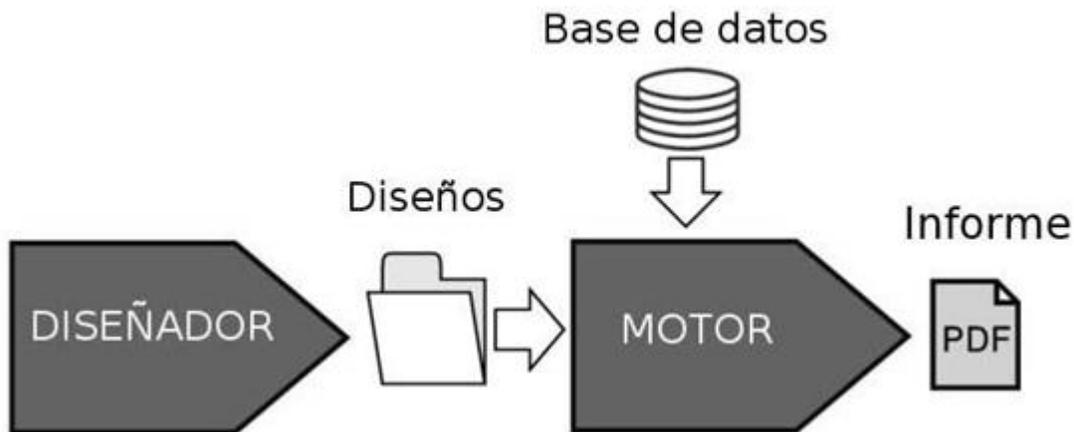


Fig. 1: Estructura general de un Sistema generador de reportes. Esquema de funcionamiento.

De aquí se puede deducir que las principales tareas que debe realizar un Sistema generador de reportes son las siguientes:

- ✓ Diseñar la apariencia de los reportes.
- ✓ Extraer la información y volcarla de forma ordenada con la apariencia diseñada.

Haciendo uso del diseñador se obtienen plantillas de diseño que posteriormente son empleadas por el motor de generación para volcar datos provenientes de la base de datos y obtener el informe final con la apariencia preestablecida en el diseño.

# Capítulo 1: Fundamentación Teórica

---

## 1.3 Herramientas para la creación de reportes

Con el propósito de conocer las principales características, ventajas y desventajas de algunos de los Sistemas generadores de reportes, se realizó un estudio de los mismos. Para el análisis se tuvo en cuenta las principales herramientas que con este fin existen en el mundo, Cuba y en la Universidad de las Ciencias Informáticas (UCI).

### Crystal Reports

Crystal Reports es una potente herramienta que posee diferentes versiones, es la herramienta de elaboración de informes estándar para Visual Studio .NET, permite crear contenido interactivo con una excelente calidad de presentación. Es un producto propietario que permite generar informes en varios tipos de formatos como HTML, PDF, XML, XLS, DOC y es multiplataforma, estando disponible para GNU/Linux, Unix y Microsoft Windows.

### Active Reports

El Active Report para .NET es un software privativo, está escrito totalmente en Visual C# y brinda una completa integración con el IDE Visual Studio .NET. Este gestor de reportes viene incluido con Microsoft Acces Report, un asistente que permite importar reportes de Acces muy fácilmente. Permite exportar a formatos populares como PDF, XLS, HTML, tanto en aplicaciones Windows como en aplicaciones web. Provee a los usuarios finales la capacidad de crear y modificar reportes y un visor web que aprovecha las ventajas de .NET para mostrar informes sin necesidad de programar código específico para exportar a formatos como HTML y PDF. (2)

### JasperReports

Es una herramienta de fuente abierta dependiente de la plataforma Java. Puede mostrar contenido de calidad en ficheros PDF, HTML, XLS y XML. La filosofía de trabajo con JasperReports parte del conocimiento de la estructura de su XML, generalmente es necesario el uso de otras herramientas visuales que ayuden en el diseño y compilación de los reportes. Puede ser utilizado en cualquier entorno o sistema operativo siempre que exista una implementación de la máquina virtual de Java para dicho entorno.

# Capítulo 1: Fundamentación Teórica

---

## PHP Report Maker 3

PHP Report Maker es una potente herramienta para la creación de reportes que permite generar páginas PHP dinámicas que construyen informes de base de datos MySQL. Es una aplicación de escritorio que posibilita crear al instante informes detallados de tablas cruzadas para sitios web con soporte para gráficos flash (columna, barra, línea, pastel, área, dona, y apiladas). Está diseñado para una alta flexibilidad, numerosas opciones permiten generar informes adaptables a las necesidades del usuario. Los códigos generados son limpios y fáciles de personalizar. PHP Report Maker no es software libre, permite utilizarlo por un período de 30 días, su uso continuado exige el pago de la licencia. (3)

## PHPReports

PHPReports es un conjunto de clases PHP, instrucciones XML y scripts XSLT para transformar archivos XML a código PHP que posibilitan la creación de reportes. Los archivos de transformación XML permiten transformar la estructura del reporte XML a código PHP. Las clases PHP son usadas para procesar el resultado de la transformación XSL y los archivos XML definen la estructura y el modo de presentación de la información, este formato sirve de plantilla y facilita además el almacenamiento e intercambio de información.

El archivo XML está basado en tres capas fundamentales:

1. Capa de informe: es la capa más externa del informe. Maneja todos los valores procesados en el informe completo.
2. Capa de la página: maneja todos los valores en la página actual.
3. Capa de grupos: maneja todos los valores en el grupo actual.

PHPReports permite el uso de templates (archivos XML estáticos) para facilitar la creación de los reportes e incluye un conjunto de plugins para manipular la presentación de la información. Está protegido bajo los términos de la Licencia Pública General de GNU y requiere servidor Apache con soporte XML/XSLT para su adecuada compilación, a pesar de que PHP corre en varios sistemas operativos, este software alcanza su máximo potencial usando GNU/LINUX.

## Sistema de Gestión de Reportes Dinámicos (SGRD)

El SGRD es una solución integral para la elaboración, generación y gestión de reportes desarrollado por el Centro de Tecnologías de Gestión de Datos (DATEC) de la UCI, provee soporte a un conjunto de

## Capítulo 1: Fundamentación Teórica

---

requisitos que satisfacen la gestión de reportes, además incluye un valor agregado relativo a la gestión de estos reportes que comprende entre otras posibilidades la distribución automatizada de los mismos. Utiliza PHPReports 0.4.9 y pChart 1.27 para la generación de reportes y gráficos respectivamente.

Está compuesto por cuatro módulos principales:

1. Diseñador de modelos: garantiza la creación de modelos y posibilita definir orígenes de datos para la conformación del modelo semántico de la base de datos.
2. Diseñador de reportes: permite el diseño, generación y publicación de los reportes.
3. Visor de reportes: permite la visualización de los reportes, así como las interfaces y servicios que brinda para ser usado por terceros.
4. Administrador de reportes: módulo responsable de la gestión de usuarios, reportes, modelos y consultas.

La elección de la herramienta más conveniente depende de las necesidades del cliente así como el tipo de arquitectura a utilizar como base para soportar una implementación que responda a dichas necesidades. Luego de haber realizado una investigación, se puede concluir que existen potentes herramientas para la generación de reportes, las cuales soportan gráficos e imágenes y permiten exportar los informes en diferentes formatos. Sin embargo, muchas de estas herramientas dependen de una plataforma específica y son, por lo general, orientadas al programador y no al usuario. Crystal Reports y Active Reports son dependientes de la plataforma .NET y propietarios, lo que implica que tienen limitadas las posibilidades de distribución y un elevado costo de adquisición. JasperReports es dependiente de la plataforma Java, la filosofía de trabajo con JasperReports parte del conocimiento de la estructura de su XML, generalmente es necesario el uso de otras herramientas visuales que ayuden en el diseño y compilación de los reportes. PHP Report Maker 3 es una herramienta para la creación de reportes que permite generar páginas PHP dinámicas que construyen informes de base de datos MySQL; es un software que permite su redistribución, sin embargo, no viene acompañado de su código fuente y, por tanto, no puede ser modificado. Además, pasado un período de tiempo, es necesario pagar una licencia para continuar usándolo. El SGRD se encuentra actualmente en desarrollo para lograr un grupo de requisitos definidos en la iteración anterior que aún no son satisfechos, entre ellos, lograr una arquitectura flexible que brinde interfaces y servicios para facilitar su uso por terceros. Este software utiliza PHPReports como motor de generación, el equipo de desarrollo explora las posibilidades de prescindir de

## Capítulo 1: Fundamentación Teórica

---

esta librería ya que no es lo suficientemente robusta y flexible para satisfacer los requerimientos de reportes complejos.

Las características de los centros biotecnológicos dificultan configurar uno de los Sistemas generadores de reportes estudiados para su uso particular, requieren una herramienta sin limitaciones de uso y distribución que permita al usuario confeccionar reportes sin la necesidad de asistencia de un desarrollador. Además, como se explicó anteriormente, la estructura de los modelos de recogida de datos de estos centros cambia con mucha frecuencia y como resultado se obtienen nuevas versiones de los mismos. Esto implica que se complejice aún más la obtención de informes de un determinado modelo debido al elevado número de versiones que se generan a partir de un ejemplar en específico.

### 1.4 Procesos de desarrollo de software

Un proceso de desarrollo de software es un conjunto de actividades necesarias para convertir los requisitos de usuarios en un conjunto consistente de artefactos que conforman un producto software (4). En los últimos años los procesos de desarrollo de software se han incrementado de manera extraordinaria llevando a cabo dos corrientes importantes, los denominados métodos pesados y métodos ligeros. Ambos métodos están encaminados a favorecer el trabajo de las personas que intervienen en el proceso de desarrollo. Los métodos pesados consiguen su objetivo por medio de orden y documentación, los métodos ligeros o ágiles tratan de mejorar la calidad del software a través de la comunicación directa e inmediata. Entre los procesos de desarrollo más conocidos se tienen: Programación Extrema (Extreme Programming, XP) y openUP clasificados como métodos ligeros, y Proceso Unificado de Desarrollo (Rational Unified Process, RUP) que se clasifica como método pesado.

#### 1.4.1 Proceso Unificado de Desarrollo (Rational Unified Process RUP)

El Proceso Unificado de Desarrollo es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, es un proceso bien definido, estructurado y adaptable a las características y necesidades de cada proyecto. Es un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, tipos de organizaciones y tamaños de proyecto. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Unifica los mejores elementos de

# Capítulo 1: Fundamentación Teórica

metodologías anteriores, preparado para desarrollar grandes y complejos proyectos, orientado a objetos y utiliza UML como lenguaje de representación visual. Se caracteriza por ser dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental. RUP se divide en cuatro fases y define nueve flujos de trabajo a realizar en cada fase del proyecto.

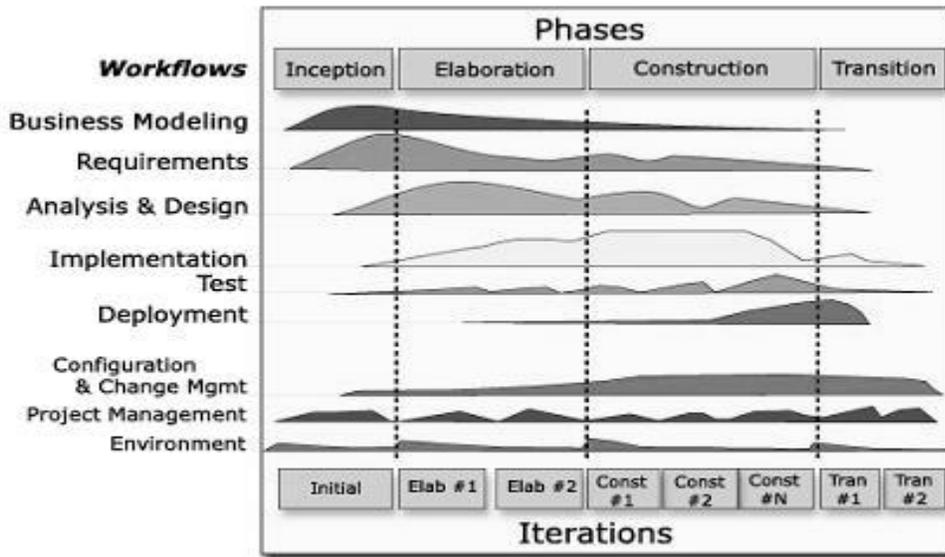


Fig. 2: Fases e iteraciones de la metodología de desarrollo RUP.

## ¿Por qué RUP?

RUP es la metodología seleccionada para guiar el proceso de desarrollo a pesar de ser una metodología tradicional, o sea, rígida y dirigida por la documentación que se genera en cada una de las actividades desarrolladas. Es una metodología muy exitosa y utilizada actualmente en proyectos de gran envergadura. El proceso de desarrollo RUP aplica varias de las mejores prácticas en el desarrollo moderno de software en una forma que se adapta a un amplio rango de proyectos y de organizaciones. Provee a cada miembro del equipo un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas del desarrollo de software. Esta metodología permite que todos los integrantes de un equipo de trabajo conozcan y compartan el proceso de desarrollo y los distintos modelos de cómo desarrollar el software utilizando un lenguaje de modelado común: UML.

# Capítulo 1: Fundamentación Teórica

---

## Roles y artefactos

Una particularidad de este proceso de desarrollo es que, en cada ciclo de iteración, se hace exigente el uso de artefactos para alcanzar un grado de certificación en el desarrollo del software. Un proceso de desarrollo se describe en partes llamadas flujos de trabajo, en los cuales los trabajadores y los artefactos son los participantes. RUP especifica varios roles que definen el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo.

Los roles a desempeñar durante el ciclo de desarrollo del software, teniendo en cuenta las características de esta investigación, serán: Analista, Diseñador de sistema, Arquitecto de software, Diseñador de base de datos y Programador, pertenecientes a los grupos de analistas y desarrolladores definido por RUP.

**Analista:** dirige y coordina la adquisición de requisitos esquematizando la funcionalidad del sistema y delimitándolo.

Artefactos que serán realizados por el Analista:

- ✓ Modelo de casos de uso: es un modelo de las funciones deseadas para el sistema y su entorno, y sirve como contrato entre el cliente y los desarrolladores. Se utiliza como entrada esencial para las actividades de análisis, diseño y prueba.
- ✓ Glosario de términos: define términos importantes que se utilizan en el proyecto.
- ✓ Actor del sistema: este artefacto define un conjunto coherente de roles que los usuarios del sistema pueden desempeñar cuando interactúan con este.
- ✓ Caso de uso del sistema: es una secuencia de acciones que lleva a cabo un sistema que producen un resultado observable de valor para un actor concreto.

**Diseñador de sistema:** este rol dirige el diseño de una parte del sistema, dentro de las restricciones de los requisitos, arquitectura y proceso de desarrollo para el proyecto.

Artefactos que serán realizados por el Diseñador de sistema:

- ✓ Clase del diseño: se define como una descripción de un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos y semánticas.
- ✓ Realización de caso de uso: describe cómo un caso de uso particular es realizado dentro del modelo de diseño, en términos de colaboración de objetos.
- ✓ Subsistema de diseño: describe una parte del sistema que encapsula comportamiento, expone un conjunto de interfaces y empaqueta otros elementos del modelo.

# Capítulo 1: Fundamentación Teórica

---

**Arquitecto de software:** es responsable de la arquitectura del software incluyendo las decisiones técnicas que restringen el diseño e implementación general del proyecto.

A continuación se describe el artefacto que será producido por el Arquitecto de software:

- ✓ Diagrama de despliegue: define cómo se distribuye físicamente un sistema. Está representado por nodos procesadores o nodos dispositivos unidos mediante conexiones de comunicación.

**Diseñador de base de datos:** dirige el diseño de la estructura de almacenamiento de datos persistentes que se utilizará en el sistema.

Artefacto que será realizado por el Diseñador de base de datos:

- ✓ Modelo de datos: describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación.

**Implementador:** desarrolla los componentes de software y efectúa las pruebas de desarrollador para la integración en subsistemas más grandes, de acuerdo con los estándares adoptados por el proyecto.

Artefacto que será usado como entrada para las actividades realizadas por el Implementador:

- ✓ Diagrama de componentes: estructuran el modelo de implementación en términos de subsistemas de implementación y muestran las relaciones entre los elementos de implementación.

## Lenguaje de modelado

RUP define como lenguaje de modelado el Lenguaje Unificado de Modelado (UML del inglés Unified Model Language). UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema y permite la modelación de sistemas con tecnología orientada a objetos. Garantiza un entendimiento haciendo uso de una notación unificada que propicia el intercambio entre los usuarios y los desarrolladores. Su uso reduce el tiempo de desarrollo, permite la modelación de sistemas orientados a objetos y garantiza una mejor planificación y control a partir de la reutilización y minimización de costos.

## 1.5 Herramientas y tecnologías de desarrollo

### 1.5.1 Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste en términos de tiempo y de dinero (5). Estas herramientas viabilizan la realización de un conjunto de tareas indispensables en el ciclo de vida de desarrollo del software tales como el proceso de realizar un diseño

# Capítulo 1: Fundamentación Teórica

---

del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras.

## Visual Paradigm

Es una herramienta colaborativa que soporta múltiples usuarios trabajando sobre el mismo proyecto. Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código. La principal ventaja del Visual Paradigm y el motivo por el cual ha sido seleccionada es por ser una herramienta multiplataforma que garantiza la calidad del software durante su desarrollo, ya que permite la comunicación entre los desarrolladores mediante un lenguaje común para todos los roles que intervienen en el proceso de desarrollo del software. Además, el equipo de desarrollo goza de una vasta experiencia con la herramienta, por ser la más usada y extendida entre los proyectos productivos de la facultad.

## 1.5.2 Framework de desarrollo

Un framework de desarrollo es una estructura conceptual y tecnológica de soporte, definida normalmente con artefactos de software concretos, mediante la cual otro proyecto de software puede ser organizado y desarrollado (6). Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Symfony 1.2 es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web y tiene disponibilidad en múltiples plataformas. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Las principales características que permitieron seleccionarlo como framework de desarrollo son las siguientes:

- ✓ Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- ✓ Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- ✓ Fácil de extender, lo que permite su integración con bibliotecas de otros fabricantes.

# Capítulo 1: Fundamentación Teórica

---

## 1.5.3 Sistema gestor de base de datos

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas, procedimientos y lenguajes que proporcionan las herramientas necesarias para trabajar con una base de datos. Permite incorporar una serie de funciones para definir registros, campos, relaciones, insertar, suprimir, modificar y consultar los datos. (7)

### PostgreSQL 8.3

PostgreSQL está considerado como uno de los gestores de bases de datos de código abierto más avanzado. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales. Tiene más de 15 años de desarrollo activo y se ha ganado la reputación de ser confiable y mantener la integridad de los datos (8). Además, corre en la mayoría de los sistemas operativos más utilizados. Cumple la prueba ACID (Atomicity, Consistency, Integrity, Durability) e incluye la mayoría de los tipos de datos de los estándares SQL92 y SQL99 (INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, TIMESTAMP, entre otros), tiene soporte para lenguajes procedurales internos y soporta almacenamiento de objetos grandes (imágenes, sonido y video). PostgreSQL se ha convertido en una herramienta muy versátil, es un SGBD de nivel empresarial, que puede ser usado en casi todo tipo de sistemas. Conserva el alto rendimiento sin sacrificar la estabilidad ni la integridad de los datos.

## 1.5.4 Lenguajes de programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina (9). Permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. A continuación se describen los lenguajes de programación utilizados para la construcción de la aplicación web:

### PHP 5

El framework de desarrollo seleccionado para el desarrollo de la aplicación web es Symfony, el cual, como se explicó anteriormente, permite crear aplicaciones PHP de forma más sencilla y rápida. PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la web, gratuito e

# Capítulo 1: Fundamentación Teórica

---

independiente de la plataforma, rápido, con una gran librería de funciones y mucha documentación. Dentro de una página web se puede incrustar código PHP que se ejecutará siempre que esta sea visitada. El código PHP es interpretado en el servidor web y genera código HTML y otros contenidos que el usuario puede ver.

## JavaScript

Lenguaje de programación que ha permitido el gran desarrollo de la web, ha sido el avance más significativo en el logro de páginas web dinámicas y exactas en cuanto a posición y presentación de su contenido, es un lenguaje robusto y a la vez ligero, el cual a pesar de ser no orientado a objetos permite implementar varias de las características de este paradigma de programación. JavaScript es un lenguaje utilizado para crear pequeños programas que luego son insertados en una página web. Gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del ratón (mouse), utilización de teclas, cargas de páginas, entre otros. Este es un lenguaje interpretado, no requiere compilación y es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera y Mozilla Firefox (10).

### 1.5.5 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la velocidad de interacción en la misma.

AJAX no constituye una tecnología en sí, es un término que engloba a un grupo de éstas que trabajan conjuntamente. AJAX incorpora:

1. XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
2. Document Object Model (DOM), para mostrar e interactuar dinámicamente con la información presentada.
3. El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web.
4. XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo JSON. (11)

## Capítulo 1: Fundamentación Teórica

---

### 1.5.6 Dojo Toolkit

Existen herramientas que facilitan la creación de interfaces gráficas de usuario, estas son conocidas como toolkits. Un toolkit consiste en una biblioteca de utilidades, las cuales pueden incorporar componentes de interfaz de usuario (IU) o widgets.

Dojo es mucho más que un framework, es un conjunto de herramientas de JavaScript que permite desarrollar aplicaciones web profesionales que utilicen tecnología AJAX de forma sencilla y más rápida (12). Provee gran cantidad de funcionalidades, las cuales divide en tres grandes paquetes: Dojo Core, Dijit y DojoX. Dijit incorpora un conjunto de widgets con los cuales se pueden crear interfaces amigables en muy poco tiempo y escribiendo casi o ningún código JavaScript. DojoX es donde se desarrollan extensiones para Dojo Toolkit, sirve de cuna para nuevas ideas y pruebas de funcionalidades adicionales para el Toolkit principal. Está disponible ya sea en virtud de los términos de la licencia BSD modificada o la Licencia Libre Académica versión 2.1; ambas licencias le otorgan amplios derechos de uso y distribución.

### 1.5.7 Entorno de desarrollo

Es un producto libre y gratuito sin restricciones de uso. NetBeans IDE 6.8 Beta proporciona herramientas para la programación en el lenguaje PHP e incluye resaltado de sintaxis, autocompletado de código, templates, documentación de PHP integrada, entre otras funcionalidades. Provee soporte para el framework Symfony y cuenta con un módulo de Subversion para facilitar la manipulación de las versiones del código durante la implementación. Por sus favorables características es el IDE seleccionado para el desarrollo de la aplicación.

### 1.5.8 Servidor web: Apache 2.0

Un servidor web es un programa que implementa el protocolo HTTP y está diseñado para transferir hipertextos, páginas web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música (13). Es un programa que se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente y que responde a estas peticiones adecuadamente mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Apache es un servidor HTTP de código abierto multiplataforma y extensible, altamente configurable, flexible, rápido, eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Permite la

# Capítulo 1: Fundamentación Teórica

---

creación de sitios web dinámicos, de lenguajes de Scripting como PHP, JavaScript, Python, Java y tiene amplias opciones de configuración para la creación y gestión de logs, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

## 1.6 Conclusiones

En este capítulo se realizó un estudio de los principales Sistemas generadores reportes existentes en el mundo y en Cuba. Se caracterizaron las principales tecnologías, metodologías, herramientas y lenguajes empleados en el desarrollo de la aplicación. Se eligió RUP como metodología de desarrollo, la herramienta CASE escogida fue Visual Paradigm, UML como lenguaje de modelado y el sistema gestor de base de datos PostgreSQL. La herramienta será desarrollada en el IDE NetBeans, el cual provee soporte para el framework Symfony e implementada en PHP5 y JavaScript. Dojo será el conjunto de herramientas de JavaScript a utilizar para desarrollar una aplicación web profesional con tecnología AJAX.

## Capítulo 2: Características del Sistema

---

### **CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA**

En este capítulo se propone la solución al problema planteado e incluye una detallada descripción de las características del sistema. Se confecciona el Modelo de dominio para lograr una mejor comprensión del contexto del sistema. Se definen los requisitos funcionales y no funcionales así como los actores y casos de usos que interactúan con el sistema.

#### **2.1 Objeto de estudio**

##### **2.1.1 Situación problemática**

Los centros e instituciones dedicados a la biotecnología en Cuba se proyectan hacia el desarrollo tecnológico y contribuyen directamente al desarrollo económico y social del país. Estos centros demandan la realización de complejos procesos de gestión de la información, aún poco estructurados, que generan grandes volúmenes de datos. Para el almacenamiento de la información se han definido modelos estándares para la recogida de datos en formato duro cuya estructura cambia constantemente. Los frecuentes cambios provocan que se generen varias versiones sobre un modelo específico, por lo que se dificulta el almacenamiento organizado, confiable y duradero de los datos.

La información generada necesita ser constantemente consultada para la generación de diversos informes, propios de labores de control y supervisión, lo que resulta muy engorroso debido al gran cúmulo de información y cambios en el formato de los modelos. Estos informes son requeridos de manera periódica, necesitan ser claros, precisos y contar con la cantidad de detalles suficientes para ser analizados y comprendidos con facilidad. Lo explicado anteriormente evidencia la deficiente estructuración de los procesos de diseño de reportes y generación de informes, necesarios para brindar la posibilidad de mostrar los resultados de la gestión de la información.

##### **2.1.2 Objeto de automatización**

###### **Proceso de diseñar un reporte**

El proceso de diseñar reportes es uno de los más importantes dentro de estos centros, el mismo comienza cuando una persona solicita realizar un reporte a partir de la información disponible en los modelos de recogida de datos, posteriormente el autor del reporte define los datos y la manera de presentación del informe.

## Capítulo 2: Características del Sistema

---

### **Proceso de obtener un informe**

Este proceso permite al usuario obtener un informe a partir de un reporte diseñado anteriormente, el mismo comienza cuando éste solicita obtener un informe determinado, posteriormente busca el reporte que necesita y obtienen el informe.

### **2.1.3 Propuesta del sistema**

Se propone el desarrollo de una herramienta web flexible, independiente de la plataforma de desarrollo y orientada al usuario, permitiendo el diseño personalizado de los reportes a partir de los modelos de recogida de datos; y además, capaz de presentar las distintas versiones de un modelo como una sola unidad, de tal forma que estas sean transparentes al usuario, es decir, permitirle al usuario final tener una vista unificada de todas las versiones de un determinado modelo para facilitar el diseño de reportes a partir de este último. Una herramienta de propósito general, adaptable al negocio de cualquier centro biotecnológico que haga uso de ella. Se debe garantizar guardar el reporte una vez diseñado y facilitar su gestión, es decir, visualizarlo, eliminarlo, modificarlo y generar el informe a partir de éste. Debe permitir además imprimir y exportar los informes obtenidos en formato PDF y XLS.

### **2.2 Descripción del modelo de dominio**

El modelo de dominio representa los conceptos más importantes relacionados con el dominio y constituye un “diccionario visual” de términos importantes en el dominio. Es una representación visual estática del entorno real del proyecto. El objetivo del modelado del dominio es contribuir a la comprensión del contexto del sistema y, por lo tanto, a la comprensión de los requisitos del sistema que se desprenden de este contexto.

Se ha decidido realizar un modelo de dominio ya que es una alternativa apropiada dado el escenario del problema. No se considera necesario un modelado completo del negocio, las fronteras del negocio no se pueden determinar con claridad y los procesos del negocio no están claramente definidos. La figura que a continuación se muestra representa el modelo de dominio:

## Capítulo 2: Características del Sistema

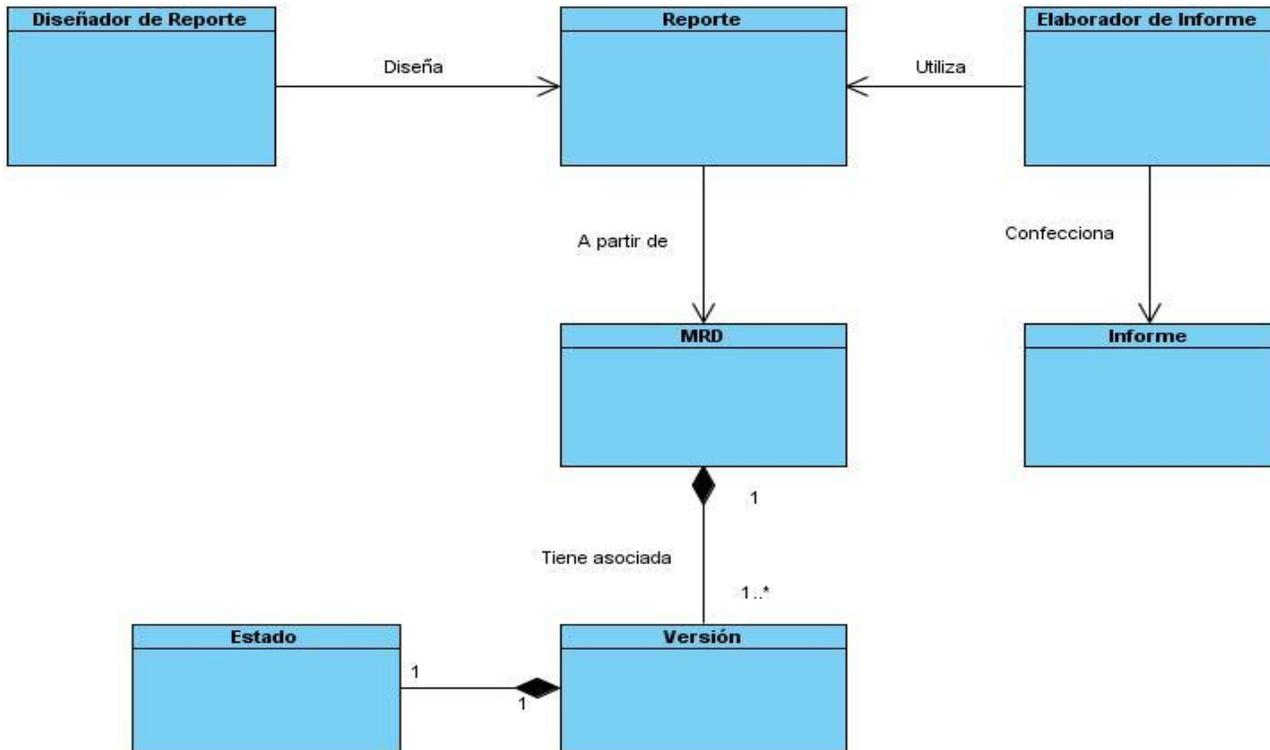


Fig. 3: Modelo de dominio.

### 2.2.1 Definición de las clases del modelo de dominio

1. **Diseñador de reporte:** persona responsable de diseñar reportes.
2. **Reporte:** plantilla que sirve como estándar para la confección de informes, define los datos y la manera de presentación de estos.
3. **MRD (Modelo de recogida de datos):** plantilla para la recogida de datos con un conjunto de variables definidas, de forma que la información de la misma pueda ser analizada y comprendida con facilidad.
4. **Versión:** modelo de recogida de datos cuya estructura ha cambiado como resultado de nuevas investigaciones y exigencias del mercado.
5. **Elaborador de informe:** persona responsable de confeccionar el informe.
6. **Informe:** conjunto de datos o resultado detallado de determinado reporte.
7. **Estado:** estado o fase en la que se puede encontrar una versión (activo, edición, obsoleto).

## Capítulo 2: Características del Sistema

---

### 2.2.2 Reglas del negocio

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, las reglas que regulan el negocio se describen a continuación:

1. Se podrá diseñar un reporte sólo con información contenida en un modelo de recogida de datos en específico y sus respectivas versiones.
2. Las versiones de un modelo de recogida de datos serán transparentes a los usuarios que utilizan los informes.

### 2.3 Requerimientos del sistema

El propósito fundamental de la captura de los requisitos es definir el ámbito del sistema y proveer a los desarrolladores un mejor entendimiento de lo que debe hacer el sistema. Para la captura de los requisitos se utilizaron las técnicas tormenta de ideas y entrevistas.

Como resultado del flujo de trabajo de Requerimientos definido por la metodología de desarrollo RUP se identifican los siguientes requisitos funcionales del sistema a implementar:

#### 2.3.1 Requisitos funcionales

- RF1** Crear reporte
- RF2** Modificar reporte
- RF3** Eliminar reporte
- RF4** Buscar y visualizar reporte
- RF5** Diseñar reporte
- RF6** Editar diseño de reporte
- RF7** Insertar elemento de diseño
- RF8** Eliminar elemento de diseño
- RF9** Modificar elemento de diseño
- RF10** Generar y visualizar informe
- RF11** Imprimir informe
- RF12** Exportar informe

## Capítulo 2: Características del Sistema

---

### 2.3.2 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades reflejan las características que hacen al producto atractivo, usable, rápido y confiable. Los requerimientos no funcionales que se identifican son:

#### Requerimientos de software

Sistema operativo Windows XP o superior, Linux (cualquier distribución). Las PC clientes de los usuarios accederán al sistema utilizando el navegador web Mozilla Firefox 3.0 o Internet Explorer 5 o versiones superiores. El servidor de Base de datos debe tener instalado Postgree SQL 8.3. En el servidor de aplicaciones se debe instalar un servidor Web Apache 2.0.

#### Requerimientos de hardware

Se deberá contar con impresora en las PC clientes o compartidas que interactúen con la aplicación.

Los requerimientos recomendados para:

Servidor de aplicaciones:

- procesador Pentium IV a 3.0 GHz
- 1 GB de RAM
- 250 GB de espacio libre en el disco duro.

Servidor de base de datos:

- procesador Pentium IV a 3.0 GHz
- 1 GB de RAM
- 500 GB de espacio libre en el disco duro.

PC cliente:

- procesador Pentium IV
- 256 MB de RAM

#### Requerimientos de apariencia o interfaz externa

La aplicación contará con una interfaz amigable con páginas no cargadas de mucha información, colores suaves y lo más cercano a una aplicación de escritorio.

## Capítulo 2: Características del Sistema

---

### Requerimientos de seguridad

*Confidencialidad:* existencia de distintos roles que establezcan que la información sólo sea vista por aquellos usuarios que posean los privilegios suficientes; restringir la ejecución de acciones a usuarios sin credenciales que intenten acceder a las mismas.

*Integridad:* validación de los datos en el servidor para evitar estados inconsistentes. La información manejada por el sistema estará protegida del acceso y divulgación no autorizada. Se debe realizar la confirmación sobre acciones irreversibles como eliminaciones.

*Disponibilidad:* el sistema estará disponible las 24 horas del día a los usuarios autorizados, garantizando el acceso a la información en cualquier momento. Los mecanismos utilizados para lograr la seguridad no obstruyen el acceso a la información.

Estos tres principios de la seguridad informática serán garantizados por el módulo de gestión de usuarios y el plugin sfDojoPlugin; este plugin permite la autenticación de los usuarios y la creación de sus credenciales.

### Requerimientos de usabilidad

La aplicación podrá ser utilizada por personal vinculado a la biotecnología, que tengan conocimientos básicos de computación y de aplicaciones web.

### Requerimientos de rendimiento

Reducción de los tiempos de respuestas y alta velocidad de procesamiento de la información a través de peticiones asincrónicas al servidor. Los tiempos de respuestas deben ser aproximadamente de 3 a 5 segundos, al igual que la velocidad de procesamiento de la información.

### Restricciones del diseño y la implementación

Para el diseño y documentación de la aplicación se utilizará la metodología RUP, usando el lenguaje de modelado UML 6.4. La Arquitectura estará basada en el patrón Modelo Vista Controlador, utilizando para el desarrollo Visual Paradigm 3.4 y NetBeans 6.8. Los lenguajes a utilizar serán JavaScript y PHP, como estándares de codificación PHPCase y CamelCase. Se hará uso de la biblioteca Dojo Toolkit.

## Capítulo 2: Características del Sistema

### Requerimientos de soporte

Garantía de instalación y prueba del sistema, además de un breve entrenamiento a los futuros usuarios. Se brindará asistencia técnica en un período de 6 meses y se realizarán pruebas de estrés durante el primer año de explotación.

### Requerimientos de persistencia

La información del sistema debe almacenarse en bases de datos con carácter imborrable con el objetivo de poder realizar análisis de la misma en cualquier momento durante el paso de los años.

### Portabilidad

Será un sistema multiplataforma, se podrá disponer del mismo tanto en el sistema operativo Linux como Windows.

### Legales

El sistema será registrado en el Centro Nacional de Derecho de Autor a través de la Dirección de Servicios Legales de la UCI. Se estará usando para el desarrollo de la aplicación herramientas de software libre con licencia GNU/GPL.

### 2.4 Modelo de casos de uso del sistema

El modelo de casos de uso del sistema es un artefacto que contiene actores, casos de uso y sus relaciones. Cada caso de uso del modelo se describe detalladamente, mostrando paso a paso el modo en que el sistema interactúa con los actores y lo que el sistema hace en el caso de uso.

Identificar los actores que interactuarán con el sistema es una de las tareas más importantes que propone el flujo de trabajo de Requerimientos, permite definir las fronteras y esquematizar la funcionalidad del sistema.

Actores del sistema identificados:

Actor	Descripción
Participante	Es la persona que interactúa con el sistema, puede realizar búsquedas de reportes y generar informes a partir de estos últimos.
Diseñador de reporte	Responsable de administrar los reportes.

## Capítulo 2: Características del Sistema

### 2.4.1 Diagrama de casos de uso del sistema

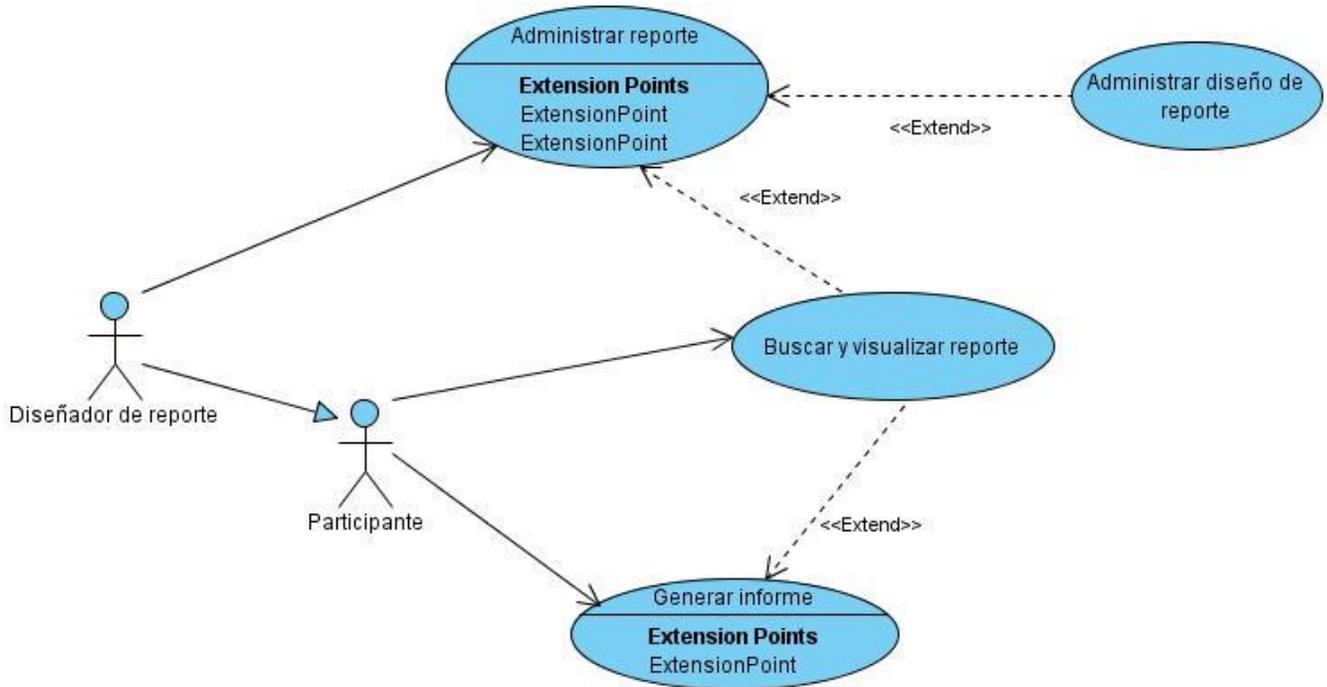


Fig. 4: Diagrama de casos de uso del sistema.

### 2.4.2 Patrones de casos de uso utilizados

Los patrones de casos de usos pueden ser utilizados como técnicas o herramientas, resultantes de la experiencia de desarrolladores, que permiten de manera ágil resolver problemas que se presentan en la modelación de sistemas, obteniendo modelos de mayor calidad de forma más rápida.

A continuación se describen los patrones de casos de uso utilizados en la modelación del sistema:

#### Concordancia (Commonality). Adición

Consta de 3 casos de uso. El primero llamado subsecuencia común, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema. (14)

## Capítulo 2: Características del Sistema

### Concordancia (Commonality). Reusabilidad interna

Cuando una subsecuencia de acciones es utilizada en diferentes lugares en un solo caso de uso no existe la necesidad de extraer la subsecuencia dentro de un caso de uso separado. Debe ser descrita entonces en una sección separada en la descripción del caso de uso. Esta sección será referenciada desde diferentes partes en la descripción del caso de uso donde las subsecuencias de acciones sean realizadas. Este patrón se utiliza cuando la subsecuencia común aparece en múltiples lugares en un mismo caso de uso. (14)

### Múltiples actores. Roles comunes

Este patrón es utilizado cuando dos actores juegan el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso. (14)

### 2.4.3 Descripciones textuales de los casos de uso del sistema

A continuación se muestran las descripciones textuales de los casos de uso del sistema:

#### CUS Administrar reporte

<b>Caso de Uso:</b>	Administrar reporte
<b>Actores:</b>	Diseñador de reporte (Inicia)
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el Diseñador de reporte va a realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>- Crear reporte: El Diseñador de reporte selecciona la opción “Crear reporte” e introduce los datos requeridos para su creación en el formulario correspondiente, finalizando el CU.</li> <li>- Modificar reporte: El Diseñador de reporte selecciona el reporte a modificar, el sistema muestra los datos correspondientes y el Diseñador de reporte los modifica, finalizando el CU.</li> <li>- Eliminar reporte: Cuando el Diseñador de reporte decide eliminar un reporte selecciona dicho reporte, indica la acción y el sistema lo elimina, finalizando el CU.</li> </ul>
<b>Precondiciones:</b>	El Diseñador de reporte debe estar autenticado.

## Capítulo 2: Características del Sistema

<b>Referencias</b>	RF1, RF2, RF3, CUS Administrar reporte (Extendido), CUS Buscar y visualizar reporte (Extendido)
<b>Prioridad</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El Diseñador de reporte selecciona la acción que desea realizar:</p> <ul style="list-style-type: none"> <li>- Crear reporte.</li> <li>- Modificar reporte.</li> <li>- Eliminar reporte.</li> <li>- Administrar diseño de reporte.</li> </ul>	<p>2. El sistema, en dependencia de la acción solicitada, hace lo siguiente:</p> <ul style="list-style-type: none"> <li>- Si decide crear un nuevo reporte, ir a Sección “Crear reporte”.</li> <li>- Si decide modificar reporte, ir a Sección “Modificar reporte”.</li> <li>- Si decide eliminar reporte ir a la Sección “Eliminar reporte”.</li> <li>- Si decide administrar diseño de reporte ir a CU extendido “Administrar diseño de reporte”.</li> </ul>
<b>Sección “Crear reporte”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El Diseñador de reporte selecciona la opción “Añadir”.</p>	<p>2. Muestra el formulario “Datos del reporte” para la creación de un nuevo reporte.</p>
<p>3. Inserta los datos requeridos para la creación del reporte:</p> <ul style="list-style-type: none"> <li>• Título.</li> <li>• Entidad base.</li> <li>• Grupo de acceso.</li> <li>• Descripción.</li> </ul>	<p>4. El sistema comprueba que los datos introducidos para la creación del reporte sean válidos.</p>
<p>5. Selecciona la opción “Guardar”.</p>	<p>6. Verifica que los campos obligatorios no estén vacíos:</p> <ul style="list-style-type: none"> <li>• Título.</li> <li>• Entidad base.</li> </ul>

## Capítulo 2: Características del Sistema

	<ul style="list-style-type: none"> <li>• Grupo de Acceso.</li> </ul>
	<p>7. Completa automáticamente los campos restantes:</p> <ul style="list-style-type: none"> <li>• Creador del reporte.</li> <li>• Fecha de creación.</li> </ul>
	<p>8. Guarda los datos y muestra el reporte creado. Finaliza el CU.</p>
<b>Flujos Alternos Sección “Crear reporte”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<p>4.1 Si los datos introducidos no son válidos, el sistema muestra el mensaje “El valor especificado no es válido”. Ir a la acción 3.</p>
	<p>6.1 Si existen campos obligatorios vacíos, el sistema muestra el mensaje “Este campo es requerido, por favor complételo.” Y lo señala en color rojo. Ir a la acción 3.</p>
<b>Sección “Modificar reporte”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. Selecciona el reporte que desea modificar del listado de reportes.</p>	
<p>2. Oprime el botón “Editar”.</p>	<p>3. Muestra un formulario con los datos requeridos para la modificación del reporte:</p> <ul style="list-style-type: none"> <li>• Título.</li> <li>• Entidad base (No editable).</li> <li>• Grupo con acceso.</li> <li>• Descripción.</li> </ul>
<p>4. Modifica los datos.</p>	<p>5. El sistema comprueba que los datos introducidos para la modificación del reporte sean</p>

## Capítulo 2: Características del Sistema

	válidos.
6. Oprime el botón "Guardar".	7. Verifica que los campos obligatorios no estén vacíos.
	8. Guarda los datos y muestra el reporte modificado. Finaliza el CU.
<b>Flujos Alternos Sección "Modificar reporte"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.1 El Diseñador de reporte desea realizar una búsqueda por filtrado (ir a CU extendido Buscar y visualizar reporte), Ir a la acción 1.	
	5.1 Si existen datos incorrectos el sistema muestra el mensaje de error "El valor especificado no es válido", ir a la acción 4.
6.1 El Diseñador de reporte no desea guardar los cambios, cancela la acción y finaliza el CU.	
	7.1 Si existen campos obligatorios vacíos, el sistema muestra el mensaje "Este campo es requerido, por favor complételo". Y lo señala en color rojo. Ir a la acción 4.
<b>Sección "Eliminar reporte"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona el reporte que desea eliminar del listado de reportes.	
2. Oprime el botón "Eliminar".	3. Muestra un cuadro de diálogo con el mensaje "Se eliminará el elemento seleccionado" solicitando la confirmación de la acción indicada.
4. Confirma que desea eliminar el reporte seleccionado.	5. Elimina el reporte seleccionado y actualiza el listado de reportes. Finaliza el CU.
<b>Flujos Alternos Sección "Eliminar reporte"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

## Capítulo 2: Características del Sistema

1.1 El Diseñador de reporte desea realizar una búsqueda por filtrado (ir a CU extendido Buscar y visualizar reporte). Ir a la acción 1.	
	3.1 Si el Diseñador de reporte no selecciona el reporte a eliminar el sistema muestra el mensaje de error “Para ejecutar esta acción antes debe seleccionar una fila”, ir a la acción 1.
4.1 El usuario confirma que no desea eliminar el reporte. Finaliza el CU.	
<b>Poscondiciones</b>	<p>En dependencia de la acción del actor:</p> <ul style="list-style-type: none"> <li>• Se crea un nuevo reporte.</li> <li>• Se modifica un reporte.</li> <li>• Se elimina un reporte.</li> </ul>

### CUS Administrar diseño de reporte

<b>Caso de Uso:</b>	Administrar diseño de reporte
<b>Actores:</b>	Diseñador de reporte (Inicia)
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el Diseñador de reporte va a realizar alguna de las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>- Diseñar reporte: El Diseñador de reporte selecciona el reporte que desea diseñar, el sistema muestra la interfaz correspondiente y permite al actor insertar, eliminar y modificar elementos de diseño, finalizando así el CU.</li> <li>- Editar diseño de reporte: El Diseñador de reporte selecciona el reporte al cual desea editar su diseño, el sistema muestra la interfaz correspondiente y permite al actor insertar, eliminar y modificar elementos de diseño, finalizando así el CU.</li> </ul>
<b>Precondiciones:</b>	El Diseñador de reporte debe estar autenticado y debe existir al menos un reporte en el listado.
<b>Referencias</b>	RF5, RF6, RF7, RF8, RF9
<b>Prioridad</b>	Crítico

### Flujo Normal de Eventos

## Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
1. El Diseñador de reporte desea realizar una de las siguientes operaciones: - Diseñar reporte. - Editar diseño de reporte.	2. El sistema, en dependencia de la acción solicitada, hace lo siguiente: - Si decide Diseñar reporte: ir a sección “Diseñar reporte”. - Si decide Editar diseño de reporte: ir a sección “Editar diseño de reporte”.
<b>Sección “Diseñar reporte”</b>	
Acción del Actor	Respuesta del Sistema
1. El Diseñador de reporte selecciona el reporte que desea diseñar.	
2. Selecciona la opción “Diseñar”.	3. Muestra el área de trabajo para el diseño de reporte.
	4. El sistema permite al Diseñador de reporte seleccionar el(los) tipo(s) de elemento(s) de diseño que formará(n) parte del diseño del reporte: <b>Caso a)</b> Campo <b>Caso b)</b> Filtro
5. Selecciona el elemento de diseño.	6. El sistema muestra la interfaz correspondiente y permite: <b>Si selecciona Caso a):</b> - Insertar elemento de diseño campo (ir a sección “Insertar elemento de diseño campo”). - Eliminar elemento de diseño campo (ir a sección “Eliminar elemento de diseño campo”). - Modificar elemento de diseño campo (ir a sección “Modificar elemento de diseño campo”). <b>Si selecciona Caso b)</b> - Insertar elemento de diseño filtro (ir a sección “Insertar elemento de diseño filtro”).

## Capítulo 2: Características del Sistema

	<ul style="list-style-type: none"> <li>- Eliminar elemento de diseño filtro (ir a sección “Eliminar elemento de diseño filtro”).</li> <li>- Modificar elemento de diseño filtro (ir a sección “Modificar elemento de diseño filtro”).</li> </ul>
	7. El sistema guarda el diseño del reporte.
<b>Sección “Editar diseño de reporte”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona el reporte para editar su diseño.	
2. Selecciona la opción “Diseñar”.	3. El sistema muestra el diseño del reporte seleccionado.
	4. El sistema permite al Diseñador de reporte seleccionar el(los) tipo(s) de elemento(s) de diseño que desea editar: <b>Caso a)</b> Campo <b>Caso b)</b> Filtro
5. Selecciona el elemento de diseño.	6.El sistema muestra la interfaz correspondiente y permite: <b>Si selecciona Caso a):</b> <ul style="list-style-type: none"> <li>- Insertar elemento de diseño campo (ir a sección “Insertar elemento de diseño campo”).</li> <li>- Eliminar elemento de diseño campo (ir a sección “Eliminar elemento de diseño campo”).</li> <li>- Modificar elemento de diseño campo (ir a sección “Modificar elemento de diseño campo”).</li> </ul> <b>Si selecciona Caso b)</b> <ul style="list-style-type: none"> <li>- Insertar elemento de diseño filtro (ir a sección “Insertar elemento de diseño filtro”).</li> <li>- Eliminar elemento de diseño filtro (ir a sección “Eliminar elemento de diseño filtro”).</li> <li>- Modificar elemento de diseño filtro (ir a sección</li> </ul>

## Capítulo 2: Características del Sistema

	“Modificar elemento de diseño filtro”).
	5. El sistema guarda los cambios efectuados, finalizando el CU.
<b>Sección “Insertar elemento de diseño campo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción “Añadir”.	2. El sistema muestra un formulario con los campos correspondientes al elemento de diseño campo. <ul style="list-style-type: none"> <li>• Nombre campo</li> <li>• Alias del campo</li> <li>• Agrupar</li> <li>• Ordenar</li> </ul>
3. El Diseñador de reporte introduce los datos del elemento de diseño.	4. El sistema valida los datos del elemento de diseño introducido.
5. Indica la opción “Guardar”.	6. Verifica que los campos obligatorios no estén vacíos: <ul style="list-style-type: none"> <li>• Ordenar.</li> <li>• Campo.</li> </ul>
	7. El sistema adiciona el elemento de diseño.
<b>Flujos Alternos Sección “Insertar elemento de diseño campo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 Si existen datos incorrectos el sistema muestra el mensaje “El valor especificado no es válido”. Ir a la acción 3.
5.1 El Diseñador de reporte no desea insertar el elemento de diseño, cancela la acción y finaliza la sección.	
	6.1 Si existen campos obligatorios vacíos el sistema muestra el mensaje de error “Campo

## Capítulo 2: Características del Sistema

	obligatorio: Este campo es requerido, por favor complételo”, ir a la acción 5.
<b>Sección “Eliminar elemento de diseño campo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona el elemento de diseño que desea eliminar e indica la opción “Eliminar”.	2. Muestra un cuadro de diálogo con el mensaje “Se eliminará el elemento seleccionado” solicitando la confirmación de la acción indicada.
3. Confirma que desea eliminar el elemento de diseño seleccionado.	4. Elimina el elemento de diseño seleccionado. Finaliza la sección.
<b>Flujos Alternos Sección “Eliminar elemento de diseño campo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.1 El Diseñador de reportes no selecciona el elemento de diseño e indica la opción “Eliminar”.	1.2 El sistema muestra el mensaje de error “Antes debe seleccionar una fila”, ir a la acción 1.
3.1 El Diseñador de reportes confirma que no desea eliminar el elemento de diseño. Finaliza la sección.	
<b>Sección “Modificar elemento de diseño campo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona el elemento de diseño que desea modificar e indica la opción “Editar”.	2. Muestra un formulario con los datos del elemento de diseño seleccionado.
3. Modifica los datos del elemento de diseño que desea: <ul style="list-style-type: none"> <li>• Alias del campo.</li> <li>• Agrupar.</li> <li>• Ordenar.</li> </ul>	4. El sistema valida los datos del elemento de diseño introducidos.
5. Indica la opción “Guardar”.	6. Verifica que los campos obligatorios no estén vacíos: <ul style="list-style-type: none"> <li>• Ordenar.</li> <li>• Campo.</li> </ul>

## Capítulo 2: Características del Sistema

	7. El sistema guarda los cambios del elemento de diseño. Finaliza la sección.
<b>Flujos Alternos Sección “Modificar elemento de diseño campo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.1 El Diseñador de reportes no selecciona el elemento de diseño e indica la opción “Editar”.	1.2 El sistema muestra el mensaje de error “Antes debe seleccionar una fila”, ir a la acción 3.
	4.1 Si existen datos incorrectos el sistema muestra el mensaje de error “El valor especificado no es válido”, ir a la acción 5.
5.1 Si el Diseñador de reporte no desea editar las propiedades de otros elementos de diseño cierra el formulario y finaliza la sección.	
	6.1 Si existen campos obligatorios vacíos el sistema muestra el mensaje de error “Campo obligatorio: Este campo es requerido, por favor complételo”, ir a la acción 5.
<b>Sección “Insertar elemento de diseño filtro”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona la opción “Añadir”.	2. El sistema muestra un formulario con los campos correspondientes al elemento de diseño filtro: <ul style="list-style-type: none"> <li>• Campo.</li> <li>• Criterio.</li> <li>• Valor.</li> <li>• Agrupar con anterior.</li> </ul>
3. El Diseñador de reporte introduce los datos del elemento de diseño.	4. El sistema valida los datos del elemento de diseño introducido.
5. Indica la opción “Guardar”.	6. Verifica que los campos obligatorios no estén vacíos: <ul style="list-style-type: none"> <li>• Campo.</li> </ul>

## Capítulo 2: Características del Sistema

	<ul style="list-style-type: none"> <li>• Criterio.</li> </ul>
	7. El sistema adiciona el elemento de diseño.
<b>Flujos Alternos Sección “Insertar elemento de diseño filtro”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	4.1 Si existen datos incorrectos el sistema muestra el mensaje “El valor especificado no es válido”. Ir a la acción 3.
5.1 El Diseñador de reporte no desea insertar el elemento de diseño, cancela la acción y finaliza la sección.	
	6.1 Si existen campos obligatorios vacíos el sistema muestra el mensaje de error “Campo obligatorio: Este campo es requerido, por favor complételo”, ir a la acción 5.
<b>Sección “Eliminar elemento de diseño filtro”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. Selecciona el elemento de diseño que desea eliminar e indica la opción “Eliminar”.	2. Muestra un cuadro de diálogo con el mensaje “Se eliminará el elemento seleccionado” solicitando la confirmación de la acción indicada.
3. Confirma que desea eliminar el elemento de diseño seleccionado.	4. Elimina el elemento de diseño seleccionado. Finaliza la sección.
<b>Flujos Alternos Sección “Eliminar elemento de diseño campo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.1 El Diseñador de reportes no selecciona el elemento de diseño e indica la opción “Eliminar”.	1.2 El sistema muestra el mensaje de error “Antes debe seleccionar una fila”, ir a la acción 1.
3.1 El Diseñador de reportes confirma que no desea eliminar el elemento de diseño. Finaliza la sección.	
<b>Sección “Modificar elemento de diseño filtro”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>

## Capítulo 2: Características del Sistema

1. Selecciona el elemento de diseño que desea modificar e indica la opción “Editar”.	2. Muestra un formulario con los datos del elemento de diseño seleccionado.
3. Modifica los datos del elemento de diseño que desea: <ul style="list-style-type: none"> <li>• Campo.</li> <li>• Criterio.</li> <li>• Valor.</li> <li>• Agrupar con anterior.</li> </ul>	4. El sistema valida los datos del elemento de diseño introducidos.
5. Indica la opción “Guardar”.	6. Verifica que los campos obligatorios no estén vacíos: <ul style="list-style-type: none"> <li>• Campo.</li> <li>• Criterio.</li> </ul>
	9. El sistema guarda los cambios del elemento de diseño. Finaliza la sección.
<b>Flujos Alternos Sección “Modificar elemento de diseño filtro”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1.1 El Diseñador de reportes no selecciona el elemento de diseño e indica la opción “Editar”.	1.2 El sistema muestra el mensaje de error “Antes debe seleccionar una fila”, ir a la acción 3.
	4.1 Si existen datos incorrectos el sistema muestra el mensaje de error “El valor especificado no es válido”, ir a la acción 5.
5.1 Si el Diseñador de reporte no desea editar las propiedades de otros elementos de diseño cierra el formulario y finaliza la sección.	
	6.1 Si existen campos obligatorios vacíos el sistema muestra el mensaje de error “Campo obligatorio: Este campo es requerido, por favor complételo”, ir a la acción 5.
<b>Poscondiciones</b>	En dependencia de la acción del actor:

## Capítulo 2: Características del Sistema

	<ul style="list-style-type: none"> <li>• Se diseña un reporte.</li> <li>• Se edita el diseño de un reporte.</li> <li>• Se inserta elementos de diseño.</li> <li>• Se elimina elementos de diseño.</li> <li>• Se modifica elementos de diseño.</li> </ul>
--	--

### CUS Buscar y visualizar reporte

<b>Caso de Uso:</b>	Buscar y visualizar reporte
<b>Actores:</b>	Participante (Inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando el Participante desea realizar una búsqueda por filtrado de reportes.
<b>Precondiciones:</b>	El Participante debe estar autenticado.
<b>Referencias</b>	RF4
<b>Prioridad</b>	Crítico

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El Participante selecciona la opción: "Buscar".	2. El sistema muestra el formulario correspondiente a la búsqueda de reporte(s) con los siguientes campos: <ul style="list-style-type: none"> <li>• Variable (título, código, fecha de creación).</li> <li>• Criterio.</li> <li>• Valor.</li> </ul>
3. Selecciona la variable de filtrado por la cual desea realizar la búsqueda de reportes, especifica el criterio correspondiente a la variable de filtrado seleccionada e introduce el valor del criterio de búsqueda.	4. El sistema comprueba que el valor introducido para la variable y criterio seleccionado sea válido.
	5. El sistema ofrece la posibilidad al Participante de adicionar otros parámetros de filtrado.

## Capítulo 2: Características del Sistema

6. Si desea adicionar otros parámetros de filtrado ir a la acción 3 de esta sección.	
7. Oprime el botón “Buscar”.	8. El sistema busca y muestra un listado con los reportes que coincidan con los parámetros introducidos para la búsqueda, finalizando el CU.
Flujos Alternos Sección “Buscar y visualizar reporte”	
Acción del Actor	Respuesta del Sistema
	4.1 Si el Participante introduce un valor incorrecto, el sistema emite un mensaje de error: “El valor especificado no es válido”. Ir a la acción 3.
6.1 Si el Participante no desea adicionar otros parámetros de búsqueda, ir a la acción 7.	
	8.1 Si no obtiene resultados al concluir la búsqueda muestra un mensaje: “No se encontraron elementos”. Finaliza el CU.
<b>Poscondiciones</b>	Se busca y visualiza el reporte.

### CUS Generar informe

<b>Caso de Uso:</b>	Generar informe
<b>Actores:</b>	Participante (Inicia)
<b>Resumen:</b>	El caso de uso se inicia cuando el Participante desea generar un informe.
<b>Precondiciones</b>	El Participante debe estar autenticado.
<b>Referencias</b>	RF10, RF11, RF12, CUS Buscar y visualizar reporte (Extendido)
<b>Prioridad</b>	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Participante selecciona un reporte del listado de reportes.	
2. Selecciona la opción “Generar informe”.	3. Genera y visualiza el informe correspondiente al

## Capítulo 2: Características del Sistema

	<p>reporte en una o varias páginas(en dependencia de la cantidad de información a presentar) y brinda la posibilidad de realizar las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>- Agrupar filas.</li> <li>- Exportar informe (exporta uno o varios archivos en dependencia de la paginación).</li> <li>- Imprimir informe.</li> </ul>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1.1 Si el Participante desea realizar una búsqueda por filtrado ir a CU extendido “Buscar y visualizar reporte”.	
<b>Poscondiciones</b>	<p>Se genera y visualiza el informe correspondiente al reporte seleccionado y en dependencia de la acción del actor:</p> <ul style="list-style-type: none"> <li>• Se agrupan las filas en correspondencia con el diseño.</li> <li>• Se imprime el informe.</li> <li>• Se exporta el informe.</li> </ul>

### 2.5 Conclusiones

Con el desarrollo de este capítulo se ha logrado una mejor comprensión de las características del sistema. El modelo de dominio contribuyó a la comprensión del contexto del sistema y, por lo tanto, a la comprensión de los requisitos del sistema. Las funcionalidades del sistema a construir quedan agrupadas en los 4 CUS identificados. Se realizó el diagrama de casos de uso del sistema, permitiendo mostrar la relación entre éstos y los actores del sistema. Se describieron detalladamente los casos de uso del sistema, facilitando el diseño de las clases y la elaboración de los diagramas correspondientes.

## **CAPÍTULO 3: DISEÑO DEL SISTEMA**

Este capítulo contiene el diseño del sistema como parte de la solución propuesta. Describe el estilo arquitectónico y patrones del diseño seleccionados, así como los diagramas de clases e interacción que satisfacen a cada uno de los requisitos funcionales y no funcionales.

### **3.1 Propósito del diseño**

Como resultado del flujo de trabajo de requerimientos se obtiene una vista externa del sistema expresado en el lenguaje del cliente, describiendo lo que se espera de él a través de casos de uso. A partir de aquí se debe profundizar en los casos de usos detallándolos de manera que permitan reflejar una vista interna del sistema, descrita con el lenguaje de los desarrolladores. Este proceso se desarrolla fundamentalmente dentro de la fase de elaboración y se corresponde principalmente con el flujo de trabajo de análisis y diseño.

En el diseño se modela el sistema para dar soporte a todos los requisitos que le suponen. Los propósitos del diseño son adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos y tecnologías de distribución; crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes capturando los requisitos o subsistemas individuales, interfaces y clases; y descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia. Esta etapa del desarrollo del software permite evolucionar hacia una arquitectura robusta y adaptar el diseño para que coincida con el ambiente de implementación, diseñando el sistema con un enfoque hacia el rendimiento.

### **3.2 Arquitectura del sistema**

La arquitectura de software define la estructura del sistema, entendida como la organización de componentes y relaciones entre ellos; los requerimientos que debe satisfacer el sistema y las restricciones a las que está sujeto, así como las propiedades no funcionales del sistema y su impacto sobre la calidad del mismo; las reglas y decisiones de diseño que gobiernan esta estructura y los argumentos que justifican las decisiones tomadas. Brinda una estructura que soporta las soluciones a cada problema que pudiera aparecer en el desarrollo y asegura que los requerimientos más importantes puedan ser evaluados e

implementados. Permite flexibilidad en el sistema pues facilita la ejecución de futuros cambios y promueve la reutilización de componentes existentes como librerías de clases y aplicaciones de terceros.

### **3.2.1 Estilo arquitectónico utilizado**

Cuando se define la arquitectura de software hay que decidir cuál o cuáles estilos arquitectónicos serán utilizados en la misma. La aplicación de dichos estilos mejora o disminuye la satisfacción de los atributos de calidad del sistema, es decir, la aplicación de un estilo depende en gran medida de los requisitos del sistema y de cómo se dará respuesta a los mismos.

Un estilo arquitectónico define las reglas generales de organización en términos de un patrón y las restricciones en la forma y la estructura de un grupo numeroso y variado de sistemas de software. En una forma más específica, un estilo determina el vocabulario de componentes y conectores que pueden ser utilizados en instancias de este estilo, con un conjunto de restricciones en las descripciones arquitectónicas. Existe una gran variedad de estilos arquitectónicos, entre ellos el de Llamada y retorno, ampliamente usado en el desarrollo moderno del software. El patrón de arquitectura utilizado es el Modelo Vista Controlador (MVC), el siguiente epígrafe ofrece una breve descripción de este patrón.

### **3.2.2 Descripción del patrón MVC**

Cuando se utiliza Symfony como framework de desarrollo es necesario emplear el patrón de arquitectura Modelo Vista Controlador, el cual pertenece a la familia del estilo arquitectónico de Llamada y retorno. Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres niveles: modelo, vista y controlador:

- ✓ El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ✓ La vista despliega la información contenida en el modelo.
- ✓ El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo. La capa del modelo se puede dividir en la capa de acceso a los datos y en la capa de abstracción de la base de datos. De esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. Así, si se cambia de sistema gestor de bases de datos, solamente es necesario actualizar la capa de abstracción de la base de datos (15). Por tanto, garantiza la independencia de la

base de datos utilizada. La capa Vista aprovecha la separación del código, separa el código específico del común a todas las interfaces mediante el uso de layouts y plantillas. Normalmente, el layout es global en toda la aplicación y la plantilla es la presentación de los datos de la acción que se está ejecutando. En la capa del controlador Symfony define un controlador frontal, único para cada aplicación, encargado del manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Se definen además acciones que incluyen el código específico de cada página.

### **3.2.3 Descripción de los patrones de diseño**

El principio más importante de la arquitectura MVC es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. La programación se puede simplificar si se utilizan patrones de diseño. De esta forma, las capas del modelo, la vista y el controlador se pueden subdividir en más capas. (15)

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares (16). Symfony mejora la calidad de las aplicaciones web mediante la aplicación de buenas prácticas y brinda la posibilidad de utilizar varios patrones de diseño. A continuación se describen los patrones GRASP utilizados en la implementación de la aplicación.

#### **Experto**

Este patrón asigna una responsabilidad al experto en información, es decir, la clase que tiene la información necesaria para cumplir con la responsabilidad. El problema que resuelve el patrón experto está referido al principio más básico mediante el cual las responsabilidades son asignadas en el diseño orientado a objetos. (17)

Symfony permite la abstracción a la base de datos y el acceso a los datos, garantizando la independencia de la base de datos utilizada. Las clases del modelo son generadas automáticamente por Symfony y quien se encarga de generarlas es la librería Propel. A partir de la tabla denominada Report (ver Fig. 15) se crean las clases: Report, BaseReport, ReportPeer y BaseReportPeer. Las clases BaseReport y BaseReportPeer son las responsables de efectuar diferentes acciones sobre esta parte de la información contenida en la BD.

### **Creador**

El patrón creador identifica el responsable de la instanciación de objetos o clases, establece que una instancia de un objeto la tiene que crear el objeto que tiene la información para ello. El empleo de este patrón se evidencia al establecer en las clases Actions, donde son definidas las acciones de cada módulo, la creación de objetos que responden a las clases del modelo que responden a dicho módulo.

### **Alta cohesión**

Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme, Symfony permite asignar responsabilidades con una alta cohesión. En la solución se asignan responsabilidades con alta cohesión, las clases Actions tienen la responsabilidad de definir las acciones para las plantillas, responsabilidad que no comparte con otras clases, el uso de este patrón posibilita que el software sea flexible frente a grandes cambios.

### **Controlador**

Asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Este patrón se pone de manifiesto en el manejo de las peticiones web, realizado por el controlador frontal, punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la dirección URL entrada por el usuario.

### **Bajo acoplamiento**

El acoplamiento es una medida de la fuerza con que una clase está relacionada a otras clases, este patrón logra que una clase no dependa de muchas otras, fortaleciendo la flexibilidad y la reutilización de código. El uso de este patrón se evidencia en las clases Action, las cuales heredan solamente de la clase sfActions para evitar la dependencia entre varias clases.

## **3.3 Modelo de diseño**

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso junto con otras restricciones relacionadas con el entorno de implementación. Sirve de abstracción a la implementación y es utilizado como una entrada fundamental a las actividades de este flujo de trabajo.

### 3.3.1 Diagramas de clases del diseño

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación; muestran un conjunto de interfaces, clases, colaboraciones y relaciones. A continuación se muestran los diagramas de clases del diseño de los casos de uso descritos en el capítulo 2:

#### Diagrama de clases del diseño: CUS Administrar reporte

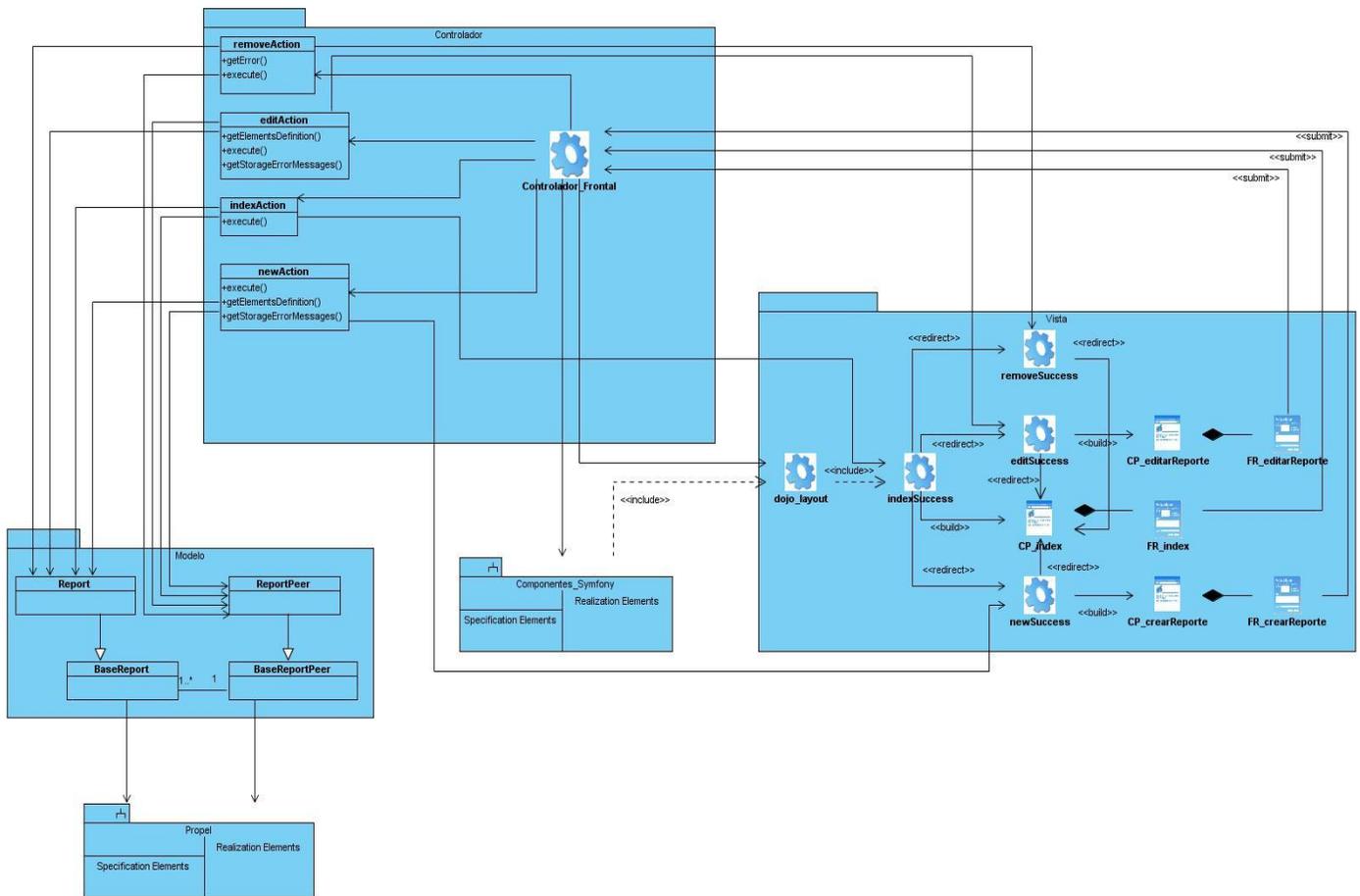


Fig. 5: Diagrama de clases del diseño: CUS Administrar reporte.



Diagrama de clases del diseño: CUS Generar informe

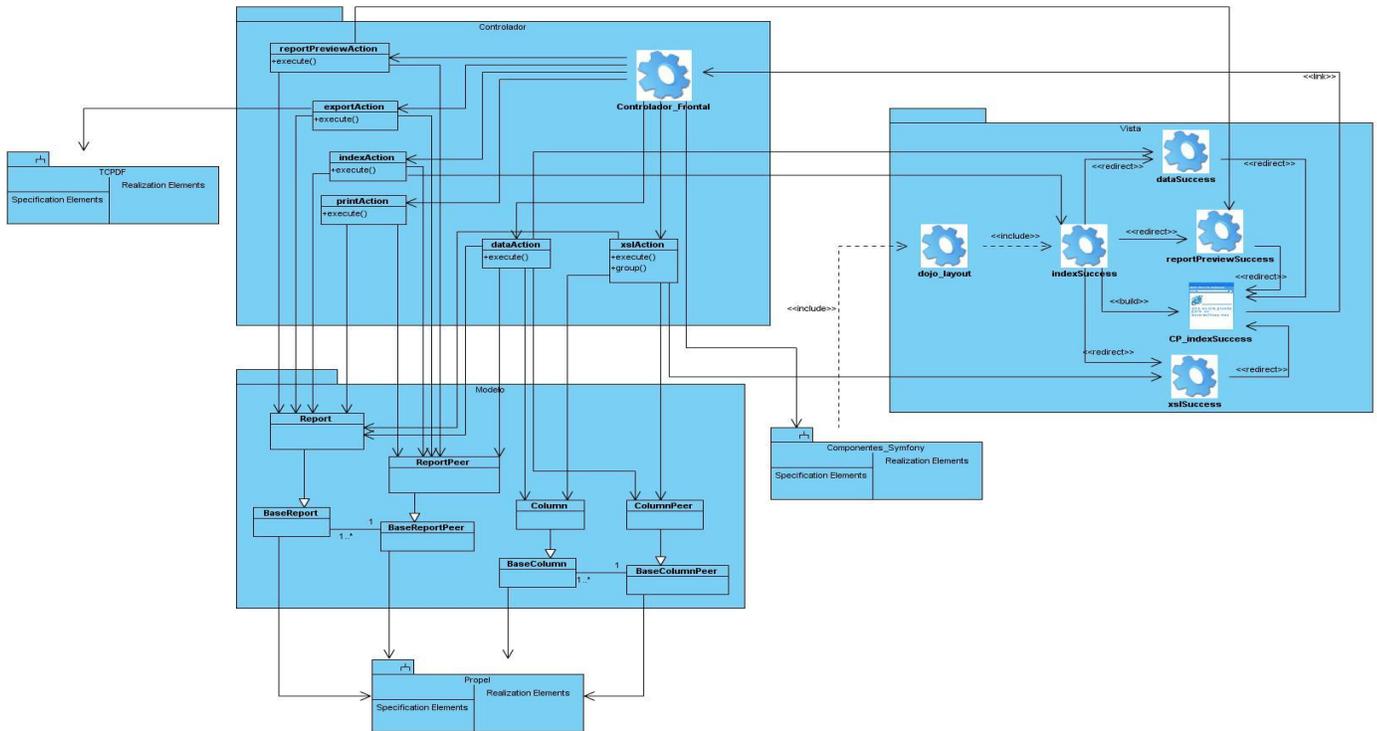


Fig. 8: Diagrama de clases del diseño: CUS Generar informe.

Diagramas de secuencia

Los diagramas de secuencia muestran las interacciones entre los objetos organizadas en una secuencia temporal. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados. (18)

Seguidamente se muestran los diagramas de secuencia de los CUS Administrar reporte, Buscar y visualizar reporte y Generar informe, exponiendo la interacción de los objetos y los mensajes intercambiados para cada escenario de estos casos de uso:

**Diagrama de secuencia CUS Administrar reporte (Escenario Crear reporte)**

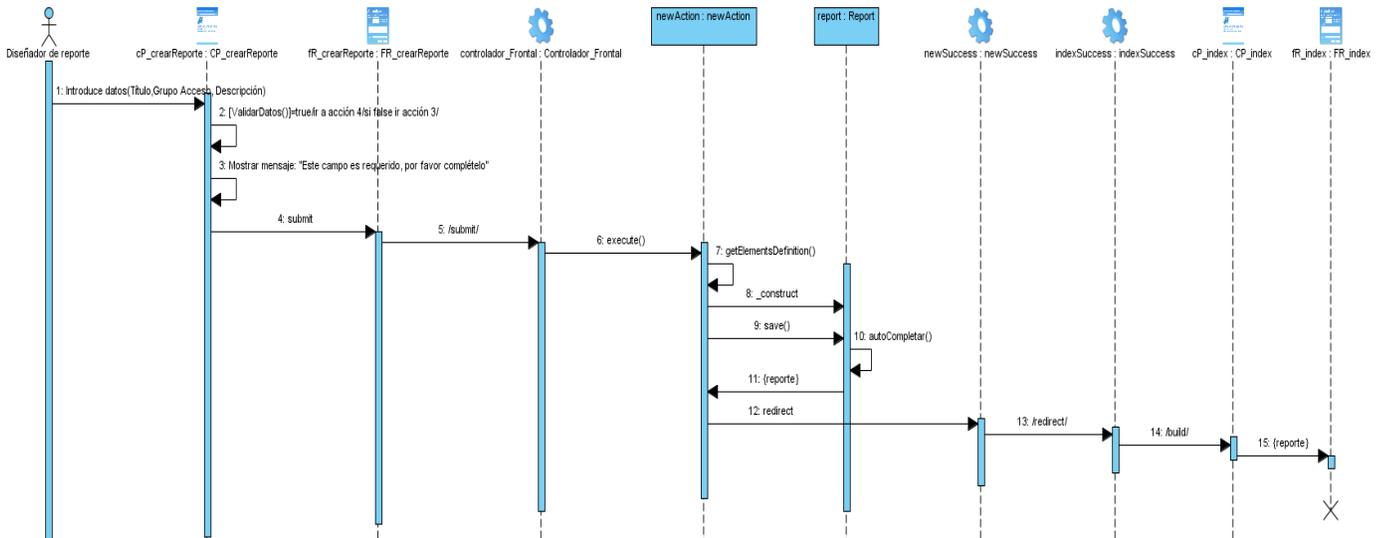


Fig. 9: Diagrama de secuencia CUS Administrar reporte (Escenario Crear reporte).

**Diagrama de secuencia CUS Administrar reporte (Escenario Modificar reporte)**

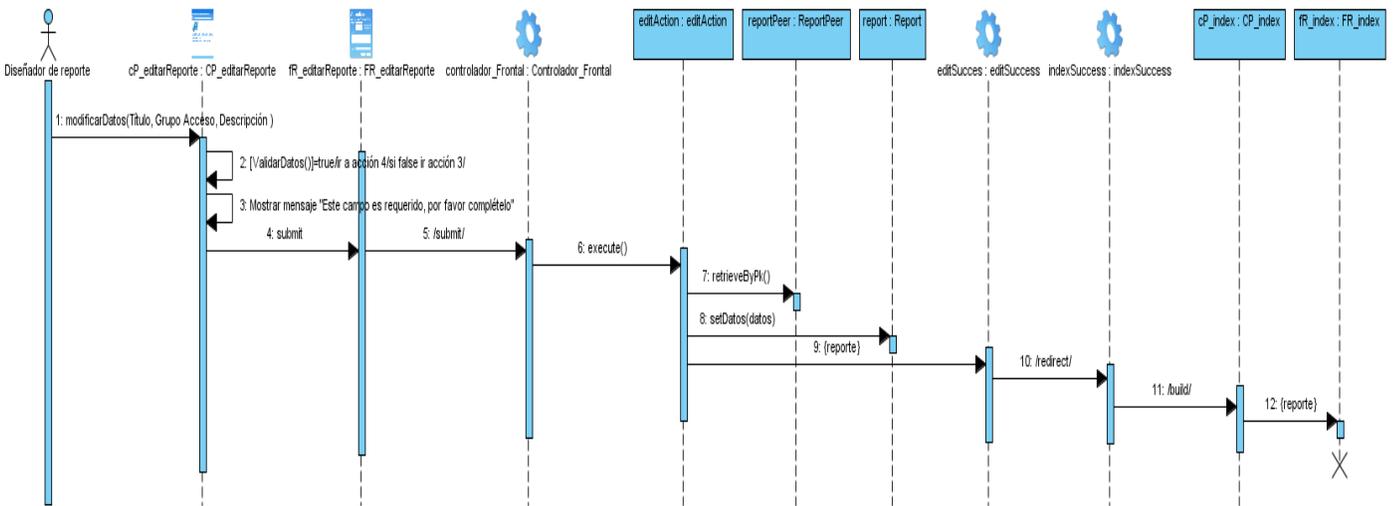


Fig. 10: Diagrama de secuencia CUS Administrar reporte (Escenario Modificar reporte).

**Diagrama de secuencia CUS Administrar reporte (Escenario Eliminar reporte)**

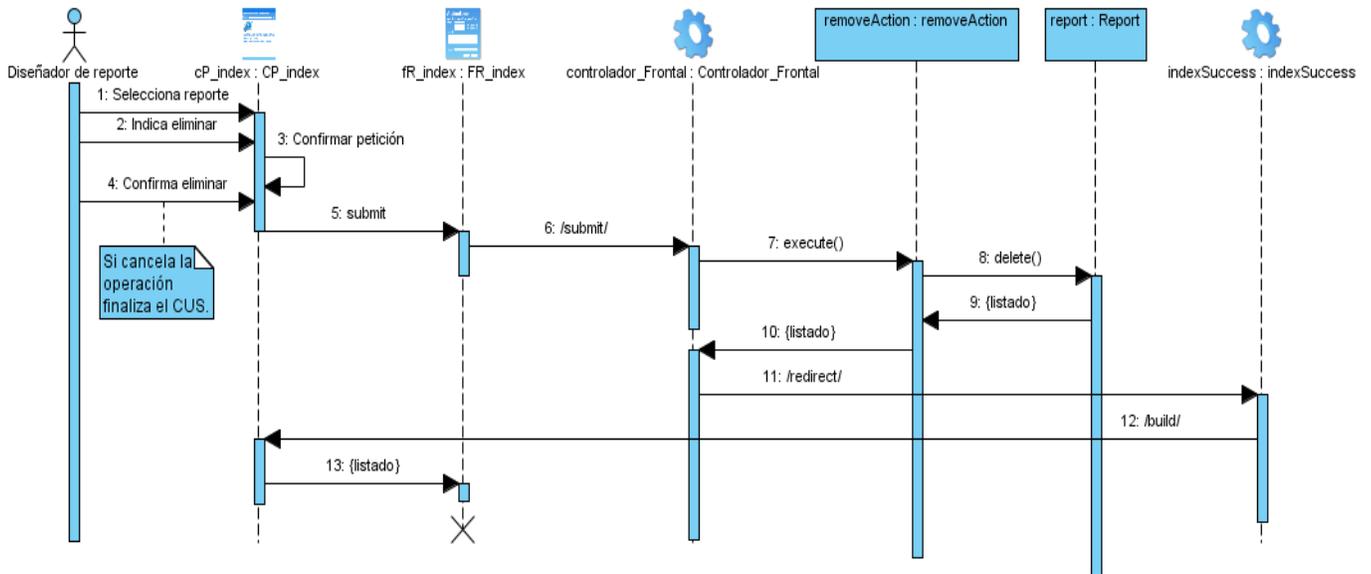


Fig. 11: Diagrama de secuencia CUS Administrar reporte (Escenario Eliminar reporte).

**Diagrama de secuencia CUS Buscar y visualizar reporte**

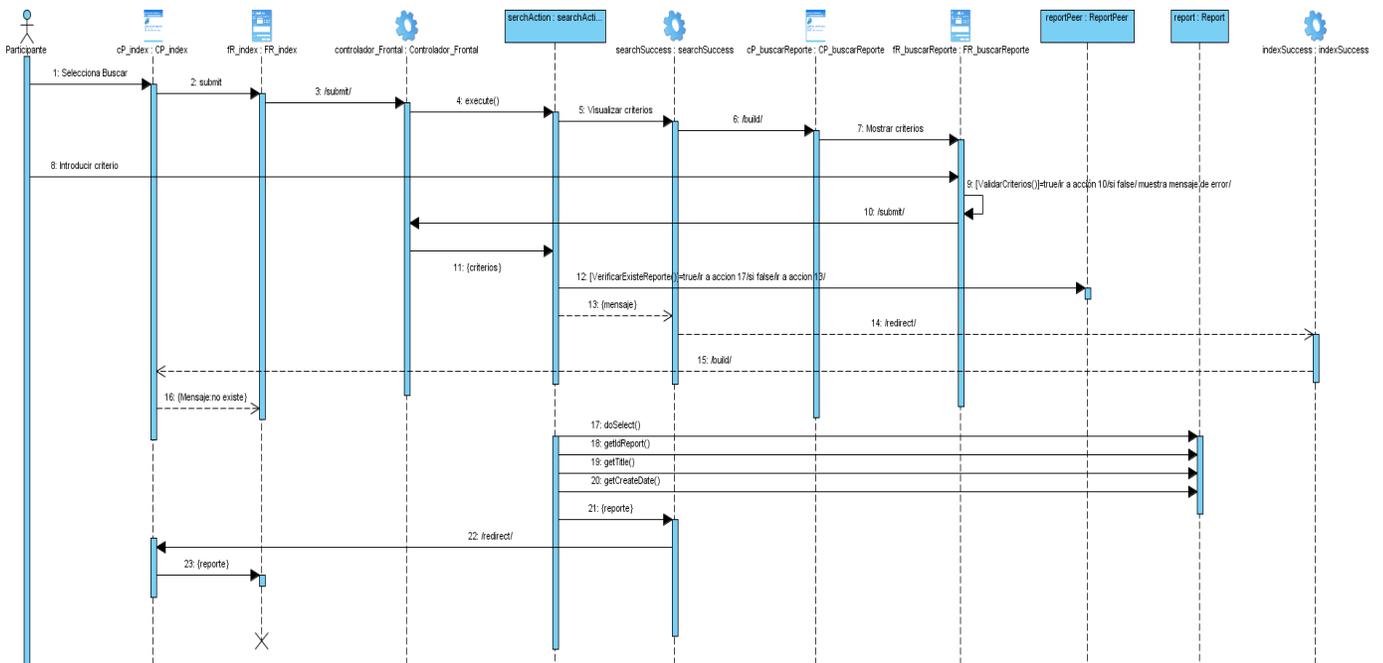


Fig. 12: Diagrama de secuencia CUS Buscar y visualizar reporte.

**Diagrama de secuencia CUS Generar informe**

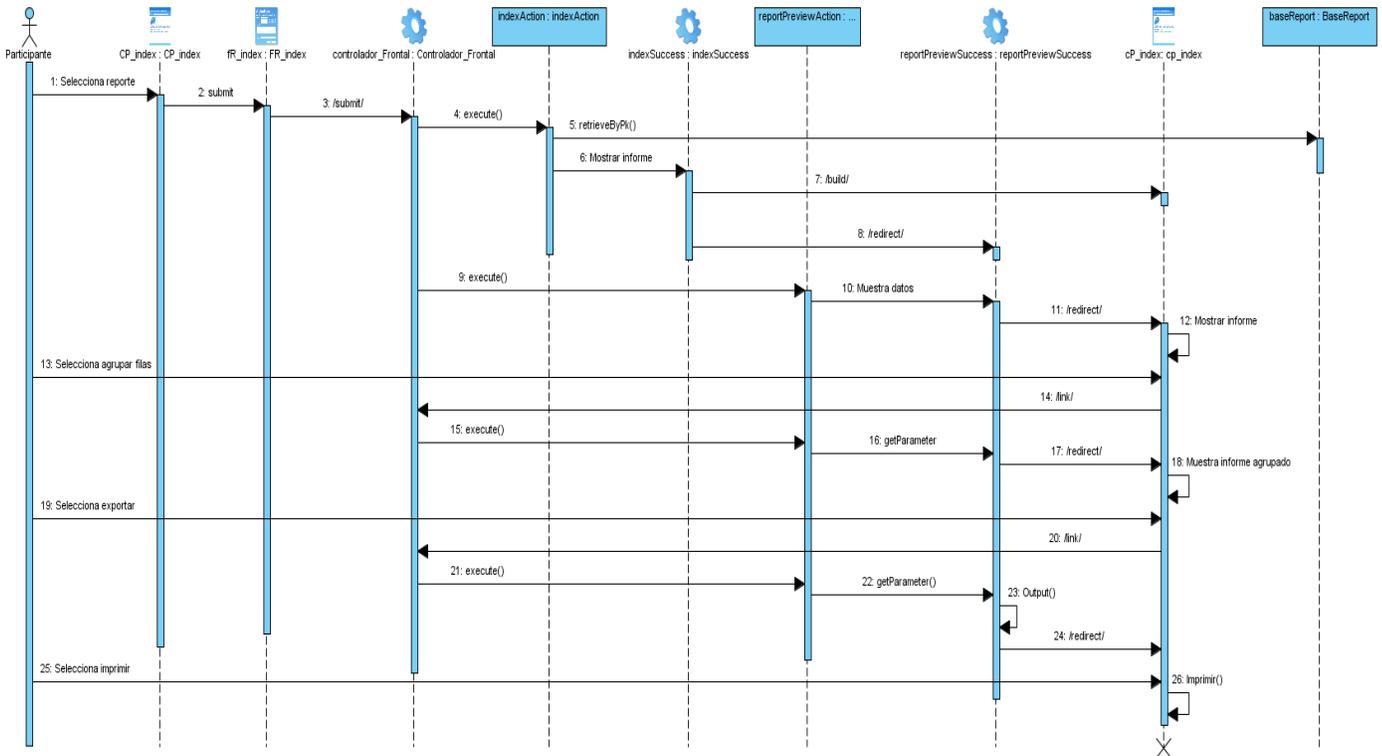


Fig. 13: Diagrama de secuencia CUS Generar informe.

**3.4 Diseño de la base de datos**

Una de las actividades cruciales en la construcción de una aplicación que maneje abundante información es, sin dudas, el diseño de la base de datos. La BD necesita de una definición de su estructura que le permita almacenar datos, reconocer el contenido, y un acceso y recuperación eficiente a la información con redundancia mínima. Para lograrlo se requiere desarrollar un conjunto de pasos bien definidos que guían el diseño de la base de datos, a partir de los cuales se obtienen dos importantes artefactos: diagrama de clases persistentes y diagrama entidad-relación.

**3.4.1 Diagrama de clases persistentes**

Todas las clases identificadas en el diseño no son persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo, es responsabilidad del diseñador identificar estas clases. El diagrama de clases persistentes captura gráficamente las clases persistentes identificadas:

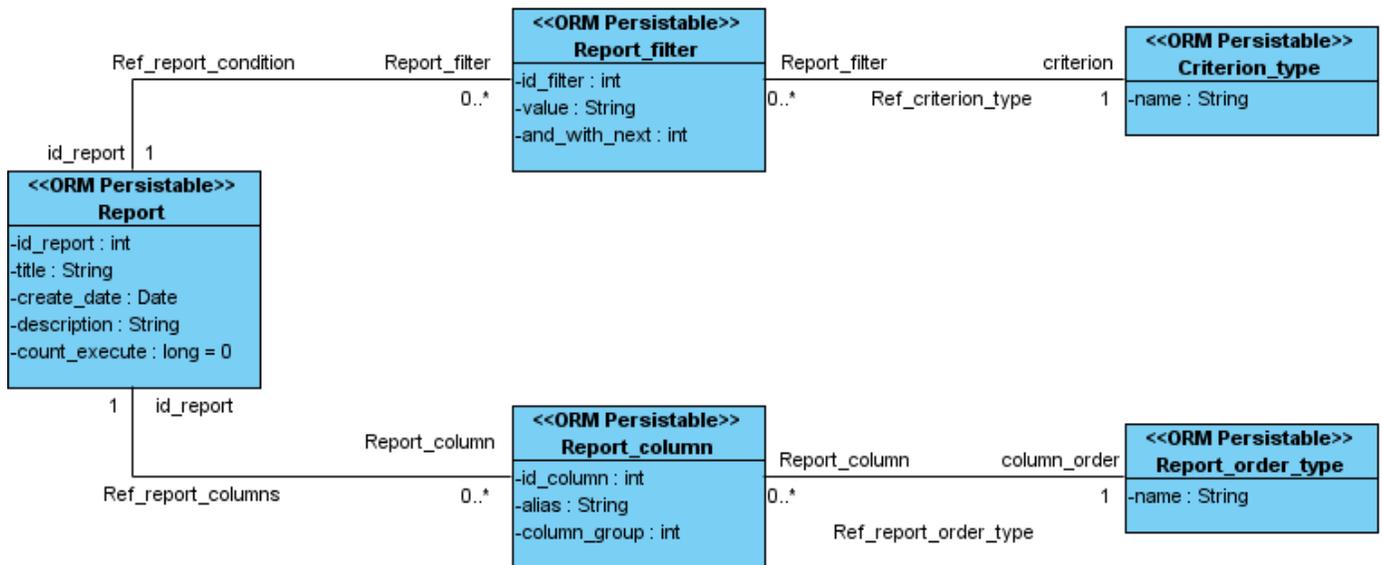


Fig. 14: Diagrama de clases persistentes.

### 3.4.2 Modelo de datos

El Modelo de datos describe las tablas que representan las distintas entidades que pertenecen al dominio del problema y serán almacenadas en la base de datos. La siguiente figura muestra el diagrama entidad-relación:

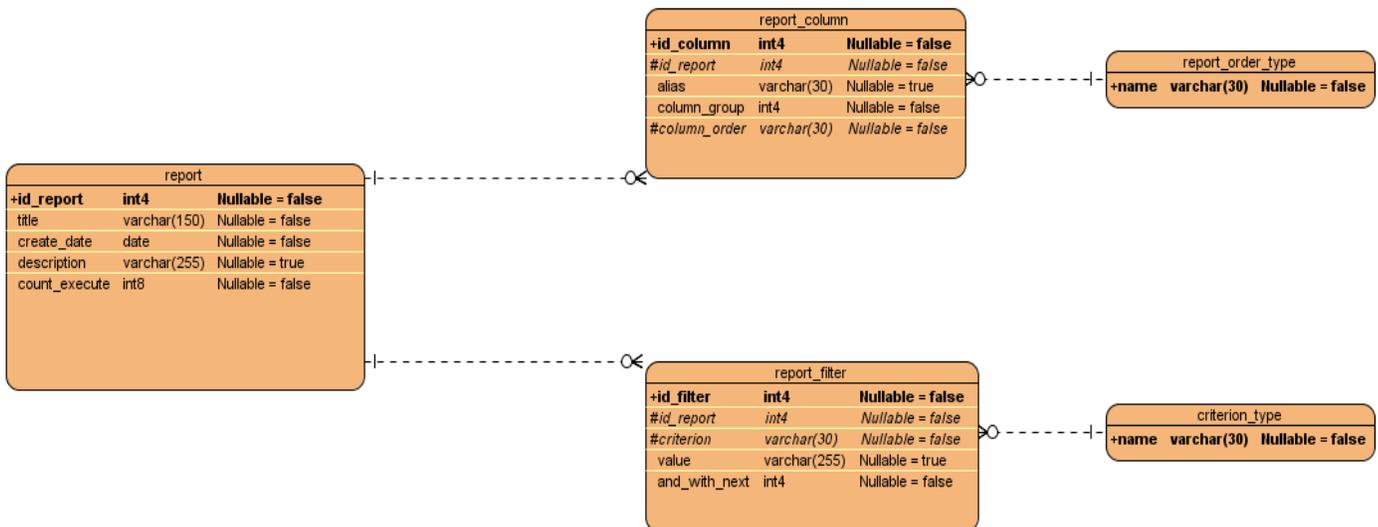


Fig. 15: Diagrama entidad-relación.

### 3.5 Diagrama de despliegue

El diagrama de despliegue representa la distribución de los nodos que forman parte de la arquitectura sobre la que se ejecuta el sistema a través de sus componentes.

El diagrama de despliegue para el Ambiente integrado para el modelado y ejecución de procesos de gestión de la información representa 4 nodos principales. El nodo procesador PC Cliente, el nodo procesador Servidor de Aplicaciones, el nodo procesador Servidor de Base de datos y el nodo dispositivo Impresora.



Fig. 16: Diagrama de despliegue.

El nodo PC Cliente estará conectado al nodo dispositivo Impresora mediante el protocolo de comunicación Universal Serial Bus (USB) y a su vez estará conectado mediante el Protocolo de Transferencia de Hipertexto (HTTP) al nodo procesador que representa al Servidor de Aplicaciones. La conexión entre el Servidor de Aplicaciones y el Servidor de Base de datos se realizará mediante el protocolo de comunicación TCP/IP.

### 3.6 Conclusiones

El desarrollo de este capítulo permitió establecer una arquitectura que da soporte a la implementación y que responde a los requisitos funcionales y no funcionales establecidos. Los patrones de diseño utilizados facilitaron el desarrollo de un sistema flexible a futuros cambios y abierto a la reutilización de código. Se definieron las clases que se corresponden con el diseño del sistema, se muestran los diagramas de clases del diseño y los diagramas de interacción para dar validez al diseño y brindar la entrada principal para la implementación de la herramienta. Se realizó el diseño de la base de datos y la distribución física del sistema.

## Capítulo 4: Implementación y Prueba

---

### CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

En este capítulo se describe cómo los elementos del modelo de diseño se implementan en términos de componentes y contiene los diagramas de componentes para cada caso de uso. También incluye las pruebas funcionales aplicadas al sistema implementado para examinar la estructura externa del software y garantizar la calidad del mismo.

El flujo de trabajo de implementación inicia con el resultado del diseño e implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El objetivo principal de esta etapa es desarrollar la arquitectura y el sistema como un todo. De forma más específica, los propósitos de la implementación son definir la organización del código, planificar las integraciones de sistema necesarias en cada iteración e implementar las clases y subsistemas encontrados durante el diseño.

#### 4.1 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros.

##### 4.1.1 Diagramas de componentes

Los diagramas de componentes conforman junto con otros artefactos generados en este flujo de trabajo el modelo de implementación, al describir los componentes a construir, su organización y dependencias. Un componente es una parte física y reemplazable de un sistema que se conforma con un conjunto de interfaces y proporciona la realización de dicho conjunto. Se usan para modelar los elementos físicos que pueden hallarse en un nodo por lo que empaquetan elementos como clases, colaboraciones e interfaces. A continuación se muestran los diagramas de componentes:

# Capítulo 4: Implementación y Prueba

## Diagrama de componentes CUS Administrar reporte

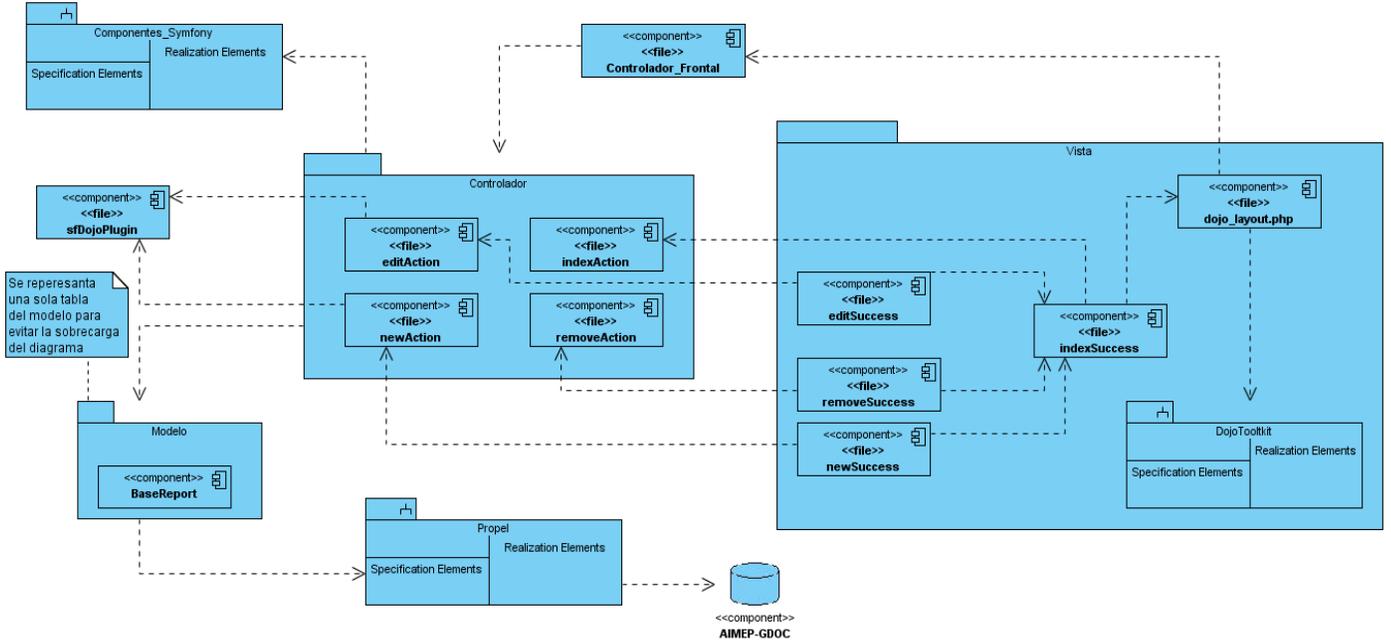


Fig. 17: Diagrama de componentes CUS Administrar reporte.

## Diagrama de componentes CUS Administrar diseño de reporte

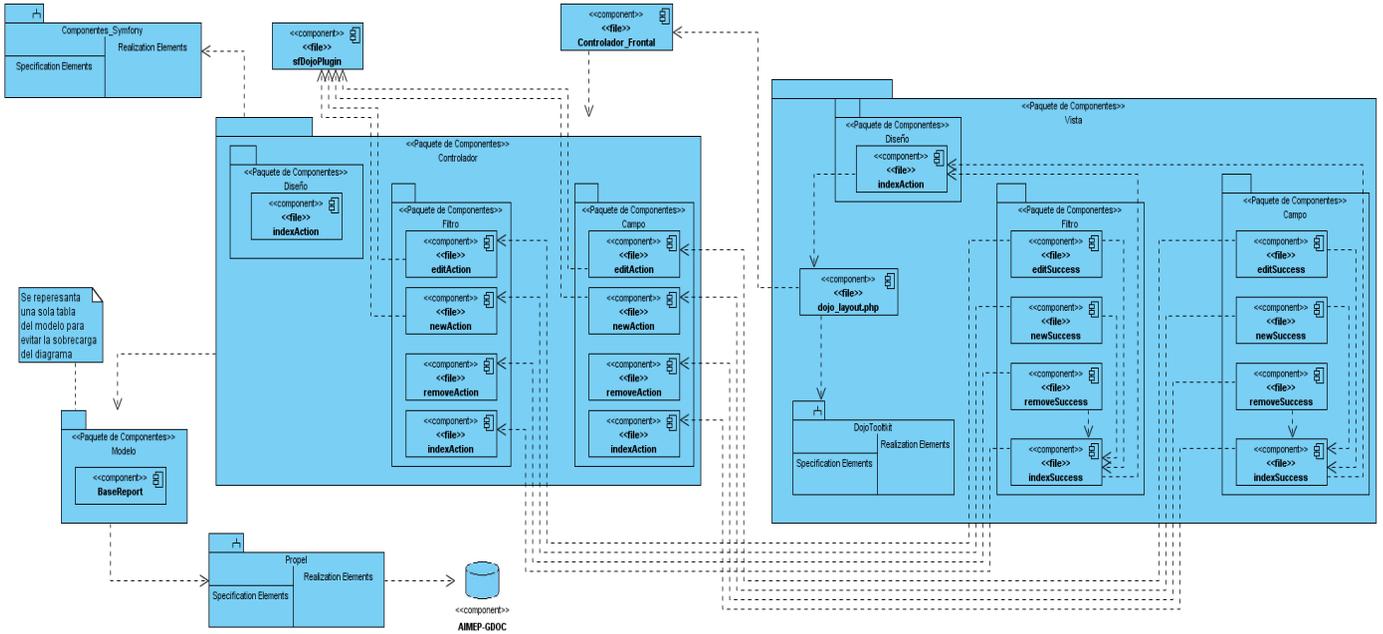


Fig. 18: Diagrama de componentes CUS Administrar diseño de reporte.

# Capítulo 4: Implementación y Prueba

## Diagrama de componentes CUS Buscar y visualizar reporte

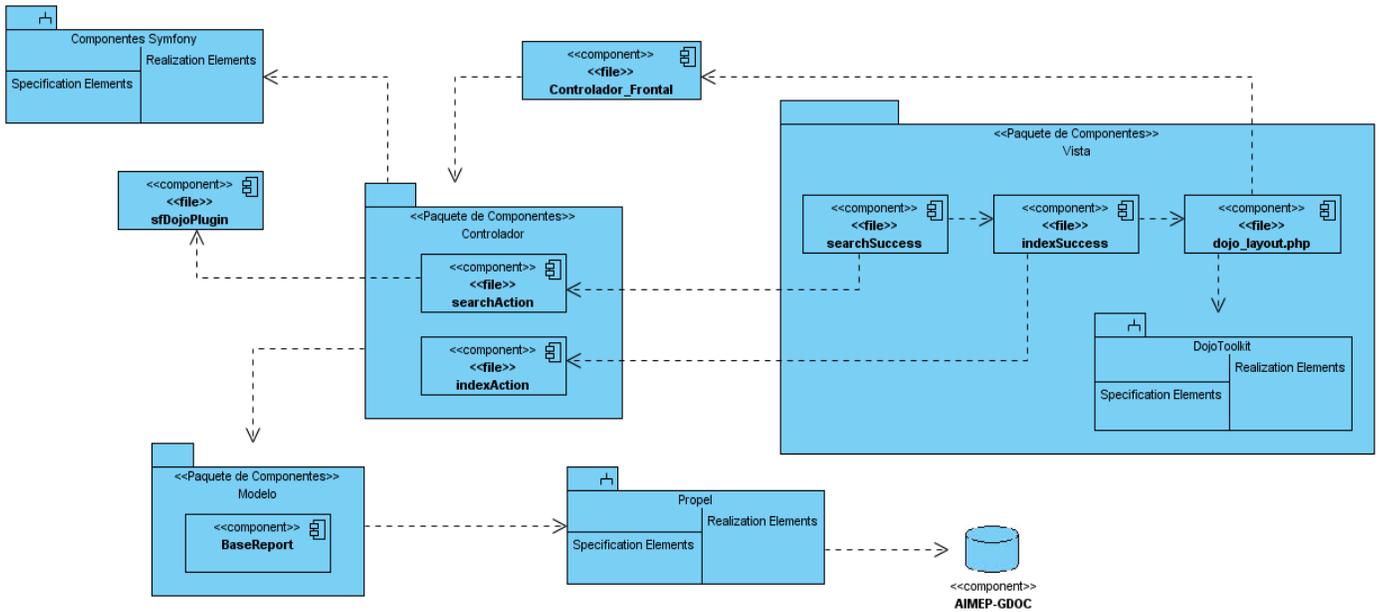


Fig. 19: Diagrama de componentes CUS Buscar y visualizar reporte.

## Diagrama de componentes CUS Generar informe

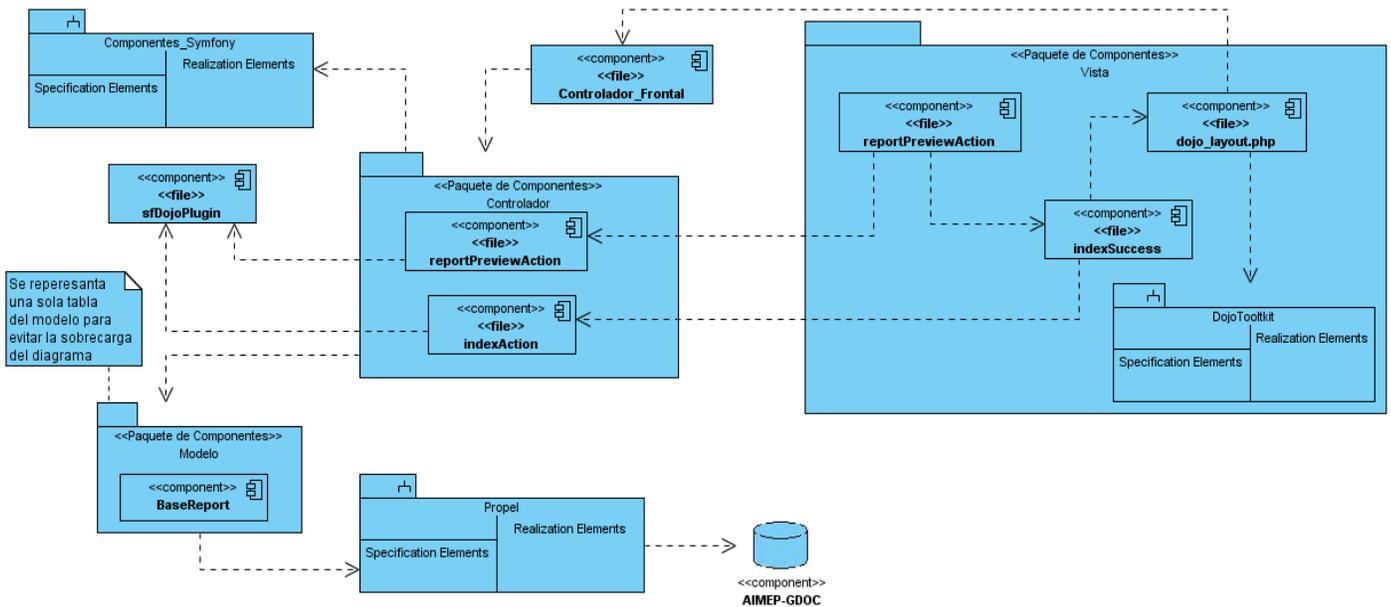


Fig. 20: Diagrama de componentes CUS Generar informe.

## Capítulo 4: Implementación y Prueba

### 4.2 Código fuente

La implementación de componentes es una actividad crítica desarrollada durante este flujo de trabajo, con el propósito de completar una parte de la implementación que podría ser entregada para la integración. En esta actividad el programador escribe código fuente, reutiliza código, compila e implementa los elementos del modelo de diseño. El siguiente fragmento de código es una función JavaScript que permite eliminar el reporte seleccionado del listado de reportes de la interfaz principal. Cuando el usuario selecciona el reporte que desea eliminar, oprime la opción eliminar. Esta función valida que, antes de eliminar, el usuario haya seleccionado el reporte, en este caso lanza el mensaje de error “Antes debe seleccionar una fila”.

```
removeReportExecute = function() {
```

```
// Función para procesar el resultado de la página ejecutada.
```

```

removeReportCallback = function(response){
  if (response.status == 'success'){
    DyGDataGrid_Report.removeSelectedRows();
  } else {
    dojoy.showTooltipError(
      response.error,          // Texto del error.
      DiFB_REMOVE_Report.domNode, // Objeto relacionado con el mensaje.
      ['above', 'below']      // Posición del mensaje.
    );
  }
}
// Obtiene los valores de la fila seleccionada.
var values = DyGDataGrid_Report.getDataSelected();
if (values.length != 0) {

  if ( confirm( '<?=__('CONFIRM_REMOVE_SELECTED_ITEM)?>' ) ) {
    // Si la confirmación es aceptada.
    dojoy.remoteExecuteInJson("<?=$url_remove?>", values, removeReportCallback);
  }
} else {
  // Si no existe una fila seleccionada lanza mensaje de error.
  dojoy.grid.firstSelectRow( DiFB_REMOVE_Report.domNode );
}
}

```

## Capítulo 4: Implementación y Prueba

### 4.3 Pantallas principales de la aplicación

La herramienta consta de una interfaz principal intuitiva y amigable, estructurada en tres áreas fundamentales. En el área central se muestra un listado de los reportes disponibles para la generación de informes y otras operaciones que pueden ser realizadas sobre estos. A la izquierda se encuentra un menú desplegable con opciones de acceso rápido para una mejor navegación web. El área superior contiene la barra de herramientas, la cual tiene como propósito facilitar el acceso a las distintas utilidades de la aplicación. Esta barra brinda las funcionalidades básicas que permiten la gestión de reportes: actualizar, añadir, editar, eliminar y diseñar como se muestra en la figura:

Código	Título	Entidad Base	Fecha de creación	Creador	Grupo con acceso
11	Datos de Trabajadores	Desempeño Laboral	18/05/10	Administrador	ProcessDesigners
12	Reporte2	Desempeño Laboral	18/05/10	Oscar Reyes	Administrators
13	Reporte3	Desempeño Laboral	18/05/10	Keyler Rodriguez Velazquez	ProcessDesigners

Fig. 21: Herramienta web para el diseño de reportes. Interfaz principal.

El área de trabajo para el diseño y edición de los reportes permite gestionar los elementos de diseño para la creación personalizada de los reportes. La barra de herramientas superior ofrece funcionalidades que posibilitan seleccionar los tipos de elementos de diseño que requiere el reporte así como generar el informe del reporte en diseño o edición. La barra de herramientas inferior permite añadir, insertar y eliminar los elementos correspondientes al tipo de elemento de diseño seleccionado como se muestra en la figura:

# Capítulo 4: Implementación y Prueba

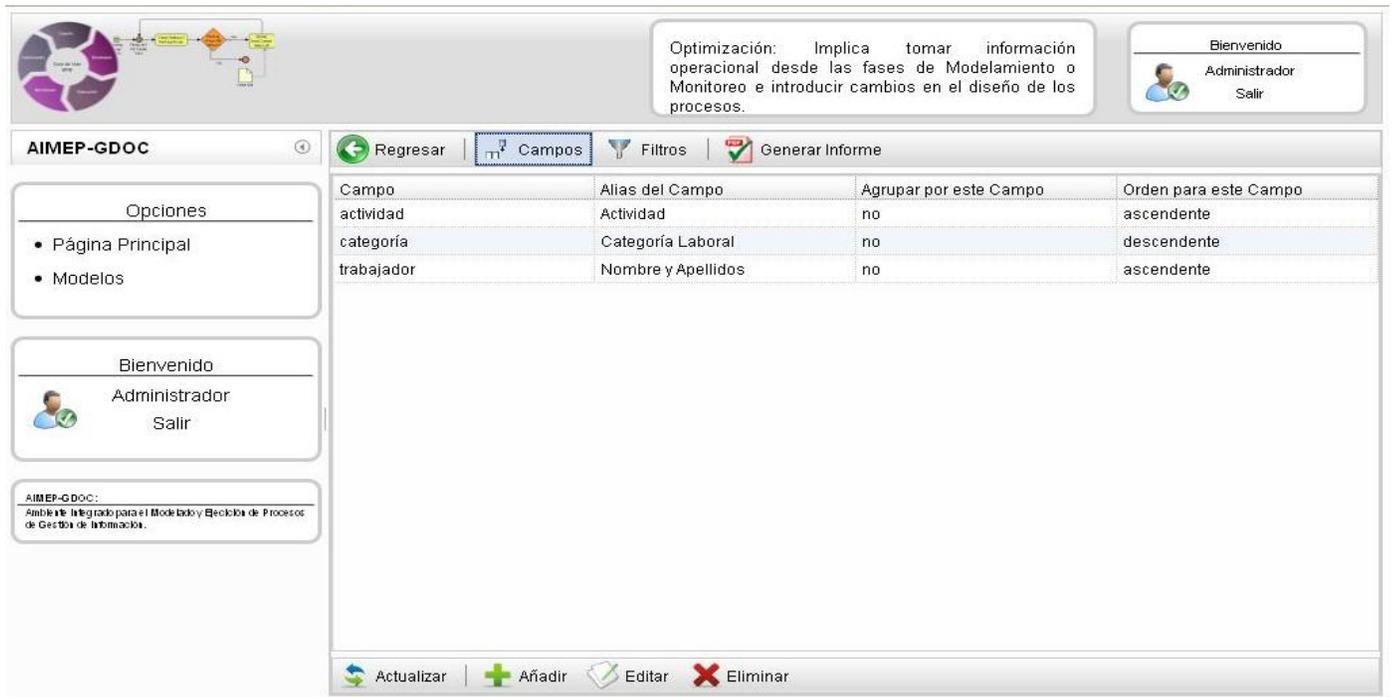


Fig. 22: Herramienta web para el diseño de reportes. Área de trabajo para el diseño.

El área de trabajo para la generación de informes muestra detalladamente en la región central el resultado de determinado reporte y una barra de herramientas en la parte superior que permite agrupar filas, imprimir y exportar el informe visualizado en formato PDF y XLS.

# Capítulo 4: Implementación y Prueba

Ejecución: Aplica alguna herramienta de software para aplicar los pasos requeridos de los procesos.

Bienvenido  
Administrador  
Salir

Regresar Actualizar Exportar PDF Exportar XLS Imprimir Agrupar filas

Nombre y Apellidos	Categoría Laboral	Actividad
Agustín Cabeza Compostizo	Trabajo Metodológico	Realización de la actividad 45
		Realización de la actividad 59
	Superación	Realización de la actividad 57
		Realización de la actividad 58
	Producción	Realización de la actividad 50
	Formación	Realización de la actividad 50
Realización de la actividad 75		
Alberto Comino Grande	Trabajo Metodológico	Realización de la actividad 15
		Realización de la actividad 85
		Realización de la actividad 95
	Superación	Realización de la actividad 52
	Producción	Realización de la actividad 45
		Realización de la actividad 51

Fig. 23: Herramienta web para el diseño de reportes. Área de trabajo para la generación de informes.

## 4.4 Prueba

Durante esta etapa cada construcción generada durante la implementación es sometida a diferentes pruebas para validar el producto software. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación (19). En todas las fases de desarrollo hay que probar el software que se va construyendo, aunque en la fase de construcción se centran los mayores esfuerzos de este flujo.

La prueba de funcionalidad se enfoca en los requerimientos para verificar que se corresponden directamente a casos de uso o funciones y reglas del negocio. Los objetivos de estas pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. Esta técnica de prueba se basa en el método de caja negra, consiste en verificar la aplicación y sus procesos interactuando con la aplicación por medio de la interfaz de usuario y analizar los resultados obtenidos. Para la obtención de todo el conjunto de condiciones de entrada que servirán para ejercitar los requisitos funcionales del programa se diseñaron los casos de prueba correspondientes.

## Capítulo 4: Implementación y Prueba

Luego de aplicar la prueba anteriormente descrita se detectaron 4 no conformidades, las cuales fueron resueltas satisfactoriamente en el tiempo esperado. A continuación se muestra la tabla de Registro de defectos y no conformidades:

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No significativa	Estado NC	Resp. equipo de desarrollo
Aplicación	1	En el CUS Buscar y visualizar reporte, cuando se realiza una búsqueda que no existe el sistema no muestra el mensaje indicado.	<a href="http://10.36.18.4:5800/ReportDesigner.php">http://10.36.18.4:5800/ReportDesigner.php</a>	Pruebas de funcionalidad.		X	PD 20/4/10 Resuelta 25/4/10	Se muestra el mensaje.
Aplicación	2	En la sección Insertar elemento de diseño filtro del CUS Administrar diseño de reporte cuando se	<a href="http://10.36.18.4:5800/ReportDesigner.php">http://10.36.18.4:5800/ReportDesigner.php</a>	Pruebas de funcionalidad.		X	PD 25/4/10 Resuelta 26/4/10	Se muestra el mensaje esperado.

## Capítulo 4: Implementación y Prueba

		deja el campo criterio vacío, no se muestra el mensaje esperado.						
Aplicación	3	En la sección Modificar reporte del CUS Administrar reporte el campo Entidad base permite ser modificado.	<a href="http://10.36.18.4:5800/ReportDesigner.php">http://10.36.18.4:5800/ReportDesigner.php</a>	Pruebas de funcionalidad.	X		PD 2/5/10 Resuelta 4/5/10	El campo Entidad base no permite ser modificado.
Aplicación	4	En la sección Exportar informe del CUS Generar informe el sistema no genera el documento en formato PDF.	<a href="http://10.36.18.4:5800/ReportDesigner.php">http://10.36.18.4:5800/ReportDesigner.php</a>	Pruebas de funcionalidad.	X		PD 5/5/10 Resuelta 7/5/10	Genera el informe en formato PDF.

Tabla 1: Registro de defectos y no conformidades.

## Capítulo 4: Implementación y Prueba

---

### 4.5 Conclusiones

El desarrollo de este capítulo permitió implementar el sistema en términos de componentes, desarrollar la arquitectura y el producto software como un todo, definir la organización del código e implementar las clases identificadas en el diseño. Durante el ciclo de desarrollo de software cada construcción generada durante la implementación fue sometida a pruebas funcionales con el objetivo de demostrar que las funciones del producto software son completamente operativas y validar la solución propuesta, las cuales arrojaron resultados satisfactorios.

## CONCLUSIONES

- Se identificaron las funcionalidades de la herramienta para el diseño de reportes permitiendo definir el ámbito del sistema.
- Se realizó el diseño de la herramienta para el diseño de reportes permitiendo establecer una arquitectura que da soporte a la implementación y que responde a los requisitos funcionales y no funcionales establecidos.
- Se implementó una herramienta web flexible y orientada al usuario, que posibilita el diseño personalizado de los reportes a partir de los modelos de recogida de datos.

# Recomendaciones

---

## RECOMENDACIONES

- Continuar el desarrollo de la herramienta con el objetivo de incorporar otras funcionalidades que permitan el diseño de reportes de mayor complejidad.
- Independizar la herramienta de la fuente y estructura de los datos para lograr un generador de reportes de propósito general y adaptable a las necesidades de cualquier empresa o institución.

## Referencias Bibliográficas

---

### REFERENCIAS BIBLIOGRÁFICAS

1. SQLMax Connections. [En línea] [Citado el: 15 de Octubre de 2009.] [http://www.sqlmax.com/reportin\\_services1.asp](http://www.sqlmax.com/reportin_services1.asp).
2. GrapeCity. [En línea] [Citado el: 05 de Diciembre de 2009.] <http://www.datadynamics.com/Products/ProductOverview.aspx?Product=ActiveReports>.
3. PHP Report Maker. [En línea] [Citado el: 27 de Marzo de 2010.] <http://www.hkvstore.com/phpreportmaker/>.
4. Tecnológico. [En línea] [Citado el: 16 de Diciembre de 2009.] <http://www.mitecnologico.com/Main/AnalisisProcesoCicloVidaSoftware>.
5. Universidad EAFIT Ingeniería Matemática. [En línea] [Citado el: 16 de Noviembre de 2009.] <http://ingenieria-matematica.eafit.edu.co/recursos/glosario/c.html>.
6. EcuRed. [En línea] [Citado el: 20 de Diciembre de 2009.] <http://www.ecured.cu/index.php/Framework>.
7. UTN - Facultad Regional La Plata. [En línea] [Citado el: 20 de Diciembre de 2009.] [http://www.frlp.utn.edu.ar/materias/info2/bases\\_de\\_datos.htm](http://www.frlp.utn.edu.ar/materias/info2/bases_de_datos.htm).
8. HTTP - Perú. [En línea] [Citado el: 20 de Diciembre de 2009.] <http://www.http-peru.com/postgresql.php>.
9. Microsoft Small Basic. [En línea] [Citado el: 30 de Marzo de 2009.] <http://programacion.fundaciondelcamino.org/wp-content/uploads/2009/10/Conceptos-de-Programaci%C3%B3n.pdf>.
10. EcuRed. [En línea] [Citado el: 29 de Noviembre de 2009.] <http://www.ecured.cu/index.php/JavaScript>.
11. Ecured. [En línea] [Citado el: 1 de Diciembre de 2009.] <http://www.ecured.cu/index.php/AJAX>.
12. **Eguíluz Pérez, Javier.** *Introducción a AJAX*. pág. 194.
13. Linux Para Todos. [En línea] [Citado el: 2009 de Diciembre de 10.] <http://www2.linuxparatodos.net/web/servidores/base-de-conocimiento/-/wiki/Main/Servidor+Web;jsessionid=FA33F05A33564D0FA82D0914C7B522C2>.
14. **Övergaard Gunnar, Palmkvist Karin.** *Use Cases Patterns and Blueprints*. s.l. : Addison Wesley Professional, 2004. Vols. Part III: Use-Case Patterns.
15. LibrosWeb.es. [En línea] [Citado el: 02 de Abril de 2010.] [http://www.librosweb.es/symfony\\_1\\_2/capitulo2/el\\_patron\\_mvc.html](http://www.librosweb.es/symfony_1_2/capitulo2/el_patron_mvc.html).
16. msdn. [En línea] [Citado el: 10 de Abril de 2010.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

## Referencias Bibliográficas

---

17. **Erick Salazar, Anaís Aponte.** Sistemas de Programas. [En línea] [Citado el: 15 de Marzo de 2010.]  
<http://www ldc.usb.ve/~teruel/ci3711/patron3a/index.html>.
18. UML. [En línea] [Citado el: 31 de Marzo de 2010.]  
[http://www.neuronsrl.com.ar/training/uml/uml\\_secuencia.html](http://www.neuronsrl.com.ar/training/uml/uml_secuencia.html).
19. Windows Live. [En línea] [Citado el: 5 de Abril de 2010.]

---

**BIBLIOGRAFÍA**

- Ajax.org. [En línea] [Citado el: 20 de Noviembre de 2009.] <http://www.ajax.org>.
- **Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M. Pattern.** *Oriented Software Architecture. A System of Patterns.* Inglaterra : Hardcover, 1996.
- EcuRed. [En línea] [Citado el: 1de Diciembre de 2009.] <http://www.ecured.cu/index.php/AJAX>.
- EcuRed. [En línea] [Citado el: 20 de Diciembre de 2009.] <http://www.ecured.cu/index.php/Framework>.
- EcuRed. [En línea] [Citado el: 29 de Noviembre de 2009.] <http://www.ecured.cu/index.php/JavaScript>.
- **Eguíluz Pérez, Javier.** *Introducción a AJAX.* pág. 194.
- **Erick Salazar, Anaís Aponte.** Sistemas de Programas. [En línea] [Citado el: 15 de Marzo de 2010.] <http://www ldc.usb.ve/~teruel/ci3711/patron3a/index.html>.
- GrapeCity. [En línea] [Citado el: 05 de Diciembre de 2009.] <http://www.datadynamics.com/Products/ProductOverview.aspx?Product=ActiveReports>.
- HTTP - Perú. [En línea] [Citado el: 20 de Diciembre de 2009.] <http://www.http-peru.com/postgresql.php>.
- Itescam. [En línea] [Citado el: 5 de Abril de 2010.] <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r56673.PDF>.
- **Jacobson, I. y Booch, G. y Rumbaugh, J.** *El Proceso Unificado de Desarrollo de software.* 2000.
- LibrosWeb.es. [En línea] [Citado el: 02 de 04 de 2010.] [http://www.librosweb.es/symfony\\_1\\_2/capitulo2/el\\_patron\\_mvc.html](http://www.librosweb.es/symfony_1_2/capitulo2/el_patron_mvc.html).
- LibrosWeb.es. [En línea] [Citado el: 02 de Abril de 2010.] [http://www.librosweb.es/symfony\\_1\\_2/capitulo2/el\\_patron\\_mvc.html](http://www.librosweb.es/symfony_1_2/capitulo2/el_patron_mvc.html).
- Linux Para Todos. [En línea] [Citado el: 2009 de Diciembre de 10.] <http://www2.linuxparatodos.net/web/servidores/base-de-conocimiento/-/wiki/Main/Servidor+Web;jsessionid=FA33F05A33564D0FA82D0914C7B522C2>.
- Microsoft Small Basic. [En línea] [Citado el: 30 de Marzo de 2009.] <http://programacion.fundaciondelcamino.org/wp-content/uploads/2009/10/Conceptos-de-Programaci%C3%B3n.pdf>.
- msdn. [En línea] [Citado el: 10 de Abril de 2010.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

- 
- **Övergaard Gunnar, Palmkvist Karin.** *Use Cases Patterns and Blueprints.* s.l. : Addison Wesley Professional, 2004. Vols. Part III: Use-Case Patterns.
  - **Övergaard, Gunnar y Palmkvist, Karin.** *Use Cases: Patterns and Blueprints.* 2004.
  - PHP Report Maker. [En línea] [Citado el: 27 de Marzo de 2010.] <http://www.hkvstore.com/phpreportmaker/>.
  - **Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico.* s.l. : McGraw-Hill/Interamericana de España, 2002.
  - SQLMax Connections. [En línea] [Citado el: 15 de Octubre de 2009.] [http://www.sqlmax.com/reportin\\_services1.asp](http://www.sqlmax.com/reportin_services1.asp).
  - Tecnológico. [En línea] [Citado el: 16 de Diciembre de 2009.] <http://www.mitecnologico.com/Main/AnalisisProcesoCicloVidaSoftware>.
  - The Apache Software Foundation. [En línea] [Citado el: 15 de Diciembre de 2009.] <http://httpd.apache.org>.
  - The Dojo Toolkit. [En línea] [Citado el: 7 de Diciembre de 2009.] <http://dojotoolkit.org>.
  - UML. [En línea] [Citado el: 31 de Marzo de 2010.] [http://www.neuronsrl.com.ar/training/uml/uml\\_secuencia.html](http://www.neuronsrl.com.ar/training/uml/uml_secuencia.html).
  - Universidad Distrital Francisco José de Caldas. [En línea] [Citado el: 25 de Abril de 2010.] <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/Pruebas/HTML%20-%20Pruebas%20de%20software/node1.html>.
  - Universidad EAFIT Ingeniería Matemática. [En línea] [Citado el: 16 de Noviembre de 2009.] <http://ingenieria-matematica.eafit.edu.co/recursos/glosario/c.html>.
  - UTN - Facultad Regional La Plata. [En línea] [Citado el: 20 de Diciembre de 2009.] [http://www.frlp.utn.edu.ar/materias/info2/bases\\_de\\_datos.htm](http://www.frlp.utn.edu.ar/materias/info2/bases_de_datos.htm).
  - Windows Live. [En línea] [Citado el: 5 de Abril de 2010.] <http://netbuzos.spaces.live.com/blog/cns!994BD929B80714FB!154.entry>.

## GLOSARIO

**CSS:** Hojas de estilo en cascada (en inglés *Cascading Style Sheets*), CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

**DOC:** Formato de archivo utilizado por el procesador de texto Microsoft Word.

**HTTP:** Protocolo de transferencia de hipertexto (HTTP por sus siglas en inglés). Es un protocolo de comunicaciones usado en cada transacción de la web, define la comunicación entre un servidor y otra máquina.

**Informe:** Conjunto de datos o resultado detallado de determinado reporte.

**MRD:** Acrónimo de Modelo de recogida de datos. Planilla con un estándar o formato previamente definido, que contienen un conjunto de variables y campos que pueden cambiar u actualizarse a partir de nuevas investigaciones.

**PDF:** Acrónimo del inglés Portable Document Format (formato de documento portátil). Es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

**Propel:** Es un ORM para PHP que facilita la labor de desarrollo de aplicaciones web.

**Reporte:** Plantilla que sirve como estándar para la generación de informes, define los datos y la manera de presentación de estos.

**UCI:** Universidad de las Ciencias Informáticas.

**USB:** El Universal Serial Bus (Bus Universal en Serie) o Conductor Universal en Serie (CUS), abreviado comúnmente USB, es un puerto para conectar periféricos a una computadora.

**XLS:** Extensión de archivo por defecto del formato Excel.

**XML:** Lenguaje de marcas extensible (XML por sus siglas en inglés). XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

**XMLHttpRequest:** Es una interfaz para realizar llamadas mediante el protocolo HTTP.

**XSLT:** Lenguaje de transformación de etiquetas extensible de hojas de estilo (XSLT por sus siglas en inglés). Presenta una forma de transformar documentos XML en otros, incluso a formatos que no son XML.