



Universidad de las Ciencias Informáticas

Facultad 9

**Diseño de un Sistema Automatizado para Mejorar
el Proceso de Gestión de Recursos del Centro de Desarrollo
“GEySED”**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autora: Nuris Riverón Alvarez

Tutor: Ing. Kizzy Yinet Escalona Labrada

Cotutor: Adrián Gracia Aguila

Ciudad de La Habana, junio, 2010

Año del 51 Aniversario del Triunfo de la Revolución

Declaración de Autoría

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 firmo la presente a los ____ días del mes de _____ del año _____.

Nuris Riverón Alvarez

Kizzy Yinet Escalona Labrada

Datos de Contacto

Generales de la Investigación

Título: Diseño de un sistema automatizado para mejorar el proceso de gestión de recursos humanos y materiales del Centro de Desarrollo “GEySED”.

Clasificación: Informe de investigación práctico del desarrollo de un rol de producción determinado.

Generales del diplomante:

Nombre y apellidos: Nuris Riverón Alvarez

Grupo: 9506

Correo electrónico: nriveron@estudiantes.uci.cu

Generales del tutor:

Ing. Kizzy Yinet Escalona Labrada: Profesor de la Universidad de las Ciencias Informáticas

Correo electrónico: kyescalona@uci.cu

Teléfono particular: 837 2322

Generales del Co-Tutor

Ing. Adrián Gracia Aguila: Profesor de la Universidad de las Ciencias Informáticas.

Correo electrónico: agracia@uci.cu

Teléfono particular: 837 2322

Generales del oponente

Ing. Yurisnel Corrales Valdés: Profesor de la Universidad de las Ciencias Informáticas.

Correo electrónico: ycvaldes@uci.cu

Teléfono particular: 835 8907

Índice

Introducción	1
Capítulo 1: Fundamentación Teórica	6
1.1 Introducción	6
1.2 Conceptos asociados al dominio del problema	6
1.3 Análisis de soluciones existentes	8
1.4 Metodología de desarrollo a utilizar	10
1.5 Lenguaje de Modelado a utilizar (UML)	13
1.6 Herramienta de modelado a utilizar (Visual Paradigm)	13
1.7 Patrones de Caso de Uso	14
1.8 Conclusiones Parciales.....	15
Capítulo 2: Presentación de La Solución Propuesta	16
2.1 Introducción	16
2.2 Requerimientos Funcionales.....	16
2.3 Requisitos No Funcionales	18
2.4 Descripción del Sistema Propuesto	20
2.4.1 Jerarquía de los Actores del Sistema	20
2.4.2 Diagrama de Caso de Uso del Sistema.....	21
2.4.3 Descripción de los Actores del Sistema	22
2.4.4 Diagrama y Descripción Textual de los Casos de Uso del Sistema.	22
2.5 Conclusiones Parciales.....	41
Capítulo 3: Análisis y Diseño de la Solución Propuesta	43
3.1 Introducción	43
3.2 Diagrama de Clases del Análisis de los Casos de Uso	43
3.3 Diagrama de Colaboración del Análisis de los Casos de Uso	44
3.4 Diagrama de Clases del Diseño de los Casos de Uso	47
3.5 Diagrama de Secuencia del Diseño de Cada caso de uso	52

3.6	Patrones.....	56
3.6.1	Patrones de diseño	56
3.6.2	Patrones de Arquitectura.....	57
3.7	Principios del diseño.....	58
3.7.1	Diseño de la interfaz de la aplicación	58
3.7.2	Estándares de la interfaz de la aplicación	59
3.8	Conclusiones Parciales.....	60
Capítulo 4 “Validación de la Solución Propuesta”		61
4.1	Introducción	61
4.2	Definiciones de Métricas	61
4.3	Aplicación de la métrica de la calidad de la especificación de requisitos.	62
4.4	Métricas de Casos de Uso del Sistema	63
4.5	Conclusiones Parciales.....	66
Conclusiones Generales		67
Recomendaciones		68
Referencias Bibliográficas.....		69
Bibliografía Consultada		71

Índice de Figuras y Tablas

Índice de Figuras

Figura 1 Software de gestión de recursos humanos.	9
Figura 2 Patrón de caso de uso Concordancia por Adición.....	14
Figura 3 Patrón de caso de uso CRUD Completo	15
Figura 4 Patrón de caso de uso Múltiples Actores_ Roles Comunes	15
Figura 5 Actores del Sistema.....	21
Figura 6 Diagrama de Casos de Uso del Sistema.....	21
Figura 7 DCUS Autenticar Usuario.....	23
Figura 8 DCUS Gestionar Departamento	26
Figura 9 DCUS Gestionar Proyecto	30
Figura 10 Diagrama de clases del análisis CUS Autenticar Usuario	43
Figura 11 Diagrama de clases del análisis CUS Gestionar Departamento	43
Figura 12 Diagrama de clases del análisis CUS Gestionar Datos de un Proyecto	44
Figura 13 Diagrama de Colaboración del Análisis del CU Autenticar Usuario.....	44
Figura 14 Diagrama de Colaboración del Análisis Adicionar un departamento	45
Figura 15 Diagrama de Colaboración del Análisis Modificar los datos de un departamento.....	45
Figura 16 Diagrama de Colaboración del Análisis Eliminar un departamento	45
Figura 17 Diagrama de Colaboración del Análisis Adicionar un proyecto.....	45
Figura 18 Diagrama de Colaboración del Análisis Mostrar los datos de un proyecto	46
Figura 19 Diagrama de Colaboración del Análisis Modificar los datos de un proyecto.....	46
Figura 20 Diagrama de Colaboración del Análisis Eliminar un proyecto	46
Figura 21 Diagrama de Clases del Diseño Autenticar Usuario.....	47

Índice de Figuras y Tablas

Figura 22 Diagrama de Clases del Diseño Adicionar Departamento.....	48
Figura 23 Diagrama de Clases del Diseño Modificar Datos de un Departamento	48
Figura 24 Diagrama de Clases del Diseño Eliminar Departamento.....	49
Figura 25 Diagrama de Clases del Diseño Adicionar Proyecto.....	49
Figura 26 Diagrama de Clases del Diseño Modificar Datos de un proyecto.....	50
Figura 27 Diagrama de Clases del Diseño Eliminar Proyecto.....	50
Figura 28 Diagrama de Secuencia del Diseño Autenticar Usuario	52
Figura 29 Diagrama de Secuencia del Diseño Adicionar Departamento.....	53
Figura 30 Diagrama de Secuencia del Diseño Modificar Departamento.....	53
Figura 31 Diagrama de Secuencia del Diseño Eliminar Departamento.....	54
Figura 32 Diagrama de Secuencia del Diseño Adicionar Proyecto.....	54
Figura 33 Diagrama de Secuencia del Diseño Modificar Datos de Proyecto.....	55
Figura 34 Diagrama de Secuencia del Diseño Eliminar Proyecto.....	55
Tabla 1 Descripción de los actores del sistema	22
Tabla 2 CUS Autenticar Usuario	23
Tabla 3 CUS Gestionar Departamento.....	26
Tabla 4 Gestionar Proyecto	31
Tabla 5 Modelo de Métricas.....	64

Introducción

En los años 90 surge el desarrollo de la teoría de los recursos humanos y sus capacidades. Hoy en día el capital humano no solo es un activo cuyo coste hay que disminuir, sino un activo de carácter vital para muchas organizaciones interesadas en el tema de Gestión de Recursos Humanos (GRH).

En la actualidad la GRH cobra cada vez mayor importancia, ya que se considera como una pieza clave para obtener, desarrollar y conservar el número del personal adecuado. La GRH tiene como objetivo fundamental, mantener y perfeccionar un grupo de individuos con diferentes habilidades y motivaciones, capacitándolos en su vida profesional para alcanzar las metas de una empresa. Es por esto, que la GRH es un elemento decisivo para que la empresa pueda adquirir altos niveles de calidad y competitividad, dependiendo de los Recursos Humanos (RRHH), ya que son los protagonistas en las tomas de decisiones que ha de hacer una empresa.

El mundo está bajo constantes transformaciones derivadas de las nuevas tecnologías de la información y la comunicación (TIC). Frente a estos retos, los RRHH se convierten en la fuente principal de ventaja competitiva sostenibles en las empresas. “Su correcta gestión se ha convertido en el fundamento de la competitividad empresarial moderna, demostrando que las personas marcan la diferencia en el funcionamiento de una organización” (Jackson and Slocum 2008).

Un país es más competitivo, mientras mejor formado tenga sus RRHH. Cuba no está exenta de estas consideraciones, pues es un país, con un alto nivel cultural y preparación en su capital humano, producto del sistema educativo cubano y de iniciativas como La Batalla de Ideas, que han permitido la formación exitosa de la fuerza de trabajo.

Por otro lado, Cuba tiene ciertas limitaciones a la hora de penetrar en el mercado internacional en condiciones competitivas, debido al bloqueo económico impuesto por el gobierno de los Estados Unidos, que dificulta el acceso de bienes materiales y financieros.

Sin embargo, Cuba cuenta con una potencia proveedora de software y servicios informáticos: La Universidad de Ciencias Informáticas, (UCI), creada con el objetivo de formar profesionales, comprometidos con su patria, calificados en la rama de la informática, a partir del estudio-trabajo vinculado a la producción. Ha logrado incrementar los ingresos de la nación por concepto de exportación de software, convirtiendo la rama de la informática en una de las más productivas del país, mediante el

aporte de soluciones tecnológicas y servicios de software relacionados con la educación, la salud, entre otras esferas de la sociedad.

La UCI nacida dentro de La Batalla de Ideas contribuye a la lucha por la liberación de los cinco héroes prisioneros del imperio, mediante la divulgación de la verdad sobre su caso y se organizan eventos relacionados con el tema. En aras de la educación y la cultura integral de nuestro pueblo, desarrolla sistemas y aplicaciones para la informatización de la sociedad cubana, un ejemplo de esto es el sistema de gestión de datos geológicos del que se beneficia la oficina de recursos minerales. La UCI respalda la misión médica cubana con un estudiante por cada Centro de Diagnóstico Integral, garantizando el correcto funcionamiento de los sistemas informáticos que desarrolla.

Este año la UCI ha hecho cambios en la infraestructura productiva con el fin de fortalecer el desarrollo de la producción de software. La infraestructura está formada por centros de desarrollo que garantizan la formación desde la producción, el desarrollo de tecnologías y la investigación asociada a cada temática especializada. Los centros de desarrollo están conformados por departamentos anteriormente conocidos como polos productivos, estos departamentos llevan a cabo líneas de investigación y proyectos temáticos. Para obtener resultados con calidad en la producción de software se hace necesario llevar el control de los recursos humanos y materiales de los centros de desarrollo ubicados en las nueve facultades que componen la sede central.

La facultad 9 posee un centro de desarrollo por dos departamentos Geoinformática y Señales Digitales, actualmente la información de los recursos materiales con que cuentan estos departamentos para llevar a cabo de forma correcta las tareas presentes en el proyecto, no se encuentra centralizada, imposibilitando que los jefes de departamento lleven el control total de la cantidad real y características de las computadoras y accesorios que las componen, así como otros medios materiales destinados a la producción.

Así mismo como sucede con los RRMM sucede con los RRHH, debido que resulta muy difícil y engorroso archivar de forma manual información relacionada en documentos físicos y pocos manejables como el Word y el Excel, provocando la existencia de documentos duplicados y algunas veces falta de información, dificultando la rapidez en la búsqueda y recuperación de datos.

Debido a esta problemática surge una necesidad que afecta el proceso de gestión de recursos humanos y materiales del centro de desarrollo y luego de la búsqueda de información pertinente y la identificación del

problema a resolver queda formulado como la **inexistencia de un sistema automatizado que gestione los recursos materiales y humanos del centro de desarrollo GEySED**. Es por esto que la investigación pretende: **diseñar un sistema automatizado que mejore el proceso de gestión de recursos humanos y materiales del centro de desarrollo GEySED**, siendo necesario el estudio del **proceso de gestión de recursos humanos y materiales en el centro de desarrollo GEySED**, centrando la atención en la **automatización del proceso de gestión de los recursos humanos y materiales del centro de desarrollo GEySED**. La investigación defiende la siguiente idea: **el diseño propuesto permitirá la futura implementación de un sistema automatizado que gestione los recursos materiales y humanos del Centro de Desarrollo GEySED, que garantizará el control y organización de los recursos asociados a dicho centro.**

Para el cumplimiento del objetivo general de la investigación es necesario realizar las siguientes tareas:

- Revisar y analizar trabajos previos acerca de la gestión de recursos.
- Construir la fundamentación teórica.
- Modelar los requisitos funcionales y no funcionales.
- Modelar el análisis del proceso de gestión de recursos.
- Diseñar el proceso de gestión de recursos.
- Evaluar los resultados obtenidos.

Como posibles resultados de la investigación se puede obtener:

- Especificación de requisitos del software.
- Diagramas correspondientes al flujo de trabajo de análisis y diseño.
- Diseño del prototipo no funcional de interfaz de usuario.

Métodos Científicos:

“El método de la investigación científica es el modo de abordar la realidad, de estudiar los fenómenos de la naturaleza, la sociedad y el pensamiento con el propósito de descubrir la esencia de los mismos y sus relaciones.” (Zayas 1995).

El método científico se puede clasificar en teóricos y empíricos. Los teóricos permiten estudiar las características del objeto de investigación y los empíricos describen y explican dichas características.

Teóricos:

➤ **Analítico – Sintético:**

A través de este método se pudo realizar el análisis de documentación y teorías del objeto de estudio, se pudo profundizar en las características fundamentales, extrayendo conceptos y elementos más importantes de la gestión de recursos, conceptos que ayudarán a comprender el resto de la investigación científica.

➤ **Análisis Histórico Lógico:**

“Reproducir un objeto en el pensamiento en toda su objetividad y concreción significa comprenderlo en su desarrollo, en su historia.” (Zayas 1995).

Mediante el método histórico se analiza la trayectoria y desarrollo de las distintas etapas por la que transcurrió la gestión de recursos, el método lógico se utilizó para extraer del estudio histórico lo fundamental del proceso de gestión de recursos.

➤ **Modelación:**

“La modelación es justamente el método mediante el cual creamos abstracciones con vistas a explicar la realidad. El modelo como sustituto del objeto de investigación se nos muestra como algo semejante a él, donde existe una correspondencia objetiva entre el modelo y el objeto...” (Zayas 1995).

Para que se logre la representación de la realidad. Este método se utiliza específicamente para generar los diagramas y modelos correspondientes en los flujos de trabajo en que se trabaja.

Empíricos:

➤ **Observación:** Se realizará un registro visual de cómo se lleva a cabo el control de los recursos humanos y materiales, en el centro de desarrollo “GEySED”, en busca de las deficiencias y las necesidades.

➤ **Entrevistas:** se realizarán entrevistas a cada uno de los jefes de departamento del centro de desarrollo “GEySED”, para obtener información sobre el funcionamiento de la administración de recursos.

Muestreo:

Población: la población comprende a todos los miembros directivos que llevan el control de los recursos humanos y materiales del Centro de Desarrollo “GEySED”.

Muestra: como muestra se tomará al subdirector de formación del centro de desarrollo, a los jefes de departamento, tanto de Geoinformática como de Señales Digitales.

Técnica de muestreo: se utilizará la no probabilística de tipo muestreo intencional, ya que esta técnica permite seleccionar de la muestra los elementos más representativos que pueden brindar mayor información referente al proceso de gestión de recursos.

Estructura del contenido

La investigación está compuesta por introducción, tres capítulos, conclusiones, recomendaciones, bibliografías y anexos.

En el capítulo 1. "Fundamentación Teórica": se plantearán los elementos teóricos relacionados con la investigación, que servirán para comprender el resto de la tesis, se realizará un estudio del proceso de gestión de recursos, evaluando sistemas automatizados existentes en el mundo, en Cuba y en La UCI. Además se abordará el tema del lenguaje modelado, la metodología de desarrollo y la herramienta de modelado a utilizar. Por último se describirán los patrones de caso de uso que se utilizarán.

En el Capítulo 2. "Propuesta de la Solución": Se reflejará una detallada descripción de los requisitos funcionales y no funcionales, se determinarán los actores del sistema y se modelarán los diagramas de casos de uso del sistema, junto con su respectiva descripción textual y el prototipo no funcional de interfaz de usuario.

En el Capítulo 3. "Análisis y Diseño de La Solución Propuesta": se reflejarán los diagramas correspondientes al flujo de trabajo análisis y diseño tales como: diagrama de clases del análisis, de colaboración, del diseño y de secuencia. Además se abordará acerca de los patrones de diseño y el patrón de arquitectura para realizar el desarrollo de la aplicación.

En el Capítulo 4 "Validación de la Solución Propuesta": se hace un análisis y valoración de los resultados obtenidos en los capítulos 2 y 3 a través de métricas aplicadas para medir la calidad de los distintos artefactos generados.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se explican los elementos teóricos relacionados con el marco del problema, para un mejor entendimiento de la situación problemática. Se aborda el análisis de otras soluciones existentes en el mundo actual, a nivel nacional y de la organización y se describe la herramienta, el lenguaje modelado y la metodología de desarrollo a utilizar. Por último se definen los patrones de caso de uso que se utilizan para hacer el trabajo más fácil en el sistema.

1.2 Conceptos asociados al dominio del problema

Gestión

- “Es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización” (Rojas Romero 2006).
- “Proceso emprendido por una o más personas para coordinar las actividades laborales de otros individuos” (Saavedra).
- "Cuando los miembros de un grupo tratan de trabajar juntos para lograr fines específicos se producen complejidades de dirección, coordinación, integración, control y comunicación. Gestión es el concepto que abarca todos estos problemas." (McFarland 1989).

Una vez analizados estos conceptos se define que gestión implica coordinar todos los recursos disponibles para conseguir determinados objetivos en una organización. Lo esencial es que es un proceso de planear, organizar, dirigir, evaluar y controlar.

Recursos Humanos

“El término de Recursos Humanos se ha aceptado de forma general durante la última década porque expresa la creencia de que los trabajadores constituyen un recurso valioso y a veces irremplazable” (Mejía, Balkin et al. 2001).

“...parte del Capital Intelectual que tiene una organización, sin los cuales esta no pudiera concretarse como tal...” (Reyes 2009).

Son aquellas personas que presentan las capacidades, habilidades y características personales, que necesita una organización para concretarse y alcanzar sus metas.

Gestión de Recursos Humanos

“La Gestión de Recursos Humanos debe entenderse como el conjunto de actividades, técnicas e instrumentos que se desarrollan y aplican en una empresa, cuyo objetivo consiste en situar a la persona idónea en el puesto adecuado en el momento oportuno, formada y motivada para contribuir eficazmente a la consecución de los objetivos de la organización.” (Serrano 2004).

“Conjunto de principios, procedimientos que procuran la mejor elección, educación y organización de los servidores de una organización su satisfacción en el trabajo y el mejor rendimiento en favor de unos y otros.”(González 2006).

En esencia la GRH es mantener y desarrollar un grupo de individuos con diferentes habilidades, motivaciones y satisfacción para que sirvan de apoyo a la organización en la búsqueda de las metas.

Ventaja Competitiva

“Aquellas habilidades, recursos, conocimientos, atributos, etc., de que dispone una empresa, de lo que carecen sus competidores y que hace posible la obtención de unos rendimientos superiores a estos”(Zaldívar).

“Ventajas competitivas son ventajas que posee una empresa ante otras empresas del mismo sector o mercado, que le permite destacar o sobresalir ante ellas, y tener una posición competitiva en el sector o mercado” (Crecenegocios.com).

Una empresa posee una ventaja competitiva cuando tiene características diferenciales respecto a sus competidores que la hacen sobresalir por encima de las demás empresas y obtienen con mayor facilidad rendimientos superiores.

Sistema Automatizado

“La automatización es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos” (Redmin.cl 2009).

“Es la tecnología utilizada para realizar procesos o procedimientos sin la ayuda de las personas” (Intelmax.com 2001).

Aplicaciones o programas que se crean con el objetivo de sustituir los procedimientos manuales administrativos, que una persona realiza en una organización de forma presencial, por un software que permita realizar los mismos procedimientos de manera no presencial y reducir el esfuerzo empleado por dicha persona.

1.3 Análisis de soluciones existentes

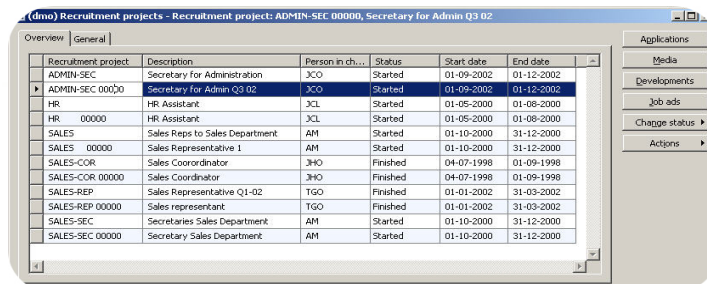
En el mundo se han desarrollado numerosos sistemas de gestión de recursos, dentro de estos los que más se destacan son los Sistemas de Planificación de Recurso Empresariales (ERP), son sistemas integrados diseñados para modelar y automatizar la mayoría de procesos y necesidades de una empresa (ÖVERGAARD and PALMKVIST 2004). Soportan las funciones de planificación y control, cubren ventas, compras, nóminas, inventarios, gestión de efectivo y contabilidad en general. Además se ocupan de la información no contables, como la gestión de proyectos, la gestión del capital humano, entre otras.

Existen una gran variedad de sistemas para el control de la información asociada a los recursos de una organización, por lo que se hace necesario un análisis de los que poseen características semejantes a las necesidades planteadas en el problema de la investigación y ver si es posible adaptarlos a los requerimientos de la solución que se desea obtener. Como resultado del análisis se encontraron en el mundo, en Cuba y en el mundo sistemas con características similares al que se desea obtener. A continuación se explica brevemente sus características y funcionalidades.

Microsoft Dynamics AX

Microsoft Dynamics AX 2009 es una solución de gestión empresarial completa para medianas y grandes organizaciones, que trabajan con el conocido software de Microsoft, para mejorar la productividad del personal. Proporciona un mayor conocimiento sobre la situación de sus empleados, una perspectiva general de toda su organización y soporta un eficaz proceso de reclutamiento, también facilita el seguimiento de ausencias y comprende las tendencias del absentismo, obteniendo fácilmente el número de ausencias y su justificación, para un período de tiempo determinado. Presenta un registro que contiene las habilidades de cada empleado, esto permite una adecuada asignación de tareas. Administra los cursos de formación especificando los grupos por objetivo, identifica los cursos, ubicación, instructor y

participantes, y la expedición de los certificados del curso. Gestiona los eventos y monitoriza la agenda y las sesiones.



The screenshot shows a software window titled "(dino) Recruitment projects - Recruitment project: ADMIN-SEC 00000, Secretary for Admin Q3 02". The window has a menu bar with "Overview" and "General". Below the menu bar is a table with the following columns: Recruitment project, Description, Person in ch..., Status, Start date, and End date. The table contains several rows of data, with the first row highlighted in blue. To the right of the table is a sidebar with buttons for "Applications", "Media", "Developments", "Job ads", "Change status", and "Actions".

Recruitment project	Description	Person in ch...	Status	Start date	End date
ADMIN-SEC	Secretary for Administration	JCO	Started	01-09-2002	01-12-2002
ADMIN-SEC 00000	Secretary for Admin Q3 02	JCO	Started	01-09-2002	01-12-2002
HR	HR Assistant	JCL	Started	01-05-2000	01-08-2000
HR 00000	HR Assistant	JCL	Started	01-05-2000	01-08-2000
SALES	Sales Reps to Sales Department	AM	Started	01-10-2000	31-12-2000
SALES 00000	Sales Representative 1	AM	Started	01-10-2000	31-12-2000
SALES-COR	Sales Coordinator	JHO	Finished	04-07-1998	01-09-1998
SALES-COR 00000	Sales Coordinator	JHO	Finished	04-07-1998	01-09-1998
SALES-REP	Sales Representative Q1-02	TGO	Finished	01-01-2002	31-03-2002
SALES-REP 00000	Sales representant	TGO	Finished	01-01-2002	31-03-2002
SALES-SEC	Secretaries Sales Department	AM	Started	01-10-2000	31-12-2000
SALES-SEC 00000	Secretary Sales Department	AM	Started	01-10-2000	31-12-2000

Figura 1 Software de gestión de recursos humanos.

El desarrollo y modificación del software se realiza mediante su propio entorno de desarrollo integrado, MorphX, que contiene varias herramientas como un depurador, analizador de código e interfaz de consulta. El lenguaje que se emplea es X++ (Corporation 2010).

El sistema Microsoft Dynamics AX no cumplen con la estrategia tecnológica que sigue el país actualmente, que es lograr la independencia tecnológica a través del software libre, puesto que es un software propietario y obtener la licencia demandaría un gasto excesivo de dinero y tiempo para el país. Por tanto no es una alternativa a tomar, ya que priva el acceso al código fuente del programa y niega el derecho de poder copiarlo y modificarlo a las necesidades requeridas.

GNU Enterprise

Es un proyecto de la comunidad de software libre implementado para la realización de un sistema completo de ERP destinado al sector empresarial. Implantado en Cuba por seguir la línea del país sobre la emigración hacia aplicaciones de código abierto. Sus funcionalidades van desde la gestión de recursos humanos, contabilidad, gestión de proyectos hasta comercio electrónico (Genix, Roblejo et al. 2009).

Este sistema a primera vista se podría aplicar porque gestiona el capital humano, pero que por sus propias especificidades son imposibles de adaptar a la solución que se desea obtener, ya que solo gestiona información relacionada con empleados de una empresa y no con estudiantes vinculados a la producción. Como tampoco integra la gestión de recursos materiales, esto conlleva a ser un software inválido para aplicarlo en el proceso de gestión de recursos del centro de desarrollo "GEySED".

CEDRUX

En la UCI se lleva a cabo la realización del sistema integral de gestión Cedrux; el primer ERP en Cuba que cuenta con diferentes subsistemas. Dentro de los subsistemas se encuentra el del capital humano que se encarga de planificar, organizar, controlar, evaluar el desempeño y logros de los trabajadores. El subsistema cuenta con un módulo de remuneración y nómina que permite contabilizar las nóminas para el pago del personal, permitiendo configurar los tipos de impuestos y los tipos de retenciones. Garantiza la actualización del submayor de vacaciones y el de retenciones, así como la confección y emisión de comprobantes de operaciones, facilitando al personal responsabilizado llevar el control de los salarios por los trabajadores en los diferentes períodos de pagos.

El sistema tiene funcionalidades muy útiles, pero como parte de la solución que se desea obtener, no es la indicada para aplicarla en el proceso de gestión de recursos del centro de desarrollo “GEySED”, ya que está destinado para gestionar pagos adicionales, puestos de trabajos e incidencias de los trabajadores y no para gestionar la información de recursos materiales y humanos vinculados a la producción.

1.4 Metodología de desarrollo a utilizar

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Las tres características fundamentales de la metodología de desarrollo RUP se describen a continuación:

- Dirigido por Casos de Uso

Un Caso de Uso es definido como un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. También guían su diseño, implementación y prueba, esto es; guían el proceso de desarrollo.

- Centrado en la arquitectura

El concepto de arquitectura software incluye los aspectos dinámicos y estáticos más significativos del sistema. La arquitectura surge de las necesidades de la empresa y se refleja en los casos de uso. Sin embargo también se ve influenciada por otros factores, como la plataforma en la que tiene que funcionar el software (arquitectura hardware, sistema operativo, sistema de gestión para bases de datos, protocolos para comunicaciones en red). Los casos de uso y la arquitectura deben equilibrarse para obtener un producto con éxito.

- Iterativo e Incremental

RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

Estos conceptos son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. La eliminación de una de estas tres ideas reduciría drásticamente el valor del Proceso Unificado.

Cada ciclo se desarrolla a lo largo del tiempo, y este a su vez, se divide en cuatro fases:

1. Inicio: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
2. Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
3. Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.
4. Transición: El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

El ciclo de vida que se desarrolla por cada iteración, es llevado bajo dos disciplinas:

Disciplina de Desarrollo:

- Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.

Disciplina de Soporte:

- Instalación: Produce release del producto y realiza actividades (empaquete, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización (JACOBSON 1998).

Considerando las características de esta metodología se puede adaptar a cualquier área de desarrollo, proyecto productivo y sirve para desarrollar un producto con éxito. Por tanto es una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software dado por la particularidad de que en cada ciclo de iteración, se hace exigente el uso de artefactos.

1.5 Lenguaje de Modelado a utilizar (UML)

El Lenguaje Unificado de Modelado es uno de los más conocidos y utilizados en la actualidad; aun cuando todavía no es un estándar oficial, está respaldado por el Object Management Group (OMG).¹ Ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Es importante resaltar que UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software Orientado a Objetos (OO) y no para describir métodos o procesos. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el RUP), pero no especifica en sí mismo qué metodología o proceso usar. Es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos (JACOBSON 1998).

1.6 Herramienta de modelado a utilizar (Visual Paradigm)

La herramienta seleccionada para realizar la modelación de este trabajo es el Visual Paradigm ya que es una poderosa herramienta CASE que al igual que el Rational Rose utiliza UML para el modelado. Es la herramienta por excelencia para ser utilizada en un ambiente de software libre. Permite crear tipos diferentes de diagramas en un ambiente totalmente visual. Es muy sencilla de usar, fácil de instalar y actualizar. Genera código para varios lenguajes, entre ellos el PHP. Tiene integrado el MS Visio y es compatible con otras ediciones. Posibilita un entorno de creación de diagramas para UML 2.0.

Visual Paradigm posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas, como el de componentes, despliegue, secuencia, casos de uso, clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las

¹Object Management Group (Grupo de Gestion de Objetos). Es un consorcio dedicado al cuidado y establecimiento de diversos estándares de tecnologías orientadas a objetos.

relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico. Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community. A diferencia de otras herramientas CASE como Rational Rose y Enterprise Architect presenta disponibilidad en múltiples plataformas y sistemas operativos, en otras palabras, es multiplataforma (Aguila 2009).

1.7 Patrones de Caso de Uso

Los patrones de caso de uso son técnicas que muestran una solución dado un problema a resolver. Son considerados herramientas que permiten resolver los problemas que se les presentan a los desarrolladores de una forma ágil y sistemática. Existen diversos tipos de patrones de caso de uso, en la investigación se aplican tres, que con la utilización de estos patrones se logrará mejores resultados de forma rápida.

- **Concordancia por Adición**

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado (ÖVERGAARD and PALMKVIST 2004). El caso de uso extendido no depende del caso base, es decir que no necesariamente tiene que cumplirse el caso de uso extendido, una vez que se efectúa el caso de uso base.

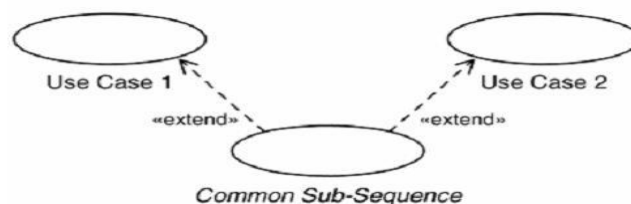


Figura 2 Patrón de caso de uso Concordancia por Adición

- **CRUD_Completo**

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual. Consta de un caso de uso llamado Gestionar Información, modela todas las operaciones sobre una parte de la información de un tipo específico, tales como crear, leer, actualizar y eliminar (ÖVERGAARD and PALMKVIST 2004).

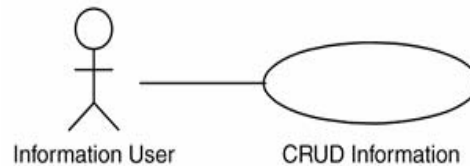


Figura 3 Patrón de caso de uso CRUD Completo

- **Múltiples Actores_ Roles Comunes**

Este patrón es representado por un actor común que de él heredan todos los actores que juegan el mismo rol sobre un caso de uso. El caso de uso extendido no depende del caso de uso base, es decir, no necesariamente tiene que llevarse a cabo el caso de uso extendido una vez que se efectuó el caso de uso base.

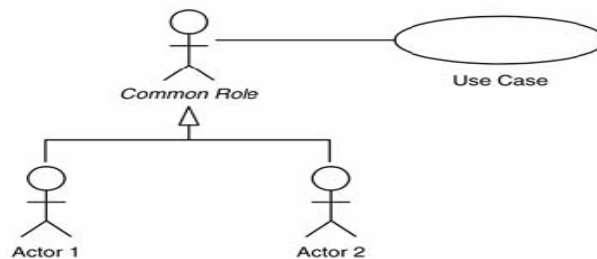


Figura 4 Patrón de caso de uso Múltiples Actores_ Roles Comunes

1.8 Conclusiones Parciales

En este capítulo se describe y explica el proceso de gestión de recurso mediante las definiciones de los principales conceptos asociados al problema. Se decidió como metodología de desarrollo RUP con UML por sus características esenciales que permiten dirigir el proceso por roles y con una mejor calidad en cada etapa del proceso, siendo Visual Parading la herramienta CASE a utilizar para modelar el sistema. El análisis de las soluciones existentes en el mundo, en Cuba y en la UCI demuestra que no existe un sistema automatizado adecuado para satisfacer las necesidades del problema a resolver. Los patrones de casos de uso como el de concordancia por adición, el CRUD completo y el de múltiples actores por roles comunes fueron utilizados como herramientas y técnicas para resolver de manera rápida los problemas que se presentan en la modelación del sistema.

Capítulo 2: Presentación de La Solución Propuesta

2.1 Introducción

En este capítulo se describen los requisitos funcionales resultantes de las entrevistas a los clientes, los requisitos no funcionales que le darán soporte al sistema, una descripción textual de los actores y los casos de uso del sistema y los diagramas de caso de uso por cada actor del sistema, así como la jerarquía que existe entre ellos.

2.2 Requerimientos Funcionales

Un sistema correcto se logra mediante la descripción de los requisitos del sistema para llegar a un acuerdo entre el cliente y los desarrolladores, es decir, los requisitos funcionales determinan lo que el sistema debe y no debe hacer. Jacobson define los requisitos funcionales como “las condiciones o capacidades que el sistema debe cumplir” (JACOBSON 1998).

RF1. Autenticar usuario.

El sistema debe validar una vez que el usuario se autentique en la aplicación, que su usuario y contraseña sean correctos.

RF2. El sistema debe permitir Gestionar Usuario.

RF2.1 Adicionar un usuario del sistema.

RF2.2 Actualizar los datos de un usuario del sistema.

RF2.3 Eliminar un usuario del sistema.

RF3. El sistema debe permitir Gestionar Nivel de Acceso.

RF3.1 Adicionar un Nivel de Acceso.

RF3.2 Modificar un Nivel de Acceso.

RF3.3 Eliminar un Nivel de Acceso.

RF4. El sistema debe permitir Gestionar Permisos.

RF4.1 Adicionar un Permiso.

RF4.2 Modificar un Permiso.

RF4.3 Eliminar un Permiso.

RF5. El sistema debe permitir gestionar un departamento.

RF5.1 Registrar un nuevo departamento.

RF5.2 Modificar los datos de un departamento.

RF5.3 Eliminar un departamento.

RF6. El sistema debe permitir gestionar un proyecto.

RF6.1 Registrar un nuevo proyecto.

RF6.2 Mostrar los datos de un proyecto.

RF6.3 Modificar los datos de un proyecto.

RF6.4 Eliminar un proyecto.

RF7. El sistema debe permitir gestionar el expediente de un proyecto.

RF7.1 Cargar un documento de un proyecto.

RF7.2 Descargar un documento de un proyecto.

RF7.3 Eliminar un documento de un proyecto.

RF8. El sistema debe permitir gestionar la asignación de un departamento a un RRHH.

RF8.1 Asignar un departamento a un RRHH.

RF8.2 Modificar la asignación de un departamento a un RRHH.

RF8.3 Eliminar la asignación de un departamento a un RRHH.

RF9. El sistema debe permitir gestionar la asignación de un proyecto a un RRHH.

RF9.1 Asignar un proyecto a un RRHH.

RF9.2 Modificar la asignación de un proyecto a un RRHH.

RF9.3 Eliminar la asignación de un proyecto a un RRHH.

RF10. El sistema debe permitir gestionar una ubicación.

RF10.1 Adicionar una Ubicación.

RF10.2 Modificar una Ubicación.

RF10.3 Eliminar una Ubicación.

RF11. El sistema debe permitir gestionar un laboratorio.

RF11.1 Adicionar un Laboratorio.

RF11.2 Modificar los datos de un Laboratorio.

RF11.3 Eliminar un Laboratorio.

RF12. El sistema debe permitir gestionar tipo de RRMM.

RF12.1 Adicionar un tipo de RRMM.

RF12.2. Modificar un tipo de RRMM.

RF12.3. Eliminar un tipo de RRMM

RF13. El sistema debe permitir gestionar los RRMM de un departamento.

RF13.1 Adicionar un recurso material a un departamento.

RF13.2 Modificar los datos de un recurso material de un departamento.

RF13.3 Eliminar un recurso material de un departamento.

RF14. El sistema debe permitir gestionar la asignación de RRMM a un Proyecto.

RF14.1 Asignar RRMM a un proyecto.

RF14.2 Modificar la asignación de RRMM a un proyecto.

RF14.3 Eliminar la asignación de RRMM a un proyecto.

RF15. El sistema debe permitir generar reporte de RRHH.

RF16. El sistema debe permitir generar reporte de RRMM.

RF17. El sistema debe permitir exportar reporte.

RF.18 El sistema debe permitir imprimir reporte.

2.3 Requisitos No Funcionales

Los requerimientos no funcionales son las características que hacen al software atractivo para el usuario, los que dicen cuán rápido o confiable debe ser el mismo, ponen las pautas a seguir y son imprescindibles para que el sistema tenga aceptación por los usuarios. En resumen según el concepto anterior, son propiedades o cualidades que el producto debe tener.

2.3.1 Interfaz

Interfaces de usuario

- El sistema debe tener una interfaz sencilla y organizada para que las personas que no tienen amplio conocimiento informático puedan acceder a ella sin problemas. Los colores, tipo de letra y tamaño deben ser agradables a la vista del usuario.

Interfaces Hardware

Para el servidor de aplicaciones se requiere de:

- Periféricos: mouse, teclado.
- Tarjeta de red.
- 512 MB de RAM o más.
- 10 GB de disco duro.

Para el servidor de Base Datos se requiere de:

- Periféricos: Mouse, teclado.
- Tarjeta de red.
- 512 MB de Ram o superior.
- 40 GB o más GB de disco.

Interfaces Software

- El sistema deberá ser multiplataforma, especialmente para Windows XP y Linux.
- El sistema deberá visualizarse en Internet Explorer 7.0 o superior y Mozilla 2.0 o superior.
- El sistema deberá tener como gestor de Base de datos PostgreSQL 8.2.
- Servidor Web Apache 2.6.0

2.3.2 Restricciones de diseño o implementación

- El sistema será implementado en el lenguaje PHP.
- Para desarrollar el sistema se utilizará el IDE Zend Development Environment.
- Para la modelación del sistema se usará el Visual Paradigm for UML 6.0 Enterprise Edition.

2.3.3 Requisitos de Fiabilidad

- El sistema debe permitir que las funcionalidades se muestren de acuerdo al nivel de accesibilidad que presente el usuario del sistema. Deberá estar protegido contra acciones que no estén autorizadas o que puedan afectar la integridad de los datos. Deberá verificar que los datos no viajen de forma transparente por la red, deberán ser encriptados.

2.3.4 Eficiencia

- El sistema deberá tener un tiempo de respuesta rápida a cualquier solicitud, para ello se utilizará al máximo los recursos de las máquinas donde esté instalado el sistema.

2.3.5 Requisitos de Usabilidad

- El sistema deberá ser utilizado por usuarios que tengan conocimiento acerca del funcionamiento y procesamiento de la información con que se trabaja en el sistema. Además el software tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades así como un mapa de la Web para una mejor orientación.

2.3.6 Requisitos de Soporte

- El sistema deberá ser revisado y actualizado cada 6 meses.

2.4 Descripción del Sistema Propuesto

2.4.1 Jerarquía de los Actores del Sistema

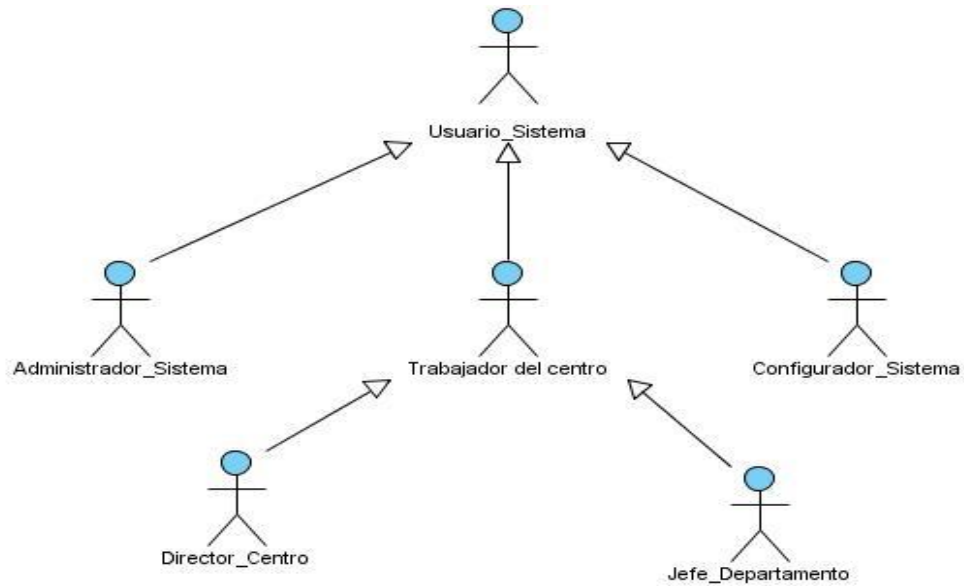


Figura 5 Actores del Sistema

2.4.2 Diagrama de Caso de Uso del Sistema

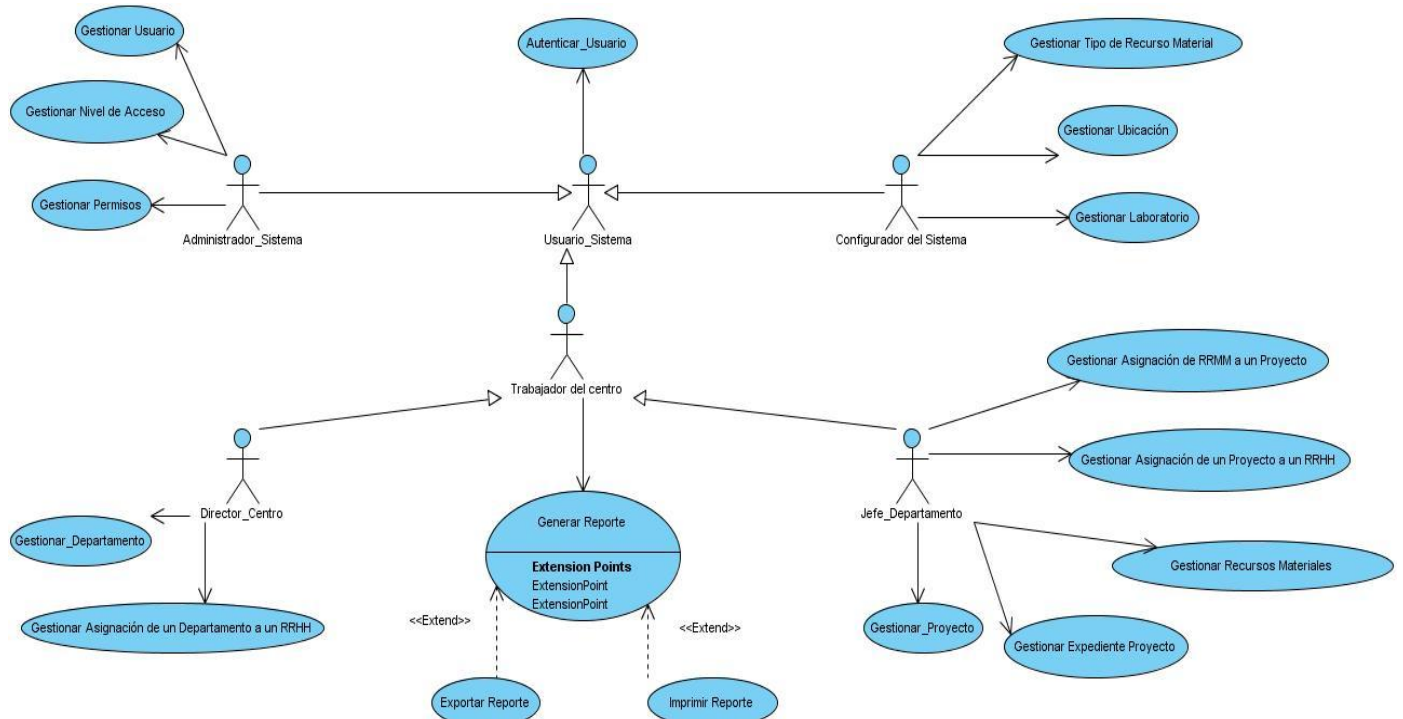


Figura 6 Diagrama de Casos de Uso del Sistema

2.4.3 Descripción de los Actores del Sistema

Tabla 1 Descripción de los actores del sistema

Actores	Descripción
Usuario del Sistema	Es la persona que accederá al sistema. Se autenticará y se le dará un nivel de acceso a la información de los recursos, según el papel que ocupa en el centro de desarrollo.
Configurador del Sistema	Es la persona que tendrá el poder de configurar la aplicación.
Administrador del Sistema	Es el encargado de otorgar permisos de accesibilidad a un usuario, en el momento que se registra en la aplicación.
Trabajador del Centro	Es cualquier miembro directivo del centro, puede generar reportes.
Director del Centro de Desarrollo	Es el encargado de gestionar los departamentos del centro y asignar un departamento a un RRHH.
Jefe de Departamento	Es el encargado de gestionar los proyectos, los expedientes de proyectos y asignar un proyecto a un RRHH, controlar los RRMM con que cuenta un departamento y asignar RRMM a un proyecto determinado.

2.4.4 Diagrama y Descripción Textual de los Casos de Uso del Sistema.

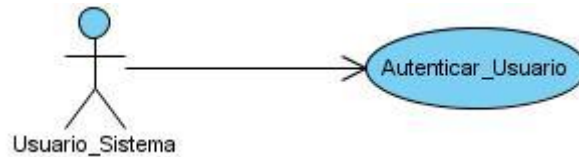



Figura 7 DCUS Autenticar Usuario

Tabla 2 CUS Autenticar Usuario

Caso de uso	Autenticar Usuario
Actores	Usuario del sistema
Resumen	El caso de uso se inicia cuando el actor introduce usuario y contraseña para comprobar que sean correctas y luego de autenticado pueda acceder a las funcionalidades del sistema según los permisos que presenta.
Precondiciones	
Referencias	RF1
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El usuario del sistema introduce el usuario y contraseña.	2. Verifica completitud de datos.
	3. Verifica que el usuario exista en el sistema.
	4. Verifica que la contraseña sea correcta.
	5. Muestra una página con todas las funcionalidades de acuerdo a los permisos que presenta el usuario y se termina el caso de uso.
Prototipo Interfaz	



Centro de Desarrollo G*GEy*SED


Usuario:

Contraseña:

Entrar

Flujos Alternos	
Actor	Sistema
	2.1. El sistema muestra un mensaje de error indicando que "Tiene que llenar todos los campos" y se termina el caso de uso.

Prototipo Interfaz



Centro de Desarrollo G*GEy*SED

Usuario:

Contraseña:

Entrar

Mensaje de Error

"Tiene que llenar todos los campos"

Aceptar

3.1. El sistema muestra un mensaje de error indicando que "No tiene acceso al sistema" y se termina el caso de uso.

Prototipo Interfaz



4.1 El sistema muestra un mensaje de error indicando que "La contraseña es incorrecta" y termina el caso de uso.

Prototipo Interfaz



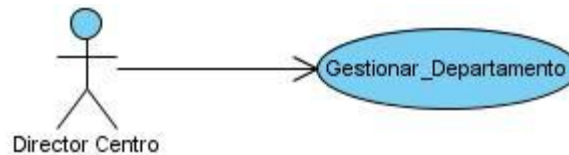


Figura 8 DCUS Gestionar Departamento

Tabla 3 CUS Gestionar Departamento

Caso de uso	Gestionar Departamento
Actores	Director del Centro
Resumen	El caso de uso se inicia cuando el director del centro se dispone a gestionar un departamento escogiendo en el menú esta opción. Luego escoge qué desea hacer, si adicionar un nuevo departamento, modificar los datos de un departamento o eliminar un departamento. Este caso de uso termina cuando el director del centro realiza la solicitud.
Precondiciones	El actor debe estar autenticado como director del centro.
Referencias	RF1, RF5, RF5.1, RF5.2, RF5.3.
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El director del centro escoge en el menú la opción "Gestionar Departamento".	2. Busca todos los departamentos que existen y los muestra en un listado con opciones de adicionar, modificar y eliminar.
3. El director del centro selecciona la opción de adicionar, ir a la sección "Adicionar un nuevo departamento".	

4. El director del centro selecciona la opción modificar, ir a la sección “Modificar datos de un departamento”.	
5. El director del centro selecciona la opción eliminar, ir a la sección “Eliminar un departamento”.	

Prototipo Interfaz



Sección “Adicionar un nuevo departamento”

Actor	Sistema
	1. Muestra una interfaz para que el director del centro introduzca los datos correspondientes a un departamento.
2. Introduce los datos correspondientes a un departamento y selecciona la acción “Adicionar”.	3. Verifica que los datos introducidos estén correctos. 4. Verifica que el departamento no exista.

5. Registra el nuevo departamento y termina el caso de uso.

Prototipo Interfaz

Flujos Alternos

Actor	Sistema
	<p>3.1. Muestra un mensaje de error indicando que “Se han introducido datos incorrectos” y se termina el caso de uso.</p> <p>4.1. Muestra un mensaje de error indicando que “El departamento existe”.</p>

Sección “Modificar datos de un Departamento”

Actor	Sistema
	<p>1. Busca los datos del departamento seleccionado y los muestra en una interfaz.</p>
<p>2. Modifica los datos que desea y</p>	<p>3. Verifica que los datos introducidos sean</p>

selecciona la acción "Modificar".	correctos.
	4. Actualiza los nuevos datos del departamento y se termina el caso de uso.

Prototipo Interfaz

Centro de Desarrollo *GEySED*

Modificar Datos de un Departamento

Nombre: Jefe de Departamento:

Descripción:

Es el encargado de editar todo lo referente a audio y video de la Universidad de las Ciencias Informáticas.

Flujos Alternos

Actor	Sistema
	3.1. Muestra un mensaje de error indicando que "Se han introducido datos incorrectos" y se termina el caso de uso.

Sección "Eliminar un Departamento"

Actor	Sistema
	1. Muestra un mensaje de confirmación para verificar si el director del centro desea realizar la acción "Si" o "No".

2. Selecciona la acción "Si".

3. Elimina el departamento seleccionado y termina el caso de uso.

Prototipo Interfaz



Flujos Alternos

Actor

Sistema

2.1. Selecciona la acción "No" y se termina el caso de uso.



Figura 9 DCUS Gestionar Proyecto

Tabla 4 Gestionar Proyecto

Caso de uso	Gestionar Proyecto
Actores	Jefe de departamento
Resumen	El caso de uso se inicia cuando el jefe de departamento se dispone a gestionar un proyecto escogiendo en el menú esta opción. Luego escoge qué desea hacer, si registrar un nuevo proyecto, mostrar, modificar los datos de un proyecto, o eliminar un proyecto. Este caso de uso termina cuando el jefe de departamento realiza la solicitud.
Precondiciones	El actor debe estar autenticado como jefe de departamento.
Referencias	RF1, RF6, RF6.1, RF6.2, RF6.3, RF6.4
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El jefe de departamento escoge en el menú la opción "Gestionar proyecto".	2. Busca todos los proyectos que existen y los muestra en un listado con opciones de adicionar, modificar y eliminar.
3. El jefe de departamento selecciona la opción adicionar, ir a la sección "Adicionar un nuevo proyecto".	
4. El jefe de departamento selecciona la opción mostrar, ir a la sección "Mostrar datos de un proyecto".	
5. El jefe de departamento selecciona la opción modificar, ir a la sección "Modificar datos de un	

proyecto”.	
6. El jefe de departamento selecciona la opción eliminar, ir a la sección “Eliminar un proyecto”.	

Prototipo de Interfaz

Centro de Desarrollo GEySED

Gestionar Proyecto | Gestionar Expediente Proyecto | Gestionar Asignación de un Proyecto a un RRHH | Gestionar RRMM de un Departar

Proyectos del Departamento de Señales Digitales

Proyectos	Modificar	Eliminar
Canal Informativo MENPET (TV Energía)		
Plataforma de VideoWeb		
ACN v 2.0		
Plataforma de Transmision de TV y Radio		
Producto PRIMICIA		

Nuevo Proyecto

Sección “Adicionar un nuevo Proyecto”

Actor	Sistema
	1. Muestra una interfaz para que el jefe de departamento introduzca los datos correspondientes a un proyecto.
2. Introduce los datos correspondientes a un proyecto y selecciona la acción “Adicionar”.	3. Verifica que los datos introducidos estén correctos.
	4. Verifica que no exista el proyecto.
	5. Adiciona los datos del nuevo proyecto y termina el caso de uso.

Prototipo Interfaz



Centro de Desarrollo G_EyS_{ED}



Adicionar un Nuevo Proyecto

Nombre:	<input type="text" value="nombre"/>	Lenguaje base:	<input type="text" value="lenguaje base"/>
Departamento:	<input type="text" value="departamento"/>	Gestor de base de datos:	<input type="text" value="gestor de BD"/>
Descripción:	<input type="text" value="descripción"/>	Sistema operativo:	<input type="text" value="sistema operativo"/>
Clasificación por estado:	<input type="text" value="clasificación por estado"/>	Herramienta de modelado:	<input type="text" value="herramienta"/>
Tipo de aplicación:	<input type="text" value="tipo de aplicación"/>	Entorno de desarrollo:	<input type="text" value="entorno de desarrollo"/>

Adicionar

Cancelar

Flujos Alternos


Actor	Sistema
	3.1. Muestra un mensaje de error indicando que se “Han introducido datos incorrectos” y se termina el caso de uso.
	4.1. Muestra un mensaje de error indicando que “El proyecto existe”.

Sección “Mostrar datos de un Proyecto”

Actor	Sistema
	1. Busca los datos correspondientes al proyecto seleccionado.
	2. Muestra los datos del proyecto y se

	termina el caso de uso.
Prototipo Interfaz	
	
Sección “Modificar datos de un Proyecto”	
Actor	Sistema
	1. Busca los datos del proyecto seleccionado y los muestra en una interfaz.
2. Modifica los datos que desea y selecciona la acción “Modificar”.	3. Verifica que los datos introducidos sean correctos.
	4. Actualiza los nuevos datos del proyecto.
	5. Muestra un mensaje indicando que “El proyecto se ha modificado correctamente” y se termina el caso de uso.

Prototipo Interfaz



Centro de Desarrollo GEySED

Modificar Datos de un Proyecto

Nombre:	<input type="text" value="ACN v 2.0"/>	Lenguaje base:	<input type="text" value="Java"/>
Departamento:	<input type="text" value="Señales Digitales"/>	Sistema operativo:	<input type="text" value="Linux"/>
Descripción:	<input type="text" value="Editor de Clases"/>	Herramienta de modelado:	<input type="text" value="Visual Paradim"/>
Clasificación por estado:	<input type="text" value="Terminado"/>	Entorno de desarrollo:	<input type="text" value="Netbeans"/>
Tipo de aplicación:	<input type="text" value="tipo de aplicación"/>		

Flujos Alternos

Actor	Sistema
	3.1. Muestra un mensaje de error indicando que “Se han introducido datos incorrectos” y se termina el caso de uso.

Sección “Eliminar un Proyecto”

Actor	Sistema
	1. Muestra un mensaje de confirmación para verificar si el jefe de departamento desea realizar la acción “Si” o “No”.
2. Selecciona la acción “Si”.	3. Elimina el proyecto seleccionado” y termina el caso de uso.

35

Prototipo Interfaz

Flujos Alternos

Actor	Sistema
2.1. Selecciona la acción "No" y se termina el caso de uso.	

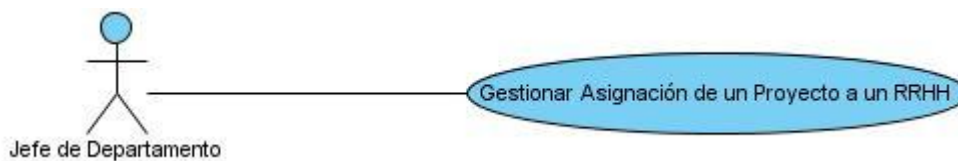


Figura 10 DCUS Gestionar Asignación de un Proyecto a un RRHH

Tabla 5 CUS Gestionar la asignación de un proyecto a un RRHH

Caso de uso	Gestionar la asignación de un proyecto a un RRHH
Actores	Jefe de Departamento
Resumen	El caso de uso se inicia cuando el jefe de departamento se dispone a gestionar la asignación

	de un proyecto a un RRHH escogiendo en el menú esta opción. Luego escoge que desea hacer, si asignar un proyecto, modificar o eliminar una asignación. Este caso de uso termina cuando el jefe de departamento realiza la solicitud.
Precondiciones	El actor debe estar autenticado como jefe de departamento.
Referencias	RF1, RF9, RF9.1, RF9.2, RF9.3.
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El jefe de departamento escoge en el menú la opción “Gestionar asignación de un proyecto a un RRHH”.	2. Muestra en una interfaz la opción de buscar un RRHH por los criterios de nombre, apellidos, grupo, usuario, género, categoría o proyecto. Además presenta tres acciones asignar, modificar y eliminar.
3. El jefe de departamento selecciona la opción asignar, ir a la sección “Asignar un proyecto a un RRHH”.	
4. El jefe de departamento escoge el criterio de búsqueda.	5. Busca los RRHH que se correspondan con el criterio de búsqueda seleccionado y los muestra.
6. El jefe de departamento selecciona el/los RRHH que desea y escoge la acción “Modificar Asignación”, ir a la sección “Modificar asignación de proyecto”.	
7. El jefe de departamento selecciona el/los RRHH que desea y escoge la acción “Eliminar Asignación”, ir a la sección “Eliminar asignación de proyecto”.	

Prototipo Interfaz



Centro de Desarrollo G.Ey.S.E.D



Gestionar Proyecto
Gestionar Expediente Proyecto
Gestionar Asignación de un Proyecto a un RRHH
Gestionar RRMM de un Departamento

Filtrar Búsqueda: Proyecto ▼

Proyecto: Genesig ▼

Seleccionar Todos



Nombre: Pedro José
Apellidos: Pérez González



Nombre: Nuris
Apellidos: Riverón Álvarez



Nombre: Yailen
Apellidos: Cano Gómez



Nombre: Dayron
Apellidos: Hernández Muñoz

Sección "Asignar un proyecto a un RRHH"

Actor	Sistema
	1. Muestra en una interfaz la opción de buscar un RRHH por los criterios de búsqueda antes mencionados exceptuando el criterio "proyecto".
2. El jefe de departamento escoge el criterio	3. Busca todos los RRHH que correspondan con

de búsqueda.	el criterio seleccionado.
4. El jefe de departamento selecciona los RRHH que desea y le asigna el proyecto que desea, selecciona la acción "Aceptar".	5. Registra la/las nuevas asignaciones y termina el caso de uso.

Prototipo Interfaz



Centro de Desarrollo GEySED

Asignar un Proyecto a un RRHH

Filtrar Búsqueda:

Grupo:

Seleccionar Todos Proyecto Ubicación Laboratorio


 Nombre: Pedro José
 Apellidos: Pérez González


 Nombre: Nuris
 Apellidos: Riverón Alvarez


 Nombre: Yalet
 Apellidos: Cano Gómez


 Nombre: Dayron
 Apellidos: Hernández Muñoz

Sección "Modificar asignación de proyecto"

Actor	Sistema
	1. Busca el proyecto que tiene asignado el/los

	RRHH y los muestra en una interfaz.
2. El jefe de departamento realiza los cambios que desea y acepta la acción "Modificar".	3. Actualiza los cambios realizados y termina el caso de uso.

Prototipo Interfaz

Centro de Desarrollo GEySED

Modificar Asignación de un Proyecto a un RRHH

Proyecto Ubicación Laboratorio

 Nombre: Pedro José Proyecto: Genesig
 Apellidos: Pérez González Ubicación: UCIFAR
 Usuario: pjgonzalez Laboratorio: 215

Sección "Eliminar asignación de proyecto"

Actor	Sistema
	1. Muestra un mensaje de confirmación para verificar si el jefe de departamento desea realizar la acción "Si" o "No".
2. El jefe de departamento selecciona la	3. Elimina la asignación de un proyecto al o los

acción "Si".	RRHH que seleccionó.
Prototipo Interfaz	
	
Flujos Alternos	
Actor	Sistema
2.1 Selecciona la acción "No" y termina el caso de uso.	

2.5 Conclusiones Parciales

En este capítulo se representó el modelado de la solución propuesta a través del diagrama de caso de uso del sistema, teniendo en cuenta la descripción de los actores y los requisitos funcionales. También se delimitaron los requisitos no funcionales que debe tener el sistema. Por último se definió que debía hacer exactamente la aplicación para dar cumplimiento a lo propuesto en los requisitos a través de las descripciones textuales de los casos de uso.

Capítulo 3: Análisis y Diseño de la Solución Propuesta

3.1 Introducción

“El análisis de un sistema se centra en la investigación del problema y no en la manera de definir una solución, mientras que el diseño pone de relieve una solución lógica: cómo el sistema cumple con los requerimientos.” (Visconti and Austidillo 2005). Es decir en esta fase del desarrollo del software se determina como se va a llevar a cabo el mismo.

En este capítulo se abordan temas fundamentales para la construcción de la solución propuesta, como los diagramas de clases del análisis y del diseño, además de los diagramas de interacción, se describe además los patrones de diseños utilizados en la realización de los diagramas de interacción y el patrón arquitectónico MVC.

3.2 Diagrama de Clases del Análisis de los Casos de Uso.

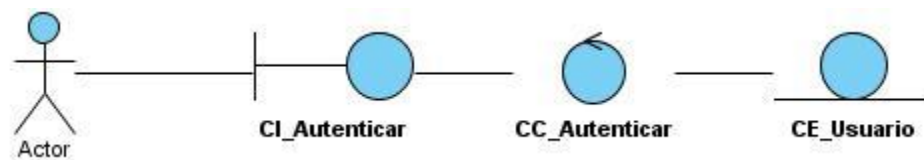


Figura 11 Diagrama de clases del análisis CUS Autenticar Usuario

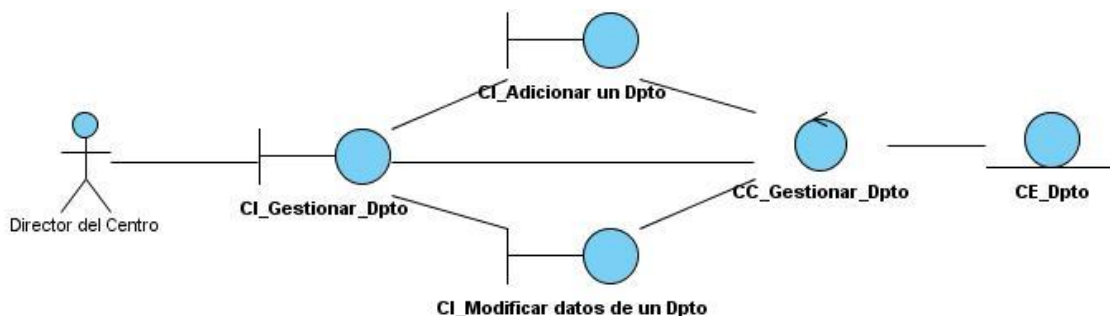


Figura 12 Diagrama de clases del análisis CUS Gestionar Departamento

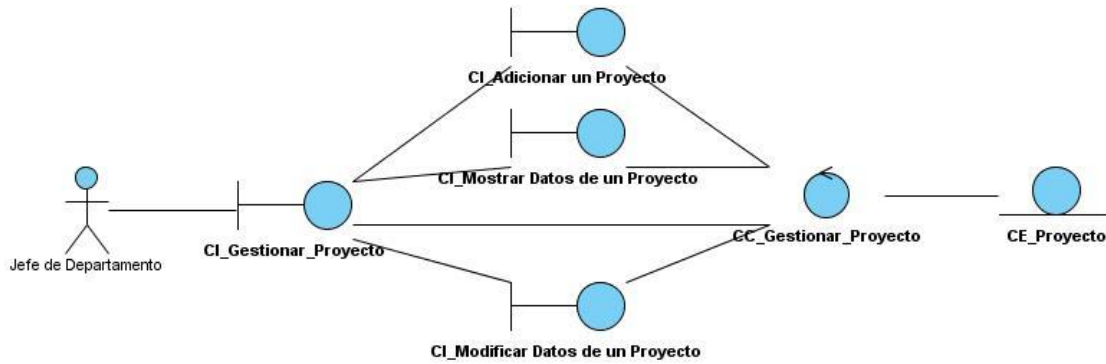


Figura 13 Diagrama de clases del análisis CUS Gestionar Datos de un Proyecto

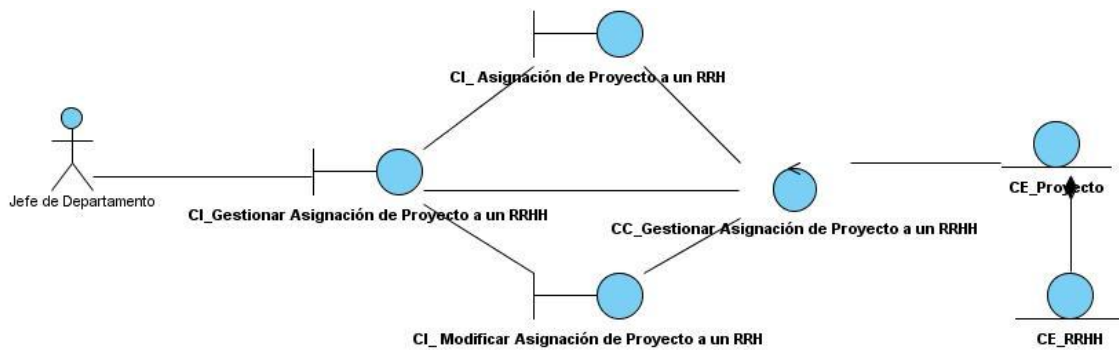


Figura 14 Diagrama de clases del análisis CUS Gestionar Asignación de un Proyecto a un RRHH

3.3 Diagrama de Colaboración del Análisis de los Casos de Uso

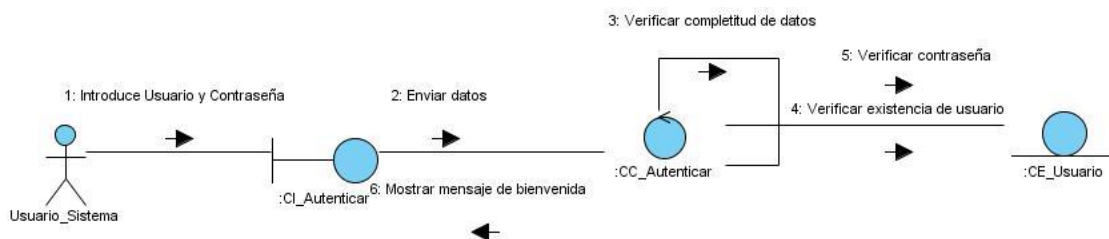


Figura 15 Diagrama de Colaboración del Análisis del CU Autenticar Usuario

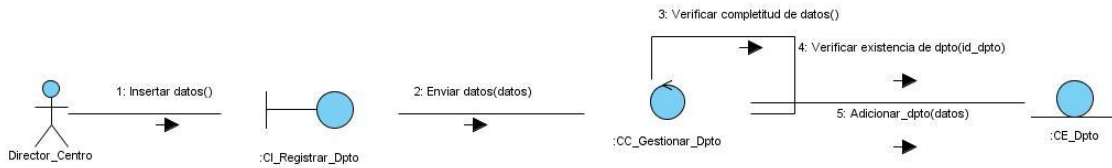


Figura 16 Diagrama de Colaboración del Análisis Adicionar un departamento

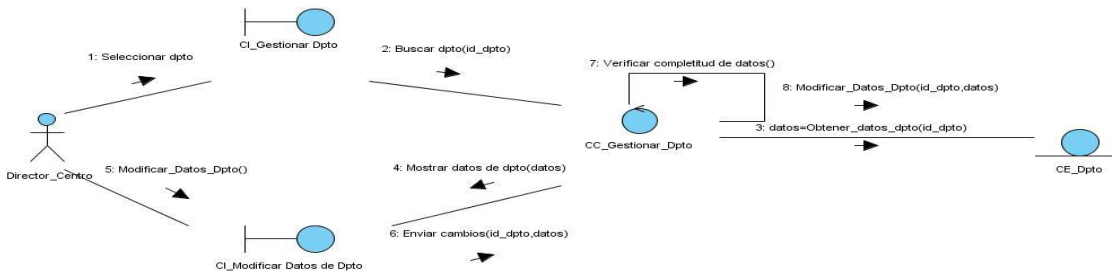


Figura 17 Diagrama de Colaboración del Análisis Modificar los datos de un departamento

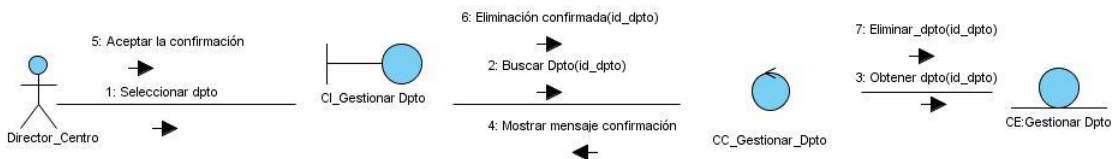


Figura 18 Diagrama de Colaboración del Análisis Eliminar un departamento

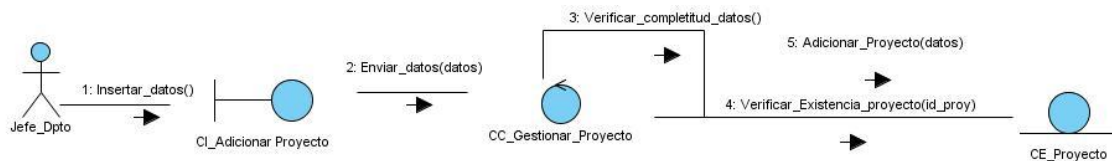


Figura 19 Diagrama de Colaboración del Análisis Adicionar un proyecto

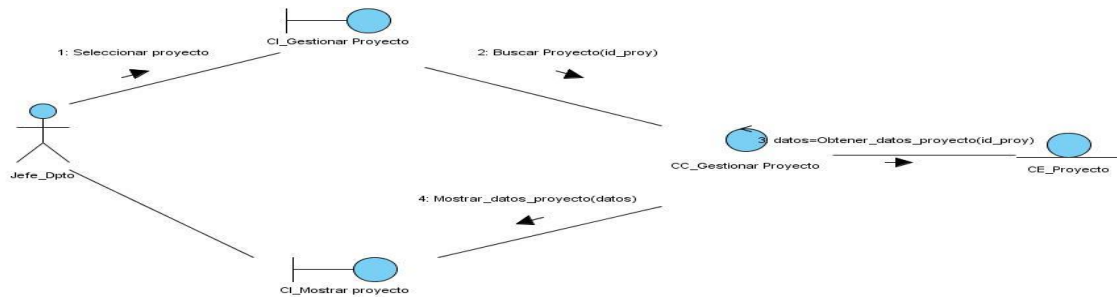


Figura 20 Diagrama de Colaboración del Análisis Mostrar los datos de un proyecto

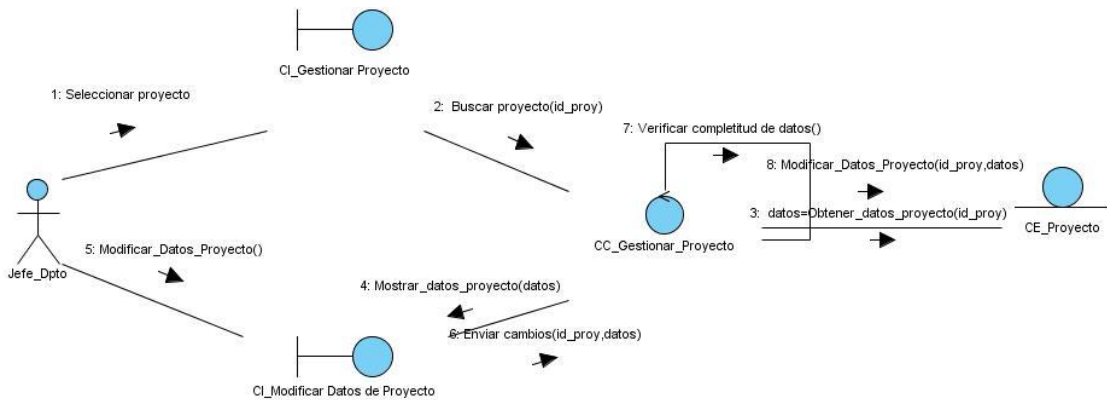


Figura 21 Diagrama de Colaboración del Análisis Modificar los datos de un proyecto

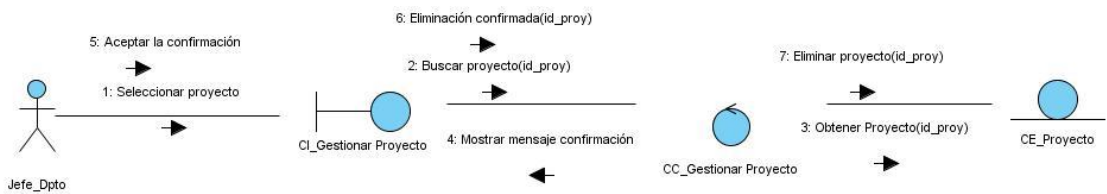


Figura 22 Diagrama de Colaboración del Análisis Eliminar un proyecto

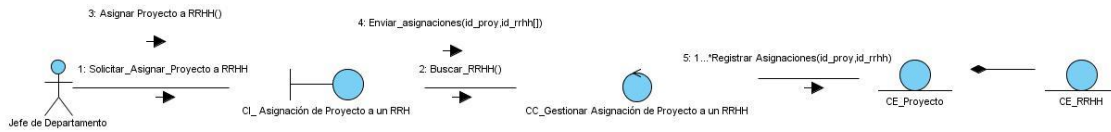


Figura 23 Diagrama de Colaboración del Análisis Asignar un Proyecto a un RRHH

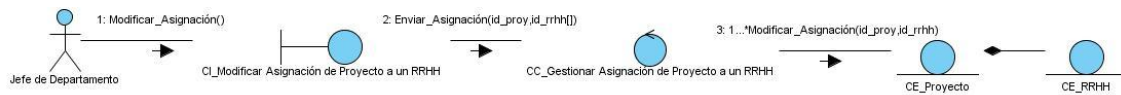


Figura 24 Diagrama de Colaboración del Análisis Modificar Asignación de un Proyecto a un RRHH

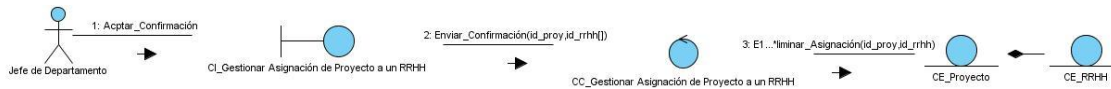


Figura 25 Diagrama de Colaboración del Análisis Eliminar Asignación de un Proyecto a un RRHH

3.4 Diagrama de Clases del Diseño de los Casos de Uso

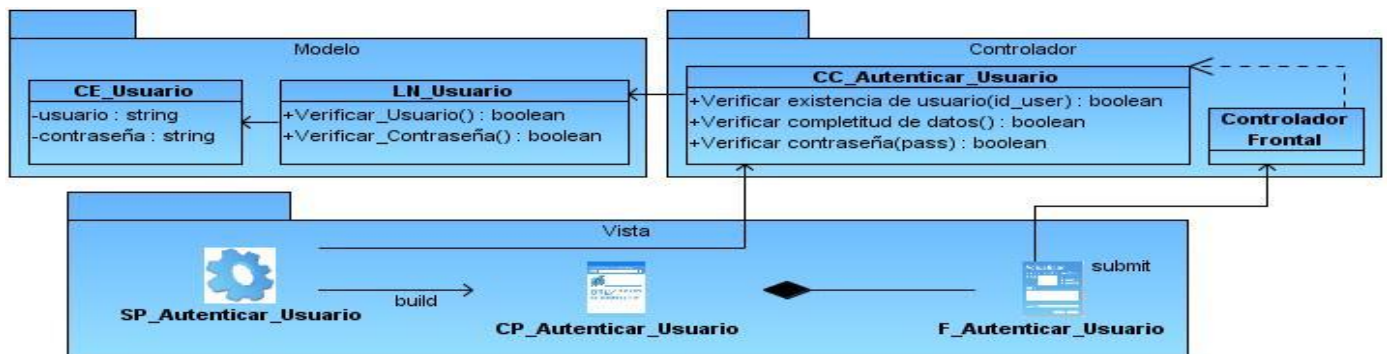


Figura 26 Diagrama de Clases del Diseño Autenticar Usuario

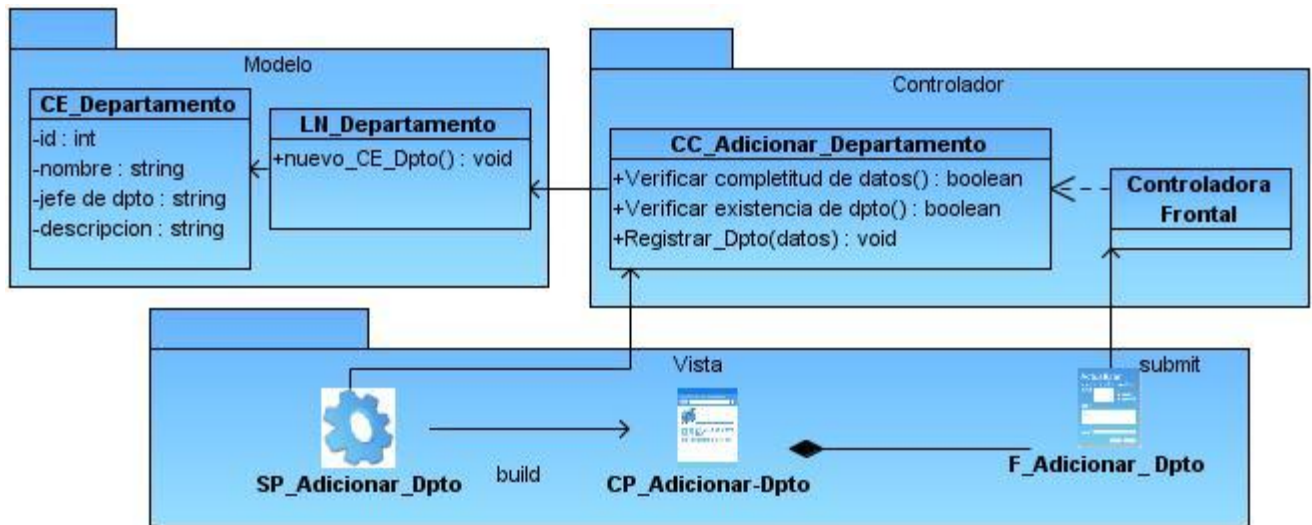


Figura 27 Diagrama de Clases del Diseño Agregar Departamento

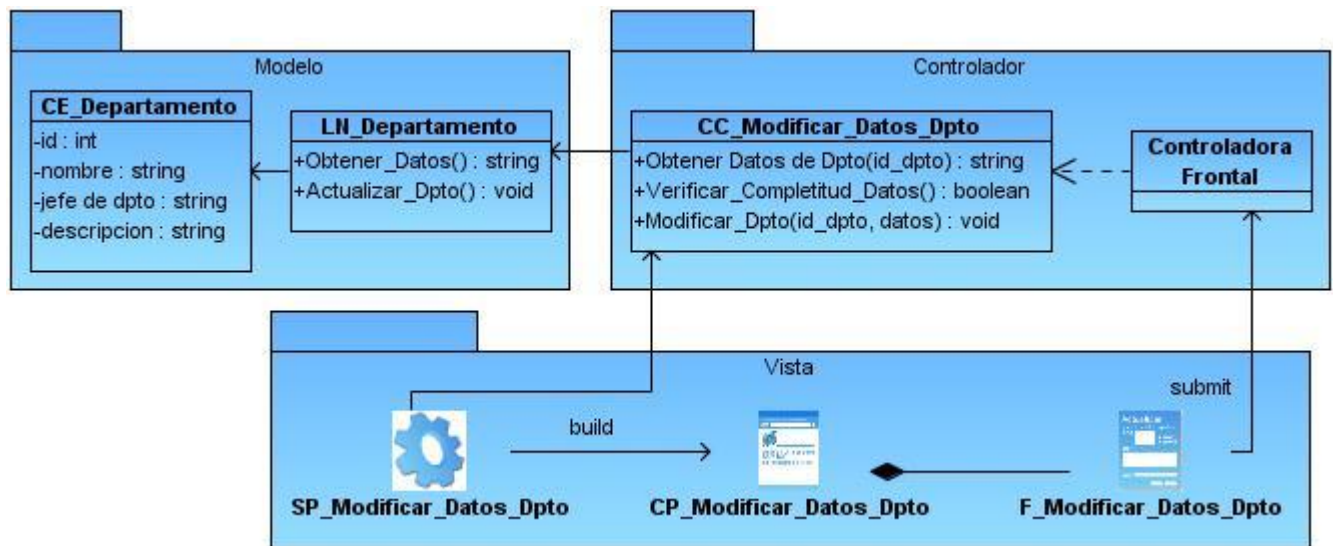


Figura 28 Diagrama de Clases del Diseño Modificar Datos de un Departamento

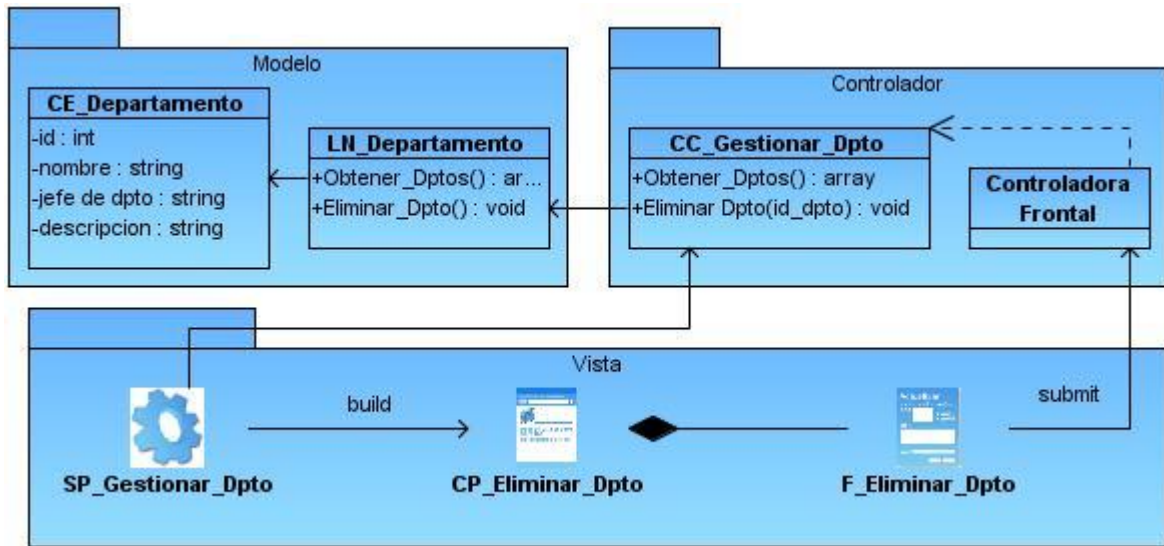


Figura 29 Diagrama de Clases del Diseño Eliminar Departamento

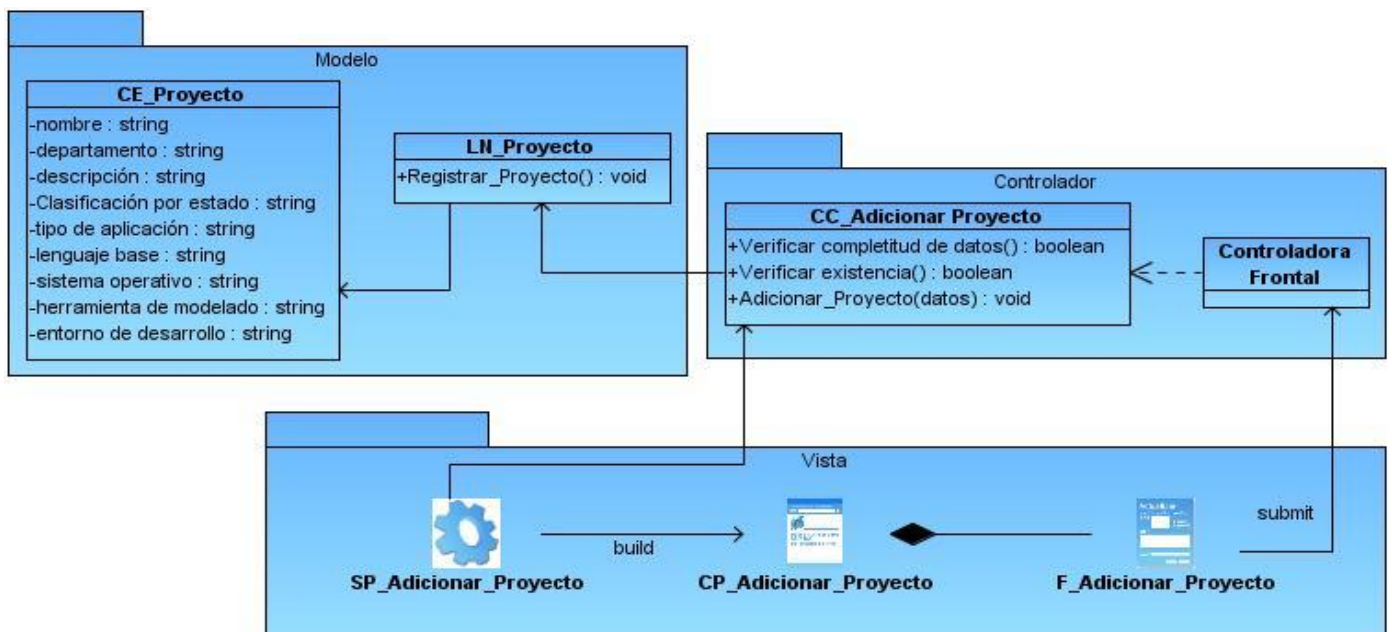


Figura 30 Diagrama de Clases del Diseño Adicionar Proyecto

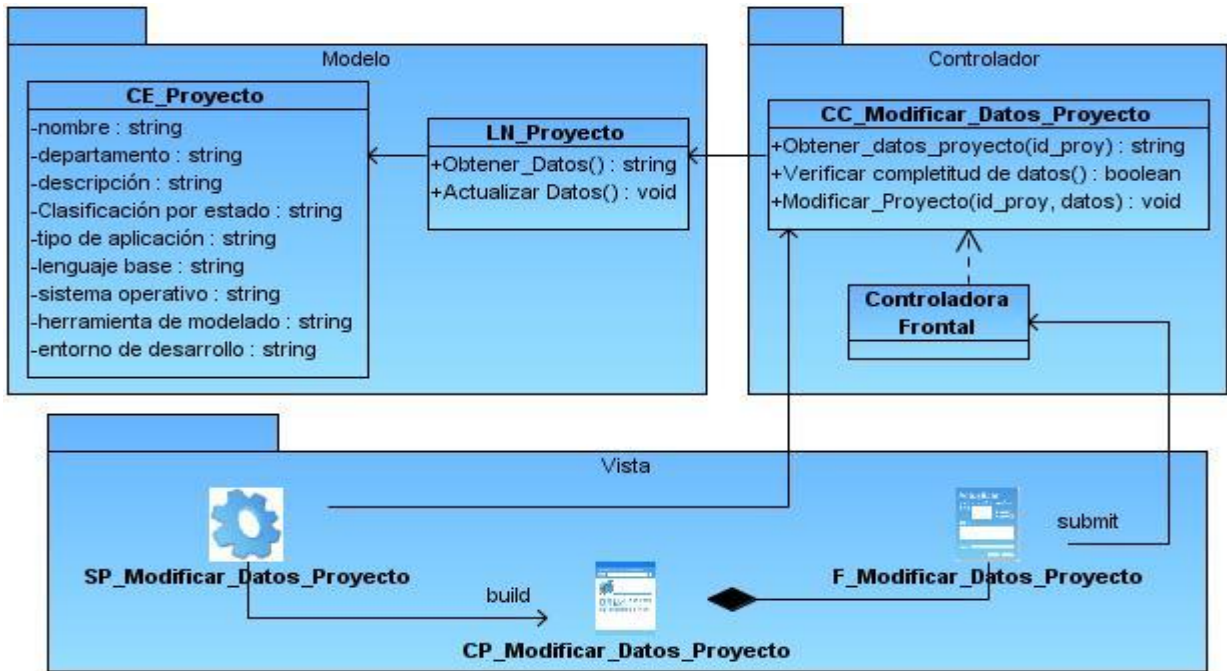


Figura 31 Diagrama de Clases del Diseño Modificar Datos de un proyecto

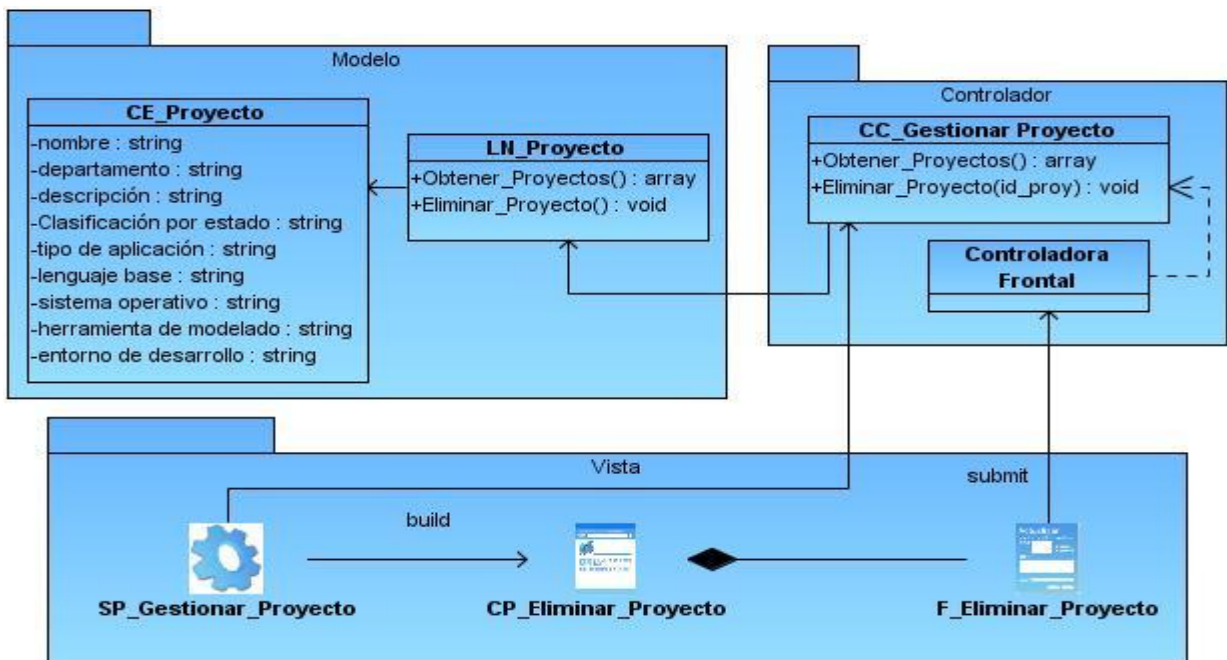


Figura 32 Diagrama de Clases del Diseño Eliminar Proyecto

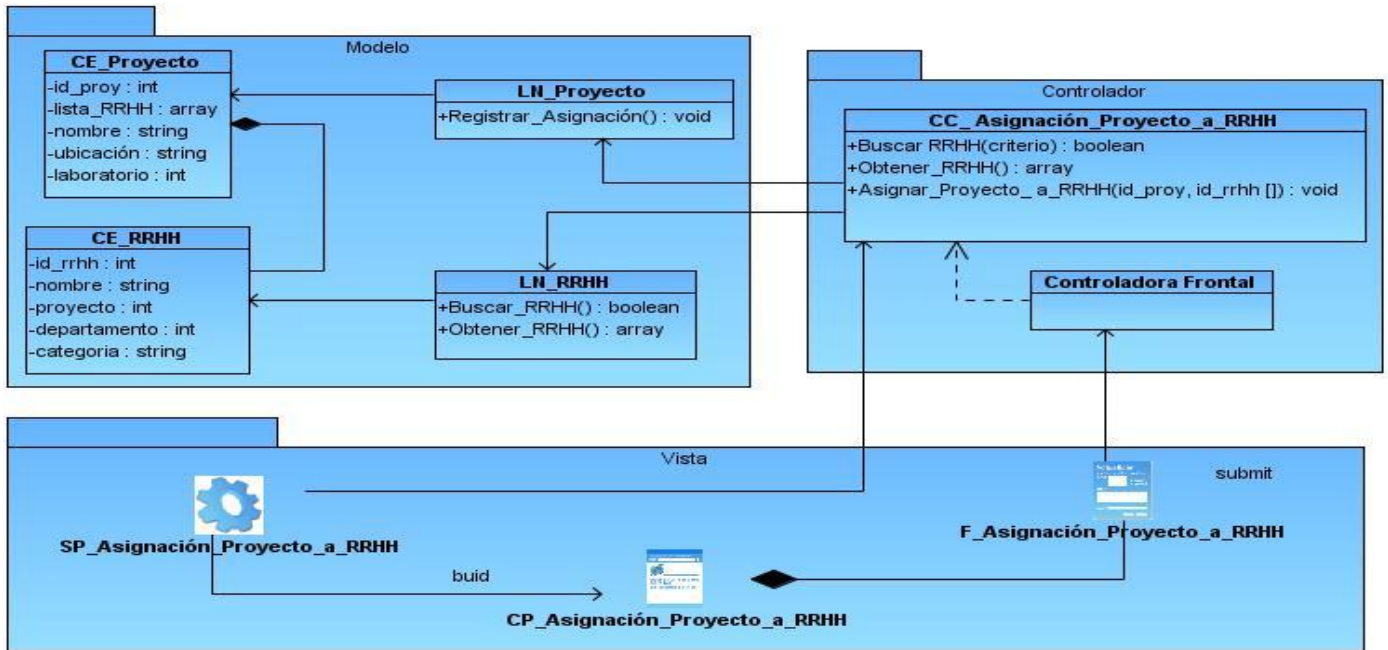


Figura 33 Diagrama de Clases del Diseño Asignar un Proyecto a un RRHH

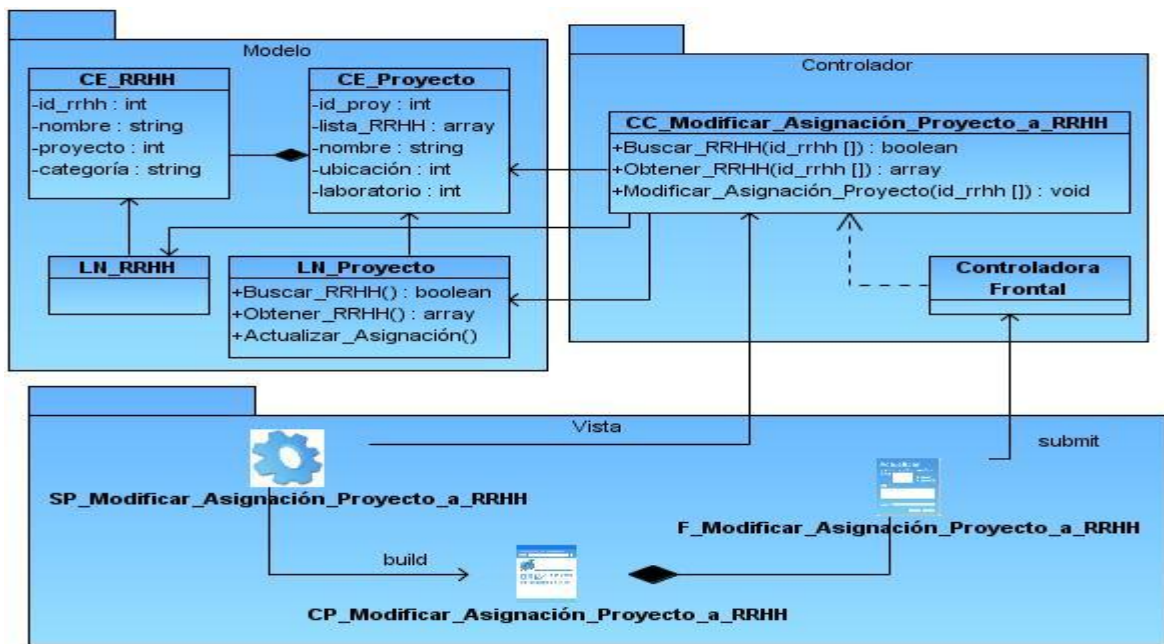


Figura 34 Diagrama de Clases del Diseño Modificar Asignación de un Proyecto a un RRHH

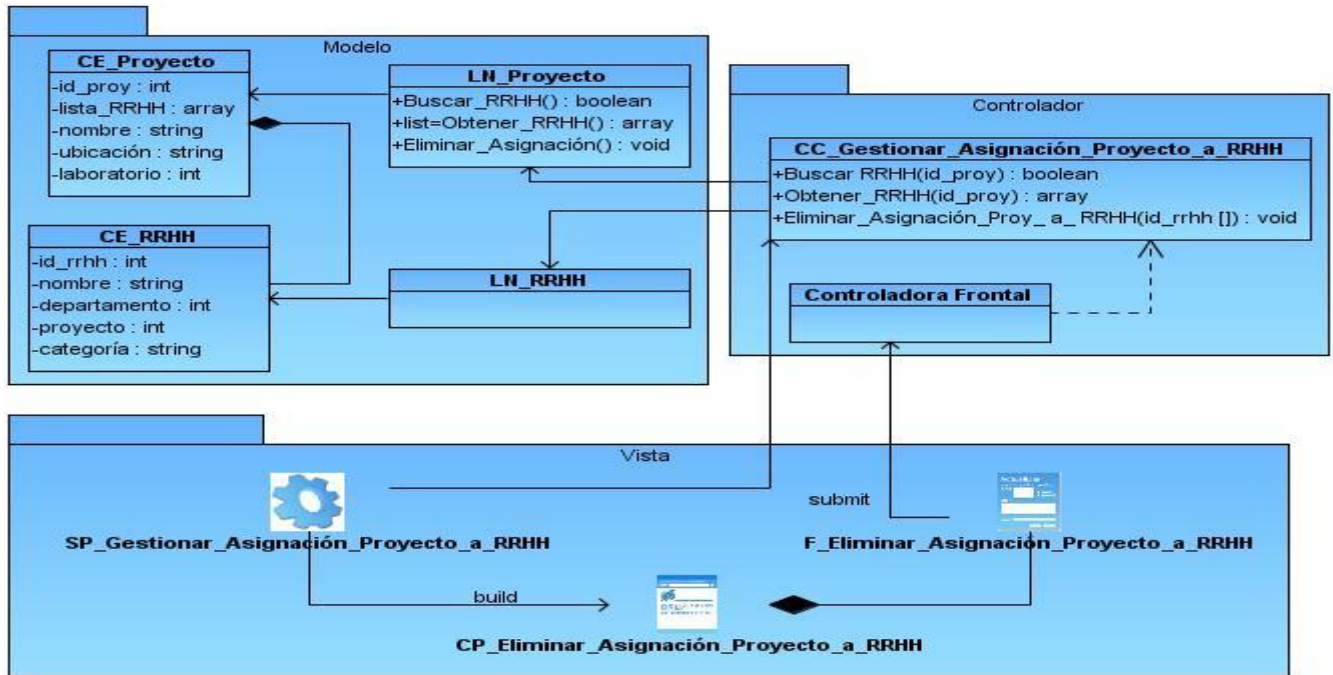


Figura 35 Diagrama de Clases del Diseño Eliminar Asignación de un Proyecto a un RRHH

3.5 Diagrama de Secuencia del Diseño de Cada caso de uso

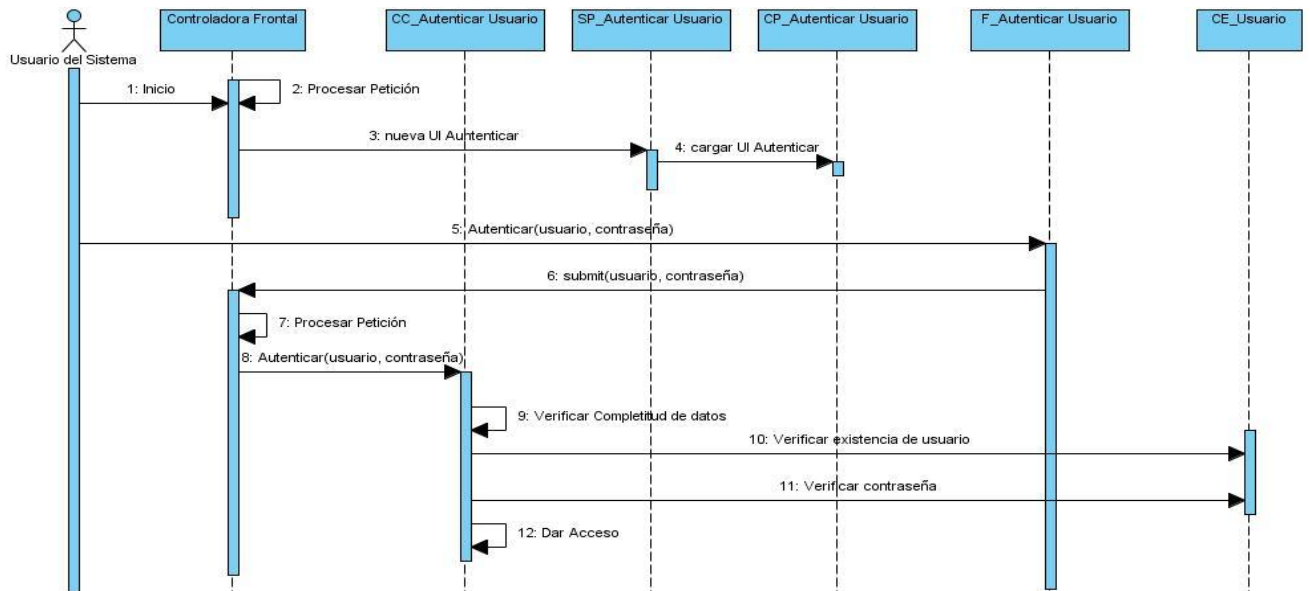


Figura 36 Diagrama de Secuencia del Diseño Autenticar Usuario

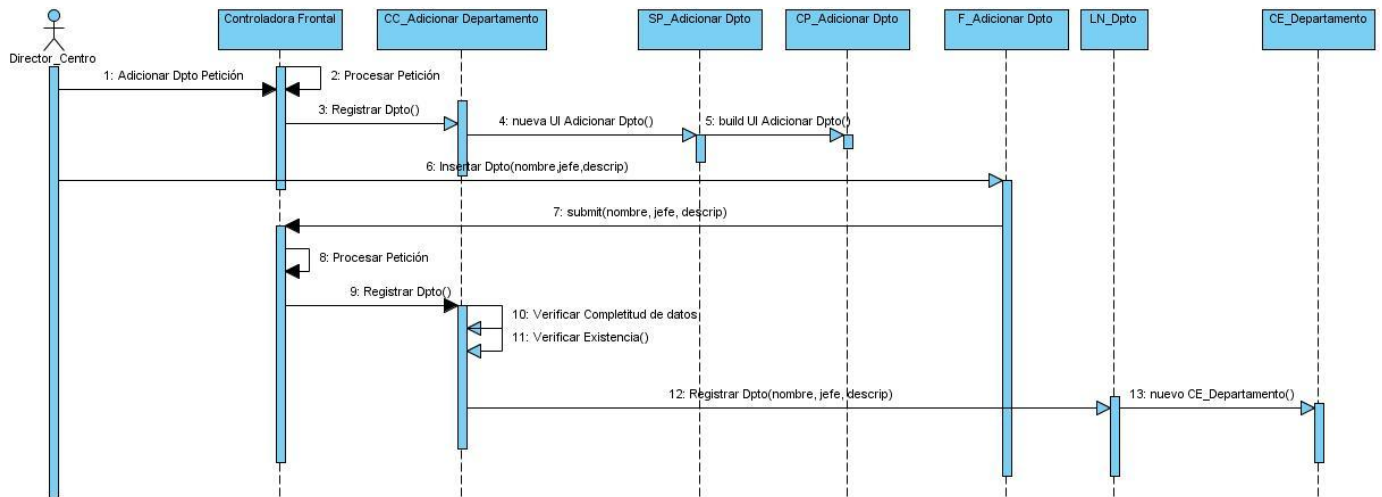


Figura 37 Diagrama de Secuencia del Diseño Adicionar Departamento

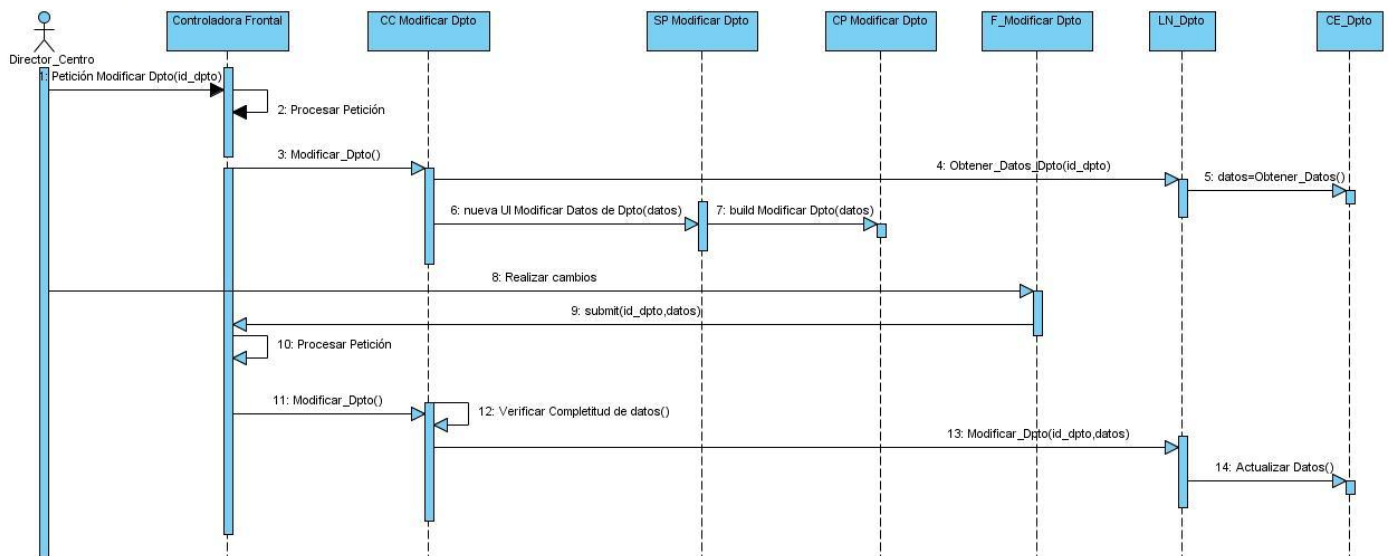


Figura 38 Diagrama de Secuencia del Diseño Modificar Departamento

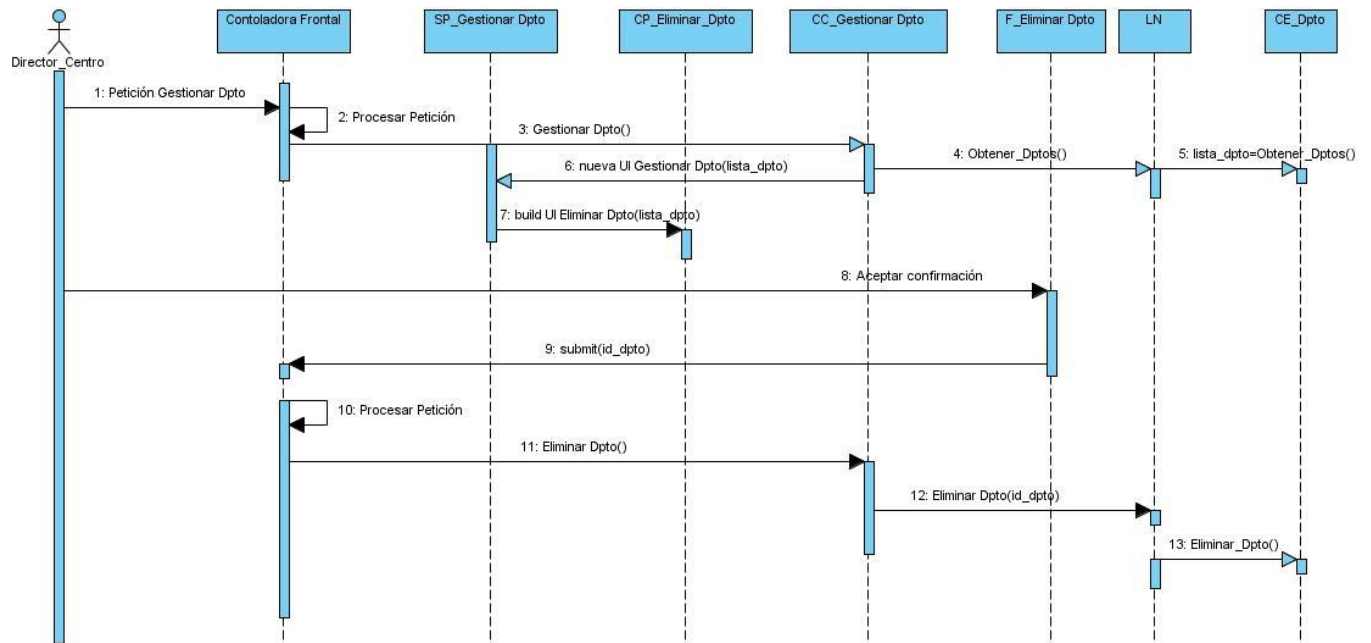


Figura 39 Diagrama de Secuencia del Diseño Eliminar Departamento

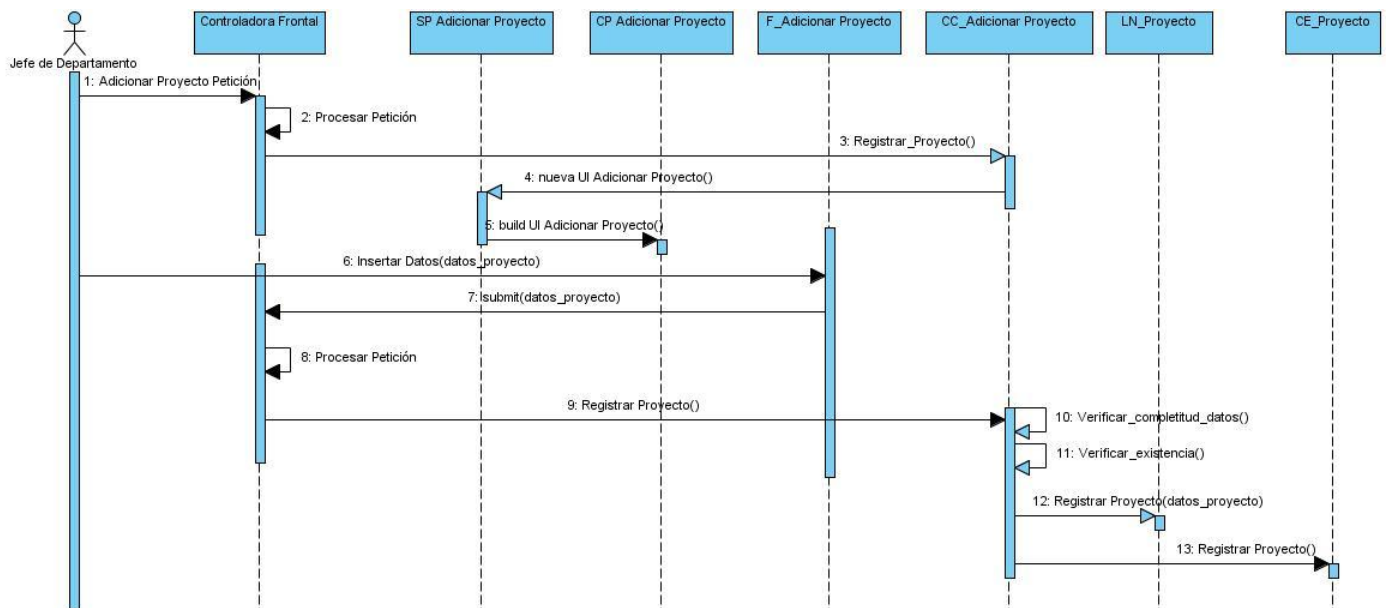


Figura 40 Diagrama de Secuencia del Diseño Adicionar Proyecto

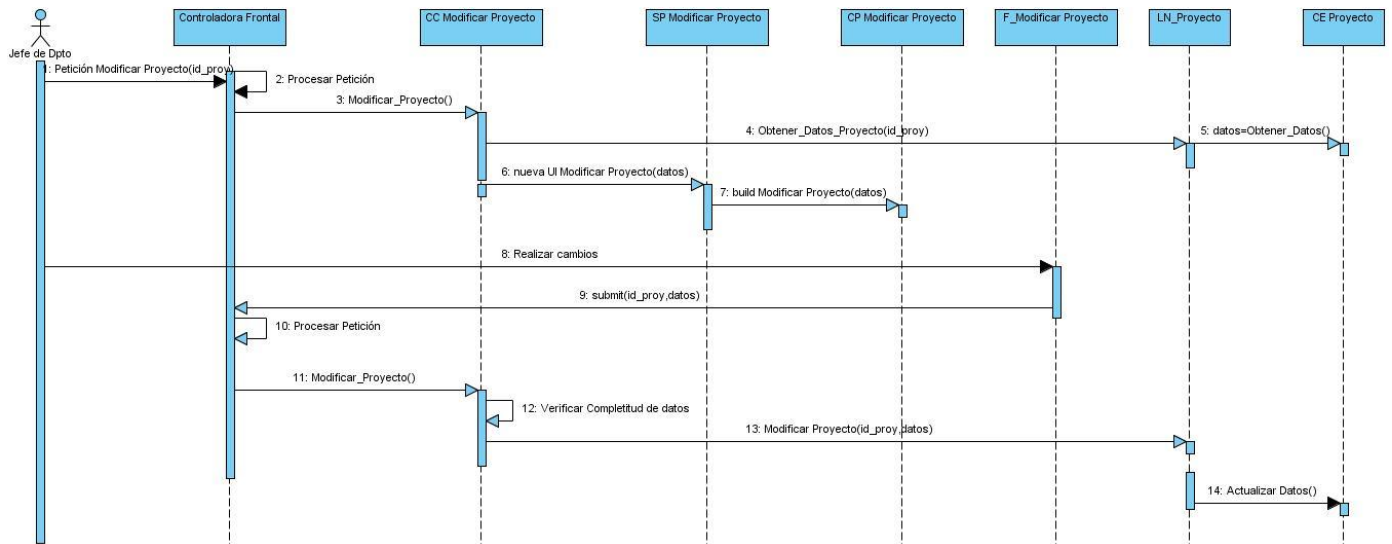


Figura 41 Diagrama de Secuencia del Diseño Modificar Datos de Proyecto

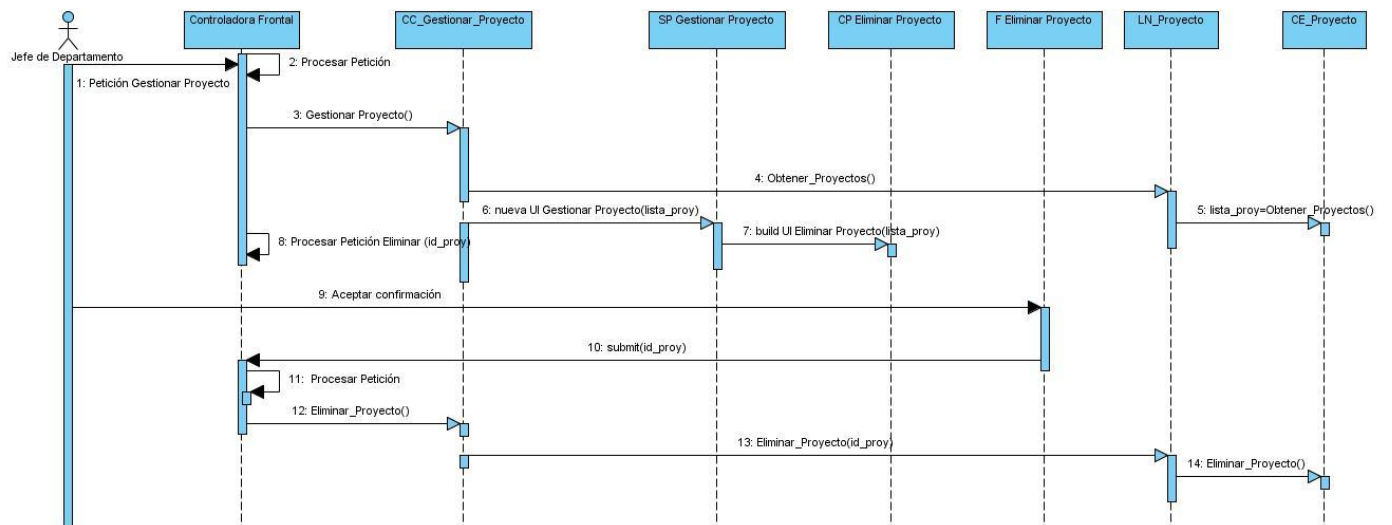


Figura 42 Diagrama de Secuencia del Diseño Eliminar Proyecto

3.6 Patrones

Un patrón es un modelo aplicable a una situación determinada para darle solución.

Los patrones:

- Son una abstracción de “problema – solución”.
- Atacan problemas que ocurren en situaciones específicas y dan una solución.
- Proporcionan vocabulario y entendimiento común.

3.6.1 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño. Para un mejor diseño de la aplicación se tuvo en cuenta los patrones GRASP (Patrones de Principios Generales para Asignar Responsabilidades) Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades. Los patrones de asignación de responsabilidades usados para el desarrollo de la aplicación fueron:

Bajo Acoplamiento: plantea que las clases tengan bajas dependencias entre ellas para cuando haya cambios en un componente no afecte a otros componentes, además sea fácil de entender y conveniente de reutilizar.

Beneficios:

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

Alta Conexión: plantea que una clase con alta conexión tiene pocos métodos y funcionalidad altamente relacionada y no realiza mucho trabajo ya que comparte el esfuerzo si la tarea es muy extensa.

Beneficios:

- Se incrementa la claridad y facilita la comprensión del diseño.
- Se simplifican el mantenimiento y las mejoras.

- Se soporta a menudo bajo acoplamiento.
- El grano fino de funcionalidad altamente relacionada incrementa la reutilización porque una clase cohesiva se puede utilizar para un propósito muy específico.

Experto: plantea que se debe asignar la responsabilidad al experto en información que en este caso sería la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Beneficios:

- Se mantiene el encapsulamiento de la información, puesto que los objetos utilizan su propia información para llevar a cabo las tareas. Normalmente, esto conlleva un bajo acoplamiento, lo que da lugar a sistemas más robustos y más fáciles de mantener.
- Se distribuye el comportamiento entre las clases que contienen la información requerida, por tanto, se estimula las definiciones de clases más cohesivas y ligeras que son más fáciles de entender y mantener.
- Se soporta normalmente una alta cohesión
- Bajo Acoplamiento.

Creador: plantea la necesidad de asignarle a una clase la responsabilidad de crear una instancia de otra clase.

Beneficios:

- Se soporta bajo acoplamiento lo que implica menos dependencias de mantenimiento y mayores oportunidades para reutilizar. Probablemente no se incrementa el acoplamiento porque la clase creada es presumible que ya sea visible a la clase creadora, debido a las asociaciones existentes que motivaron su elección como creador.

Controlador: es la clase responsable de recibir mensajes de eventos del sistema, normalmente no hace el trabajo sino lo delega a otras clases (Larman 2002).

3.6.2 Patrones de Arquitectura

El sistema propuesto se ha diseñado teniendo en cuenta el patrón arquitectónico MVC, que no es más que un estilo de arquitectura de software que separa una aplicación interactiva en tres componentes distintos: modelo, vista y controlador.

Modelo: contiene los datos y la funcionalidad esencial.

Vista: despliega la información al usuario, es decir, presentación de información por pantalla.

Controlador: recibe los eventos de la interfaz, invoca al negocio según la acción lanzada desde la interfaz, espera la respuesta del negocio y actualiza la interfaz.

Desarrollar una aplicación siguiendo este patrón arquitectónico tiene muchas ventajas:

- El componente del modelo se desarrolla de manera independiente del componente vista.
- Las vistas siempre van a estar actualizadas.
- Si se desea hacer modificaciones en el modelo como agregar datos o métodos, solo debe modificarse el modelo y las interfaces del mismo, sin afectar la aplicación en su totalidad.
- Las modificaciones a las vistas no afectan los otros módulos de la aplicación.

Por tales motivos y por permitir bajo acoplamiento entre los componentes de modelo, vista y controlador fue utilizado dicho patrón arquitectónico para el diseño realizado en la investigación.

3.7 Principios del diseño

3.7.1 Diseño de la interfaz de la aplicación

Existen dentro del diseño Web aparte de la combinación de colores y la apariencia estética otros elementos a tener en cuenta, por ejemplo se necesita que los servicios que se brinden en la aplicación se puedan usar y que la forma de presentarse la información sea útil para los usuarios. Una aplicación con una interfaz bien diseñada debe tener, además de un buen diseño

gráfico, una buena navegabilidad y usabilidad. Siguiendo estos objetivos planteados se propone que la aplicación web a desarrollar cumpla con los siguientes principios:

- Hacer sentir al usuario que tiene el control de la interacción incluyendo siempre la posibilidad de que el usuario pueda cancelar la acción una vez que la inicie.
- Dar al usuario diversas formas de realizar una acción relacionada con la interfaz y disculparse con elegancia ante un error de la aplicación.
- Colocar la información fundamental en la parte superior izquierda de la pantalla.
- Destacar la legibilidad y comprensión del texto.
- Restringir la cantidad de colores a utilizar, pues muchos colores disocian al usuario de su tarea a realizar.
- Restringir el uso de fuentes como cursiva.
- Mantener los componentes (como los botones y formularios) del mismo tamaño
- Requerir un mínimo proceso de aprendizaje y utilización de la misma desde el primer momento, por cualquier usuario que tenga dominio aunque sea escaso de computación.
- El usuario debe tener conocimiento en cualquier momento de su navegación por la aplicación, sobre qué punto de la estructura del sistema se encuentra, también debe tener a la mano el mecanismo de cómo regresar al inicio o a puntos claves de la misma.
- Mostrarle solamente al usuario aquellas opciones a las que dado su nivel de acceso tiene establecido acceder. (Schmulle 2000)

3.7.2 Estándares de la interfaz de la aplicación

Con el objetivo de lograr un diseño uniforme de la interfaz de la aplicación, el sistema debe estar conformado por una plantilla que contenga el banner y el logo establecido para una mejor identificación. Debe contar con un menú de secciones ubicado en la parte superior de cada página con los vínculos necesarios para gestionar los recursos humanos y materiales del centro de

desarrollo “GEySED”, y en el área de interacción ubicada debajo del menú de secciones se mostrarán los distintos formularios que permiten al usuario ingresar información y desarrollar la actividad que la aplicación ofrece llevar a cabo. Se deben utilizar pequeños íconos para una mayor comprensión de las acciones. Para una adecuada visualización las letras deben tener un tamaño adecuado para personas con problemas de visión disminuida y el color de los textos deben contrastar con el color de fondo.

3.8 Conclusiones Parciales

En este capítulo fueron modelados los artefactos que tienen lugar en los flujos de trabajo de requerimientos y análisis y diseño, sustentándose de esta forma la base para la implementación del sistema propuesto.

Capítulo 4 “Validación de la Solución Propuesta”

4.1 Introducción

El objetivo fundamental de la ingeniería de software es obtener un sistema o aplicación con alta calidad. Para alcanzar este objetivo se utilizan diferentes métricas de validación de los artefactos que se producen durante los flujos de trabajo de la ingeniería de software.

En el desarrollo del capítulo se utilizan diferentes métricas para evaluar la calidad de los artefactos generados en la investigación. Los artefactos son: especificación de requisitos y modelo de caso de uso del sistema.

4.2 Definiciones de Métricas

Para entender mejor el concepto de métricas es fundamental aclarar primero que los términos medida, medición y métricas representan definiciones muy diferentes.

Dentro del contexto de la ingeniería del software, se definen estos tres conceptos como:

“Una medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones o capacidad algunos atributos de un proceso o producto.” (Presman 2005)

“La medición es el acto de determinar una medida.” (Presman 2005)

Y por último métrica se define como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.” (Presman 2005)

Se define las métricas del software como: “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos”.(Doria 2001)

4.3 Aplicación de la métrica de la calidad de la especificación de requisitos.

Para la validación de los requisitos existen diversos factores como la especificidad, corrección, comprensión, concisión, entre otros, que son conocidos como medida de la calidad. En la investigación se aplicará la especificidad que no es más que la ausencia de ambigüedad en los requisitos capturados.

Primeramente se tiene Nr que representa el total de requisitos:

$$Nr = Nf + Nnf$$

Donde Nf es el número de requisitos funcionales y Nnf es el número de requisitos no funcionales.

Para determinar la especificidad (ausencia de ambigüedad) de 18 requisitos. Se sugiere una métrica basada en la consistencia de la interpretación de los revisores de requisitos:

$$Q = Nui / Nr$$

Donde Nui representa el número de requisitos para los que los revisores tuvieron interpretaciones idénticas, es importante resaltar que cuando más cerca de 1 este el valor Q, menos ambigüedad presentará la especificación.

En el presente trabajo se realizaron dos revisiones para llegar a obtener un análisis sin ambigüedad en los requisitos, y para que se reflejara con más claridad las necesidades del cliente, en las cuales se obtuvieron los siguientes resultados:

Revisión 1:

Durante la revisión de un total de **26** requisitos, entre funcionales y no funcionales, se detectaron de los **18** requisitos funcionales, tres que no estaban claramente especificados, es el caso de gestionar ubicación, gestionar nivel de acceso y gestionar permisos, ya que los requisitos no expresaban con claridad su objetivo. Por lo que en todos los requisitos menos en 3, los revisores tuvieron interpretaciones bastante parecidas.

$$Q = Nui / Nr$$

$$Q = 23 / 26$$

$$Q = 0.88$$

Revisión 2:

Durante esta revisión no se detectaron ambigüedades en los requisitos revisados, debido a la realización de un profundo análisis. Por lo que de los **26** requisitos con los que se cuentan, los revisores tuvieron interpretaciones bastante parecidas.

$$Q = Nui / Nr$$

$$Q = 26 / 26$$

$$Q = 1$$

Después de haber realizado las dos revisiones se puede llegar a la conclusión de que los requisitos obtenidos presentan una ausencia de ambigüedad en su totalidad, debido a que el valor de Q es igual a 1.

Con este resultado y con la aplicación de la lista de chequeo especificación de requisitos que se hizo a las funcionalidades del sistema se pone de manifiesto el grado de calidad que presentan los requisitos especificados, artefacto fundamental para el desarrollo de un software que sea capaz de cumplir con las expectativas y necesidades de los clientes.

4.4 Métricas de Casos de Uso del Sistema

Durante el flujo de trabajo de requerimientos es común utilizar los casos de uso para especificar los requisitos funcionales. La técnica de utilizar el lenguaje natural para la descripción textual de los casos de uso provoca la existencia de ambigüedad en las interpretaciones de los actores de casos de uso. Es por esto que se hace necesaria la validación de estos a través de un modelo de métricas, el cual está compuesto por 4 atributos fundamentales (completitud, comprensibilidad, concisión, no trivialidad).

Compleitud

Un caso de uso es completo si especifica todo lo que deben hacer el actor y el sistema para alcanzar el objetivo del caso de uso.

Comprensibilidad

Un caso de uso es comprensible si todos los tipos de lectores pueden entenderlo fácilmente con una mínima explicación del autor.

Concisión

Un caso de uso es conciso si no incluye información innecesaria.

No trivialidad

Un caso de uso es no trivial si su secuencia de pasos conduce al actor a conseguir el objetivo que persigue la realización del caso de uso. (Bernárdez, Durán et al. 2004)

Tabla 6 Modelo de Métricas

Métricas	Preguntas
Compleitud	<ol style="list-style-type: none">1. ¿Hay respuestas a todas las peticiones que el actor del caso de uso hace al sistema y viceversa?2. ¿Se contemplan todos los posibles escenarios para poder alcanzar el objetivo del caso de uso?3. ¿Se especifican todas las secuencias alternativas a la secuencia normal?4. ¿Se contemplan todas las posibles excepciones a la secuencia normal?
Comprensibilidad	<ol style="list-style-type: none">1. ¿Es posible leer el caso de uso sin volver atrás en repetidas ocasiones?2. ¿Es difícil seguir la secuencia normal del caso de uso por la

	<p>presencia de las relaciones inclusión o extensión?</p> <ol style="list-style-type: none"> ¿Es difícil seguir la secuencia de pasos por la existencia de demasiados pasos alternativos? ¿Se han desglosado demasiado los pasos de algún actor o del sistema provocando que el caso de uso avance a un ritmo muy lento? ¿Aparecen pasos condicionales para expresar que el sistema comprueba una situación que permite al caso de uso continuar su realización?
Concisión	<ol style="list-style-type: none"> ¿Podría el caso de uso ser expresado con menos palabras? ¿Existen elementos que se pueden obviar o aparecen anotaciones innecesarias y que dificultan la lectura del caso de uso? ¿Aparecen demasiadas interacciones entre el actor principal del caso de uso y otros elementos del entorno?
No trivialidad	<ol style="list-style-type: none"> ¿Expresa el nombre del caso de uso un objetivo de un usuario que el sistema debe implementar? ¿Conduce el caso de uso al actor a conseguir alguno de sus objetivos sin representar un conjunto de interacciones triviales?

Al aplicar las métricas expuestas anteriormente a los casos de uso los revisores tuvieron las siguientes interpretaciones:

- En el caso de uso Autenticar se detectó que es difícil seguir la secuencia de pasos por la existencia de demasiados pasos alternativos.

- En el caso de uso Gestionar Departamento se detectó que no se contemplan todos los posibles escenarios para poder alcanzar el objetivo del caso de uso y no se especifican todas las secuencias alternativas a la secuencia normal.
- En el caso de uso Gestionar Proyecto no se especifican todas las secuencias alternativas a la secuencia normal.
- En el caso de uso Gestionar asignación de departamento a RRHH se detectó que el caso de uso podría ser expresado con menos palabras, existen elementos que se pueden obviar y aparecen anotaciones innecesarias que dificultan la lectura del caso de uso.
- En el caso de uso Gestionar Expediente de Proyecto no se contemplan todas las posibles excepciones a la secuencia normal.

Luego de sacar un promedio de las imprecisiones que se obtuvieron de las revisiones a los casos de uso resultó ser que estos cumplen con un 96.4% con las respuestas a las preguntas hechas con anterioridad.

4.5 Conclusiones Parciales

Al medir la calidad de los artefactos generados a través de las métricas especificadas en este capítulo, se logró una especificación exitosa de los requisitos, además se garantizó la funcionalidad del sistema, alcanzando la satisfacción del cliente y sus necesidades. Las métricas aplicadas ofrecen la manera correcta en que deben estar los artefactos implementados por el rol de analista de sistema.

Conclusiones Generales

Con la realización del diseño de un sistema automatizado, se dio respuesta a al problema planteado cumpliéndose el objetivo general para el cual se desarrollaron las diferentes tareas de manera satisfactoria.

Una vez culminado el proceso investigativo se arribó a las siguientes conclusiones:

- La metodología de desarrollo, lenguaje y herramienta de modelado utilizada corresponden de acuerdo con las premisas de soberanía tecnológica impulsada por la Universidad.
- El rol analista de sistema juega un papel muy importante en el desarrollo de software, pues es el encargado de transformar la información obtenida por los clientes en un lenguaje entendible para el equipo de desarrollo.
- El uso de las estrategias de captura de requisitos facilitó la identificación de los requisitos del sistema de una manera eficiente y confiable.
- Se obtuvieron los artefactos correspondientes a los flujos de trabajo: Requerimientos y Análisis y Diseño según establece la metodología de desarrollo de software utilizada.
- Los artefactos obtenidos posibilitaron detallar los procesos que se llevan a cabo en el centro de desarrollo "GEySED" con el objetivo de lograr un entendimiento común entre el cliente y el equipo de desarrollo.
- Se evaluó la calidad de los principales artefactos generados, mediante métricas definidas por autores reconocidos, lo que permitió confirmar el 100% de calidad de los requisitos.
- Al final se sentaron todas las bases para una futura implementación minimizando las trabas que puedan manifestarse para garantizar la calidad del software.

Recomendaciones

La autora de la presente investigación recomienda:

- Que la investigación sea utilizada para una posterior implementación del sistema propuesto, en aras de tener la información de los recursos humanos y materiales del centro de desarrollo “GEySED” centralizada y controlada.
- Refinar los requisitos funcionales con el cliente en próximas versiones del producto, con el objetivo de aumentar las funcionalidades del sistema a desarrollar y lograr una mayor satisfacción del cliente.

Referencias Bibliográficas

Referencias Bibliográficas

- Aguila, A. G. (2009). Análisis y diseño de un sistema automatizado para el control de los recursos humanos en los polos productivos de la facultad 9. Cuba, UCI.
- Bernárdez, B., A. Durán, et al. (2004) "UNA PROPUESTA PARA LA VERIFICACIÓN DE REQUISITOS BASADA EN MÉTRICAS." Volume, DOI:
- Corporation, M. (2010). "Gestión de Recursos Humanos " Retrieved 15,12, 2009, from http://www.microsoft.com/spain/businesssolutions/dynamics/productos/ax/area_negocio/rh.h.msp
- Crecenegocios.com. Retrieved 30, 08, 2009, from <http://www.crecenegocios.com/ventajas-competitivas/>
- Doria, H. G. (2001). Las Métricas de Software y su Uso en la Región.
- Genix, A. d. I. C. I., Y. G. Roblejo, et al. (2009) "Estudio sobre la información que se gestiona de los recursos humanos, en los ERP." Volume, DOI:
- González, Z. (2006) "DEFINICIÓN DE ADMINISTRACIÓN DE RECURSOS HUMANOS " Volume, DOI:
- Intelmax.com. (2001). "Automatización." Retrieved 18,11, 2009, from www.intelmax.com/ensanluis/images/capitulo3.ppt
- Jackson, S. and J. V. Slocum. (2008). "Administración: Un enfoque basado en competencias." Retrieved 15 de abril, 2010, from <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r47773.DOC>.
- JACOBSON, I. (1998). El Proceso Unificado de Desarrollo de Software.
- Larman, C. (2002). UML y Patrones.
- McFarland, D. E. (1989). Administración de personal. México.
- Mejía, G., L. R. Balkin, et al. (2001). Dirección y Gestión de los Recursos Humanos. 2009.
- ÖVERGAARD, G. and K. PALMKVIST (2004). Use Cases: Patterns and Blueprints, Addison Wesley.
- Presman, R. S. (2005). Ingeniería de Software. Un enfoque práctico.
- Redmin.cl. (2009). Retrieved 14,11, 2009, from <http://redmin.cl/?a=9265>
- Reyes, L. d. T. (2009) "Los recursos humanos en la gestión de proyectos como capital humano." Volume, DOI:

Referencias Bibliográficas

- Rojas Romero, J. D. (2006). "Glosario de administración de empresas." Retrieved 5, 11, 2009, from <http://www.slideshare.net/cindyjvera/conceptos-basicos-de-administracion-de-servicios-de-informacion>
- Saavedra, G. R. "Gestión, Calidad y Agregación de valor en información." Retrieved 05, 11, 2009, from <http://b3.bibliotecologia.cl/ar-gestion.htm>
- Schmuller, J. (2000). Aprendiendo UML en 24 horas.
- Serrano, J. C. R. (2004). El Modelo de Gestión de Recursos Humanos.
- Visconti, M. and H. Austidillo (2005) "Fundamento de Ingeniería de Software." Volume, DOI:
- Zaldívar, M. "La ventaja competitiva a través de la gestión de recursos humanos." Volume, DOI:
- Zayas, C. Á. d. (1995). "METODOLOGIA DE LA INVESTIGACION CIENTIFICA." Retrieved 02,12, 2009, from <http://eva.uci.cu/mod/resource/view.php?id=2740>

Bibliografía Consultada

- Aguila, A. G. (2009). Análisis y diseño de un sistema automatizado para el control de los recursos humanos en los polos productivos de la facultad 9. Cuba, UCI.
- Bernárdez, B., A. Durán, et al. (2004) "UNA PROPUESTA PARA LA VERIFICACIÓN DE REQUISITOS BASADA EN MÉTRICAS." Volume, DOI:
- Corporation, M. (2010). "Gestión de Recursos Humanos " From http://www.microsoft.com/spain/businesssolutions/dynamics/productos/ax/area_negocio/rh.h.mspx
- Crecenegocios.com. From <http://www.crecenegocios.com/ventajas-competitivas/>
- Doria, H. G. (2001). Las Métricas de Software y su Uso en la Región.
- Genix, A. d. I. C. I., Y. G. Roblejo, et al. (2009) "Estudio sobre la información que se gestiona de los recursos humanos, en los ERP." Volume, DOI:
- González, Z. (2006) "DEFINICIÓN DE ADMINISTRACIÓN DE RECURSOS HUMANOS " Volume, DOI:
- Intelmax.com. (2001). "Automatización." From www.intelmax.com/ensanluis/images/capitulo3.ppt
- Jackson, S. and J. V. Slocum. (2008). "Administración: Un enfoque basado en competencias." From <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r47773.DOC>.
- JACOBSON, I. (1998). El Proceso Unificado de Desarrollo de Software.
- Larman, C. (2002). UML y Patrones.
- McFarland, D. E. (1989). Administración de personal. México.
- Mejía, G., L. R. Balkin, et al. (2001). Dirección y Gestión de los Recursos Humanos. 2009.
- ÖVERGAARD, G. and K. PALMKVIST (2004). Use Cases: Patterns and Blueprints, Addison Wesley.
- Presman, R. S. (2005). Ingeniería de Software. Un enfoque práctico.
- Redmin.cl. (2009). From <http://redmin.cl/?a=9265>

Bibliografía Consultada

Reyes, L. d. T. (2009) "Los recursos humanos en la gestión de proyectos como capital humano." Volume, DOI:

Rojas Romero, J. D. (2006). "Glosario de administración de empresas." From <http://www.slideshare.net/cindyjvera/conceptos-basicos-de-administracion-de-servicios-de-informacion>

Saavedra, G. R. "Gestión, Calidad y Agregación de valor en información." From <http://b3.bibliotecologia.cl/ar-gestion.htm>

Schmuller, J. (2000). Aprendiendo UML en 24 horas.

Serrano, J. C. R. (2004). El Modelo de Gestión de Recursos Humanos.

Visconti, M. and H. Austidillo (2005) "Fundamento de Ingeniería de Software." Volume, DOI:

Zaldívar, M. "La ventaja competitiva a través de la gestión de recursos humanos." Volume, DOI:

Zayas, C. Á. d. (1995). "METODOLOGIA DE LA INVESTIGACION CIENTIFICA." From <http://eva.uci.cu/mod/resource/view.php?id=2740>