

005-12
Bri
A
TD-1645-08

FEU ...Universitarios
Cubanos



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

ANÁLISIS Y DISEÑO DEL MÓDULO REPORTE DEL PORTAL NACIONAL DE LA FEU

Trabajo de Diploma para Optar por el Título de:
Ingeniero en Ciencias Informáticas

Autor: Reinaldo Brito Cabrera

Tutores: Ing. Maidileydys Castellano Báez
Ing. Reldy Pérez Morales

“Año 50 de la Revolución”
Junio 2008

Agradecimientos comunes

A nuestro Comandante en Jefe y a la Revolución por propiciarme la oportunidad de estudiar en un centro como este.

Por ser el paradigma de todo joven revolucionario.

A la Universidad de las Ciencias Informáticas por formarme como profesional.

A mi tutora Ing Maidileydy's Castellanos Báez por su paciencia y dedicación.

A mi tutor Ing Reldy Pérez Morales por su ayuda incondicional.

A todas las personas que me ayudaron y siempre estuvieron a mi lado en todo momento.

Agradecimientos

A todos mis amigos del grupo por el apoyo que me dieron, por compartir todo momento en estos 5 años en donde compartimos momentos buenos y malos, gracias por todo: Roberto, Osniel, Yoan, Orlando, Juan José, Adrian...

A mi familia por estar siempre pendiente de mis estudios y a todos los que de una forma u otra han hecho realidad este sueño...

Dedicatoria

A mi mamá y a mi papá por su amor, apoyo y confianza plena en mí. Por enseñarme a vencer los obstáculos que se presentan en el camino de la vida.

A mi abuelito Nardo que es mi padre mayor y mi ejemplo. Por constituir este trabajo uno de sus más anhelados sueños.

A mis familiares que la vida no les dio la oportunidad de verme graduado.

A mi hermanita Elizabeth por todo el amor de hermano que siento por ella y porque deseo que tenga un buen futuro.

A mi abuelita Olga que me ha dado todo el amor y el cariño que necesita un nieto.

A mis tíos que siempre han estado presentes para ayudarme en todo momento.

A mi novia porque sin su cariño, ayuda y comprensión no lo hubiese logrado.

Rey

No pienso nunca en el futuro porque llega muy pronto.

Albert Einstein



Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 17 días del mes de Junio del año 2008.



Reinaldo Brito Cabrera



Ing. Maidileydys Castellano Báez



Ing. Reldy Pérez Morales

Opinión del tutor del trabajo de diploma

Título: Análisis y Diseño del Módulo Reportes del Portal Nacional de la FEU.

Autor: Reinaldo Brito Cabrera

Los tutores del presente Trabajo de Diploma consideran que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

El diplomante desarrolló su trabajo con alta calidad, independencia y responsabilidad. Demostró durante todo el proceso sus conocimientos en la rama de la informática y su laboriosidad a la hora de enfrentar todas las tareas para dar cumplimiento a los objetivos trazados.

Con la originalidad y creatividad que lo caracterizó en todo momento, obtuvo un trabajo que cuenta con una correcta organización, una estructura coherente, un desarrollo claro y fluido de las ideas, una adecuada concordancia y buena ortografía.

De esta excelente labor se obtuvo el análisis y diseño del Módulo de Reportes del Portal Nacional de la FEU. Este resultado contribuirá en gran medida a la implementación del futuro sistema, lo que mejorará la gestión eficiente de los reportes que se obtienen en el desarrollo de los procesos del Sistema Nacional de la FEU.

Destacar la calidad científico técnica que presenta el trabajo y el grado de importancia del mismo, el cual apoya el desarrollo de Kainos uno de los proyectos productivos de la facultad 1 de la Universidad de las Ciencias Informáticas.

Además se considera que el trabajo tiene valor para ser publicado.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero en Ciencias Informática; y propongo que se le otorgue al Trabajo de Diploma la calificación de ___ puntos.

___ días del mes de Junio del año 2008.

Ing. Maidileydys Castellano Báez

Fecha

Ing. Reldy Pérez Morales

Resumen

Resumen

En el presente trabajo se realiza un estudio de la situación actual que presenta el Sistema Nacional de la FEU, centrandó la investigación en el proceso de gestión de la información en dicho sistema.

El Sistema Nacional de la Federación Estudiantil Universitaria (FEU) es el encargado de gestionar todos aquellos procesos que estén relacionados con el estudiantado universitario, entre los cuales podemos mencionar la entrega de reportes a los diferentes niveles de dicha organización, sin embargo, presenta un gran número de dificultades en el tratamiento de los procesos estudiados, de ahí la necesidad de realizar el análisis y diseño de una aplicación Web que contribuya en la posterior implementación del sistema que mejorará la gestión de dichos procesos.

Durante el desarrollo de la investigación se realiza un análisis de toda la lógica del negocio, así como el fundamento de los lenguajes, tecnologías, herramientas y metodologías que se utilizan. Se explican las funcionalidades que tendrá el futuro sistema para poder erradicar los problemas encontrados y se realiza un estudio de la factibilidad de la investigación con el objetivo de obtener el costo estimado de la realización de la misma.

Por último, se propone el diseño de un módulo de reportes que gestione la información, en el cual se especifican los principales aspectos que se deben tener en cuenta a la hora del desarrollo del módulo, tomando siempre como guía las características de la arquitectura seleccionada.

Palabras Claves: Módulo de Reportes, reportes.

Índice

Índice

Introducción	1
Capítulo 1 Fundamentación teórica	5
1.1 Introducción	5
1.2 Contenido	5
1.3 Sistemas de Gestión de la Información en América Latina.....	5
1.3.1 SGA-UPL	6
1.3.2 Sistema Académico para Facultades.....	6
1.4 Sistemas de Gestión de la Información en Cuba.....	7
1.4.1 SIG-ESAC Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba	7
1.4.2 Sistema GNU	7
1.5 Herramientas para crear reportes.....	8
1.5.1 DynamicJasper	8
1.5.2 JavaEye Reporting Tool	9
1.5.3 JasperReports.....	10
1.6 Herramientas, lenguajes y metodologías utilizadas en el desarrollo del Sistema de Gestión de la Información: Kainos	10
1.6.1 J2EE.....	11
1.6.2 IDE Eclipse.....	11
1.6.3 PostgreSQL.....	12
1.6.4 Java.....	13
1.7 Software Libre.....	14
1.8 Arquitectura basada en Hibernate+Spring+JSF	17
1.8.1 Diseño de la Arquitectura de Alto Nivel.....	17

Índice

1.8.2 Arquitectura Multi-Capa.....	18
1.8.3 La Capa de Presentación y Java Server Faces	18
1.8.4 La Capa de Lógica de Negocio y el Marco de Trabajo Spring	19
1.8.5 Manejo de Beans con contexto de aplicación	19
1.8.6 Integración con Hibernate	19
1.8.7 La Capa de integración de Hibernate.....	20
1.9 Visual Paradigm.....	20
1.10 UML (Unified Model Language)	21
1.11 RUP	21
1.12 Conclusiones	22
Capítulo 2 Características del sistema.....	23
2.1 Introducción	23
2.2 Descripción de la situación actual.....	23
2.3 Información que se maneja.....	24
2.4 Propuesta del sistema	24
2.5 Modelo del negocio.....	25
2.5.1 Actores y trabajadores que intervienen en el negocio.....	25
2.5.2 Diagrama de casos de uso del negocio	26
2.5.3 Diagrama de clases del modelo de objetos.....	27
2.5.4 Diagrama de actividades <Solicitar información>	28
2.5.5 Descripción de los casos de uso del negocio.....	30
2.6 Especificación de los requisitos	32
2.6.1 Requerimientos funcionales	32
2.6.2 Requerimientos no funcionales	34
2.7 Modelo del sistema	35
2.7.1 Actores del sistema	35

Índice

2.7.2 Diagrama de casos de uso general.....	36
2.8 Descripción de los casos de uso.....	37
2.10 Conclusiones	63
Capítulo 3 Análisis y Diseño del sistema	64
3.1 Introducción	64
3.2 Descripción de la arquitectura a utilizar	64
3.2.1 Patrones de diseño utilizados	65
3.3 Modelado del diseño.....	66
3.3.1 Diagrama de diseño web del sistema.....	66
3.3.2 La vista de interacción.....	70
3.3.3 Interacción.....	71
3.3.4 Diagrama de secuencia.....	71
3.4 Modelo de Datos.....	76
3.5 Modelo de Despliegue	79
3.6 Estimación de costo-beneficio	76
3.7 Concepción general de la ayuda.....	81
3.8 Conclusiones	82
Conclusiones generales.....	83
Recomendaciones	84
Referencias bibliográficas.....	85
Bibliografía.....	86
Glosario de términos.....	87

Índice de Tablas

Índice de Tablas

Tabla 2.1 Actores y Trabajadores del Negocio	26
Tabla 2.2 Descripción de los casos de uso del negocio	30
Tabla 2.3 Actores del sistema	36
Tabla 2.4 Descripción del caso se uso: Gestionar Reporte	37
Tabla 2.5 Descripción de caso de uso: Gestionar Carpeta	43
Tabla 2.6 Descripción del caso se uso: Consultar Carpeta	50
Tabla 2.7 Descripción del caso se uso: Consultar Reporte	53
Tabla 2.8 Descripción del caso de uso: Crear Reporte.....	55
Tabla 3.1 Para calcular UAW.....	80
Tabla 3.2 Para calcular el UUCW	81
Tabla 3.3 Para calcular la complejidad técnica.....	82
Tabla 3.4 Para calcular el factor ambiente.....	83
Tabla 3.5 Para calcular el esfuerzo total.....	84

Índice de Figuras

Índice de Figuras

Figura 1 Diagrama de Casos de Uso del Negocio	27
Figura 2 Diagrama de clases del modelo de objetos	27
Figura 3 Diagrama de actividades Primera Parte	28
Figura 4 Diagrama de actividades Segunda Parte	29
Figura 5 Diagrama de CU del sistema	36
Figura 6 Arquitectura de tres capas	65
Figura 7 Diagrama de clases Web Consultar Carpeta 1 Parte	67
Figura 8 Diagrama de clases Web Consultar Carpeta 2 Parte	68
Figura 9 Diagrama de clases Web Crear Carpeta 1 Parte	69
Figura 10 Diagrama de clases Web Crear Carpeta 2 Parte	70
Figura 11 Diagrama de secuencia Consultar Carpeta (Escenario 1) 1 Parte	72
Figura 12 Diagrama de secuencia Consultar Carpeta (Escenario 1) 2 Parte	73
Figura 13 Diagrama de secuencia Crear Carpeta (Escenario 1) 1 Parte	74
Figura 14 Diagrama de secuencia Crear Carpeta (Escenario 1) 2 Parte	75
Figura 15 Diagrama de clases persistentes	77
Figura 16 Diagrama Entidad Relación	78
Figura 17 Modelo de Despliegue	79

Introducción

Introducción

En la Universidad de las Ciencias Informáticas (UCI) se están desarrollando proyectos productivos que cumplen con una de las ideas de nuestro Comandante "...las producciones intelectuales serán el sustento fundamental de Cuba...". La idea es convertir la Informática en una de las ramas más productivas de recursos para la nación" (1). Kainos es uno de los proyectos que surge para informatizar el trabajo de la (FEU) en todo el país, que propone la creación de un Portal Web que facilite el control de los datos de esta organización y brinde un espacio para la comunicación entre los universitarios. Una de las proyecciones de este proyecto está encaminada a desarrollar y poner en marcha un sitio que gestione la información de la FEU en la UCI.

La Federación Estudiantil Universitaria, es una organización estudiantil que representa los intereses y hace valer los derechos del estudiantado cubano. Entre sus objetivos actuales está agrupar a los universitarios para continuar la obra de la Revolución Cubana; promueve el perfeccionamiento del nivel y el rigor docente; y hace de la extensión universitaria la vía más importante para llevar el deporte, la cultura y el pensamiento intelectual a la sociedad.

La FEU desde su fundación ha estado en constante cambio a la par de los procesos revolucionarios cubanos. Hoy en día, sigue este ritmo poniéndose a la altura de la batalla de ideas y la universalización de la enseñanza donde tiene un papel protagónico.

La generación de reportes es quizás la tarea fundamental de todo sistema de gestión de datos.

Lamentablemente el punto débil de muchos sistemas de gestión es contar sólo con un conjunto cerrado de reportes, pues con la evolución de un determinado proceso de negocio aparece la necesidad de nuevos reportes. El Sistema Automatizado de Gestión de la Información (Kainos) y el Portal Nacional de la Federación Estudiantil Universitaria (FEU) fue diseñado a partir de las necesidades de dicha organización.

Cuenta hasta el momento con tres módulos (Seguridad, Funcionamiento y Estadística), haciéndose necesaria la creación de un módulo de Reportes Dinámico. El mismo debe

Introducción

permitir la obtención de información veraz y actualizada a los distintos niveles de la FEU en el país.

A partir de esta necesidad se identificó como **Situación Problemática** lo siguiente: En Cuba, las organizaciones de masas tienen dificultades para llevar el control de las estadísticas; ya sea de sus integrantes, o de la propia institución. Actualmente los procesos en la Federación Estudiantil Universitaria, a lo largo de nuestro país no se encuentran automatizados, por lo que los partes llegan errados, fuera de tiempo y en el peor de los casos nunca llegan. Las vías para entregar los mismos son el teléfono, el correo, o por valijas, corriendo un alto riesgo de perder la integridad y confidencialidad de la información que se desea transmitir, por lo que los reportes que deben recibir los interesados son poco confiables.

El Sistema Nacional de la FEU actualmente cuenta con un Portal Nacional de la FEU (Kainos) que no brinda la posibilidad de confeccionar de forma dinámica los reportes, lo que ocasiona que este sistema pierda funcionalidad.

Hoy en día los reportes en Kainos se confeccionan manualmente y se muestran en una página del sistema, esto provoca que el usuario final reciba en ocasiones información innecesaria, además de un trabajo engorroso para el personal que tiene que confeccionar el reporte.

De esta situación se deriva el siguiente **Problema Científico**: ¿Cómo lograr la gestión eficiente de reportes en los procesos del Sistema Nacional de la Federación Estudiantil Universitaria (FEU)?

El **Objeto de Estudio** de la investigación lo componen los procesos de gestión de la información en el Sistema Nacional de la FEU.

El **Campo de Acción** es: El proceso de gestión de reportes en el Sistema Nacional de la FEU.

El **Objetivo General** es: Realizar el análisis y diseño del Módulo de Reporte para el Portal Nacional de la FEU (Kainos).

Para dar respuesta al problema planteado se propone la siguiente **Idea a defender**: La realización del análisis y diseño del Módulo de Reportes del Portal Nacional de la FEU,

Introducción

facilitará su futura implementación, lo que contribuirá a la gestión eficiente de los reportes que se obtienen en el desarrollo de los procesos del Sistema Nacional de la FEU.

De lo anterior se derivan las siguientes **Tareas de la Investigación:**

1. Realización de un estudio preliminar de la situación en el actual Sistema Nacional de la FEU en Cuba.
2. Estudio de los sistemas de Cuba y América Latina que llevan a cabo la gestión de reportes.
3. Análisis de la bibliografía contemporánea de las herramientas más utilizadas en el trabajo con reportes, así como las utilizadas en el Portal Nacional de la FEU.
4. Captura de los requisitos; utilización de técnicas, métodos y plantillas.
5. Desarrollo del análisis y diseño del módulo de Reporte de Kainos.

Diseño Metodológico

Se utilizarán métodos teóricos y empíricos que garanticen los datos estadísticos para conducir adecuadamente la investigación.

1. Métodos Teóricos:

- **Analítico-Sintético:** Posibilitó el análisis de los procesos de gestión de la información en el Sistema Nacional de la FEU, permitiendo identificar los errores existentes durante la ejecución de dichos procesos, también una vez realizado el análisis de la bibliografía referentes a los sistemas de gestión de la información estadística y en especial el funcionamiento de las herramientas de confección de reportes, permitió descubrir las diferencias entre ellas.
- **Análisis histórico – lógico:** Mediante este método serán analizados trabajos y ficheros que abarquen información necesaria para el perfeccionamiento de la investigación.

Introducción

2. Métodos Empíricos:

- **Entrevistas:** Permitió adquirir información precisa de la lógica del negocio, viabilizando el entendimiento profundo del problema a resolver, así como las aristas principales hacia las cuales debe estar dirigida la investigación. También reuniones con representantes de dicha organización estudiantil con el objetivo que informen según el levantamiento de requisitos realizado, la información que se desea que se muestre en los reportes estadísticos.

El presente informe de investigación se estructura en 3 capítulos, que incluyen todo lo relacionado con el estudio y diseño de Reportes Dinámicos.

Capítulo 1. Fundamentación teórica: Incluye una fundamentación teórica sobre los sistemas que gestionan el funcionamiento de organizaciones políticas y de masas a nivel nacional e internacional y el de nuestra universidad; el análisis de la herramienta de modelado Visual Paradigm, Eclipse, RUP, PostgreSQL, Apache Tomcat y las diferentes herramientas para la construcción de reportes como el JasperReport, DynamicJasper etc, los cuales se estudiarán a fondo para determinar si es necesario su uso.

Capítulo 2. Características del sistema: En este capítulo se realiza un estudio del problema dado, que está orientado a fundamentar la necesidad de confeccionar un sistema que permita la creación de reportes dinámicos para el Portal Nacional de la Federación Estudiantil Universitaria (FEU).

Capítulo 3. Análisis y Diseño del módulo: En este capítulo se aborda el análisis y diseño de la aplicación en los aspectos más importantes, en función de satisfacer los requerimientos esperados.

Capítulo 1. Fundamentación teórica

1

Capítulo 1 Fundamentación teórica

1.1 Introducción

Este capítulo contiene el estudio de la generación de reportes en los sistemas de gestión de la información en el mundo y en Cuba. Se abordan las características de las herramientas que se utilizan en el desarrollo del sistema así como las tendencias actuales de las mismas y se hace una propuesta de una herramienta para crear reportes.

1.2 Contenido

El Sistema de Gestión Nacional de la FEU plantea una idea novedosa sin ningún precedente en nuestro país y con pocos antecedentes a escala internacional. En la investigación previa a este trabajo se aprecia que este sistema puede servir de paradigma para proyectos futuros, sobre todo en la gestión del trabajo de cualquier estructura.

Un reporte es una plantilla confeccionada en donde se encuentra una información determinada, esta plantilla puede ser elaborada estáticamente o dinámicamente.

El Portal Nacional de la Federación Estudiantil Universitaria (FEU) muestra los reportes estáticamente lo que dificulta la visualización de la información que es necesaria para el usuario final.

1.3 Sistemas de Gestión de la Información en América Latina

En América Latina existe un gran número de países que cuentan con sistemas que gestionan la información ya sea estadística o académica, y otros que investigan la posibilidad de producirlos, por la importancia de los procesos informativos en la sociedad.

Capítulo 1. Fundamentación teórica

1.3.1 SGA-UPL

Universidad Técnica de Loja, Ecuador. Las necesidades de la UPL se han incrementado permitiendo que se inicie un proyecto para implementar un proceso de desarrollo basado en Microsoft Solution Framework (MSF) para la construcción del Sistema de Gestión Académica; el cual le proporciona a la Universidad un solo entorno donde puede disponer de la información de todos los estudiantes en sus diferentes modalidades. El sistema facilita el acceso a través de su interfaz Web, también tiene como ventaja la cómoda actualización de las versiones, que son realizadas únicamente en el servidor. Presenta un módulo de reportes que permite la obtención eficiente de toda la información que se maneja. (2)

1.3.2 Sistema Académico para Facultades

Universidad Nacional de Asunción, Paraguay. El objetivo principal del sistema es proveer a la universidad de mecanismos automatizados que faciliten la planificación, organización, gestión y control académico. La arquitectura del sistema está pensada para su uso en un entorno multiusuario dando alcance a los diferentes sectores que componen la organización de la universidad. Entre los objetivos del sistema está registrar, controlar y emitir los resultados de cursos y exámenes de las asignaturas curriculares de todas las carreras de la universidad y apoyar la evaluación de las actividades educativas desarrolladas por la universidad mediante varios informes estadísticos.

Cuenta con varias funcionalidades y módulos, dentro de ellos está el módulo de Plan de estudio, que constituye la estructura de información principal del sistema, en torno a él giran gran parte de los controles que se realizan. Cuenta con el Módulo Reportes estadísticos, este módulo del sistema permite a la institución la obtención de información estadística en base a todos los datos cargados en el sistema. Los reportes generados tienen formato de planilla y también en algunos casos pueden generarse reportes gráficos comparativos y la inclusión de nuevos reportes es totalmente factible, entre otros. (3)

Capítulo 1. Fundamentación teórica

1.4 Sistemas de Gestión de la Información en Cuba

1.4.1 SIG-ESAC Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba

SIG-ESAC surge con el objetivo de facilitar la gestión de la estadística de salud. Se confecciona una aplicación de un Sistema de Información Geográfica (SIG), el cual permite cartografiar y hacer diferentes tipos de análisis de importantes indicadores de salud: morbilidad, mortalidad, demográficos, recursos y servicios.

El desarrollo y la diversificación de los Sistemas de Información Geográfica (SIG) posibilitan que actualmente esta poderosa herramienta sea aplicable en un campo tan sensible e importante como la salud pública, teniendo en cuenta fundamentalmente la gran cantidad de datos que se genera en esta actividad y que estos puedan ser representados para gestionar dicha información. Los SIG en la salud pública se han utilizado principalmente para procesar la estadística médica y en investigaciones epidemiológicas que estudian la magnitud y distribución de distintos problemas sanitarios en las poblaciones, así como en la vigilancia, análisis, monitoreo, evaluación de intervenciones, gestión y toma de decisiones vinculadas con este campo. Incluye un módulo de reportes estadísticos con el fin de obtener de forma eficaz toda la información que dicho sistema procesa. (4)

1.4.2 Sistema GNU

El Ministerio de Educación Superior (MES) carece de un sistema informatizado para la organización de la Educación Superior, por lo que dificulta la preparación de datos e informes, tanto para el Gobierno, como para uso interno. Actualmente con la colaboración de varias universidades se está desarrollando un sistema que contará con 15 módulos: Matrícula, Secretaría Docente, Plan de Estudio, Profesor, Recuperación de Información Web, Estadísticas, Alumno Ayudante, Estipendio, Estudiantes Extranjeros, Planificación Docente, Archivo Histórico, Becas, Extracurricular, Guardia y Reporte.

Capítulo 1. Fundamentación teórica

Cuenta con una plataforma de software libre y multiplataforma J2EE / DBMS PostgreSQL /servidor de aplicaciones JBOSS /Generador AndroMDA (pueden funcionar sobre Windows y LINUX-UNIX). (5)

1.5 Herramientas para crear reportes

Los usuarios tienen la necesidad de gestionar información de manera fácil y con mantenimiento simplificado. Para la generación de reportes se analizó la posibilidad de utilizar las siguientes herramientas: el DynamicJasper, JavaEye Reporting Tool y el JasperReports.

1.5.1 DynamicJasper

(DJ) es un proyecto open-source que permite al desarrollador crear rápidamente una gran variedad de reportes a través de una intuitiva API escrita en Java. Esta permite definir las columnas, grupos, totales, gráficos (charts), sub-reportes, el formato de salida (PDF, Excel, HTML, etc.) en tiempo de ejecución. La API maneja todo lo relacionado con la diagramación y posicionamiento de los elementos del reporte haciendo el proceso de diseño fácil y automático.

DynamicJasper tiene como objetivo abarcar el 99% de los reportes que se basan en columnas como así también los que tienen grupos (cortes de control). La API permite agregar variables en las cabeceras y pie de las columnas y grupos con operaciones tales como sumar, contar, etc. Se puede definir en tiempo de ejecución el orden de aparición de las columnas, los grupos, las variables, los estilos, estilos condicionales, sub-reportes, etc. Existe una aplicación web online de ejemplo que muestra las características principales de DynamicJasper.

Características:

- DJ usa directamente la API de JasperReports, por lo que no hay necesidad de archivos jrxml. Sin embargo se pueden seguir utilizando como plantilla (marca de agua, logo de la compañía, etc.) y permitir a DJ hacer la disposición de los elementos del reporte.

Capítulo 1. Fundamentación teórica

- Columnas dinámicas: Se pueden definir las columnas del reporte y su orden de aparición en tiempo de ejecución. Cada una de ellas puede tener su propio estilo (Font de la cabecera, de los datos, colores, bordes, etc.) El ancho de las columnas se acomoda automáticamente hasta alcanzar el ancho de la hoja.
- Grupo: Se pueden crear grupos dinámicamente usando una columna como criterio, o expresiones más complejas (programadas en Java). Cada grupo tendrá su cabecera y pie, los cuales pueden tener variables mostrando subtotales de operaciones de las columnas.
- Formato condicional: DJ provee una manera sencilla de agregar formato condicional a los elementos de una columna. Se pueden utilizar expresiones sencillas o expresiones customizadas escritas en Java para definir cuando utilizar un estilo en particular.
- Opciones de reporte dinámico configurables en tiempo de ejecución tales como tamaño y orientación de página, márgenes, columnas (tipo guía telefónica), etc. Los elementos siempre se acomodan a las nuevas condiciones.
- Exportar a los formatos más populares no es un problema, ya que DJ se basa en JasperReports y puede exportar todo lo que JasperReports (Excel, PDF, OpenOffice, Word, HTML, etc.) Aunque DJ agrega valor al poder exportar a Excel sin formateo de los datos, sin salto de página, sin celdas vacías en el medio, lo cual es ideal para usuarios que utilizaran las hojas de cálculo para hacer otras operaciones. Esto es difícil de lograr simplemente utilizando editores como iReport. (6)

Existe en el mundo poca documentación para el estudio sobre esta herramienta de construcción de reportes.

1.5.2 JavaEye Reporting Tool

Es un producto web desarrollado en Java que se puede instalar bajo Tomcat, Los reportes se definen como una consulta SQL, pero al usuario se le muestra el nombre del reporte, se le pueden pasar parámetros a las consulta SQL, asignándole el tipo de datos y un valor por defecto, los resultados se muestran compaginado para no consumir demasiados recursos, y

Capítulo 1. Fundamentación teórica

existe una opción para imprimir que muestra todos los datos de la consulta, por otra parte se pueden exportar los datos a Excel. Además cuenta con seguridad, pudiendo definir usuarios, grupos, y permisos de acceso a las consultas. No es muy utilizado en el mundo. (7)

1.5.3 JasperReports

JasperReports es la mejor herramienta de código libre en Java para generar reportes. Puede entregar ricas presentaciones o diseños en la pantalla, para la impresora o para archivos en formato PDF, HTML, RTF, XLS, CSV y XML.

Está completamente escrita en Java y se puede utilizar en una gran variedad de aplicaciones de Java, incluyendo J2EE o aplicaciones Web, para generar contenido dinámico.

JasperReports trabaja en forma similar a un compilador y a un intérprete. El usuario diseña el reporte codificándolo en XML de acuerdo a las etiquetas y atributos definidos en un archivo llamado `jasperreports.dtd` (parte de JasperReports). Usando XML el usuario define completamente el reporte, describiendo dónde se deben colocar textos, imágenes, líneas, rectángulos, cómo adquirir los datos, cómo realizar ciertos cálculos para mostrar totales, etc.

iReport es un diseñador visual de código libre para JasperReports escrito en Java. Es un programa que ayuda a los usuarios y desarrolladores que usan la librería JasperReports para diseñar reportes visualmente. A través de una interfaz rica y simple de usar, iReport provee las funciones más importantes para crear reportes amenos en poco tiempo. (8)

Después de un estudio profundo se puede concluir que el JasperReports es la herramienta que más se ajusta a las características de los reportes que se requieren, dicha herramienta se ha trabajado y probado.

1.6 Herramientas, lenguajes y metodologías utilizadas en el desarrollo del Sistema de Gestión de la Información: Kainos

Por la rapidez que se requería para la confección del sistema y la inexperiencia del equipo seleccionado en una Universidad recién creada se decidió utilizar las siguientes herramientas:

Capítulo 1. Fundamentación teórica

- Almacenamiento de datos: PostgreSQL.
- Plataforma de Desarrollo: J2EE.
- Lenguaje de programación: Java.
- Metodología para el desarrollo del software: RUP.

1.6.1 J2EE

Java 2 Enterprise Edition, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Java con arquitectura de N niveles distribuidos, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. Java EE es también considerada informalmente como un estándar. Es por esto que se decidió utilizarlo.

La utilización de Servidores de Aplicaciones y la ejecución de aplicaciones basadas en tecnología J2EE impactan en mayor o menor medida en distintas disciplinas dentro de una organización. Algunas de las más relevantes:

1. Modelado de Datos.
2. Análisis de Requerimientos.
3. Arquitectura y Diseño.
4. Codificación.
5. Testing.
6. Entorno y Operación.
7. Administración de Proyectos.

1.6.2 IDE Eclipse

Como Entorno Integrado de Desarrollo (IDE) se propone la utilización del Eclipse, que no es más que un software creado originalmente por IBM. Actualmente la Fundación Eclipse es una entidad sin ánimo de lucro, que se encarga de llevar a cabo el desarrollo de este proyecto. La

Capítulo 1. Fundamentación teórica

definición que da la Fundación Eclipse acerca de su software es: "una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular".

Eclipse es un entorno integrado de desarrollo (IDE, Integrated Development Environment) multiplataforma englobado en el movimiento de software libre para crear aplicaciones clientes de cualquier tipo.

Beneficios

- Es open-source.
- Soporta la construcción de una variedad de herramientas para el desarrollo de aplicaciones.
- Soporta el desarrollo de aplicaciones basadas en GUI y non-GUI.
- Soporta herramientas que manipulan diferentes tipos de archivos como por ejemplo Java, C, C++, EJB, HTML, GIF, etc.
- Corre en una gran cantidad de sistemas operativos incluyendo Windows y Linux.
- Provee a los desarrolladores, herramientas (ej.- PDE) que facilitan la creación de plugins.

Desventajas

Si bien Eclipse es multiplataforma, los plugins no tienen por qué serlo. Existen plugins que sólo corren en una plataforma, o que aún no han sido desarrollados para más de una. Al ser una herramienta open-source, se desarrollan plugins que no tienen todas las funcionalidades que tienen en otras herramientas comerciales, como ser IBM Websphere.

1.6.3 PostgreSQL

PostgreSQL, es un motor de base de datos, servidor de base de datos relacional libre, liberado bajo una licencia de software libre.

Algunas de sus características fundamentales son:

- Alta concurrencia

Capítulo 1. Fundamentación teórica

- Amplia variedad de tipos nativos
- Los usuarios pueden crear sus propios tipos de datos.
- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (foreign keys).
- Disparadores (triggers).
- Vistas.
- Integridad referencial.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Funciones.

Inconvenientes:

- Consume recursos y carga el sistema.
- Límite del tamaño de cada fila de las tablas a 8k.
- Es de dos a tres veces más lento que MySQL.
- Menos funciones en PHP.

1.6.4 Java

Se propone que la implementación sea en el lenguaje Java dadas sus características y potencialidades.

Java está totalmente basado en clases y objetos. Todo en Java (excepto algunos tipos primitivos) es un objeto. Contrariamente a lenguajes híbridos como C++ o el popular Visual Basic, en Java no se permite programar fuera de un objeto o clase. No hay módulos ni funciones globales.

Una característica más distintiva de Java es su capacidad multiplataforma. Lenguajes como C o COBOL se han implementado en múltiples plataformas, pero siempre han necesitado recopilaciones o adaptaciones al pasar de una a otra. Dicha capacidad no es sólo a nivel de

Capítulo 1. Fundamentación teórica

código fuente, sino también a nivel de código compilado. El programa que se escribe puede compilarse en Windows o en Linux, y funciona. Además hasta el programa compilado puede ejecutarse sin más preparación en distintas máquinas, debido a que Java se compila y ejecuta no en un procesador o entorno en particular, sino en lo que se llama una "virtual machine", una máquina virtual. Por otra parte podrá ejecutarse en cualquier sistema operativo que tenga una máquina virtual Java compatible.

No es solamente un lenguaje, es una tecnología. Al estar basado en clases y objetos, viene acompañado de un conjunto de éstos, que sirve como base para la programación de aplicaciones, de texto, gráficas o que se ejecuten en una página Web como "applets".

Java maneja prácticamente todas las bases de datos relacionales, desde Microsoft SQL Server, Oracle, Sybase, Informix, como Access, MySQL y PostgreSQL.

1.7 Software Libre

Software libre (en inglés free software) es la denominación del software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso, se refiere a cuatro libertades de los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo). (9)

"El software libre, es un movimiento tecnológico que ha revolucionado la sociedad por sus implicaciones propias. Presenta características especiales que han permitido (y hecho necesaria) la experimentación de nuevas formas de desarrollo y mantenimiento de programas, así como de modelos económicos normas legales.

El Software Libre es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en "libre" como en "libertad de expresión".

Capítulo 1. Fundamentación teórica

Software Libre se refiere a cuatro libertades de los usuarios del software:

- Usar el programa, con cualquier propósito (libertad 0).
- Estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- Distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- Mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.”

Para que las libertades de hacer modificaciones y de publicar versiones mejoradas tengan sentido, se debe tener acceso al código fuente del programa. Por lo tanto, la posibilidad de acceder al código fuente es una condición necesaria para el Software Libre.

Para que estas libertades sean reales, deben ser irrevocables mientras no se haga nada incorrecto; si el desarrollador del software tiene el poder de revocar la licencia aunque no se le haya dado motivos, el software no es libre.

Son aceptables, sin embargo, ciertos tipos de reglas sobre la manera de distribuir Software Libre, mientras no entren en conflicto con las libertades centrales. Por ejemplo, copyleft [“izquierdo de copia”]; expresado muy simplemente es la regla que implica que, cuando se redistribuya el programa, no se pueden agregar restricciones para denegar a otras personas las libertades centrales. Esta regla no entra en conflicto con las libertades centrales, sino que más bien las protege.

Lo anteriormente expresado da la medida de la viabilidad económica de este sistema, que por ser libre no es necesariamente gratuito, sino que nos da la posibilidad de comercializarlo, regalarlo, prestarlo con total libertad y protegido legalmente. Evidentemente, a pesar de las lagunas técnicas que pueda tener el Software Libre, es la alternativa para los países subdesarrollados.

Capítulo 1. Fundamentación teórica

Ventajas

- Costo: el costo total de propiedad del sistema operativo libre Linux es menos de la mitad que el de Windows. Gran parte del ahorro proviene de no tener que pagar licencia y de sus menores costos de administración.
- Innovación tecnológica: el desarrollo en comunidad de este sistema y el conocimiento del Código Fuente, propician que a cada instante, un desarrollador necesite nuevas actualizaciones y las realice él mismo, proponiendo nueva funcionalidad al programa.
- Requisitos de hardware y durabilidad de las soluciones: un mejor soporte en general para las versiones antiguas de software y de plataformas de hardware o software minoritarias. No necesitas comprar nuevas versiones para estar actualizado.
- Escrutinio público: el proceso de revisión pública al que está sometido el desarrollo del software libre imprime un gran dinamismo al proceso de corrección de errores. Cada mejora es socializada libremente, la comunidad puede cambiar la realidad de las innovaciones.
- Independencia del proveedor: el Software Libre garantiza una independencia con respecto al proveedor gracias a la disponibilidad del código fuente.
- Desarrollo de la industria local: en el Software Libre no hay costo de licencia debido al derecho a copia, y al disponer del código fuente de la aplicación es posible desarrollar internamente las mejoras o las modificaciones necesarias. De este modo, se contribuye a la formación de profesionales en nuevas tecnologías y al desarrollo local y de la industria nacional de software.
- Datos personales, privacidad y seguridad: el Software Libre, por su carácter abierto, dificulta la introducción de código malicioso, espía o de control remoto, debido a que el código lo revisan muchos usuarios y desarrolladores que pueden detectar posibles puertas traseras. En el mundo del software libre, cualquier programador puede realizar una auditoría para comprobar que no se ha introducido ningún código malicioso y a su vez, cualquier entidad puede añadir libremente encriptación adicional a la aplicación que utilice para proteger sus datos.

Capítulo 1. Fundamentación teórica

- Adaptación del software: el Software Libre permite personalizar, gracias al hecho de que se dispone del código fuente, los programas tanto como sea necesario hasta que cubran exactamente cualquier necesidad.
- Lengua: los principales proyectos libres, tienen numerosas traducciones, gracias a que no precisan autorización de ningún propietario y cualquier persona o institución puede realizarlas.

Desventajas

Una de las principales desventajas de los Sistemas Libres, (que a veces constituye una ventaja), radica en el hecho de que existe más de una imagen y cada desarrollador o usuario utiliza la que le conviene. Sin embargo, la comunidad nunca se pone de acuerdo hacia cual versión se debe migrar.

Para los especialistas en tecnologías de la informática, esto no representa un problema, pero si hablamos de usuarios finales (secretarías, profesionales de otras ramas que usan la computadora solo como medio auxiliar, estudiantes, población en general), si cada vez que ven una PC, tiene una versión diferente (Debian, Ubuntu, Nova, etcétera), entonces se les hace muy difícil su trabajo.

En esta investigación optamos por un entorno mayormente de software libre analizando también que la política de nuestro país y en particular nuestra Universidad, va encaminada a la migración de todos los sistemas hacia esta plataforma.

1.8 Arquitectura basada en Hibernate+Spring+JSF

1.8.1 Diseño de la Arquitectura de Alto Nivel

La arquitectura de alto nivel implica subdividir la aplicación en componentes funcionales y particionarlos en capas. El diseño de la arquitectura de alto nivel es neutral a las tecnologías utilizadas.

Capítulo 1. Fundamentación teórica

1.8.2 Arquitectura Multi-Capa

Una arquitectura multicapa particiona todo el sistema en distintas unidades funcionales: cliente, presentación, lógica-de-negocio, acceso a datos, y bases de datos. Esto asegura una división clara de responsabilidades y hace que el sistema sea más mantenible y extensible. Los sistemas con tres o más capas se han probado como más escalables y flexibles que un sistema cliente-servidor, en el que no existe la capa central de lógica-de-negocios.

La capa del cliente es donde se consumen y presentan los modelos de datos. Para una aplicación Web, la capa cliente normalmente es un navegador Web. Los clientes pequeños basados-en-navegador no contienen lógica de presentación; se trata en la capa de presentación.

La capa de presentación expone los servicios de la capa de lógica-de-negocio a los usuarios. Sabe cómo procesar una petición de cliente, cómo interactuar con la capa de lógica-de-negocio, y cómo seleccionar la siguiente vista a mostrar.

La capa de la lógica-de-negocio contiene los objetos y servicios de negocio de la aplicación. Recibe peticiones de la capa de presentación, procesa la lógica de negocio basada en las peticiones, y media en los accesos a los recursos de la capa de datos. Los componentes de la capa de lógica-de-negocio se benefician de la mayoría de lo servicios a nivel de sistema como el control de seguridad, de transacciones y de recursos.

La capa de acceso a datos es el puente entre la capa de lógica-de-negocio y la capa de datos. Encapsula la lógica para interactuar con la capa de datos.

1.8.3 La Capa de Presentación y Java Server Faces

JSF es un marco de trabajo de componentes de interface de usuario del lado del servidor para aplicaciones Web basadas en Java. JSF contiene un API para representar componentes UI y manejar sus estados, sus eventos, la validación del lado del servidor, y la conversión de datos, definir la navegación entre páginas, soportar internacionalización y accesibilidad; y proporcionar extensibilidad para todas estas características. También contiene dos librerías de

Capítulo 1. Fundamentación teórica

etiquetas JSP (JavaServer Pages) personalizadas para expresar componentes UI dentro de una página JSP y para conectar componentes a objetos del lado del servidor.

La capa de presentación recoge la entrada del usuario, presenta los datos, controla la navegación por las páginas y delega la entrada del usuario a la capa de la lógica-de-negocio. La capa de presentación también puede validar la entrada del usuario y mantener el estado de sesión de la aplicación.

1.8.4 La Capa de Lógica de Negocio y el Marco de Trabajo Spring

Los objetos y servicios de negocio existen en la capa de lógica-de-negocio. Un objeto de negocio no sólo contiene datos, también contiene la lógica asociada con ese objeto específico.

Los servicios de negocio interactúan con objetos de negocio y proporcionan una lógica de negocio de más alto nivel. Se debería definir una capa de interface de negocio formal, que contenga los interfaces de servicio que el cliente utilizará directamente.

1.8.5 Manejo de Beans con contexto de aplicación

Spring puede organizar de forma efectiva nuestros objetos de la capa central y manejar las conexiones por nosotros. Spring puede eliminar la proliferación de solitarios y facilita unas buenas prácticas de programación orientada a objetos, por ejemplo utilizando interfaces.

1.8.6 Integración con Hibernate

Spring no nos fuerza a utilizar su potente característica de abstracción JDBC. Se integra bien con marcos de trabajo de mapeo O/R, especialmente con Hibernate. Spring ofrece un manejo seguro y eficiente de sesiones Hibernate, maneja la configuración de la SessionFactory de Hibernate y las fuentes de datos JDBC en el contexto de la aplicación, y hace que la aplicación sea más fácil de testear.

Capítulo 1. Fundamentación teórica

1.8.7 La Capa de integración de Hibernate

Hibernate es un marco de trabajo de mapeo Open Source que evita la necesidad de utilizar el API JDBC. Hibernate soporta la mayoría de los sistemas de bases de datos SQL. El Hibernate Query Language, diseñado como una extensión mínima, orientada a objetos, de SQL, proporciona un puente elegante entre los mundos objeto y relacional. Hibernate ofrece facilidades para recuperación y actualización de datos, control de transacciones, repositorios de conexiones a bases de datos, consultas programáticas y declarativas, y un control de relaciones de entidades declarativas.

1.9 Visual Paradigm

Las aplicaciones informáticas que facilitan el trabajo dentro del ciclo de desarrollo del software son conocidas como herramientas CASE (Ingeniería de Software Asistida por Computadora) y se emplean para aumentar la productividad del desarrollo del software disminuyendo los tiempos de construcción y el costo de los mismos. La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad en el desarrollo de sistemas de información.

Como herramienta CASE a utilizar, fue seleccionado el Visual Paradigm debido a las grandes ventajas que presenta y que se relacionan a continuación, haciéndola idónea para el desarrollo. Para la elección se tuvo presente su carácter de software libre, además de que la universidad posee la licencia para su empleo.

Su uso se ha extendido entre los analistas de todo el mundo, dadas sus facilidades de integración con distintos entornos de desarrollo, incluye además UML 2.0, brindando la posibilidad de modelar el sistema orientado a procesos en el futuro.

Presenta varios beneficios entre los que se destacan:

- Poderosa herramienta de generación de PDF/HTML a partir de diagramas UML.
- Sincronización entre el código fuente y el modelo en tiempo real.
- Soporte para toda la notación UML.
- Soporta UML 2.0.

Capítulo 1. Fundamentación teórica

- Presenta un diseño basado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Ofrece capacidades de ingeniería directa e inversa.
- Se integra al IDE de Eclipse

1.10 UML (Unified Model Language)

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software (Rumbaugh, y otros, 1998). UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y cuentan con reglas para combinar tales elementos. Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso o metodología, sino un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

1.11 RUP

Se entendió utilizar la metodología RUP por las tendencias actuales que existen, las cuales hablan mucho de la utilización de XP o Extreme Programming y proponen un ciclo de desarrollo de software de manera ágil y novedosa, contando siempre con un cliente interno que tendrá un contacto directo con el equipo de desarrollo, razón por la cual no es apropiado para el sistema precisamente por las características del cliente. RUP es la que mejor se adapta a este caso de estudio y a las condiciones del equipo de desarrollo, teniendo en cuenta que el organigrama de equipo del mismo es descentralizado democrático y la comunicación es abierta. RUP es uno de los procesos más generales y abarcadores de los existentes actualmente, ya que en realidad está pensado englobar todo el desarrollo del mismo, además de ser una metodología probada y segura. RUP se basa en casos de uso para describir lo que se espera del software y está orientado a la arquitectura del sistema, documentándose lo mejor posible, basándose en UML (Unified Modeling Language) como herramienta principal. (10)

Capítulo 1. Fundamentación teórica

1.12 Conclusiones

En este capítulo se detallaron las condiciones que rodean al objeto de estudio y se argumentaron los problemas existentes en el campo de acción.

Se describieron algunos de los sistemas de gestión de la información de nuestro país y del mundo. Se estudiaron las herramientas con que se desarrolló el sistema y su estado del arte, lo cual facilitó el entendimiento de nuestro sistema en general y permitió conocer como se comporta la tecnología actual y se logra hacer una propuesta después del estudio de la herramienta para crear reportes para gestionar información de forma simplificada.

Capítulo 2. Características del sistema

2

Capítulo 2 Características del sistema

2.1 Introducción

En este capítulo se hace una propuesta del Módulo de Reportes del Portal Nacional de la FEU a partir del problema planteado. Se analiza el negocio actual, obteniendo el modelo del negocio y las descripciones de los Casos de Uso del mismo. Se muestran los requerimientos funcionales y los no funcionales, se definen los actores y los casos de uso que intervienen en el sistema.

2.2 Descripción de la situación actual

Gran parte de la información que se maneja en los distintos niveles de la Federación Estudiantil Universitaria se adquiere a través del Sistema Automatizado de Gestión de la Información – Kainos. Este sistema no cuenta con un módulo de reporte por lo que no posibilita el diseño y publicación de estos.

Actualmente muchos usuarios no pueden consultar el grueso de la información del sistema. Por otra parte los reportes no satisfacen las necesidades de los usuarios al no disponer de la información necesaria.

Cuando el Presidente Nacional de la FEU solicita una información esta viaja escalonadamente, primeramente el Presidente Nacional se la solicita al Organizador Nacional, luego pasa al Secretario de Funcionamiento Provincial, este se la solicita al Secretario de Funcionamiento Municipal o CES, luego al Secretario de Funcionamiento de Facultades o Sedes y finalmente llega a la base que es el Jefe de Brigada, el Jefe de Brigada confecciona un reporte que viajará nuevamente escalonadamente, en cada nivel se consulta y se confecciona un nuevo reporte hasta llegar al Presidente Nacional.

Capítulo 2. Características del sistema

2.3 Información que se maneja

La Información que se maneja en los niveles de FEU, sea Nacional, Provincial, Municipal, Facultades o Sedes o Brigada tienen un contenido diferente que queda almacenado en las siguientes planillas.

Datos de la Brigada: Esta planilla posee los datos de la brigada como son: nombre de la brigada, cantidad de militantes, y la cantidad según los datos de cada militante (sexo, raza, entre otros).

Datos de la Facultades o Sedes: Esta planilla posee los datos fundamentales de una Facultad o de una Sede, como cantidad de brigadas, nombres de las brigadas, cantidad de militantes en dicha Facultad o Sede, y la cantidad según los datos de cada militante (sexo, raza, entre otros).

Datos de Municipio o CES: Esta planilla posee los datos del Municipio o CES, como cantidad de Sedes Universitarias, cantidad de militantes que existen en el municipio, y la cantidad según los datos de cada militante (sexo, raza, entre otros).

Datos de Provincia: Esta planilla posee los datos de la Provincia, se encuentra registrada en ella la cantidad de Facultades que existe en la provincia, nombre de cada facultad, así como el número total de militantes, y la cantidad según los datos de cada militante (sexo, raza, entre otros).

Datos de la Nación: En esta plantilla se encuentra registrada la información fundamental de la nación, como cantidad de facultades de país, nombre de cada facultad, cantidad de militantes en el país, y la cantidad según los datos de cada militante (sexo, raza, entre otros).

2.4 Propuesta del sistema

El trabajo con los reportes en el sistema actual no siempre resulta ameno. El usuario dedica mucho tiempo para realizar un reporte, por eso se pretende que el diseño del módulo esté encaminado a facilitar la futura implementación del mismo lo que ofrecerá:

Capítulo 2. Características del sistema

- Opciones al usuario para crear diferentes tipos de reportes.
- Darle la opción al usuario de eliminar un reporte.
- Brindarle la opción al usuario de modificar los reportes creados con el objetivo de obtener nuevos datos.
- Personalizar el formato de salida de los reportes según las necesidades de los usuarios finales.

2.5 Modelo del negocio

El proceso de modelación del negocio permite obtener una visión de lo que se necesita para definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos, especificándose aquí qué procesos del negocio soportará el sistema.

2.5.1 Actores y trabajadores que intervienen en el negocio

Un actor del negocio es cualquier individuo, grupo, organización o máquina que interactúa con el negocio.

Un trabajador del negocio se utiliza en el modelado de negocio como punto de partida para derivar un primer conjunto de actores y casos de uso del sistema de información en construcción.

Capítulo 2. Características del sistema

Tabla 2. 1 Actores y Trabajadores del Negocio

Actores del Negocio	Justificación
Presidente de la FEU Nacional	Interviene en el proceso de obtener información.
Trabajadores del Negocio	Justificación
Organizador Nacional.	Interviene en la organización y planificación de las actividades de la Federación Estudiantil Universitaria.
Secretario de Funcionamiento Provincial.	Interviene en el trabajo con las actas y los datos de la Provincia de la Federación Estudiantil Universitaria.
Secretario de Funcionamiento Municipal o CES.	Interviene en el trabajo con las actas y los datos del Municipio o CES de la Federación Estudiantil Universitaria.
Secretario de Funcionamiento de Sedes o Facultades.	Interviene en el trabajo con las actas y los datos de las Sedes o Facultades de la Federación Estudiantil Universitaria.
Jefe de Brigada.	Interviene en el trabajo de gestión de datos estadísticos y de actas de funcionamiento de la Brigada.

2.5.2 Diagrama de casos de uso del negocio

El modelo del negocio describe el mismo en términos de casos de usos del negocio, lo cual corresponde con lo que generalmente se le llama procesos.

El modelo de Casos de Uso del Negocio describe los procesos del mismo (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes. Describe las funciones que el negocio pretende realizar y su objetivo básico, es describir cómo el negocio es utilizado por sus clientes y socios.

Capítulo 2. Características del sistema

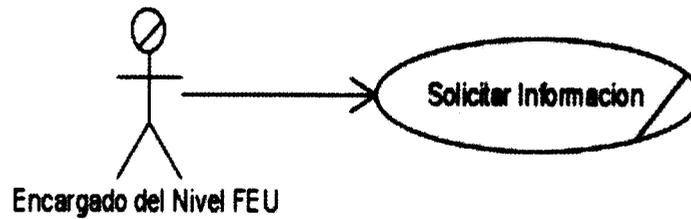


Figura 1 Diagrama de Casos de Uso del Negocio

2.5.3 Diagrama de clases del modelo de objetos

El diagrama de objetos describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso del mismo, mostrando la relación entre ellos.

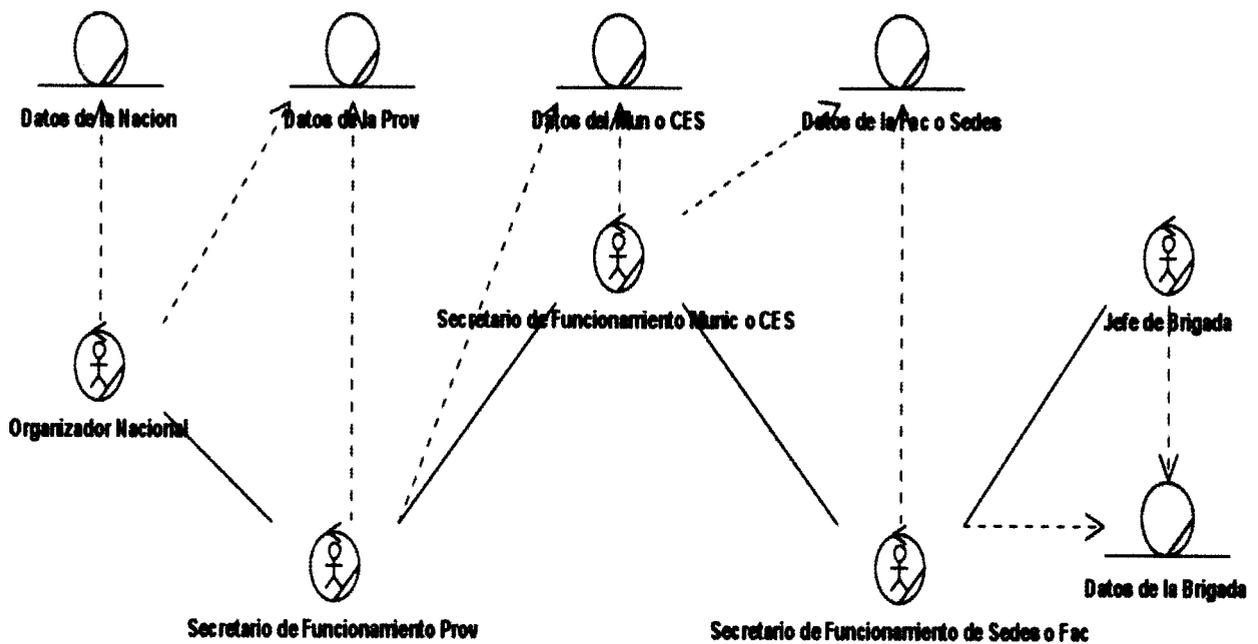


Figura 2 Diagrama de clases del modelo de objetos

Capítulo 2. Características del sistema

2.5.4 Diagrama de actividades <Solicitar información>

El diagrama de actividades describe gráficamente la estructura del flujo de actividades.

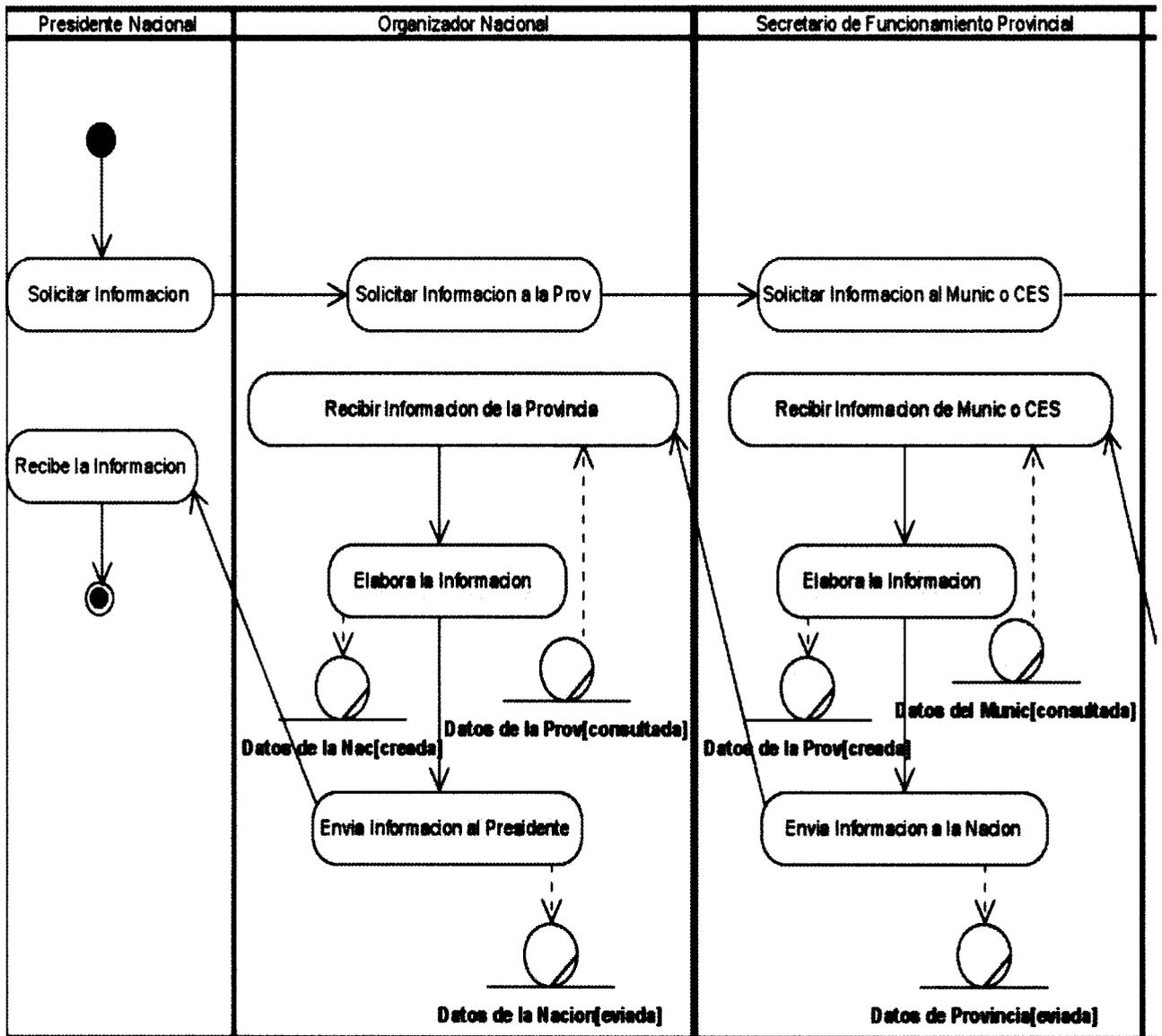


Figura 3 Diagrama de actividades Primera Parte

Capítulo 2. Características del sistema

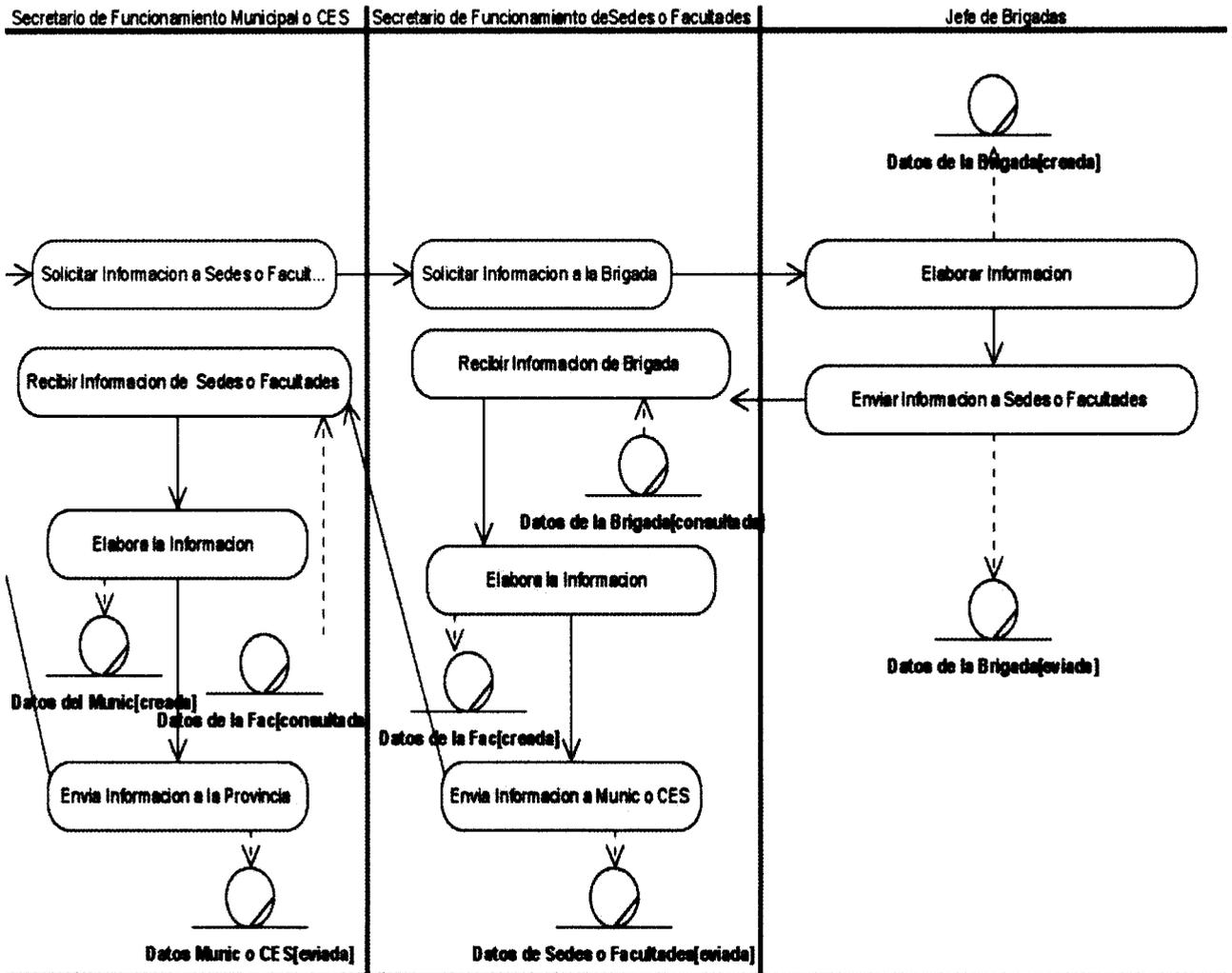


Figura 4 Diagrama de actividades Segunda Parte

Capítulo 2. Características del sistema

2.5.5 Descripción de los casos de uso del negocio

Un caso de uso del negocio describe el comportamiento de un actor particular del negocio.

Tabla 2. 2 Descripción de los casos de uso del negocio

Nombre del Caso de Uso: Solicitar Información	
Actores:	Presidente Nacional o del Nivel del FEU (Inicia)
Propósito:	Solicitar y Obtener la información que necesita el actor.
Resumen:	El caso de uso se inicia cuando el Presidente Nacional de la FEU o el Presidente del Nivel solicita alguna información en forma de reporte, esta solicitud pasa por cada nivel hasta la base que es la brigada, esta elabora un reporte y lo envía la información al nivel que se encuentra por encima y así sucesivamente hasta llegar al Presidente Nacional.
Precondiciones	
Poscondiciones	La información de la Organización Estudiantil es recibida por el Presidente Nacional.
Prioridad	Alta
Curso Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El Presidente solicita una información.	2. El Organizador solicita la información al nivel de Provincia. 3. El Secretario de Funcionamiento Provincial solicita la información al nivel Municipal o CES. 4. El Secretario de Funcionamiento Municipal o CES solicita la información al nivel de Sedes o Facultades. 5. El Secretario de Funcionamiento de Sede o Facultad solicita la información

Capítulo 2. Características del sistema

	<p>al Jefe de Brigada.</p> <p>6. El Jefe de Brigada elabora un informe con los datos de la Brigada.</p> <p>7. El Jefe de Brigada envía la información al nivel superior (Nivel Sedes o Facultades).</p> <p>8. El Secretario de Funcionamiento de Sede o Facultades recibe la información de la Brigada.</p> <p>9. El Secretario de Funcionamiento de Sedes o Facultades elabora un informe con los datos de la Sede o bien de una Facultad.</p> <p>10. El Secretario de Funcionamiento de Sedes o Facultades envía la información al nivel superior (Nivel Municipal o CES).</p> <p>11. El Secretario de Funcionamiento Municipal o CES recibe la información de la Sede o Facultad.</p> <p>12. El Secretario de Funcionamiento Municipal o CES elabora un informe con los datos del Municipio o de una CES.</p> <p>13. El Secretario de Funcionamiento Municipal o CES envía la información al nivel superior (Nivel Provincial).</p> <p>14. El Secretario Provincial recibe la información del Municipio o de una CES.</p> <p>15. El Secretario Provincial elabora un informe con los datos de la Provincia.</p> <p>16. El Secretario Provincial envía la información al Organizador Nacional.</p> <p>17. El Organizador Nacional recibe la información de la Provincia.</p> <p>18. El Organizador Nacional elabora la información final.</p> <p>19. El Organizador Nacional se la envía al Presidente Nacional.</p>
20. El Presidente recibe la información.	

Capítulo 2. Características del sistema

2.6 Especificación de los requisitos

El objetivo de la captura de requisitos es guiar el desarrollo de software hacia el sistema correcto, definiendo objetivos generales concretos de manera tal que tanto el negocio como sus actores se beneficien. Se utilizan dos técnicas para lograr este objetivo: la entrevista y los cuestionarios.

2.6.1 Requerimientos funcionales

Los Requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

Los requisitos funcionales que se identificaron son los siguientes:

R1. Gestionar reportes

R1.1 Crear reportes.

1.1.1 Especificar el tipo de reporte que se quiere. Los reportes son:

- Reporte de tipo Cruzado.
- Reporte de tipo Listado.
- Reporte de tipo Conteo.

1.1.2 Definir el nivel (Nacional, Provincial, Municipal o CES, Facultad o Sede, Brigada) de acuerdo al rol que posee el usuario.

1.1.3 Registrar los datos del reporte creado.

R1.2 Eliminar reportes.

1.2.1 Mostrar el listado de carpetas.

1.2.2 Seleccionar una carpeta.

1.2.2.1 Mostrar los reportes que hay dentro de una carpeta seleccionada.

1.2.2.2 Seleccionar el reporte a eliminar.

1.2.2.3 Registrar la operación realizada.

Capítulo 2. Características del sistema

R1.3 Modificar un reporte.

1.3.1 Mostrar el listado de carpeta.

1.3.2 Seleccionar una carpeta.

1.3.2.1 Mostrar el listado de reportes que existen dentro de una carpeta.

1.3.2.2 Seleccionar el reporte que se desea modificar.

1.3.2.3 Cargar los datos del reporte seleccionado.

1.3.2.4 Modificar los datos del reporte seleccionado.

1.3.2.5 Registrar la modificación realizada en el reporte.

R2 Gestionar carpeta

R2.1 Crear una carpeta.

2.1.1 Introducir el nombre de la carpeta.

2.1.2 Crear un reporte si se desea. (Ver requisitos del R1.1 al 1.1.3)

2.1.3 Registrar los datos de la carpeta creada.

R2.2 Eliminar una carpeta.

2.2.1 Mostrar el listado de carpetas.

2.2.2 Seleccionar la carpeta que se desea eliminar.

2.2.3 Eliminar el contenido de la carpeta.

2.2.4 Registrar la operación realizada.

R2.3 Modificar una carpeta.

2.3.1 Mostrar el listado de carpetas.

2.3.2 Seleccionar la carpeta que se desea modificar.

2.3.3 Renombrar la carpeta seleccionada.

2.3.4 Registrar las modificaciones realizadas en la carpeta.

Capítulo 2. Características del sistema

R3 Consultar carpeta

R3.1 Mostrar el listado de carpeta.

R4.1 Seleccionar la carpeta a consultar.

4.1.1 Mostrar el listado de reportes en la carpeta.

4.1.2 Consultar un reporte si se desea. (Ver requerimientos del R5.1 al 5.2.1)

R5 Consultar reporte

R5.1 Mostrar el listado de reportes en la carpeta seleccionada.

R5.2 Seleccionar el reporte a consultar.

5.2.1 Visualizar los datos del reporte seleccionado.

2.6.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener.

Requisitos de Seguridad

- Realizar un control estricto de las acciones que lleven a cabo los usuarios en el sistema.

Requisitos de Implementación

- Emplear la tecnología J2EE como plataforma de desarrollo.
- Utilizar PostgreSQL como Sistema de Gestión de Base de Datos.
- Utilizar IDE Eclipse como entorno integrado de desarrollo.

Requisitos de Apariencia

- Poseer una interfaz amigable.
- Mostrar el explorador con los reportes distribuidos por carpetas.
- Explorar de manera sencilla.
- Mostrar las opciones que puede efectuar dentro del sistema según el rol del usuario.

Capítulo 2. Características del sistema

Requisitos de Software

Cliente:

- Sistema operativo con interfaz gráfica y soporte para red.
- Navegador Web.

Servidor:

- Windows 2000 Professional Service Pack 2.0 o superior.
- Apache Tomcat
- Bases de datos PostgreSQL

Requisitos de Hardware

Para el desarrollo:

- Pentium 2.4 GHz o superior.
- 512 MB de memoria RAM.
- 20 GB o más de disco duro.

Legales

El empleo del sistema debe estar regido por un manual de normas y procedimientos, que será aprobado por la Dirección de la Universidad y tiene que estar basado en las disposiciones legales vigentes.

2.7 Modelo del sistema

2.7.1 Actores del sistema

Los actores se corresponden en la mayoría de los casos con trabajadores y actores del negocio y representan los usuarios del sistema y otras aplicaciones que interactúan con él.

Capítulo 2. Características del sistema

Tabla 2.3 Actores del sistema

Actores del Sistema y su Justificación.	
Actores	Justificación
Usuario de Reporte	Es el encargado de realizar reportes beneficiándose con la obtención de la información en el sistema.

2.7.2 Diagrama de casos de uso general

El diagrama de casos de uso explica gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre estos.

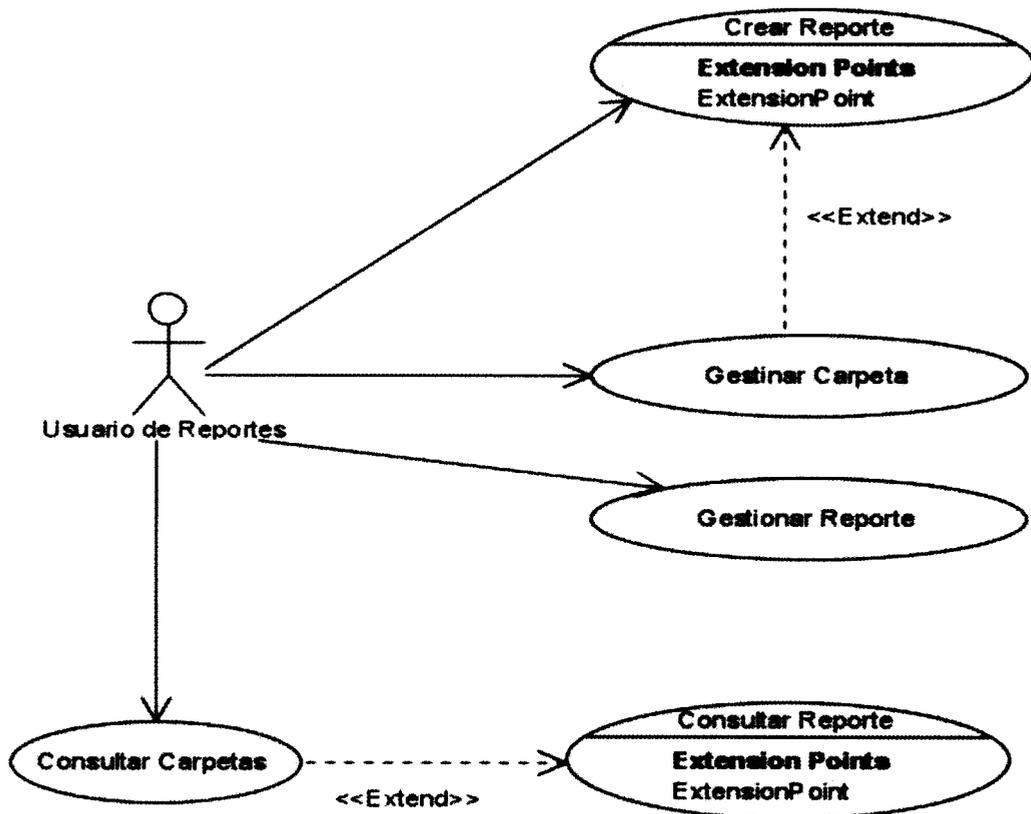


Figura 5 Diagrama de CU del sistema

Capítulo 2. Características del sistema

2.8 Descripción de los casos de uso del sistema

El objetivo principal de detallar cada caso de uso es describir su flujo de sucesos en detalle, incluyendo como comienza, termina e interactúa con los actores.

Tabla 2.4 Descripción del caso se uso: Gestionar Reporte

Especificación de Caso de Uso de Sistema	
Nombre	Gestionar Reporte
Descripción	El caso de uso se inicia cuando el Usuario de Reportes decide modificar o eliminar un reporte en el sistema. Posteriormente el Usuario de Reportes introduce los datos necesarios para realizar la operación deseada y el sistema modifica o elimina un reporte según la decisión del Usuario de Reportes, finalizando el caso de uso.
Actores	Usuario de Reportes
Precondiciones	El Usuario de Reportes se encuentra autenticado y ha obtenido los permisos necesarios para realizar las actividades según el rol que le corresponde.
Poscondiciones	El reporte queda modificado o eliminado en el sistema.
Relaciones	Include NA
	Extend NA
	Uses NA
Prioridad de Implantación	Crítica
Requisitos Funcionales	R1.1, 1.1.1, 1.1.2, 1.1.3, R1.2, 1.2.1, 1.2.2, 1.2.2.1, 1.2.2.2, 1.2.2.3, R1.3, 1.3.1, 1.3.2, 1.3.2.1, 1.3.2.2, 1.3.2.3, 1.3.2.4, 1.3.2.5.
Flujo de Eventos	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de gestionar	2. El sistema muestra las siguientes

Capítulo 2. Características del sistema

<p>reporte.</p>	<p>opciones:</p> <ul style="list-style-type: none"> a. Modificar reporte. b. Eliminar reporte.
<p>3. Selecciona la opción deseada.</p>	<p>4. El sistema realiza determinadas acciones en dependencia de la opción seleccionada.</p> <ul style="list-style-type: none"> a. Si selecciona la opción modificar reporte va a la sección "Modificar reporte". b. Si selecciona la opción eliminar reporte va a la sección "Eliminar reporte".
<p>Sección "Modificar reporte"</p>	
<p>Flujo Básico</p>	
Acción del Actor	Respuesta del Sistema
	<ul style="list-style-type: none"> 1. Comprueba que existen carpetas en el sistema, sino ver flujo alternado 1 de la sección "Modificar reporte". 1. Muestra las carpetas existentes. 2. Muestra una opción para seleccionar una carpeta.
<p>3. Selecciona la carpeta deseada.</p>	<ul style="list-style-type: none"> 4. Comprueba que existen reportes dentro de la carpeta, sino ver flujo alternado 2 de la sección "Modificar reporte" 5. Muestra los reportes existentes. 6. Muestra una opción para seleccionar el reporte deseado. 7. Muestra una opción para modificar el

Capítulo 2. Características del sistema

	<p>reporte seleccionado.</p> <p>8. Muestra una opción para cancelar la operación.</p>
<p>9. Selecciona el reporte a modificar.</p> <p>10. Selecciona la opción para modificar el reporte seleccionado, sino ver flujo alternado 3 de la sección "Modificar reporte".</p>	<p>11. Obtiene el tipo de reporte seleccionado.</p> <p>12. Carga los datos del reporte a modificar según el tipo.</p> <p>13. Muestra los datos del reporte según su tipo.</p> <p>14. Muestra una opción para modificarlos datos del reporte según su tipo.</p> <p>15. Muestra una opción para cancelar la operación.</p>
<p>16. Seleccionar la opción para modificar los datos del reporte según su tipo, sino ver flujo alternado 4.</p>	<p>17. Comprueba que no existen campos sin llenar, sino ver flujo alternado 5 de la sección "Modificar reporte".</p> <p>18. Registra los datos modificados.</p> <p>19. Muestra un mensaje indicando que la operación se realizó satisfactoriamente, finalizando el caso de uso.</p>
Flujo alternado 1	
Acción del Actor	Respuesta del Sistema
	<p>1. Comprueba que no existen carpetas en el sistema.</p> <p>2. Muestra un mensaje indicando que no existe una carpeta en el sistema.</p> <p>3. Va a la acción 2 del flujo básico.</p>
Flujo alternado 2	

Capítulo 2. Características del sistema

Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que no existen reportes dentro de la carpeta. 2. Muestra un mensaje indicando que no existen reportes dentro de la carpeta y va a la acción 2 del flujo básico.
Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Selecciona la opción para cancelar la operación. 	<ol style="list-style-type: none"> 2. Va a la acción 2 del flujo básico.
Flujo Alternado 4	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Selecciona la opción para cancelar la operación. 	<ol style="list-style-type: none"> 2. Va a la acción 9 del flujo básico de la sección "Modificar reporte".
Flujo Alternado 5	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que existe campos sin llenar. 2. Muestra un mensaje indicando cuales son los campos que deben ser llenados para completar la operación. 3. Va a la acción 16 del flujo básico de la sección "Modificar reporte"
Sección "Eliminar reporte"	
Flujo Básico	
Acción del Actor	Respuesta del Sistema

Capítulo 2. Características del sistema

	<ol style="list-style-type: none"> 1. Comprueba que existen carpetas en el sistema, sino ver flujo alternado 1 de la sección "Eliminar reporte". 2. Muestra todas las carpetas existentes. 3. Muestra una opción para seleccionar la carpeta.
4. Selecciona la carpeta deseada.	<ol style="list-style-type: none"> 5. Comprueba que existen reportes dentro de la carpeta, sino ver flujo alternado 2 de la sección "Eliminar reporte". 6. Muestra todos los reportes que existen en la carpeta seleccionada. 7. Muestra una opción para seleccionar lo(s) reporte(s) a eliminar. 8. Muestra una opción para eliminar lo(s) reporte(s) seleccionado(s). 9. Muestra una opción para cancelar la operación.
<ol style="list-style-type: none"> 10. Selecciona lo(s) reporte(s) a eliminar. 11. Selecciona la opción para eliminar lo(s) reporte(s) seleccionado(s), sino ver flujo alternado 2 de la sección "Modificar reporte". 	<ol style="list-style-type: none"> 12. Comprueba que al menos fue seleccionado un reporte, sino ver flujo alternado 3 de la sección "Eliminar carpeta". 13. Elimina el reporte, finalizando el caso de uso.
Flujo Alternado 1	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que no existen carpetas en el sistema. 2. Muestra un mensaje indicando que no existe una carpeta en el sistema.

Capítulo 2. Características del sistema

	3. Va a la acción 2 del flujo básico.
Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que no existen reportes dentro de la carpeta seleccionada. 2. Muestra un mensaje indicando que no existe un reporte dentro de la carpeta seleccionada. 3. Va a la acción 2 del flujo básico.
Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que no fue seleccionado ningún reporte. 2. Muestra un mensaje indicando que debe seleccionar un reporte 3. Va a la acción 10 del flujo básico de la sección "Eliminar reporte".
Interfaces del caso de uso	

Capítulo 2. Características del sistema

The image displays three sequential screenshots of a web application interface for report management, labeled 1, 2, and 3.

- Screenshot 1:** Shows a menu titled "Gestionar Reporte" with three options: "Crear Reporte", "Modificar Reporte", and "Eliminar Reporte".
- Screenshot 2:** Shows the "Eliminar Reporte" form. It includes a text input field for "Nombre del reporte", a dropdown menu for "Carpeta a seleccionar" (currently showing "carpeta nacional"), and two buttons: "Eliminar" and "Cancelar".
- Screenshot 3:** Shows the "Modificar Reporte" form. It includes a text input field for "Nombre del reporte", a dropdown menu for "Carpeta a seleccionar" (currently showing "carpeta nacional"), a dropdown menu for "Reporte a seleccionar" (currently showing "reporte 1"), and two buttons: "Cargar" and "Cancelar".

Tabla 2.5 Descripción de caso de uso: Gestionar Carpeta

Especificación de Caso de Uso de Sistema	
Nombre	Gestionar Carpeta
Descripción	El caso de uso se inicia cuando el Usuario de Reportes decide crear, modificar o eliminar una carpeta en el sistema. Posteriormente el Usuario de Reportes introduce los datos necesarios para realizar la operación deseada y el sistema crea, modifica o elimina una carpeta según la decisión del Usuario de Reportes, finalizando el caso de uso.
Actores	Usuario de Reportes
Precondiciones	El Usuario de Reportes se encuentra autenticado y ha obtenido los permisos necesarios para realizar las actividades según el rol que le corresponde.
Poscondiciones	La carpeta queda creada o modificada o eliminada en el sistema.
	Include NA

Capítulo 2. Características del sistema

Relaciones	Extend	Crear Reporte
	Uses	NA
Prioridad de Implantación		Crítica.
Requisitos Funcionales		R2, R2.1, 2.1.1, 2.1.2, R1.1, 1.1.1, 1.1.2, 1.1.3, 2.1.3, R2.2, 2.2.1, 2.2.2, 2.2.3, 2.2.4, R2.3, 2.3.1, 2.3.2, 2.3.3, 2.3.4.
Flujo de Eventos		
Flujo Básico		
Acción del Actor		Respuesta del Sistema
1. Selecciona la opción de gestionar carpeta.		2. El sistema muestra las siguientes opciones: <ul style="list-style-type: none"> a. Crear carpeta. b. Modificar carpeta. c. Eliminar carpeta.
3. Selecciona la opción deseada.		4. El sistema realiza determinadas acciones en dependencia de la opción seleccionada. <ul style="list-style-type: none"> a. Si selecciona la opción crear carpeta va a la sección "Crear carpeta". b. Si selecciona la opción modificar carpeta va a la sección "Modificar carpeta". c. Si selecciona la opción Eliminar carpeta va a la sección "Eliminar carpeta".
Sección "Crear carpeta"		
Flujo Básico		
Acción del Actor		Respuesta del Sistema

Capítulo 2. Características del sistema

	<ol style="list-style-type: none"> 1. Muestra los campos para introducir los siguientes datos: <ul style="list-style-type: none"> • Nombre de la carpeta 2. Muestra una opción para registrar la operación. 3. Muestra una opción para cancelar la operación.
<ol style="list-style-type: none"> 4. Introduce los datos necesarios para crear la carpeta. 5. Selecciona la opción para registrar la operación, sino ver flujo alternado 1 de la sección "Crear carpeta". 	<ol style="list-style-type: none"> 6. Comprueba que no existen campos sin llenar, sino ver flujo alternado 2 de la sección "Crear carpeta". 7. Comprueba que no existe otra carpeta con el mismo nombre, sino ver flujo alternado 3 de la sección "Crear carpeta". 8. Registra los datos de la carpeta: <ul style="list-style-type: none"> • Nombre de la carpeta. 9. Muestra un mensaje indicando que la operación se realizó satisfactoriamente. 10. Muestra una opción para crear un reporte dentro de la nueva carpeta. 11. Muestra una opción para concluir la operación.
<ol style="list-style-type: none"> 12. Selecciona la opción para crear un nuevo reporte, sino ver flujo alternado 4 de la sección "Crear Carpeta". 	<ol style="list-style-type: none"> 13. Va a la acción 13 del flujo básico del caso de uso extendido Crear Reporte, finalizando el caso de uso.
Flujo Alternado 1	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Selecciona la opción para cancelar la operación. 	<ol style="list-style-type: none"> 2. Va a la acción 2 del flujo básico.

Capítulo 2. Características del sistema

Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que existen campos sin llenar. 2. Muestra un mensaje indicando cuales son los campos que deben ser llenados para completar la operación. 3. Va a la acción 4 del flujo básico de la sección "Crear carpeta".
Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que existe otra carpeta con el mismo nombre. 2. Muestra un mensaje indicando que existe una carpeta con el nombre introducido. 3. Va a la acción 4 del flujo básico de la sección "Crear carpeta".
Flujo Alternado 4	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción para concluir la operación y finaliza el caso de uso.	
Sección "Modificar carpeta"	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que existen carpetas en el sistema, sino finaliza el caso de uso. 2. Muestra todas las carpetas existentes.

Capítulo 2. Características del sistema

	<ol style="list-style-type: none"> 3. Muestra una opción para seleccionar una carpeta. 4. Muestra una opción para modificar la carpeta seleccionada. 5. Muestra una opción para cancelar la operación.
<ol style="list-style-type: none"> 6. Selecciona la carpeta a modificar. 7. Selecciona la opción para modificar la carpeta seleccionada, sino ver flujo alternado 1 de la sección "Crear carpeta". 	<ol style="list-style-type: none"> 8. Comprueba que no existen campos sin llenar, sino ver flujo alternado 2 de la sección "Modificar carpeta". 9. Comprueba que no existe otra carpeta con el mismo nombre, sino ver flujo alternado 3 de la sección "Modificar carpeta" 10. Registra los datos modificados de la carpeta: <ul style="list-style-type: none"> • Nombre de la carpeta. 11. Muestra un mensaje indicando que la operación se realizó satisfactoriamente, finalizando el caso de uso.
Flujo Alternado 1	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que no existen carpetas en el sistema. 2. Muestra un mensaje indicando que no existen carpetas en el sistema y va a la acción 2 del flujo básico.
Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que existen campos sin llenar. 2. Muestra un mensaje indicando cuales son los campos que deben ser llenados para

Capítulo 2. Características del sistema

	<p>completar la operación.</p> <p>3. Va a la acción 6 del flujo básico de la sección "Modificar carpeta".</p>
Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que existe otra carpeta con el mismo nombre. 2. Muestra un mensaje indicando que existe una carpeta con el nombre introducido. 3. Va a la acción 6 del flujo básico de la sección "Modificar carpeta".
Sección "Eliminar carpeta"	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que existen carpetas en el sistema, sino ver flujo alternado 1 de la sección "Modificar carpeta" 2. Muestra todas las carpetas existentes. 3. Muestra una opción para seleccionar la(s) carpeta(s) a eliminar. 4. Muestra una opción para eliminar la(s) carpeta(s) seleccionada(s). 5. Muestra una opción para cancelar la operación.
<ol style="list-style-type: none"> 6. Selecciona la(s) carpeta(s) a eliminar. 7. Selecciona la opción para eliminar la(s) carpeta(s) seleccionada(s), sino ver flujo alternado 1 de la sección "Crear 	<ol style="list-style-type: none"> 8. Comprueba que al menos fue seleccionada una carpeta, sino ver flujo alternado 1 de la sección "Eliminar carpeta". 9. Comprueba que la(s) carpeta(s)

Capítulo 2. Características del sistema

carpeta”.	<p>seleccionada(s) tiene(n) contenido, sino ver flujo alternado 2 de la sección “Eliminar carpeta”.</p> <p>10. Elimina el contenido de la carpeta.</p> <p>11. Elimina la carpeta, finalizando el caso de uso.</p>
Flujo Alternado 1	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que no fue seleccionada ninguna carpeta. 2. Muestra un mensaje indicando que debe seleccionar una carpeta. 3. Va a la acción 6 del flujo básico de la sección “Eliminar carpeta”.
Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que la(s) carpeta(s) seleccionada(s) no tiene(n) contenido. 2. Va a la acción 11 del flujo básico de la sección “Eliminar carpeta”.
Interfaces del caso de uso	

Capítulo 2. Características del sistema

1

2

3

4

Tabla 2.6 Descripción del caso se uso: Consultar Carpeta

Especificación de Caso de Uso de Sistema	
Nombre	Consultar Carpeta
Descripción	El caso de uso se inicia cuando el Usuario de Reportes decide consultar el contenido de una carpeta del sistema, se muestra el contenido de la misma y finaliza el caso de uso.
Actores	Usuario de Reportes
Precondiciones	El Usuario de Reportes se encuentra autenticado y ha obtenido los permisos necesarios para realizar las actividades según el rol que le corresponde.
Poscondiciones	Queda consultado el contenido de la carpeta.

Capítulo 2. Características del sistema

Relaciones	Include	NA
	Extend	Consultar Reporte
	Uses	NA
Prioridad de Implantación		Crítica.
Requisitos Funcionales		R3.1, R4.1, 4.1.1, 4.1.2, R5.1, R5.2, 5.2.1.
Flujo de Eventos		
Flujo Básico		
Acción del Actor		Respuesta del Sistema
<p>1. Solicita consultar el contenido de una carpeta.</p>		<p>2. Comprueba que existen carpetas en el sistema, sino ver flujo alternado 1.</p> <p>3. Muestra las carpetas existentes en el sistema.</p> <p>4. Muestra una opción para seleccionar la carpeta a consultar.</p> <p>5. Muestra una opción para consultar la carpeta seleccionada.</p> <p>6. Muestra una opción para cancelar la operación.</p>
<p>7. Selecciona la carpeta a consultar.</p> <p>8. Selecciona la opción para consultar la carpeta, sino ver flujo alternado 2.</p>		<p>9. Comprueba que existen reportes dentro de la carpeta seleccionada, sino ver flujo alternado 3.</p> <p>10. Muestra una opción para seleccionar el reporte que se desea consultar.</p> <p>11. Muestra una opción para consultar el reporte.</p> <p>12. Muestra una opción para cancelar la</p>

Capítulo 2. Características del sistema

	operación.
13. Selecciona el reporte que se desea consultar. 14. Selecciona la opción para consultar el reporte, sino ver flujo alterado 4.	15. Va al caso de uso extendido (Consultar Reporte).
Flujo Alternado 1	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que no existen carpetas en el sistema. 2. Muestra un mensaje indicando que no existen carpetas en el sistema, finalizando el caso de uso.
Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción para cancelar la operación, finalizando el caso de uso.	
Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que no existen reportes dentro de la carpeta seleccionada. 2. Muestra un mensaje indicando que no existen reportes dentro de la carpeta seleccionada. 3. Va a la acción 7 del flujo básico.

Capítulo 2. Características del sistema

Flujo Alternado 4	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción para cancelar la operación.	2. Va a la acción 7 del flujo básico.
Interfaces del caso de uso	
<p>Consultar Carpeta</p> <p>Seleccionar carpeta</p> <p>carpeta 1 ▾</p> <p>Aceptar Cancelar</p> <p>1</p>	<p>Consultar Carpeta</p> <p>Seleccionar reporte</p> <p>reporte 1 ▾</p> <p>Aceptar Cancelar</p> <p>2</p>

Tabla 2.7 Descripción del caso se uso: Consultar Reporte

Especificación de Caso de Uso de Sistema	
Nombre	Consultar Reporte
Descripción	El caso de uso se inicia cuando el Usuario de Reportes decide consultar el contenido de un reporte que existe dentro de una carpeta, el sistema muestra el contenido del reporte y finaliza el caso de uso.
Actores	Usuario de Reportes
Precondiciones	El Usuario de Reportes se encuentra autenticado y ha obtenido los permisos necesarios para realizar las actividades según el rol que le corresponde. Se realizó en el sistema una consulta de carpetas.
Poscondiciones	Queda consultado el contenido del reporte.
	Include NA

Capítulo 2. Características del sistema

Relaciones	Extend	NA
	Uses	NA
Prioridad de Implantación		Crítica.
Requisitos Funcionales		R5.1, R5.2, 5.2.1.
Flujo de Eventos		
Flujo Básico		
Acción del Actor		Respuesta del Sistema
		<ol style="list-style-type: none"> 1. Obtiene los datos del reporte seleccionado. 2. Muestra el los datos del reporte seleccionado. 3. Muestra una opción de Imprimir. 4. Muestra una opción para cancelar la operación.
5. Selecciona la opción para imprimir el reporte, sino ver flujo alternado 1.		6. Imprime el reporte finalizando el caso de uso.
Flujo Alternado 4		
Acción del Actor		Respuesta del Sistema
1. Selecciona la opción para cancelar la operación y finaliza el caso de uso.		
Interfaces del caso de uso		

Capítulo 2. Características del sistema

Nombre del reporte: Reporte 1		
Nivel: Bridaga		
Sexo		
F	M	
15	16	
Raza		
B	N	M
8	6	17

Tabla 2.8 Descripción del caso de uso: Crear Reporte

Especificación de Caso de Uso de Sistema		
Nombre	Crear reporte.	
Descripción	El caso de uso se inicia cuando el Usuario de Reportes decide crear un reporte, para lo cual introduce todos los datos necesarios para realizar la operación en el sistema y este registra la operación, finalizando el caso de uso.	
Actores	Usuario de Reportes	
Precondiciones	El Usuario de Reportes se encuentra autenticado y ha obtenido los permisos necesarios para realizar las actividades según el rol que le corresponde.	
Poscondiciones	Queda creado un reporte en el sistema.	
Relaciones	Include	NA
	Extend	NA
	Uses	NA

Capítulo 2. Características del sistema

Prioridad de Implantación	Crítica.
Requisitos Funcionales	R1.1, 1.1.1, 1.1.2, 1.1.3.
Flujo de Eventos	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. Solicita crear un reporte.	2. Comprueba que existen carpetas en el sistema, sino ver flujo alternado 1. 3. Muestra las carpetas existentes en el sistema. 4. Muestra una opción para seleccionar la carpeta a consultar. 5. Muestra una opción para consultar la carpeta seleccionada. 6. Muestra una opción para cancelar la operación.
7. Selecciona la carpeta a consultar. 8. Selecciona la opción para consultar la carpeta, sino ver flujo alternado 2.	9. Muestra una opción para seleccionar el tipo de reporte. 10. Muestra una opción para seleccionar el nivel de la organización de la cual se desea obtener el reporte. 11. Muestra una opción para aceptar la operación. 12. Muestra una opción para cancelar la operación.
13. Selecciona el tipo de reporte que se desea crear.	16. Muestra las opciones según el tipo de reporte seleccionado.

Capítulo 2. Características del sistema

<p>14. Selecciona el nivel de la organización.</p> <p>15. Selecciona la opción para aceptar la operación, sino ver flujo alternado 3.</p>	<p>a. Si selecciona el tipo de reporte “Cruzado” va a la sección “Reporte Cruzado”.</p> <p>b. Si selecciona el tipo de reporte “Listado” va a la sección “Reporte Listado”.</p> <p>c. Si selecciona el tipo de reporte “Conteo” va a la sección “Reporte Conteo”.</p>
Flujo Alternado 1	
Acción del Actor	Respuesta del Sistema
	<p>1. Comprueba que no existen carpetas en el sistema.</p> <p>2. Muestra un mensaje indicando que no existen carpetas en el sistema, finalizando el caso de uso.</p>
Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
<p>1. Selecciona la opción para cancelar la operación, finalizando el caso de uso.</p>	
Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
<p>1. Selecciona la opción para cancelar la operación.</p>	<p>2. Va a la acción 7 del flujo básico.</p>
Sección “Reporte Cruzado”	

Capítulo 2. Características del sistema

Flujo Básico	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Muestra una opción para introducir el nombre del reporte. 2. Muestra el campo A donde se introducirá el parámetro para realizar el filtrado. 3. Muestra el campo B donde se introducirá el parámetro para realizar el filtrado. 4. Se muestra una opción para aceptar la operación. 5. Se muestra una opción para cancelar la operación.
<ol style="list-style-type: none"> 6. Introduce el nombre del reporte. 7. Introduce el campo A. 8. Introduce el campo B. 	<ol style="list-style-type: none"> 9. Comprueba si no existe un reporte con el mismo nombre, sino ver flujo alternado 1 de la sección "Reporte cruzado". 10. Comprueba que los campos estén llenos, sino ver flujo alterno 2 de la sección "Reporte cruzado". 11. Selecciona la opción aceptar, guardando la operación, sino ver flujo alternado 3 de la sección "Reporte cruzado". 12. Guarda los datos del reporte, finalizando el caso de uso.
Flujo Alternado 1	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba si existe un reporte con el mismo nombre. 2. Muestra un mensaje indicando que existe

Capítulo 2. Características del sistema

	<p>una carpeta con el mismo nombre.</p> <p>3. Va a la acción 6 de la sección "Reporte cruzado."</p>
Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que los campos están sin llenar. 2. Muestra un mensaje indicando que los campos están sin llenar. 3. Va a la acción 16 del flujo básico.
Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Selecciona la opción para cancelar la operación. 	<ol style="list-style-type: none"> 2. Va a la acción 16 del flujo básico.
Sección "Reporte Listado"	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Muestra una opción para introducir el nombre del reporte. 2. Muestra el campo A donde se introducirá el parámetro para realizar el filtrado. 3. Se muestra una opción para aceptar la operación. 4. Se muestra una opción para cancelar la operación.

Capítulo 2. Características del sistema

<p>5. Introduce el nombre del reporte.</p> <p>6. Introduce el campo A.</p>	<p>7. Comprueba si no existe un reporte con el mismo nombre, sino ver flujo alternado 1 de la sección "Reporte listado".</p> <p>8. Comprueba que los campos estén llenos, sino ver flujo alterno 2 de la sección "Reporte cruzado".</p> <p>9. Selecciona la opción aceptar, guardando la operación, sino ver flujo alternado 3 de la sección "Reporte listado".</p> <p>10. Guarda los datos del reporte, finalizando el caso de uso.</p>
Flujo Alternado 1	
Acción del Actor	Respuesta del Sistema
	<p>1. Comprueba si existe un reporte con el mismo nombre.</p> <p>2. Muestra un mensaje indicando que existe una carpeta con el mismo nombre.</p> <p>3. Va a la acción 6 de la sección "Reporte listado."</p>
Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
	<p>1. Comprueba que los campos están sin llenar.</p> <p>2. Muestra un mensaje indicando que los campos están sin llenar.</p> <p>3. Va a la acción 16 del flujo básico.</p>

Capítulo 2. Características del sistema

Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción para cancelar la operación.	2. Va a la acción 16 del flujo básico.
Sección "Reporte Conteo"	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
	1. Muestra una opción para introducir el nombre del reporte. 2. Muestra el campo A donde se introducirá el parámetro para realizar el filtrado. 3. Se muestra una opción para aceptar la operación. 4. Se muestra una opción para cancelar la operación.
5. Introduce el nombre del reporte. 6. Introduce el campo A.	7. Comprueba si no existe un reporte con el mismo nombre, sino ver flujo alternado 1 de la sección "Reporte conteo". 8. Comprueba que los campos estén llenos, sino ver flujo alterno 2 de la sección "Reporte conteo". 9. Selecciona la opción aceptar, guardando la operación, sino ver flujo alternado 3 de la sección "Reporte conteo". 10. Guarda los datos del reporte, finalizando el caso de uso.
Flujo Alternado 1	

Capítulo 2. Características del sistema

Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba si existe un reporte con el mismo nombre. 2. Muestra un mensaje indicando que existe una carpeta con el mismo nombre. 3. Va a la acción 6 de la sección "Reporte conteo."
Flujo Alternado 2	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Comprueba que los campos están sin llenar. 2. Muestra un mensaje indicando que los campos están sin llenar. 3. Va a la acción 16 del flujo básico.
Flujo Alternado 3	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. Selecciona la opción para cancelar la operación. 	<ol style="list-style-type: none"> 2. Va a la acción 16 del flujo básico.
Interfaces del caso de uso	

Capítulo 2. Características del sistema

The diagram illustrates a five-step process for report management:

- 1. Gestionar Reporte:** A menu with options:
 - Crear Reporte
 - Modificar Reporte
 - Eliminar Reporte
- 2. Crear Reporte:** A form with fields:
 - Tipo de Reporte: Reporte Conteo
 - Nivel: Nacional
 - Botón: Siguiente
- 3. Reporte "Listado":** A form with fields:
 - Nombre del reporte
 - Campo 1: Sexo
 - Botones: Crear, Ir Atrás
- 4. Reporte "Listado":** A form with fields:
 - Nombre del reporte
 - Campo 1: Sexo
 - Botones: Crear, Ir Atrás
- 5. Reporte "Cruzado":** A form with fields:
 - Nombre del reporte
 - Campo 1: Sexo
 - Campo 2: Raza
 - Botones: Crear, Ir Atrás

2.10 Conclusiones

En este capítulo se ha comprendido con profundidad las características del sistema a automatizar, teniendo como resultado el modelo del negocio y los actores que intervienen en este. El caso de uso del negocio se describió en formato expandido y se elaboró su diagrama de actividades. Se realizó la captura de requisitos y se hizo una propuesta del sistema según las necesidades de los usuarios. Además se obtuvo el diagrama de casos de uso del sistema y la descripción de cada uno de ellos. Al terminar este flujo de trabajo se puede comenzar el análisis y diseño a partir de los artefactos obtenidos.

3

Capítulo 3 Análisis y Diseño del sistema

3.1 Introducción

En este capítulo se presenta el modelo del análisis y el modelo del diseño, donde son expuestas las realizaciones de los casos de uso definidos en el capítulo anterior mediante diagramas de clases y diagramas de interacción. Se describen las clases que se obtienen en el diseño y se realiza la estimación de costo-beneficio por Puntos de Caso de Uso obteniendo un resultado que no reporta beneficio monetario.

3.2 Descripción de la arquitectura a utilizar

El modelado del diseño va a tener como base una arquitectura de tres capas, donde en cada capa se hace referencia a los componentes que presentan los niveles inferiores.

Presentación: contiene todos los aspectos del software que tienen que ver con las interfaces y la interacción con los diferentes tipos de usuarios. Estos aspectos típicamente incluyen el manejo e imagen de las ventanas, el formato de los reportes, gráficos y elementos multimedia en general.

Capa de Negocio: reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican.

Capa de Acceso a Datos: encargada del manejo de los datos persistentes, por lo que también se le denomina capa de persistencia.

En la figura # queda representada la arquitectura a utilizar.

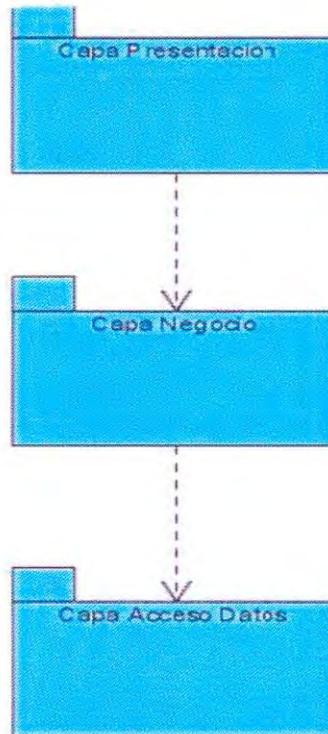


Figura 6 Arquitectura de tres capas

3.2.1 Patrones de diseño utilizados

Un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. Un patrón de diseño es básicamente una solución (un diseño) que surge de la experimentación práctica con varios proyectos, y los equipos de desarrollo han encontrado que se puede aplicar en diversos contextos.

Los patrones utilizados para el diseño fueron principalmente el patrón experto y el creador, ambos pertenecientes al grupo de patrones de asignación de responsabilidades. El experto consiste en asignar la responsabilidad a aquella clase que tenga la información necesaria para cumplir con la misma, mientras que el creador se basa en asignar la responsabilidad de crear una instancia de una clase a otra clase, si esta última contiene, agrega, registra, utiliza o posee los datos de inicialización de la primera.

Capítulo 3. Análisis y Diseño del sistema

3.3 Modelado del diseño

El diseño se ha escrito como un proceso multifase en el que se sintetizan representaciones de la estructura de datos, la estructura del programa, las características de la interfaz y los detalles procedimentales desde los requisitos de la información. (Pressman).

El modelo de diseño intenta preservar la estructura definida por el modelo de análisis. Es un modelo de objetos que describe la realización física de los casos de uso, además de tener en cuenta los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, tecnologías de gestión de transacciones, entre otras. Otro de los aspectos a tener en cuenta del diseño es que crea una entrada apropiada y un punto de partida para actividades de implementación.

O sea en el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, que proporciona una comprensión detallada de los requisitos. (11)

3.3.1 Diagrama de diseño web del sistema

Durante la construcción de este diagrama es imposible aplicar la forma tradicional para modelar los diagramas de clases. Esto se debe fundamentalmente a que en esta modelación una página Web es considerada una clase y por tanto a la hora de diseñar la aplicación Web no permite identificar qué atributos y operaciones se mantienen en ejecución, además no se puede especificar cuáles de estos se activan al estar el usuario interactuando con dicha página.

A continuación se muestran ejemplos de diagramas de clases Web, pertenecientes a los casos de usos Consultar Carpeta y Gestionar Carpeta (Crear Carpeta).

Detalles sobre los restantes diagramas de diseño web [Ver anexo 1]

Capítulo 3. Análisis y Diseño del sistema

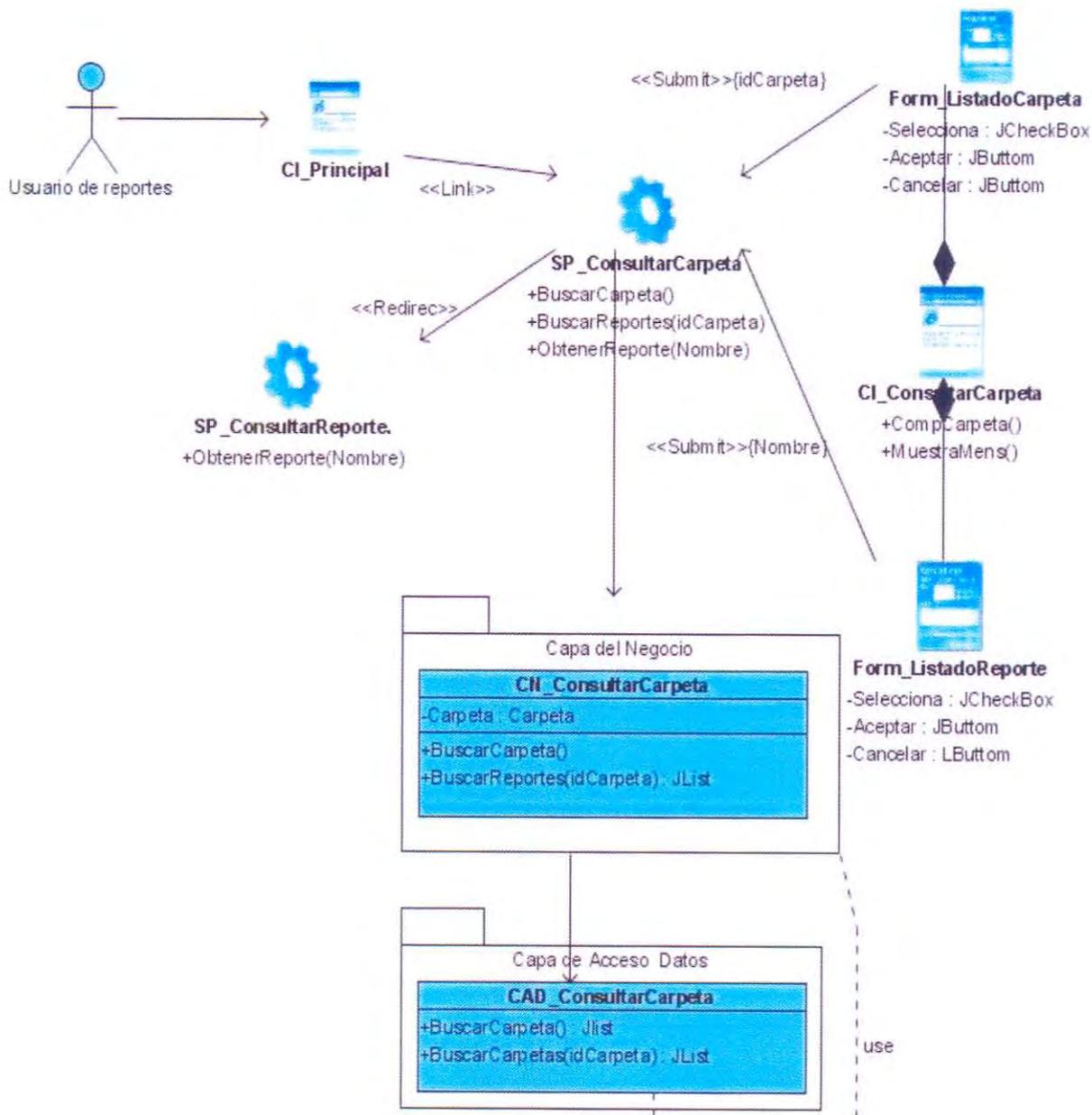


Figura 7 Diagrama de clases web Consultar Carpeta 1 Parte

Capítulo 3. Análisis y Diseño del sistema

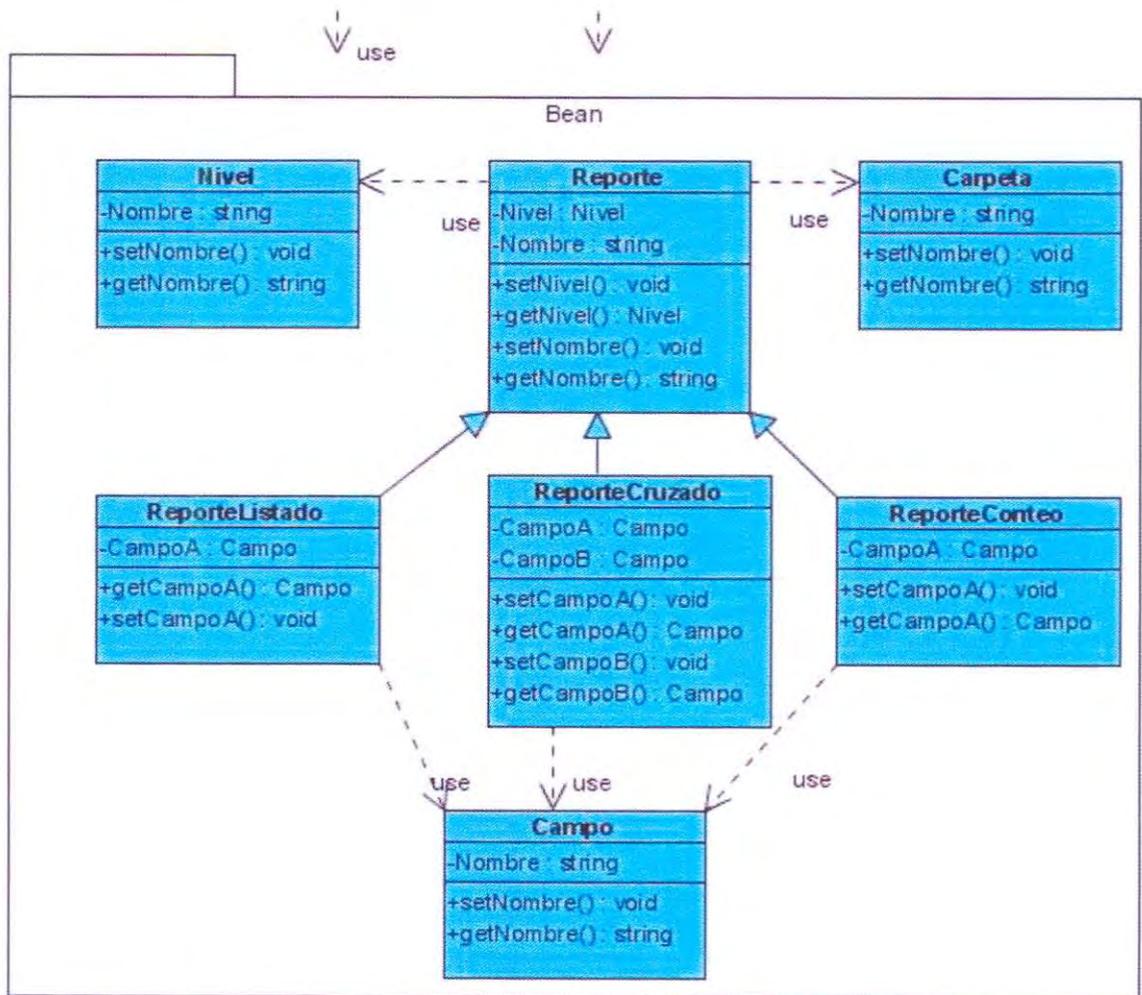


Figura 8 Diagrama de clases web Consultar Carpeta 2 Parte

Capítulo 3. Análisis y Diseño del sistema

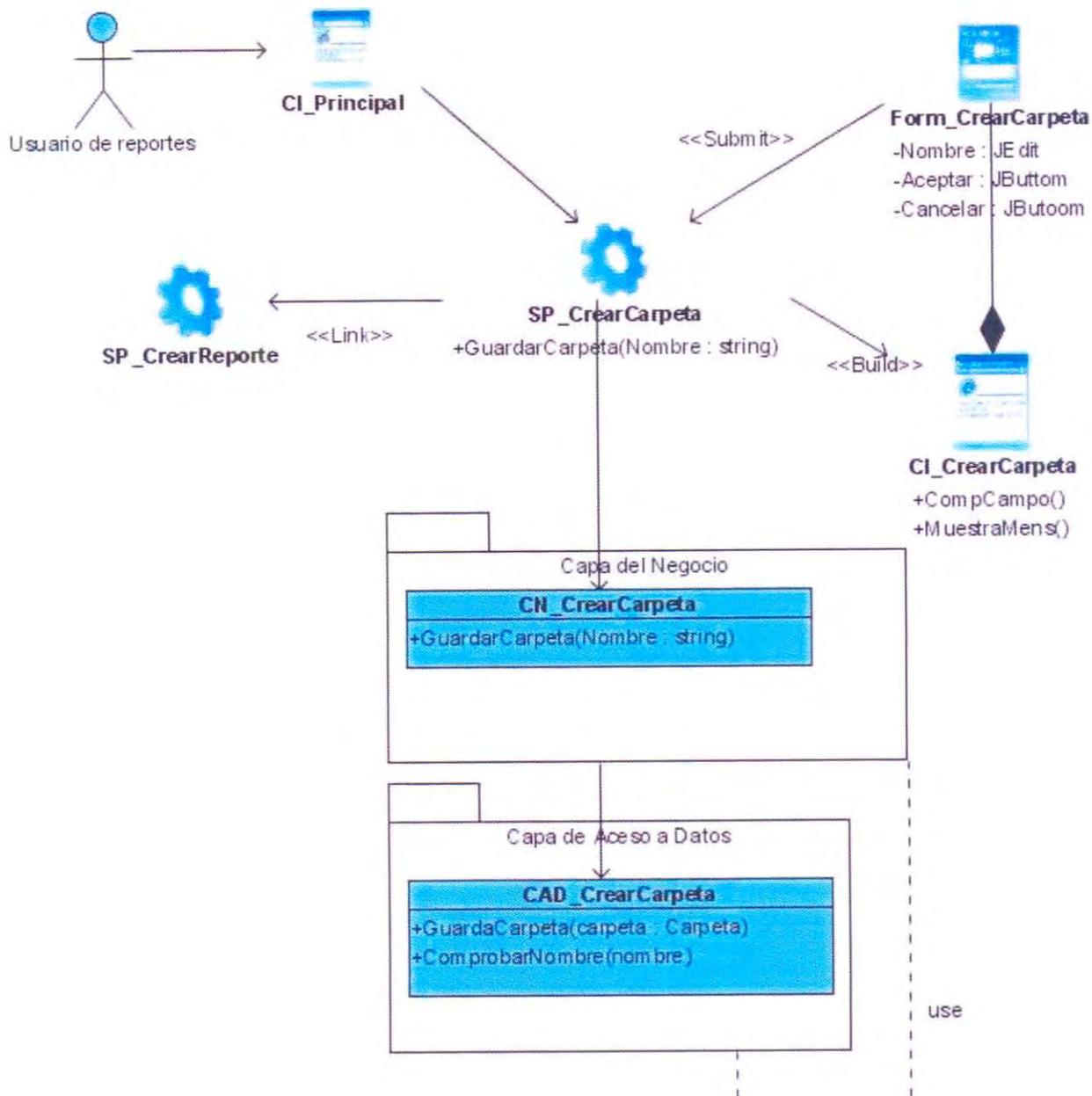


Figura 9 Diagrama de clases web Crear Carpeta 1 Parte

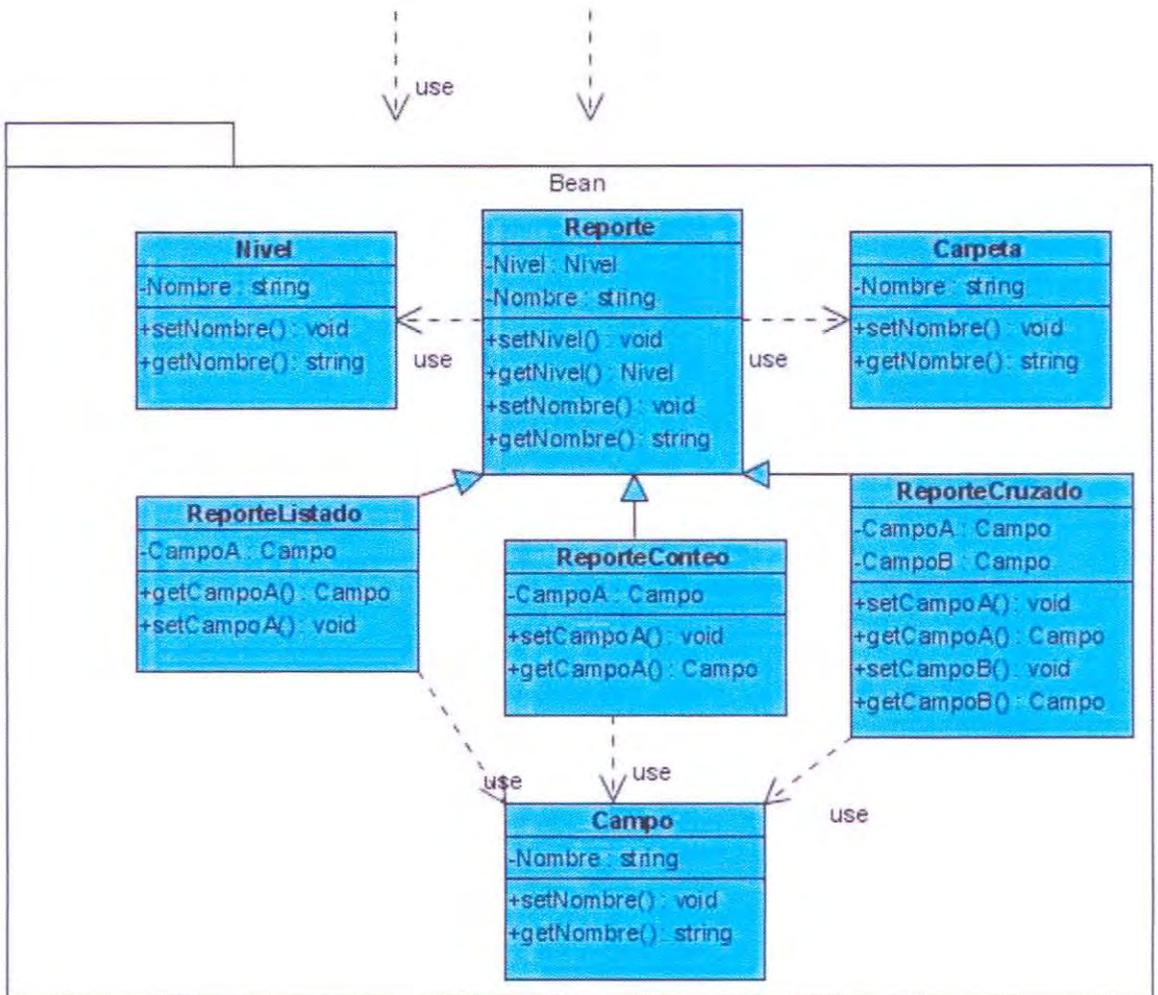


Figura 10 Diagrama de clases web Crear Carpeta 2 Parte

3.3.2 La vista de interacción

Los objetos obran recíprocamente para implementar comportamiento. Esta interacción se puede describir de dos maneras complementarias, una de ellas se centra en los objetos individuales y la otra en una colección de objetos cooperantes. Una máquina de estados es una vista estrecha y profunda del comportamiento, una vista reduccionista que mira a cada objeto individualmente.

Capítulo 3. Análisis y Diseño del sistema

Una especificación de la máquina de estados es exacta y conduce inmediatamente al código. Sin embargo, puede ser difícil entender el funcionamiento total de un sistema, debido a que una máquina de estados se centra en un solo objeto a la vez, y se deben combinar los efectos de muchas máquinas de estado para determinar el comportamiento de todo el sistema. La vista de interacción proporciona una visión más integral del comportamiento de un sistema de objetos. Esta vista es modelada por colaboraciones.

3.3.3 Interacción

Una interacción es un conjunto de mensajes dentro de una colaboración que son intercambiados por roles de clasificador a través de roles de asociación. Una interacción modela la ejecución de una operación, caso de uso, u otra entidad de comportamiento. Un mensaje es una comunicación unidireccional entre dos objetos, un flujo de objeto con la información de un remitente a un receptor. Un mensaje puede tener parámetros que transporten valores entre los objetos. Un mensaje puede ser una señal (una comunicación explícita entre objetos, con nombre, y asíncrona) o una llamada (la invocación síncrona de una operación con un mecanismo para el control, que retorna posteriormente al remitente). La creación de un nuevo objeto se modela como un evento causado por el objeto creador y recibido por la propia clase.

3.3.4 Diagrama de secuencia

El tipo de diagrama seleccionado para construir los diagramas de interacción fue el de secuencia, debido a que muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente.

Detalles sobre los restantes diagramas de secuencia [Ver anexo 2]

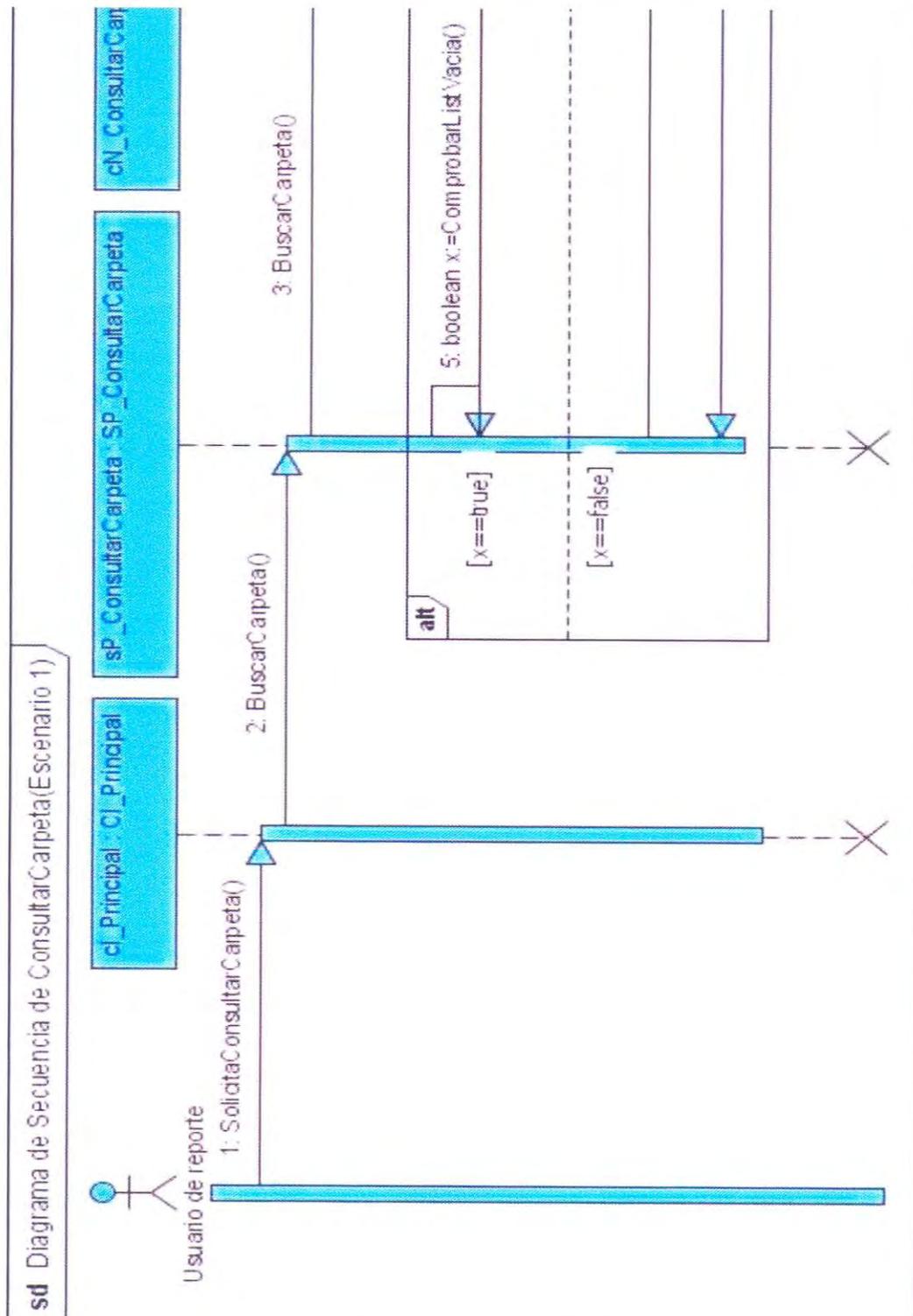


Figura 11 Diagrama de secuencia Consultar Carpeta (Escenario 1) 1 Parte

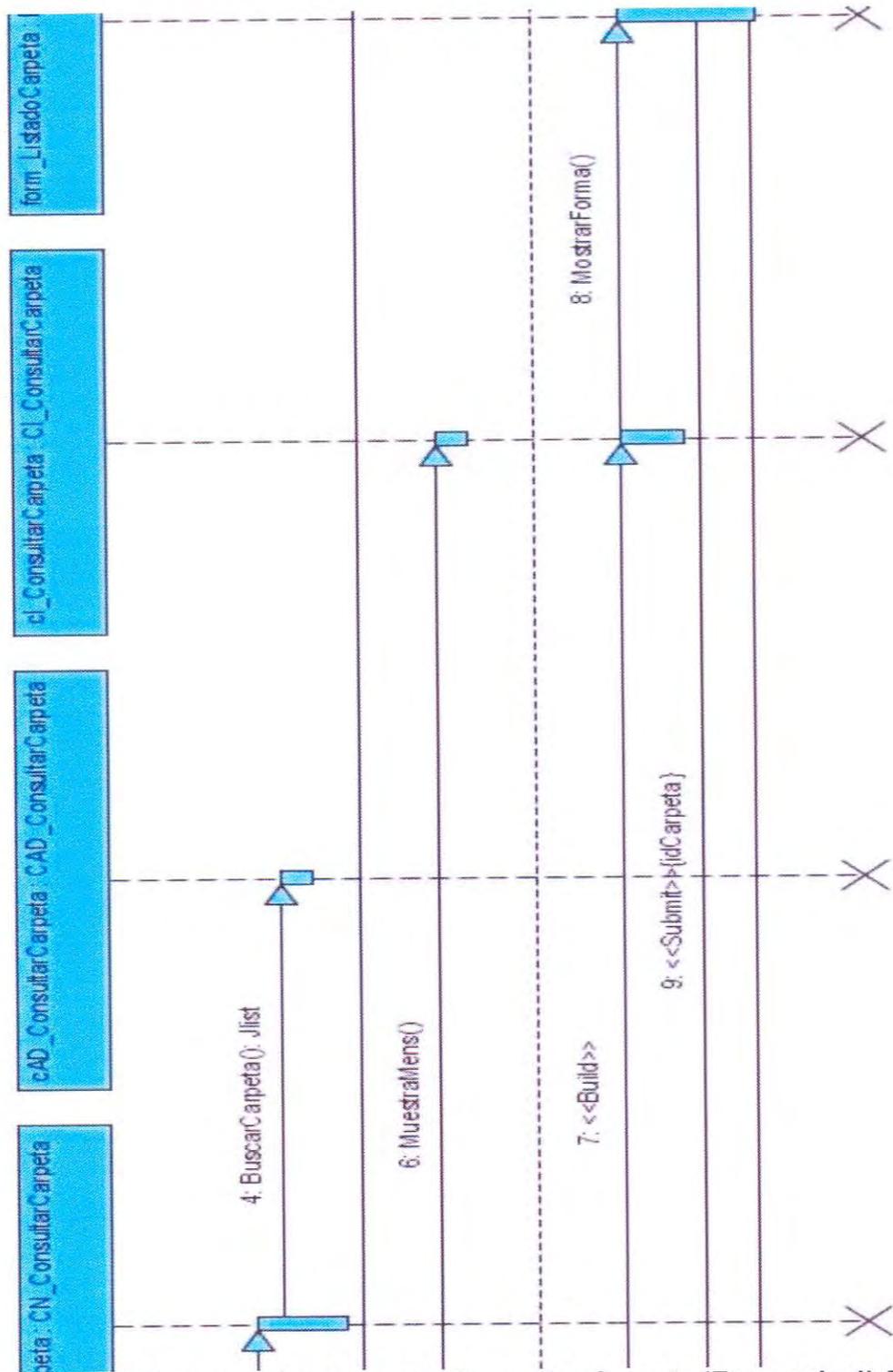


Figura 12 Diagrama de secuencia Consultar Carpeta (Escenario 1) 2 Parte

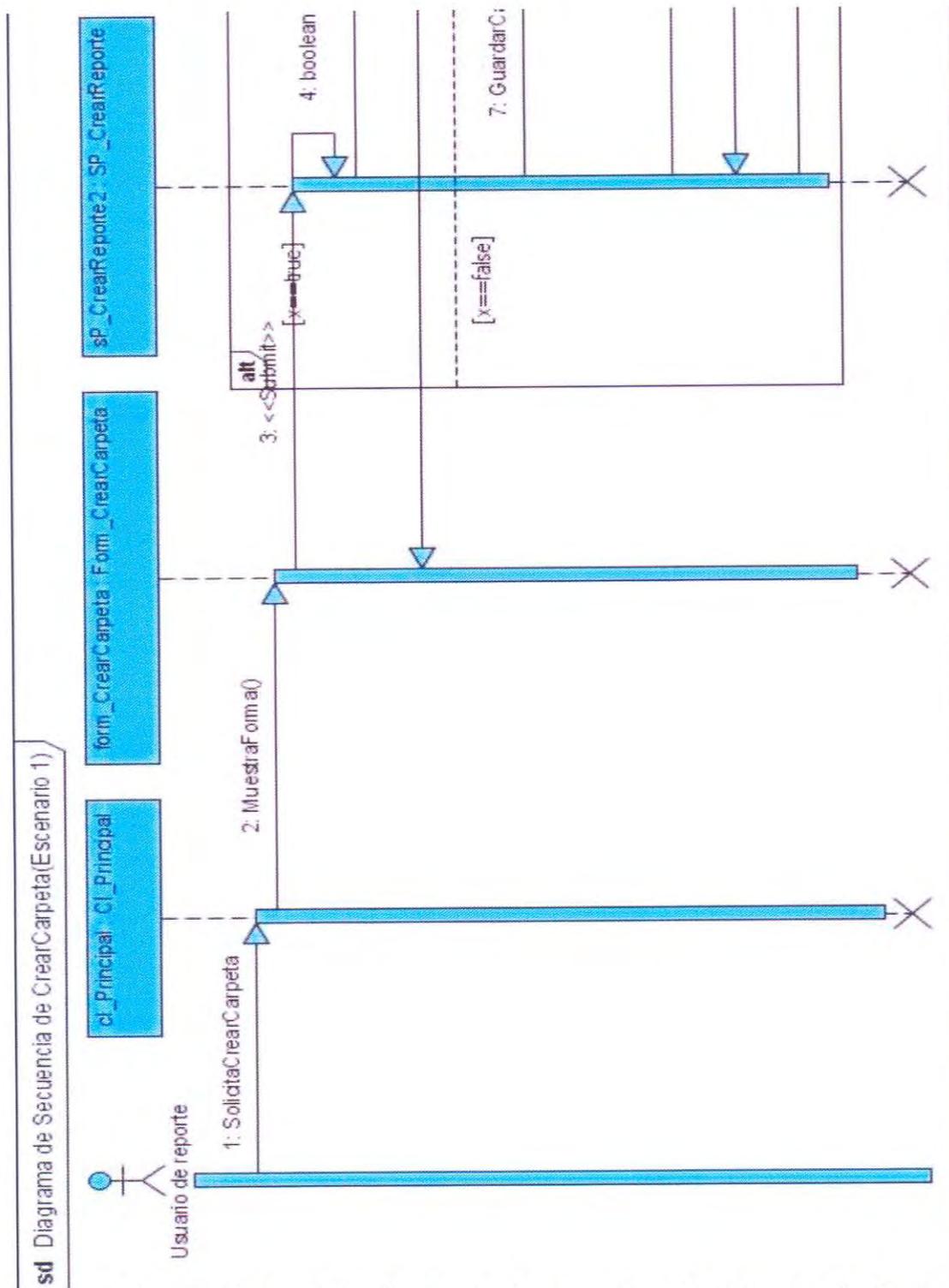


Figura 13 Diagrama de secuencia Crear Carpeta (Escenario 1) 1 Parte

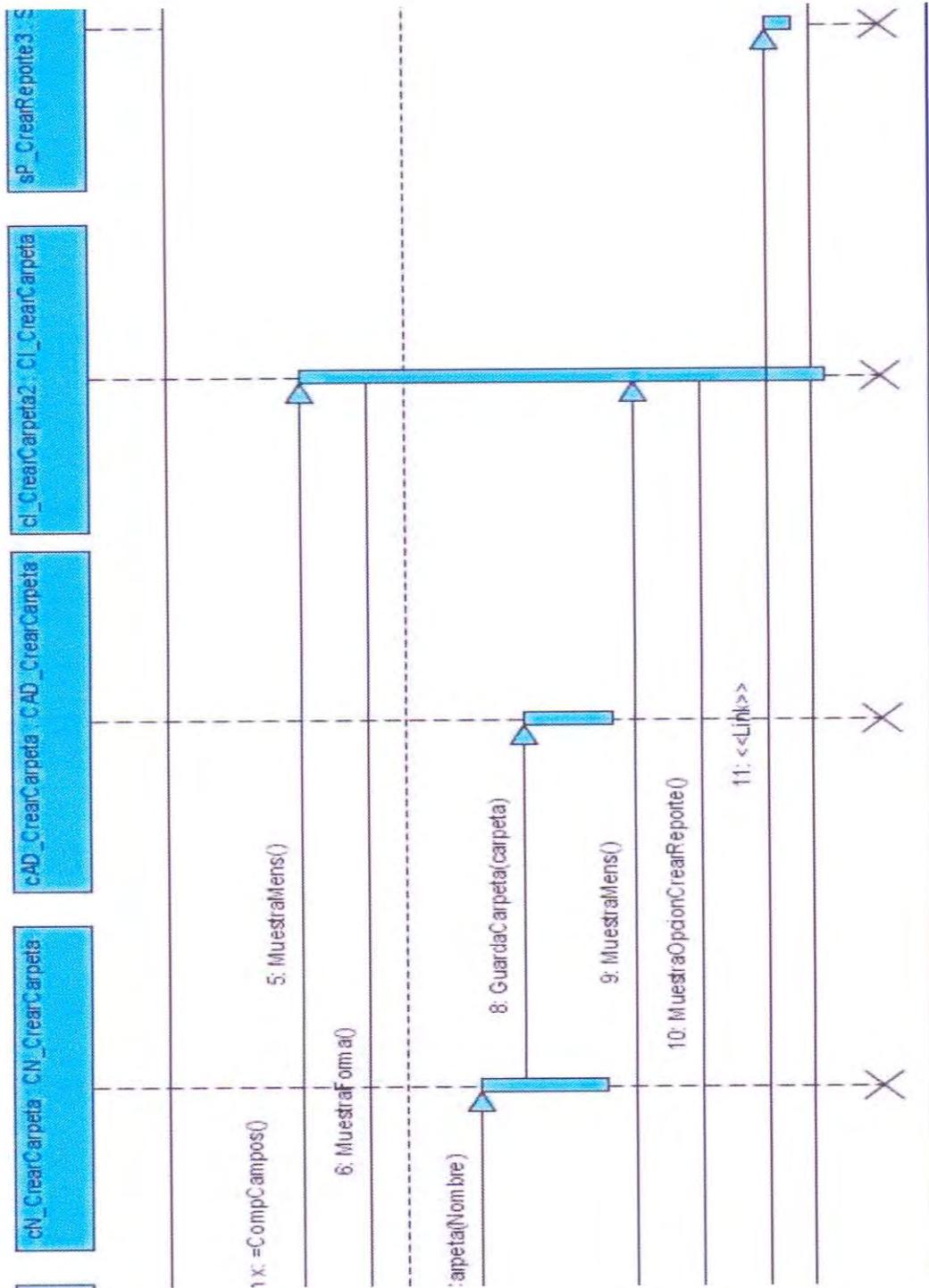


Figura 14 Diagrama de secuencia Crear Carpeta (Escenario 1) 2 Parte

Capítulo 3. Análisis y Diseño del sistema

3.4 Modelo de Datos

A la hora de realizar el diseño de la base de datos se debe tener en cuenta si se satisfacen los requisitos funcionales que presenta el sistema y ver que se logre mantener la integridad de los datos, que no exista redundancia de información y si se brinda la seguridad que se necesita. Los modelos que se utilizan para diseñar la base de datos del sistema se denominan, modelos lógicos y físicos de datos.

Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información. La estructura tiene que ser desarrollada para la necesidad de las aplicaciones que la usaran, esto nos puede ayudar a realizar un proceso del negocio para alcanzar un valor agregado para el cliente.

El módulo de reportes utilizará casi toda la base de datos en la obtención de información que fue gestionada en los demás módulos que le corresponderán. Este diseño asegura que el modelo de datos no sufra cambio alguno. El módulo en sí tiene un modelo de datos independiente.

Las clases persistentes que a continuación se muestran cubren las necesidades que el sistema pueda presentar.

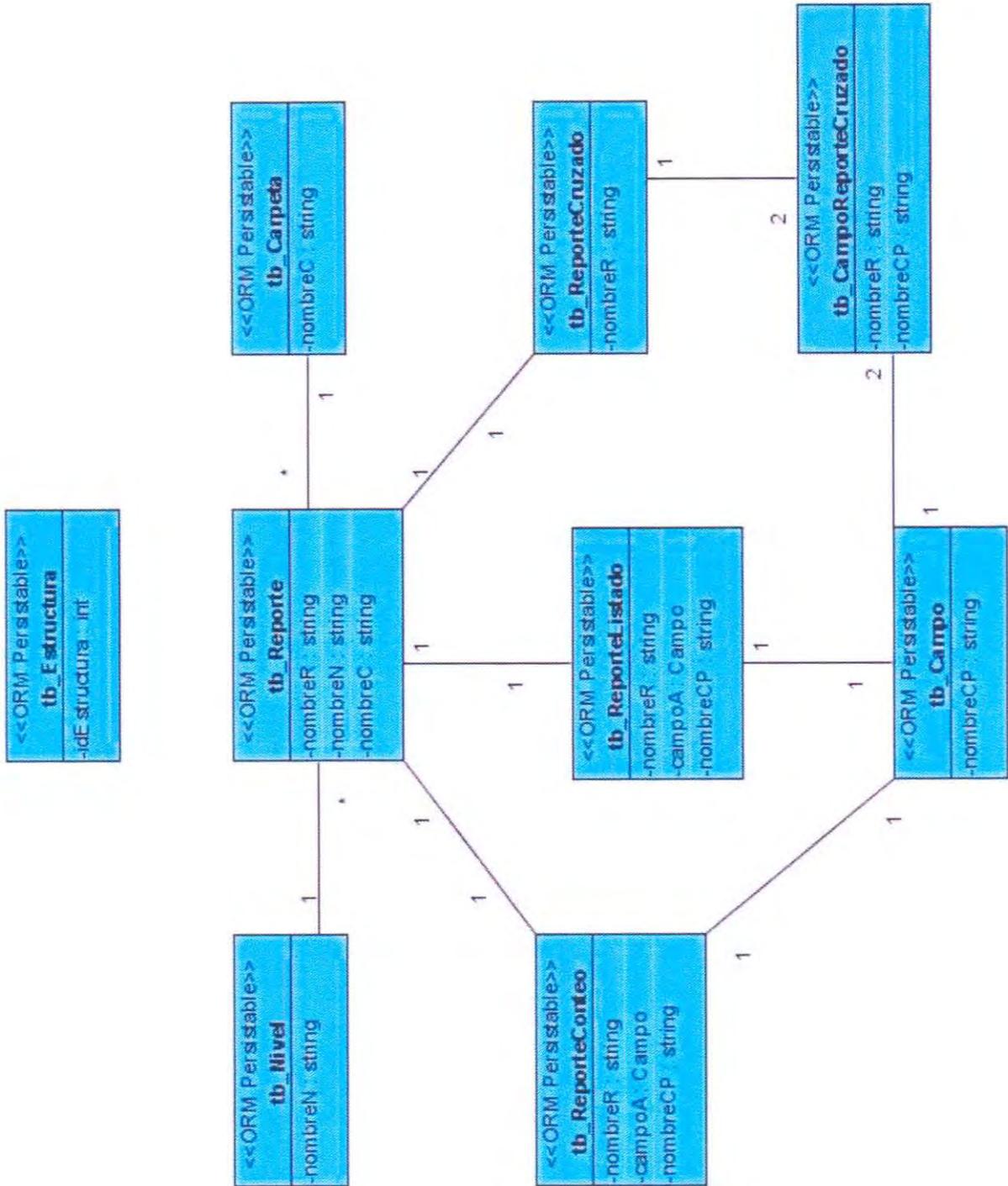


Figura 15 Diagrama de clases persistentes

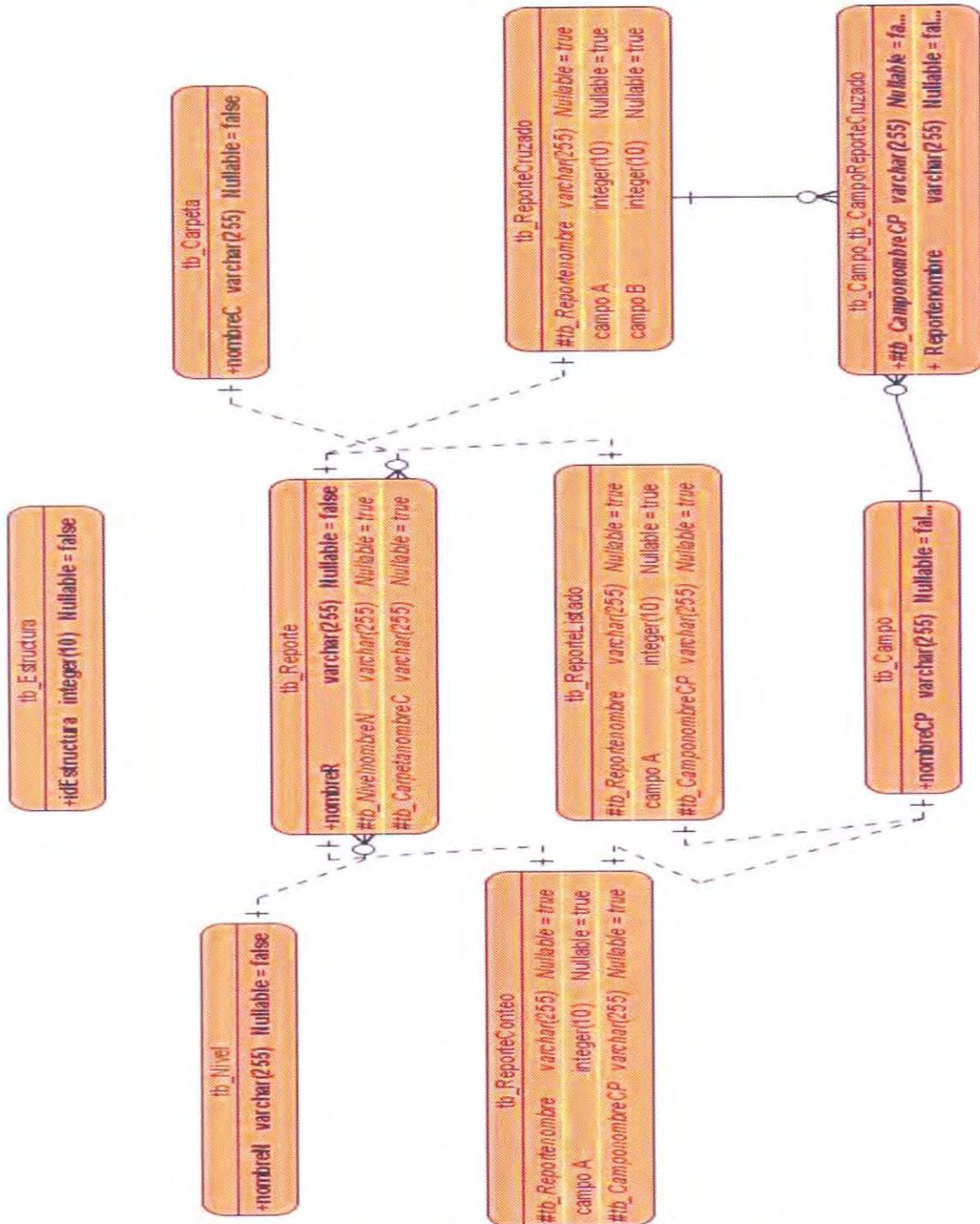


Figura 16 Diagrama entidad relación

3.5 Modelo de Despliegue

El modelo de despliegue permite describir la distribución física que tendrá el sistema estableciendo una correspondencia entre la arquitectura software y la arquitectura hardware.

El diagrama de despliegue elaborado para los procesos diseñados se muestra a continuación:

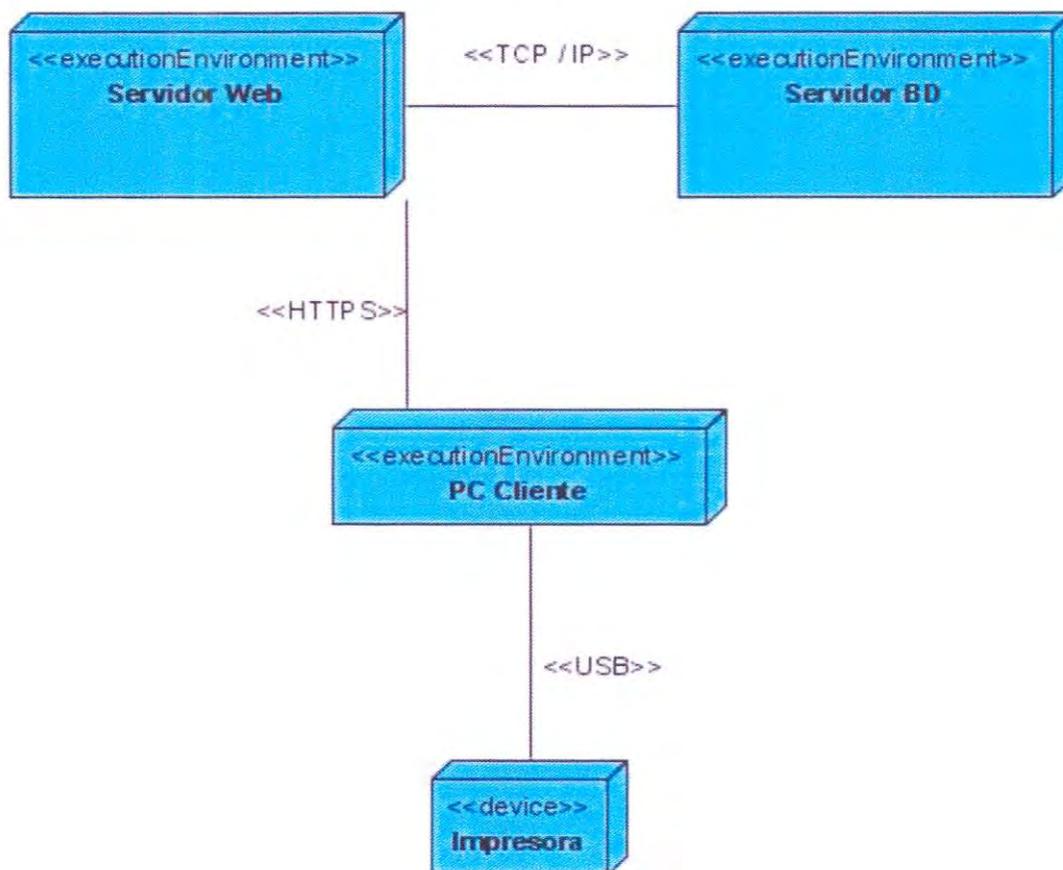


Figura 17 Modelo de Despliegue

Capítulo 3. Análisis y Diseño del sistema

3.6 Estimación de costo-beneficio

Estimación por Puntos de Caso de Uso

La Estimación por Puntos de Caso de Uso es un método de estimación de esfuerzo de un proyecto de desarrollo de software a partir de los casos de uso.

Puntos de Casos de Uso sin ajustar

UUCP = UAW + UUCW donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin Ajustar.

Tabla 3.1 Para calcular UAW

Actores	Complejidad	Factor de Peso
Usuario de Reportes	Complejo	3

UAW = Sumatoria de la multiplicación de la cantidad de actores de un tipo con su factor de peso.

$$\mathbf{UAW} = 3 + 1 = 4$$

Cantidad de actores de tipo complejo: 1

$$\mathbf{UAW} = 3 \times 1$$

$$\mathbf{UAW} = 3$$

Capítulo 3. Análisis y Diseño del sistema

Tabla 3.2 Para calcular el UUCW

Caso de Uso	Tipo de Caso de Uso	Factor de Peso
Gestionar Reporte	Simple	5
Gestionar Carpeta	Simple	5
Consultar Carpeta	Simple	5
Consultar Reporte	Simple	5
Crear Reporte	Medio	10

UUCW = Casos de uso del sistema y su complejidad.

$$\mathbf{UUCW} = 4 \times 5 + 1 \times 10$$

$$\mathbf{UUCW} = 30$$

$$\mathbf{UUCP} = \mathbf{UAW} + \mathbf{UUCW}$$

$$\mathbf{UUCP} = 4 + 30$$

$$\mathbf{UUCP} = 34$$

Puntos de Casos de Uso ajustados

$$\mathbf{UCP} = \mathbf{UUCP} \times \mathbf{TCF} \times \mathbf{EF}$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Capítulo 3. Análisis y Diseño del sistema

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i)$$

Tabla 3.3 Para calcular la complejidad técnica

Factor	Descripción	Peso	Valor Asignado
T1	Sistema distribuido	2	1
T2	Tiempo de respuesta	1	1
T3	Eficiencia del usuario final	1	2
T4	Procesamiento interno complejo	1	3
T5	El código debe ser reutilizable	1	3
T6	Facilidad de instalación	0.5	1
T7	Facilidad de uso	0.5	3
T8	Portabilidad	2	1
T9	Facilidad de cambio	1	1
T10	Concurrencia	1	2
T11	Incluye objetivos especiales de seguridad	1	2
T12	Provee acceso directo a terceras partes	1	1
T13	Se requieren facilidades especiales de entrenamiento	1	0

$$TCF = 0.6 + 0.01 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$TCF = 0.6 + 0.01 \times \Sigma (2 \times 1 + 1 \times 1 + 1 \times 2 + 1 \times 3 + 1 \times 3 + 0.5 \times 1 + 0.5 \times 3 + 2 \times 1 + 1 \times 1 + 1 \times 2 + 1 \times 1 + 1 \times 1 + 1 \times 0)$$

$$TCF = 0.85$$

Capitulo 3. Análisis y Diseño del sistema

Tabla 3.4 Para calcular el factor ambiente

Factor	Descripción	Peso	Valor Asignado
E1	Familiaridad con el modelo de proyecto utilizado	1.5	1
E2	Experiencia en la aplicación	0.5	2
E3	Experiencia en trabajo orientado a objetos	1	3
E4	Capacidad del analista líder	0.5	3
E5	Motivación	1	4
E6	Estabilidad de los requerimientos	2	3
E7	Personal a tiempo completo	-1	3
E8	Dificultad del lenguaje de programación	-1	2

El Factor de ambiente se calcula:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso} \times \text{Valor asignado})$$

$$EF = 1.4 - 0.03 \times \Sigma (1.5 \times 1 + 0.5 \times 2 + 1 \times 3 + 0.5 \times 3 + 1 \times 4 + 2 \times 3 + (-1 \times 3) + (-1 \times 2))$$

$$EF = 1.04$$

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 30 \times 0.85 \times 1.04$$

$$UCP = 26.52$$

Para calcular el esfuerzo en horas-hombre

$$E = UCP \times CF$$

Donde:

$$E = \text{Esfuerzo}$$

Capítulo 3. Análisis y Diseño del sistema

UCP= Puntos de Casos de Uso ajustados.

CF= Factor de conversión (para este tipo de proyecto 20 horas-hombre/Punto de Casos de Uso)

$$E = UCP \times CF$$

$$E = 26.52 \times 20$$

$$E = 530.4$$

Tabla 3.5 Para calcular el esfuerzo total

Actividad	Porcentaje	Esfuerzo
Análisis	10%	53.04
Diseño	20%	106
Implementación	40%	530.4
Prueba	15%	79.56
Sobrecarga	15%	79.56
Total	100%	848.56

Esfuerzo Total (Horas--Hombre)	848.56
Esfuerzo Total (Mes--Hombre)	1.17
Salario Promedio	100
Cantidad de Hombres (Integrantes de la tesis)	1
Costo Hombre--Mes	100
Costo Total	848.56

El proyecto requerirá de 848.56 horas-hombre a desarrollar por una persona en un tiempo de 250 días; tomando que se trabajará en el proyecto como promedio 8 horas diarias. Por lo cual

Capítulo 3. Análisis y Diseño del sistema

se puede decir que se terminará el proyecto en aproximadamente 10 meses ajustándose a los requerimientos del cliente.

3.7 Concepción general de la ayuda

Teniendo en cuenta que la futura aplicación se encontrará en constante intercambio con la mayoría de los trabajadores que laboran en las diferentes tareas, los que en su mayoría poseen escasos o ningún conocimiento de computación, se hace necesaria la realización de un manual de usuario.

La existencia de este manual de usuario donde se expliquen todas las tareas a desarrollar, se planteen y se especifiquen algunos procedimientos básicos y se muestren además las preguntas frecuentes sobre la funcionalidad de la aplicación y sus respuestas, será de gran ayuda en la capacitación y adiestramiento del personal, garantizando un funcionamiento eficiente y con ello un servicio de calidad.

3.8 Conclusiones

A partir de la investigación realizada se arriba a las siguientes conclusiones:

En el transcurso del presente capítulo se realizó un estudio detallado de la factibilidad de la investigación, además se obtuvo los principales artefactos pertenecientes al flujo de trabajo de análisis y diseño, dentro de los que se encuentran: el diagrama de diseño, el diagrama de clases persistentes. Por último se enunciaron las concepciones generales de la ayuda que deberán tenerse en cuenta para su desarrollo.

Conclusiones generales

Conclusiones generales

Con la realización del presente trabajo investigativo se evidenció la necesidad de realizar la implementación del módulo de reportes del Portal Nacional de la FEU. Por tal motivo, se realizó el diseño de una aplicación Web para la gestión de la información, con el objetivo fundamental de facilitar la posterior implementación de la misma, contribuyendo así a la solución de los errores para lograr una mayor calidad en la prestación de los servicios al cliente y posibilitando que se agilice de forma notable el trabajo de los involucrados en el desempeño de las actividades dentro de estos procesos.

Se fundamentó la utilización de Java como lenguaje de programación, PostgreSQL como gestor de bases de datos, UML como lenguaje para el modelado visual, RUP como metodología para el desarrollo de software y Visual Paradigm como herramienta para el modelado visual. Se modeló el negocio, identificando los actores, trabajadores, así como las actividades que son objeto de automatización. Se definieron los requerimientos del sistema, tanto funcionales como no funcionales y posteriormente, se estructuró el modelo de casos de uso del sistema, describiéndose cada caso de uso para una mejor comprensión de las funcionalidades que brindan.

Se realizó el diseño del módulo, generándose artefactos como son los diagramas de clases Web, diagramas de secuencia del diseño, entre otros. Se estructuró el modelo de datos, que constituye la representación física de la base de datos del sistema.

Además, se efectuó un estudio de la factibilidad de la investigación con el cual se obtuvo el costo estimado de la realización de la misma.

Recomendaciones

Recomendaciones

Una vez realizado este trabajo y cumplidos los objetivos propuestos, se exponen las siguientes recomendaciones con vistas a la futura implementación y puesta en marcha de la aplicación:

- Realizar la implementación de manera tal que se logre un módulo de reportes, adaptable y ameno para el usuario.
- Incorporación en un futuro nuevas funcionalidades al módulo, como la realización de reportes de funcionamiento.

Referencias bibliográficas

1. **JHONES, D. A. R.** La Universidad de las Ciencias Informáticas. [Online] 03, 02, 2006. [Cited: 2 2, 2007] <http://www.intranet.uci.cu>
2. **MIER, C.** SGA-UTPL. [Online] 1, 25, 2001. [Cited: 20 3, 2007] <http://www.utpl.edu.ec>
3. **ROMERO, RAMÓN.** Desarrollo de Software - Sistemas Paraguay. [Online] 6, 4, 2003. [Cited: 10 4, 2007] <http://www.cnc.una.py/cms/cnc/index.php>
4. **M., RAFAEL. D.** SIG-ESAC: Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba. [Online] 9, 3, 2000. [Cited: 1 11, 2007] http://bvs.sld.cu/revistas/hie/vol44_3_06/hie03306.htm
5. **HECTOR.P.M.** MES. Presentación del Proyecto Nacional: "Sistema para la Informatización de la Gestión de la Nueva Universidad" Sistema GNU. En Taller Nacional de Especialistas. Sistema GNU. [Online] 5, 17, 2005. [Cited: 4 1, 2007] <http://www.mes.cu>
6. **MARTIN.C.E.** JAVA HISPANO. [Online] 2, 3, 2007. [Cited: 4 6, 2007] <http://www.javahispano.org>
7. **VIERGÍNIA.P.P.** Petroglifo. [Online] 6, 23, 2003. [Cited: 16 7, 2007] <http://petroglifo.blogspot.com/2006/09/herramienta-para-generar-reportes.html>.
8. **ORTEGA.A.C.** MYGNET JasperReports. [Online] 5, 23, 2006. [Cited: 7 10, 2007] <http://www.mygnet.net/articulos/java>
9. **STALLMAN, R. M.** "El movimiento del Software Libre". [Online] 2, 14, 2007. [Cited: 4 9, 2007] <http://www.opensource.com>.
10. **BARRETOS.M.V.** "Concepto de RUP". [Online] 2, 5, 2005. [Cited: 6 5, 2007] <http://www.monografias.com>
11. **UCI.** Definición sobre Diseño. [Online] 3, 15, 2006. [Cited: 3 7, 2007] <http://www.biblioteca.uci.cu/tesis>

Bibliografía

Bibliografía

1. **GONZÁLEZ, Y. C.** Portal de las Misiones Sociales de la República Bolivariana de Venezuela. [Online] 5 14, 2006.
2. **SANCHEZ, M. A. M.** "Metodologías De Desarrollo De Software". [Online] 3 2, 2004. <http://www.informatizate.net/articulos/pdfs/>
3. **CRISTIAN.P.C.** Java Servlet Technology. [Online] 9 26, 2006. [Disponible en: <http://java.sun.com/products/servlet>
4. **BARAHONA, J. M. G.** La evolución del software libre. 2005. p.
5. **COMUNICACIONES, M. D.** Servicios de Correos y Telegráficos. Cuba, Ministerio de comunicaciones, 1985. 89 p.
4. **ESPIÑOZA, J. M.** Servicio Postal Universal. Consideraciones generales para su establecimiento.2003. <http://www.mtc.gob.pe/portal/comunicacion/portal.pdf>
5. **GUINAND, E. O.** El día mundial del software libre: por qué uso software libre, 2006. <http://arcadia.homelinux.org/~emilio/wordpress/archives/470>
6. **OMAR.D.B.** Historia y concepto de las herramientas CASE 2003. <http://web.madritel.es/personales3/edcollado/ingsw/tema2/2-6.htm>
7. **http-peru.** Proveedor de Tecnologías de Información. 2007. <http://www.http-peru.com/postgresql.php>
8. **OPENHOST.** Soporte Técnico, Data Base, 2008. <http://www.openhost.es/content/view/22/18/>
9. **OSORIO, G. P.** El software libre hace tambalear a Microsoft y su Explorer, 2005. <http://www.softwarelibre.gob.mx/?q=node/75>
10. **PÉREZ, J. E.** Introducción a AJAX. 2008. p.
11. **STALLMAN., R. M.** Software libre para una Sociedad Libre. 2004. p.
12. **UNIVERSIA.** Software Libre como alternativa, 2004. http://www.universia.es/portada/actualidad/noticia_actualidad.jsp?noticia=76596

Glosario de Términos y Siglas

HTML: Acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

HTTP: Es el conjunto de reglas para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

Servlets: Los servlets son objetos que corren dentro del contexto de un contenedor de servlets (Ej. Tomcat) y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones.

SGBD: Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

API: Una API (Del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones, interfaz de programación de la aplicación) es un conjunto de especificaciones de comunicación entre componentes software. Representa un método para conseguir abstracción en la programación.

Base de Datos: Una base de datos de consta de una colección de tablas que contienen datos y otros objetos, como vistas, índices, procedimientos almacenados y desencadenadores, que se definen para poder llevar a cabo distintas operaciones con datos.

Cliente-Servidor: Cuando se menciona este término no se piensa en otra cosa más que en bases de datos, dado que generalmente (Y de manera incorrecta) este término se usa como sinónimo de esto. Este término, en su más amplia definición, se usa para describir una aplicación en la cual dos o más procesos separados trabajan juntos para completar una tarea. El proceso cliente solicita al proceso servidor la ejecución de alguna acción en particular. Esta operación se conoce como Proceso Cooperativo, dado que dos procesos separados cooperan para completar la tarea en particular.

Glosario de términos

Componentes UI: Componentes de interfaz de usuario.

CU: Caso de uso.

DAO: Clases de acceso a datos.

Hibernate: Es el puente entre nuestra aplicación y la BD, sus funciones van desde la ejecución de sentencias SQL hasta la creación, modificación y eliminación de objetos persistentes.

Interface: Conexión entre dos dispositivos de hardware, entre dos aplicaciones o entre un usuario y una aplicación que facilita el intercambio de datos.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

Intranet: Una red interna, conectando a múltiples usuarios a través de la tecnología de Internet, usualmente dentro de una organización. Una Intranet es típicamente usada para incrementar el acceso a la información, comunicación, y trabajo entre usuarios individuales y la organización.

JSF: Es un Framework de desarrollo basado en el patrón Modelo Vista Controlador.

JSP: Es una tecnología Java que permite a los programadores generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo. Las JSP's permiten al código Java y a algunas acciones predefinidas ser incrustadas en el contenido estático del documento web. En las JSP se escribe el texto que va a ser devuelto en la salida (normalmente, código HTML) incluyendo código java dentro de él, para poder modificar o generar contenido dinámicamente.

Microsoft: Compañía que manufactura los sistemas de operación DOS y Windows. Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para

Glosario de términos

computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

Proceso: Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí que, a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido.

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

Sistema: Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a determinado objeto.

Sitio Web: Sistema de computación que corre un servidor Web y que se ha establecido para editar documentos en Web.

UML y UML2.0: El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

La diferencia mas notable entre UML y UML 2.0 es que el primero es orientado a objetos y el segundo es orientado a procesos, lo cual muestra su principal ventaja en que en el mundo real la mayoría de las cosas son procesos.

Web (WWW): Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.