

**SISTEMA DE GESTIÓN DE TIEMPO DE LOS ESTUDIANTES
VINCULADOS AL PROYECTO PROCYON.**

TESIS DE GRADO EN INGENIERIA EN LAS CIENCIAS INFORMÁTICAS.



Tesistas: Niurys Álvarez Delgado.

Lisbettes Guilbeaux Gregorich.

Tutor: Ing. Mayra de Rojas Pérez.

Ciudad de la Habana, Julio 2007.

“Año 49 de la Revolución

La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Lisbettes Guilbeaux Gregorich.

Mayra de Rojas Pérez.

Niurys Alvarez Delgado.

Dedicatoria

A mis padres Ángela Berenice Gregorich Rondón y Filiberto Guilbeaux Terrero, mi razón de ser y mi inspiración.

De Lisbettes Guilbeaux Gregorich

Mi mayor alegría es poder dedicar este trabajo a mis padres Nancy Delgado González y Roberto Alvarez Vienes y a mi hermano Roberto Alvarez Delgado, por ser mi razón de ser y por el inmenso amor que me han brindado siempre. Sin ellos tanto sacrificio no hubiese sido posible. Les debo todo lo que he alcanzado hasta hoy.

De Niurys Alvarez Delgado

Agradecimientos

A mi mamá, por acompañarme en todo momento, por comprenderme, aconsejarme, apoyarme y por todo su amor incondicional.

A mi papá, por todos estos años de sacrificio, por todos sus consejos y por todo su amor.

A mi novio Carlos Capote, por todo su amor y apoyo incondicional.

A mis tíos, por su constante preocupación, apoyo y darme ánimo en todo momento.

A mis primos por todo su amor y cariño.

A mis suegros, por haberme acogido como a una más de la familia.

A mis hermanos que aunque no han estado conmigo para compartir momentos importantes en mi vida, me han enseñado muchas cosas.

A mi abuela Jorgelina que aunque ya no esta conmigo siempre esta en mi corazón.

A mi abuelito que aunque no puede ya aconsejarme, lo hizo cuando pudo.

A Antonio, por dedicarme tantas horas de su valioso tiempo.

A Liliam, por haber sido mí amiga en estos años de universidad.

A mis compañeros de tesis, por tantas dudas aclaradas y ayuda brindada en momentos difíciles.

A todos mis profesores, que me transmitieron conocimientos y valores que me permitieron formarme como profesional.

A Arnaldo Noa, por sus consejos y su apoyo en los momentos más difíciles de mi carrera.

A Mayra, tutora de esta investigación, por haberme transmitido sus conocimientos y ayudarme a desarrollar este trabajo.

A la revolución por permitirme estudiar en esta universidad de excelencia.

A Nuestros Comandante Fidel Castro Ruz, por haber echo este sueño realidad y por todo lo que me a enseñado en toda mi vida.

Gracias a todos he logrado que mi sueño se haya hecho realidad.

De Lisbettes Guilbeaux Gregorich.

Agradecimientos

Quiero agradecer a aquellas personas que han estado a mi lado en momentos difíciles, brindándome su cariño y apoyo incondicional.

A mi mamá por estar a mi lado siempre, porque sin sus consejos no hubiese hecho realidad este gran sueño. Por ser mi guía. Por su amor incondicional.

A mi papá por confiar en mí y apoyarme en todo momento. Por el inmenso cariño que siempre me ha brindado. Por dedicarme su vida.

A mi hermano por ser mi fiel ejemplo de sacrificio. Por enseñarme que en la vida hay que luchar por alcanzar todos nuestros sueños y metas. Por estar conmigo a cada instante en mi corazón.

A mi familia por el apoyo y cariño que me brindan a diario. Por ayudarme en todos estos años de estudio.

A Marlon Rodríguez, por estar a mi lado en uno de los momentos más importantes de mi carrera. Por su cariño e inmenso corazón.

A Antonio Marrero, por su paciencia, por brindarme su ayuda incondicional, imprescindible en la realización de este trabajo.

A mis amistades por estar a mi lado en todo momento.

A mis profesores por enseñarme y ayudarme en el plano profesional y personal.

A mi tutora Mayra De Rojas por confiar en nosotras y ayudarnos en todos estos meses de tanto trabajo.

A nuestro Comandante Fidel Castro por haberme dado la posibilidad de estudiar en una Universidad como la UCI, de la cual me siento orgullosa.

De Niurys Alvarez Delgado

Resumen

Aprovechar al máximo el tiempo efectivo de trabajo es uno de los más importante elementos de productividad. La entidad PROCYON, quien se dedica a elaborar productos de software para las telecomunicaciones móviles y a la prestación de servicios en esta área, no está ajeno a esta situación. Para dicha institución el registro y control de todos los datos productivos de sus trabajadores resulta de vital importancia en la toma de decisiones de trabajo rápidas y justas que involucren a los miembros del proyecto.

Una vez que los estudiantes pasan a formar parte de la empresa son considerados trabajadores de la misma aunque con características propias. Es por esto que surge la necesidad de contar con un sistema automatizado que permita administrar la información necesaria para dar cumplimiento al convenio y su posterior seguimiento. Se necesita además, mantener un registro actualizado del tiempo de trabajo de los estudiantes vinculados al proyecto. Partiendo de esta idea se define el objetivo general del trabajo:

Implementar un sistema informático para la automatización de la gestión del tiempo de los estudiantes vinculados a PROCYON.

El sistema garantiza en todo momento, ofrecer a los usuarios autorizados una información detallada sobre los proyectos en que trabaja el estudiante, la tarea que desempeña en el mismo y la fase en la que se encuentra. Permite a través de la integración del sistema con otros recursos de la red de PROCYON, como el sistema de control de versiones y el Proxy conocer la actividad realizada por el estudiante en Internet y en el repositorio. Además posibilita realizar evaluaciones periódicas a los estudiantes según su comportamiento y resultados en el trabajo, quedando registradas en la base de datos del sistema. El sistema propuesto gestiona toda la información relacionada con los datos del convenio de forma rápida, requiriendo un menor esfuerzo por parte del personal encargado de esta actividad a la hora de organizar toda la información.

Índice

Índice	V
Introducción	1
Antecedentes	2
Objeto de estudio	2
Objetivo General	2
Objetivos Específicos	3
Tareas	3
Estructura del documento	4
Capítulo 1: Fundamentación Teórica	6
1.1 Introducción	6
1.2 Gestión de Recursos Humanos	6
1.3 Gestión de la Evaluación	7
1.4 Gestión del Tiempo	8
1.4.1 Gestionar el tiempo de los empleados monitoreando el tráfico de Internet	8
1.4.2 Servidores Proxy para monitorear los accesos a Internet	9
1.4.3 Software que procesan la información de los ficheros logs	10
1.4.3.1 Software existente en Cuba	10
1.5 Software utilizados en PROCYON	10
1.5.1 Proxy Squid	11
1.5.2 Sistema de Control de Versiones TortoiseSVN	11
1.5.3 Windows Server 2003	11
1.6 Tecnología seleccionada para la solución del problema	12
1.6.1 Tecnología Java	12
1.6.2 Java Development Kit	13
1.6.3 Máquina virtual Java	14
1.6.4 Servidor Web Apache Tomcat	14
1.6.5 Servidor de Base de Datos PostgradeSQL	15
1.6.6 Definición de la Arquitectura Software	16
1.6.6.2 Tecnologías elegidas para cada capa	17
1.6.7 Proceso Unificado de Desarrollo de Software (RUP)	18
1.7 Herramientas utilizadas	19
1.7.1 Entorno integrado de desarrollo Eclipse	19
1.7.2 My Eclipse	19
1.7.3 Hibernate	19
1.7.4 Rational Rose	20
1.7.5 Diseño de Interfaz. Macromedia Dreamweaver	20
1.8 Conclusiones	21
Capítulo 2. Características del Sistema	22
2.1 Introducción	22
2.2 Objeto de estudio	22
2.3 Problema y Situación Problemática	22
2.3.1 Objetivos estratégicos de la organización y procesos de negocio que los soportan	22

2.3.2 Flujo actual de los procesos involucrados en el campo de acción. Causas que originan la situación problemática.....	23
2.4.1 Procesos que serán objeto de automatización.	24
2.4.2 Sistemas automatizados que existen en la empresa para la Gestión del Tiempo.....	24
2.4.2.1 Time Tracker.	25
2.5 Información que se maneja.	25
2.6 Propuesta de sistema.....	25
2.6.1 Descripción general de la propuesta de sistema.	25
2.6.2 Análisis comparativo de otras soluciones existentes con la propuesta.	25
2.6.2.1 Time Tracker v0.5.	26
2.7 Modelo del negocio.	26
2.7.1 Descripción general de los procesos de negocio.....	26
2.7.2 Definición de los actores y trabajadores del negocio.	26
2.7.3 Diagrama de Casos de Uso del Negocio.....	27
2.7.4 Descripción textual del Caso de Uso de Negocio Asignar Proyecto. Ver Anexo 1.....	28
2.7.5 Diagrama de Actividad del Caso de Uso de Negocio Asignar Proyecto.....	28
Ver Anexo 2.....	28
2.7.6 Descripción textual del Caso de Uso de Negocio Firmar Convenio. Ver Anexo 3.....	28
2.7.7 Diagrama de Actividad del Caso de Uso de Negocio Firmar Convenio.....	28
Ver Anexo 4.....	28
2.8 Especificación de los requisitos de software.	28
2.8.1 Dependencias y relaciones con otros software.....	28
2.8.2 Requerimientos funcionales.	28
2.8.3 Requerimientos no Funcionales.	29
2.8.3.1 Requerimientos de Software.	30
2.8.3.3 Requerimientos de Portabilidad.....	30
2.8.3.4 Requerimientos de Apariencia o interfaz externa.	30
2.8.3.5 Requerimientos de Usabilidad.....	30
2.8.3.6 Requerimientos de Rendimiento.	30
2.8.3.7 Requerimientos de Seguridad y Privacidad.	30
2.9 Modelo de Casos de Uso del Sistema.....	31
2.9.1 Definición de los actores del sistema a automatizar.	31
2.9.2 Diagrama de Casos de Uso del Sistema a automatizar.....	32
2.9.3 Descripción de los Casos de uso del Sistema. Ver Anexo 6.....	33
2.10 Conclusiones.....	33
Capítulo 3. Análisis y Diseño del Sistema.	34
3.1 Introducción.	34
3.2 Modelo de Análisis.	34
3.2.1 Diagrama de Clases de Análisis. CU "Autenticar Usuario".....	35
3.2.2 Diagrama de Clases de Análisis. CU "Gestionar Usuario".	35
3.2.3 Diagrama de Clases de Análisis. CU "Administrar Estudiante".	36
3.2.5 Diagrama de Clases de Análisis. CU "Mostrar Actividad Estudiante".....	37
3.2.6 Diagrama de Clases de Análisis. CU "Gestionar Tiempo".	37
3.2.4 Diagrama de Clases de Análisis. CU "Gestionar Evaluación Estudiante".	38
3.2.7 Diagrama de Clases de Análisis. CU "Administrar Información Proyecto".	39

3.3 Modelo de Diseño.....	39
3.3.1 Diagramas de Interacción (Secuencia).....	39
3.3.3 Diagramas de Clases Diseño Web.....	41
3.3.3.1 Diagramas de Clases de Diseño Web. CU Autenticar Usuario.....	41
3.2.3.2 Diagramas de Clases de Diseño Web. CU Administrar Estudiante.....	42
3.2.3.3 Diagramas de Clases de Diseño Web. CU Gestionar Evaluación Estudiante.....	43
3.2.3.4 Diagramas de Clases de Diseño. CU Mostrar Actividad Estudiante.....	44
3.2.3.5 Diagramas de Clases de Diseño. CU Gestionar Tiempo.....	45
3.2.3.6 Diagramas de Clases de Diseño. CU Administrar Información Proyecto.....	46
3.2.3.7 Diagramas de Clases de Diseño. CU Gestionar Usuario.....	47
3.3 Diseño de la BD.....	48
3.3.1 Modelo lógico de datos (diagrama de clases persistentes).....	48
3.3.2 Modelo físico de datos (modelo de datos).....	49
3.4 Definiciones de diseño de la interfaz de usuario.....	50
3.5 Tratamiento de errores.....	50
3.6 Seguridad.....	51
3.7 Conclusiones.....	52
Capítulo 4. Implementación.....	53
4.1 Introducción.....	53
4.2 Modelo de Implementación.....	53
4.2.1 Diagrama de despliegue.....	53
4.2.2 Diagrama de componentes.....	54
4.2.2.1 Diagramas de componentes. CU Administrar Información Proyecto.....	54
4.2.2.2 Diagramas de componentes. CU “Autenticar Usuario”.....	55
4.2.2.3 Diagramas de componentes. CU “Administrar Estudiante”.....	55
4.2.2.4 Diagramas de componente. CU “Gestionar Evaluación Estudiante”.....	56
4.2.2.5 Diagrama de componente. CU “Mostrar Actividad Estudiante”.....	56
4.2.2.6 Diagrama de componente. CU “Gestionar Tiempo”.....	57
4.2.2.7 Diagrama de componente. CU “Gestionar Usuario”.....	57
4.3 Conclusiones.....	58
5.1 Introducción.....	59
5.2 Planificación basada en casos de uso.....	59
5.2.1 Paso 1. Cálculo de los Puntos de casos de uso Desajustados.....	59
5.2.1 Paso 2. Cálculo de los Puntos de casos de uso ajustados.....	61
5.2.3 Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso.....	64
5.2.4 Paso 4. Calcular esfuerzo de todo el proyecto.....	65
5.3 Beneficios tangibles e intangibles.....	66
5.4 Análisis de costos y beneficios.....	66
5.5 Conclusiones.....	67
Conclusiones Generales.....	68
Referencia Bibliográfica.....	70
Anexos.....	73
Anexo1.....	73
Descripción del caso de uso Asignar Proyecto.....	73
Tabla 1.1: Descripción del caso de uso Asignar Proyecto.....	73

Anexo 2.....	75
Diagramas de Actividad del caso de uso Asignar Proyecto.....	75
Anexo 3.....	76
Anexo 3.....	76
Descripción del caso de uso Firmar Convenio.	76
Tabla 3.1: Descripción del caso de uso Firmar Convenio.....	76
Anexo 4.....	77
Diagrama de Actividad del caso de uso Firmar Convenio.	77
Anexo 5.....	77
Diagrama de Clases del Modelo de Objetos	77
Anexo 6	78
Descripción de los casos de uso del sistema.	78
Tabla 6.1: Descripción del caso de uso Autenticar Usuario.....	78
Tabla 6.2: Descripción del caso de uso Gestionar Usuario.....	80
Tabla 6.3: Descripción del caso de uso Administrar Estudiante	84
Tabla 6.4: Descripción del caso de uso Mostrar Actividad Estudiante	88
Tabla 6.5: Descripción del caso de uso Gestionar Evaluación Estudiante.....	91
Tabla 6.6: Descripción del caso de uso Gestionar Tiempo.....	95
Tabla 6.7: Descripción del caso de uso Administrar Información Proyecto.....	97
Anexo 7.....	100
Fig. 7.1 Prototipo funcional del sistema.....	100
Fig. 7.2 Reportes del sistema.....	101
Fig. 7.3 Componentes utilizados.	101
Fig. 7.3.1 Componente Select One.	101
Fig. 7.3.2 Componente Select Many	102
Fig. 7.4 Mensajes de errores.....	102
Anexo 8.....	102
Glosario de términos y siglas.	103

Introducción.

PROCYON, firmó recientemente un convenio de colaboración con la Facultad 2, y no cuenta con un sistema automatizado para administrar la información necesaria para dar cumplimiento al mismo.

Se necesita de un sistema para la gestión del tiempo y los expedientes laborales de los estudiantes vinculados al proyecto, lo cual constituye el **problema a resolver** a través del presente trabajo.

Partiendo de la existencia de este problema, la **necesidad del trabajo** está dada por el hecho de que a PROCYON le urge implementar un sistema que integre la información de los tiempos de trabajo reportados y ejecutados por los estudiantes vinculados al proyecto, así como tener un registro de los proyectos en los que el estudiante trabaja, además de tener la información de cómo utiliza el tiempo y los servicios de la red mientras está en la entidad. Actualmente se cuenta con un sistema de Gestión de Tiempo de los estudiantes, un sistema de control de versiones (SVN Tortoise) y un Proxy (Squid), pero se hace necesario acceder a la información que nos brindan estos servicios de la red para tenerla en cuenta a la hora de evaluar el trabajo de los estudiantes y el uso que hacen del tiempo y de los servicios de la red local. En el proyecto no se cuenta con una herramienta automatizada para la administración de las evaluaciones de los estudiantes vinculados al proyecto, tal y como se establece en el Convenio de colaboración firmado entre la Facultad 2 y PROCYON. Los datos relacionados con los estudiantes se encuentran dispersos, lo que supone un gasto de tiempo innecesario y un mayor esfuerzo por parte del personal encargado de la toma de decisiones en un momento dado.

Antecedentes

Actualmente en el proyecto PROCYON, para la gestión del tiempo de los estudiantes, se utiliza el software WR Time Tracker v0.5 que es un sistema de monitoreo de proyectos basado en la Web, que permite hacer el seguimiento de los tiempos de trabajo dedicados a un proyecto, para cada uno de los colaboradores. Es fácil y rápido de usar. El software permite crear cuentas de usuarios, organizarlos en grupos de trabajo, crear y aportar todo tipo de modificaciones a múltiples proyectos, hacer el seguimiento en horas de trabajo de la dedicación de cada uno al proyecto, generar informes de trabajo y facturas, incluso enviar mensajes de correo electrónico.

En el desarrollo del presente trabajo se aborda una solución al problema existente en PROCYON, lo cual constituye los **aportes prácticos esperados del trabajo**.

Basados en el problema existente se definen en el presente trabajo:

Objeto de estudio

Registro y control del tiempo, las evaluaciones, el uso de los recursos de la red, y proyectos en los que han trabajado los estudiantes vinculados al proyecto PROCYON en la Universidad de las Ciencias Informáticas.

Aprovechar al máximo el tiempo efectivo de trabajo, es el más importante elemento de productividad.

Objetivo General

Implementar un sistema informático para la automatización de la gestión del tiempo de los estudiantes vinculados a PROCYON.

Objetivos Específicos

- Gestionar la información referente a los estudiantes.
- Registrar las actividades de los estudiantes y el tiempo dedicado a las mismas.
- Consultar el tiempo de trabajo de los estudiantes.
- Obtener las evaluaciones sistemáticas de los estudiantes.
- Registrar los proyectos en los que trabaja cada estudiante.
- Emitir reportes de las evaluaciones.
- Consultar los recursos de la red que utilizan los estudiantes.

Para dar cumplimiento a estos objetivos se desarrollan las siguientes tareas:

Tareas

- Selección de la metodología con la cual se trabajará.
- Interacción con el cliente donde se definen las bases del Modelo de Negocio.
- Especificación de los Requerimientos definidos a partir del Modelo de Negocio.
- Análisis de los requerimientos del cliente y propuesta de una solución.
- Especificación de la Interfaz de Usuario del sistema.
- Diseño de la Interfaz de Usuario del sistema.
- Presentación del prototipo de Interfaz de usuario.
- Investigación de interacción con control de versiones.
- Investigación de interacción con el Proxy.
- Definición de la arquitectura del sistema.
- Implementación de los componentes del sistema a automatizar.

Estructura del documento.

El presente documento está organizado de la siguiente forma. La introducción aborda una visión del problema a resolver, así como la actualidad y necesidad del trabajo a desarrollar. Muestra también los antecedentes utilizados como fuente de información, y un breve resumen de los aportes prácticos esperados del trabajo. Se detalla el objeto de estudio, objetivos generales y específicos del trabajo, y una síntesis de las tareas trazadas para cumplir con los objetivos.

El Capítulo 1 presenta la Fundamentación Teórica del trabajo. Muestra un estado del arte del tema de la Gestión del Tiempo, de la Evaluación, a nivel internacional, nacional y de la Universidad. Se selecciona la tecnología para dar solución al problema, definiendo la arquitectura del sistema y la metodología que se utilizó.

El Capítulo 2 muestra las características del sistema propuesto para dar solución al problema. Se detallan los objetivos estratégicos de la organización y los procesos de negocio que los soportan. Se describe el flujo actual de los procesos involucrados en el campo de acción y se realiza un análisis crítico de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problemática y las consecuencias. Se analizan los procesos que serán objeto de automatización, así como los sistemas automatizados que existen en la empresa y que están vinculados con el campo de acción. Se realiza un profundo análisis sobre la propuesta del sistema, cómo debe funcionar y un análisis comparativo de otras soluciones existentes con la propuesta, llegando a conclusiones sobre los aspectos en los que la propuesta se diferencia de las otras y las supera. Se describen los casos de uso del negocio, donde se indica quiénes son los actores y trabajadores del negocio y por qué lo son. Se presentan los diagramas de actividades del negocio y diagrama de clases del modelo de objetos. Se especifican los requerimientos de software, requerimientos funcionales y no funcionales, dependencias y relaciones con otros sistemas. Se definen y describen los casos de uso del sistema, los actores del sistema y la justificación de por qué lo son.

En el Capítulo 3 presenta los resultados de la etapa de análisis y diseño del sistema, guiándose por la metodología RUP. Se realiza la definición de los diagramas de clases de análisis y diseño respectivamente, así como los diagramas de interacción del diseño. Se define el Modelo de Datos, quedando especificados el diagrama Entidad Relación de la BD. Finalmente se explica el tratamiento de

errores, los principios generales en el diseño de la interfaz de usuario y la concepción del sistema de la ayuda y seguridad.

El Capítulo 4 muestra los artefactos correspondientes al flujo de trabajo de Implementación. Se presentan los diagramas de componentes y una propuesta final sobre el diagrama de despliegue, que permite el entendimiento de la distribución física del sistema con sus correspondientes nodos, y las relaciones entre estos.

En el Capítulo 5 se realiza el Estudio de Factibilidad del sistema propuesto. Se muestra la planificación del proyecto. Se especifican los beneficios tangibles e intangibles, cuantificándose los tangibles. Se analiza el costo requerido para la realización de la aplicación de acuerdo a los beneficios que la misma reporta. Por último se determina la factibilidad del sistema, de acuerdo a todos los datos analizados.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción.

En el presente capítulo se realiza un análisis de la tendencia actual a nivel internacional, en Cuba y en PROCYON de la Administración del personal, específicamente de la Gestión del Tiempo monitoreando el tráfico de Internet. Además se define la tecnología que se utiliza para darle solución al problema planteado anteriormente, se selecciona el lenguaje de programación, el servidor web, el gestor de Base de Datos, quedando definida la arquitectura del sistema y la metodología para el desarrollo del software.

En estos momentos, el libre mercado y la competencia tanto tecnológica como económica se incrementan a un ritmo acelerado, la capacidad competitiva de las empresas está, cada vez más, determinada por la definición e implantación de una estrategia adecuada, que permita, de forma clara, orientar a las organizaciones hacia la consecución de objetivos concretos. El valor de las organizaciones ya no reside tanto en los bienes tangibles sino en lo que se ha denominado capital intelectual o conocimiento. La Gestión de Recursos Humanos se ha vuelto un arma imprescindible para el buen funcionamiento de las empresas. Sobre todo en las empresas que se dedican al desarrollo de Software, donde el Conocimiento es su bien máspreciado. La Gestión del Tiempo contribuye al buen funcionamiento de la empresa, al tener conocimiento de en qué los empleados invierten su tiempo de trabajo.

1.2 Gestión de Recursos Humanos.

El término *dirección de recursos humanos* surgió a finales de los 70 en Estados Unidos. Fueron diversos factores los que contribuyeron a que las políticas de personal fueran más complejas y pasaran a integrarse en la estrategia empresarial y, posteriormente, dieran lugar a la aparición de la interpretación actual de los recursos humanos. Desde entonces, la gestión de los recursos humanos ha ido evolucionando y adaptándose a la realidad social y empresarial de cada momento. Así, del mero reclutamiento y administración de personal, basados en la gestión administrativa y burocrática, se ha ido avanzando hacia la conciliación estratégica de las necesidades económicas con las necesidades sociales, para llegar a la consideración de los recursos humanos como componente esencial para la calidad y competitividad de la empresa actual. (1)

1.3 Gestión de la Evaluación.

La Gestión de la Evaluación constituye un instrumento de gran importancia en las empresas actuales. La evaluación de rendimiento es útil para validar y refinar las actividades de la empresa, tal como la selección, capacitación y para brindar información a los empleados que desean mejorar su futuro rendimiento. La selección, capacitación y casi todas las prácticas de administración, pueden evaluarse tomando en cuenta los datos de rendimiento.

En la evaluación de desempeño el supervisor examina el rendimiento laboral de un empleado y comparte con éste el análisis de los resultados obtenidos.

Las fases principales que hay que abordar para conseguir evaluaciones eficaces son: determinar el tipo de datos que se va a utilizar así como quién va a realizar la evaluación, establecer una estrategia de evaluación, tener presentes, para poder evitarlas, las deficiencias habituales del proceso de evaluación, e implicar a los empleados en la toma de decisiones relativas a cambios futuros en el desempeño.

La información obtenida por medio de las evaluaciones de desempeño también pueden utilizarse para: desarrollar descripciones de puestos más correctos y planificar programas para la formación de los empleados ya contratados, otorgar aumentos u otros beneficios, establecer sesiones de asesoramiento laboral, y adoptar medidas disciplinarias o procedimientos de despido.

Los siguientes pasos permiten lograr evaluaciones útiles de desempeño:

- 1) Seleccionar el tipo de datos para evaluar el desempeño.
- 2) Determinar quién efectuará la evaluación.
- 3) Decidir sobre una filosofía de valuación.
- 4) Superar deficiencias de valuación.
- 5) Diseño de un instrumento de evaluación.
- 6) Retroalimentación de información a los empleados

Capítulo 1: Fundamentación Teórica

Para el fin que nos ocupa, se puede optar entre diversas técnicas de recolección de datos y de evaluación, o escalas de valoración. Lo importante es que el instrumento utilizado proporcione información significativa tanto a los empleados como a la dirección. (2)

1.4 Gestión del Tiempo.

La capacidad de gestionar bien el tiempo es una habilidad que las empresas valoran cada vez más en sus trabajadores, debido a que este factor mejora la productividad y la competitividad de la organización.

Si existe algo que se debe apreciar en nuestro trabajo es el tiempo. Apreciarlo, por su importancia y porque su incorrecta gestión, puede influir negativamente en la toma de decisiones, en el trabajo realizado, en abordar nuevos asuntos y en definitiva, en la marcha global de la empresa.

El tiempo es inflexible, pasa y no se detiene, aunque a veces se tenga la sensación de todo lo contrario. Es, en apariencia, una variable que no se puede modificar. No se puede, alargarlo, estirarlo, comprarlo o detenerlo. Sin embargo, se puede controlar. (3) (4)

1.4.1 Gestionar el tiempo de los empleados monitoreando el tráfico de Internet.

Internet se ha convertido en un medio masivo de comunicación. Desde una computadora conectada a un proveedor de servicios se pueden hacer compras, reservaciones, leer noticias, ver vídeos y oír música. En las empresas se espera que se use Internet como una herramienta de trabajo... pero no siempre es así. Se pierde tiempo en los centros laborales accediendo a sitios no relacionados con el contenido de trabajo. Aunque es cierto que siempre que un usuario visita un sitio en Internet esa información completa queda reflejada en múltiples lugares, que van desde su propia computadora hasta los servidores de los proveedores del servicio, también lo es el hecho de que es prácticamente imposible hacer un seguimiento efectivo de esa información. La situación anterior crea la sensación de impunidad para aquellos que hacen un mal uso de Internet. Se pudiera pensar que ese tiempo perdido es insignificante, a continuación se expondrán ejemplos que demuestran lo contrario.

En México el periódico El Universal Online informa que los empleados que tienen computadoras conectadas a internet afectan la productividad de las compañías mexicanas, porque utilizan entre 75 por

Capítulo 1: Fundamentación Teórica

ciento y 80 por ciento del tiempo de navegación para atender asuntos personales, como consulta del correo electrónico y Chat.

En España, según un estudio que ha realizado la firma Inology, los trabajadores españoles dedican más de 16 días al año a navegar por la Red durante la jornada laboral, una media de 34,6 minutos diarios. Según el estudio, durante algo más de 18 minutos, los trabajadores visitan páginas Web relacionadas con el ocio. Según recoge este estudio, este tipo dedicado a consultar Internet para aspectos no relacionados con el trabajo repercute en unos gastos para la empresa de más de 1.100 euros anuales por cada trabajador. (5)

1.4.1.1 Uso de Internet en Cuba.

Cuba no está ajena a esta problemática, aunque quizás no a los niveles de otros países.

En Cuba no existen estudios profundos sobre el uso que hacen los empleados del acceso a Internet en sus centros laborales. En el Instituto Central de Investigación Digital (ICID), como parte de la aplicación de la política de seguridad informática, se realizó un estudio sobre este tema. El ICID es un instituto de investigación científica dedicado fundamentalmente al desarrollo de equipos y software médico de alta tecnología y proyectos de automatización. La labor investigativa del ICID exige una utilización intensiva de Internet. De las 293 computadoras del ICID 119 tienen permiso para acceder a Internet, hay 182 usuarios con permisos de navegación. Durante este estudio se comprobó que de 3379 sitios distintos visitados en un trimestre sólo el 48,57% estaban relacionados directamente con el trabajo. Estos datos confirman las tendencias internacionales, por lo tanto se puede inferir que las pérdidas de tiempo sean muy similares, pudiéndose fijar en el orden de las 5 horas semanales por usuario. (5)

1.4.2 Servidores Proxy para monitorear los accesos a Internet.

Los servidores Proxy cada vez que un usuario accede a un sitio en Internet almacenan un nuevo registro en un fichero texto, a estos ficheros texto se les llama logs. Técnicamente el "log de visitas" es un archivo creado por el servidor donde se registran las acciones que los usuarios generan en la web. El formato de los logs varía dependiendo del servidor Web.

Una limitación de los logs es que el archivo en bruto presenta un formato bastante denso y su tamaño puede ser un problema a la hora de manejarlo (un log de solo 20.000 visitas puede ocupar 3 megas).

Capítulo 1: Fundamentación Teórica

Cada visita, cada HTML, cada imagen dentro de ese HTML generará una línea. Es esta "redundancia" de información lo que hace que el log sea bastante difícil de manejar en bruto.

1.4.3 Software que procesan la información de los ficheros logs.

A continuación se realiza un estudio de software desarrollados en Cuba que permiten transformar la información generada por los ficheros logs en información útil y entendible para quienes lo requieran.

1.4.3.1 Software existente en Cuba.

En Cuba se ha comercializado un software desarrollado por Segurmática con propósitos similares "AAInternet", aunque su concepción es diferente. En primer lugar es una aplicación Windows que debe ser instalada en la computadora en la que se va a utilizar, hay que copiar hacia la máquina donde esté instalada la aplicación todos los Logs generados por el servidor Proxy o instalar la aplicación donde estos se encuentren. Además no se almacenan los registros en base de datos para consultas posteriores ni se establecen Jerarquías para acceso a la información de tráfico. Una sola persona es la que maneja ese gran volumen de información. Pueden realizarse consultas y reportes a la información de los logs de Internet.

ISAWEB es otro de los software utilizados en Cuba desarrollado en el ICID, aborda los problemas de seguimiento al acceso a Internet mediante la lectura de logs, al ser una aplicación Web puede ser consultada por cualquier usuario desde cualquier Terminal de la red. Las bases de datos son centralizadas, y el proceso de inserción de datos es independiente a la aplicación principal. Lo más importante: se establece una jerarquía que permite que la labor de seguridad informática se descentralice entre todos los directivos, e incluso participe el trabajador. (5)

1.5 Software utilizados en PROCYON.

Se hace mención de los software utilizados actualmente en PROCYON relacionados con el campo de acción del problema definido en la introducción del trabajo. Se describe el servidor proxy usado para lograr un eficiente y racional uso de Internet, el sistema de control de versiones que facilita el trabajo en equipo, permitiendo a cada usuario hacer actualizaciones en cualquier versión de un determinado producto. Se menciona además el servidor utilizado en dicha institución.

Capítulo 1: Fundamentación Teórica

1.5.1 Proxy Squid.

Para lograr un uso racional y controlado de Internet en PROCYON se utiliza el Servidor Proxy Squid, software de libre distribución para realizar la tarea de un servidor Proxy con prestaciones muy profesionales, es un aplicativo de alto rendimiento que realiza las funciones de servidor Proxy-Cache. Como diferencia con respecto al software tradicional de cache, Squid procesa todas las peticiones en un solo proceso transparente y no bloqueante.

Squid es el servidor Proxy más popular y extendido entre los sistemas operativos basados sobre UNIX®. Es muy confiable, robusto y versátil. Al ser *software libre*, además de estar disponible el código fuente, está libre del pago de costosas licencias por uso o con restricción a un uso con determinado número de usuarios.

En la actualidad Squid es capaz de hacer proxy-cache de los protocolos HTTP, FTP, GOPHER, SSL y WAIS, pudiendo realizar listas de control de acceso (ACLs) y registro de logs completos. Squid se puede configurar de forma jerárquica para incrementar el ancho de banda salvado, utilizando Internet Cache Protocol. (6)

1.5.2 Sistema de Control de Versiones TortoiseSVN.

El Sistema de Control de Versiones utilizado en PROCYON es el TortoiseSVN el cual maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un *repositorio* central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. TortoiseSVN se integra perfectamente en el shell de Windows (por ejemplo, el explorador). Esto significa que puede seguir trabajando con las herramientas que ya conoce. Y que no tiene que cambiar a una aplicación diferente cada vez que necesite las funciones del control de versiones.

1.5.3 Windows Server 2003.

El Servidor utilizado en PROCYON es el Windows Server 2003, es la versión de Windows para servidores lanzada por Microsoft en el año 2003. Está basada en el núcleo de Windows XP, al que se le han añadido una serie de servicios, y se le han bloqueado algunas características (para mejorar el rendimiento, o simplemente porque no serán usadas). En términos generales, Windows Server 2003 es un Windows XP

Capítulo 1: Fundamentación Teórica

simplificado, no con menos funciones, sino que estas están deshabilitadas por defecto para obtener un mejor rendimiento y para centrar el uso de procesador en las características de servidor. (8)

1.6 Tecnología seleccionada para la solución del problema.

Teniendo en cuenta el problema existente y las características del entorno donde se aplicara la solución propuesta hacemos un estudio y análisis de la tecnología actual, concluyendo con la selección de la que utilizaremos.

1.6.1 Tecnología Java.

Java fue el lenguaje de programación seleccionado para la solución del problema. Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros. JavaScript, un lenguaje interpretado, comparte un nombre similar y una sintaxis similar, pero no está directamente relacionado con Java.

Java comenzó como un proyecto llamado "Green" y su objetivo inicial era crear un lenguaje que fuera capaz de ejecutarse en electrodomésticos que tuvieran microprocesadores pero se dieron cuenta que ese tipo de tecnología estaba aun muy lejos de poder existir. El proyecto dio como resultado un lenguaje muy parecido a C/C++ al cual le llamaron "Oak" (en referencia al roble que se encontraba en el exterior de las oficinas de Sun Microsystems) por James Gosling en junio de 1991 para usarse en un proyecto de receptor digital externo, pero descubrieron que ya existía un lenguaje con este nombre, luego alguien sugirió el nombre de Java (se cree que es por un tipo de café y otros piensan que son siglas) y fue ese nombre el que quedó.

La primera implementación pública fue Java 1.0 en 1995. Prometía "Escribir una vez, ejecutar en cualquier parte" ("*Write once, run anywhere*"), proporcionando ningún coste extra en el tiempo de ejecución en las plataformas populares. Era bastante seguro y su seguridad era configurable, permitiendo restringir el acceso a archivos o a una red. Los principales navegadores web pronto incorporaron la capacidad de

Capítulo 1: Fundamentación Teórica

ejecutar "applets" Java seguros dentro de páginas web. Java adquirió popularidad rápidamente. Con la llegada de "Java 2", las nuevas versiones tuvieron múltiples configuraciones pensadas para diferentes tipos de plataformas. Por ejemplo, J2EE era para aplicaciones de empresa y la versión reducida J2ME era para aplicaciones para móviles. J2SE era la designación para la Edición Estándar. En 2006, las nuevas versiones "J2" fueron renombradas a Java EE, Java ME y Java SE, respectivamente.

En 1997, Sun se dirigió al cuerpo de estándares ISO/IEC JTC1 y más tarde a Ecma International para formalizar Java, pero pronto se retiró del proceso. Java permanece como un estándar de facto propietario que está controlado a través del Java Community Process. Sun hace disponibles la mayoría de sus implementaciones Java sin cargo alguno, generando los ingresos con productos especializados como el Java Enterprise System.

El 13 de noviembre de 2006, Sun liberó partes de Java como software libre/de código abierto, bajo la GNU General Public License (GPL). La publicación del código fuente completo bajo la GPL se espera que ocurra en la primera mitad de 2007. (9)

1.6.2 Java Development Kit.

El Java Development Kit (JDK) es un compilador y conjunto de herramientas de desarrollo para la creación de programas independientes y applets java.

Un compilador es un programa que, a su vez, traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente. Usualmente el segundo lenguaje es código máquina, pero también puede ser simplemente texto. Este proceso de traducción se conoce como compilación.

La razón principal para querer usar un compilador es querer traducir un programa de un lenguaje de alto nivel (como Java), a otro lenguaje de nivel inferior (típicamente lenguaje máquina). De esta manera un programador puede diseñar un programa en un lenguaje mucho más cercano a como piensa un ser humano, para luego compilarlo a un programa más manejable por una computadora.

Los lenguajes de Alto Nivel se caracterizan por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de a la capacidad ejecutora de las máquinas. En los primeros lenguajes de alto nivel la limitación era que se orientaban a un área específica y sus instrucciones requerían de una sintaxis predefinida. Se clasifican como lenguajes procedimentales. Otra limitación de

Capítulo 1: Fundamentación Teórica

los lenguajes de alto nivel es que se requiere de ciertos conocimientos de programación para realizar las secuencias de instrucciones lógicas. Los lenguajes de muy alto nivel se crearon para que el usuario común pudiese solucionar tal problema de procesamiento de datos de una manera más fácil y rápida.

1.6.3 Máquina virtual Java.

La máquina virtual de Java (en inglés *Java Virtual Machine*, JVM) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable incluso como lenguaje de entrada para un microprocesador físico.

La gran ventaja de la máquina virtual java es aportar portabilidad al lenguaje de manera que desde SUN se han creado diferentes máquinas virtuales java para diferentes arquitecturas y así un programa .class escrito en un Windows puede ser interpretado en un entorno Linux. Tan solo es necesario disponer de dicha máquina virtual para dichos entornos.

La máquina virtual de Java puede estar implementada en software, hardware, una herramienta de desarrollo o un Web browser; lee y ejecuta código precompilado bytecode que es independiente de la plataforma multiplataforma. La JVM provee definiciones para un conjunto de instrucciones, un conjunto de registros, un formato para archivos de clases, la pila, un heap con recolector de basura y un área de memoria. Cualquier implementación de la JVM que sea aprobada por SUN debe ser capaz de ejecutar cualquier clase que cumpla con la especificación.

Existen varias versiones, en orden cronológico, de la máquina virtual de Java. En general la definición del Java bytecode no cambia significativamente entre versiones, y si lo hacen, los desarrolladores del lenguaje procuran que exista compatibilidad hacia atrás con los productos anteriores. (10)

1.6.4 Servidor Web Apache Tomcat.

El Servidor Web Apache Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat

Capítulo 1: Fundamentación Teórica

implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Es un servidor de aplicaciones con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que su uso de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. Es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software Licence*. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 6.x, que implementan las especificaciones de Servlet 2.4 y de JSP 2.0. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

1.6.5 Servidor de Base de Datos PostgradeSQL.

PostgreSQL es un sistema de gestión de bases de datos Objeto-Relacionales (ORDBMS) libre, liberado bajo la licencia BSD (Berkeley Software Distribution). Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebirdy MaxDB), así como sistemas propietarios como Oracle y SQLServer. El mismo ha sido desarrollado de varias formas desde 1977.

Presenta alta concurrencia. Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

En cuanto a sus funciones, posee bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientación

Capítulo 1: Fundamentación Teórica

a objetos o la programación funcional. Los disparadores (triggers en inglés) son funciones enlazadas a operaciones sobre los datos.

Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL (Tool Command Language) como lenguaje procedural embebido.

Usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL. Soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés).

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés). (11)

1.6.6 Definición de la Arquitectura Software.

A continuación se define la arquitectura de alto nivel, necesaria en el diseño del sistema propuesto, por tratarse de una aplicación Web. Esto implica subdividir la aplicación en componentes funcionales y particionar estos componentes en capas. El diseño de la arquitectura de alto nivel es neutral a las tecnologías utilizadas.

1.6.6.1 Arquitectura Multi-capa.

Una arquitectura multicapa particiona todo el sistema en distintas unidades funcionales: cliente, presentación, lógica-de-negocio, integración, y sistema de información empresarial (EIS). Esto asegura una división clara de responsabilidades y hace que el sistema sea más mantenible y extensible. Los sistemas con tres o más capas se han probado como más escalables y flexibles que un sistema cliente-servidor, en el que no existe la capa central de lógica-de-negocios. (13)(14)

Capítulo 1: Fundamentación Teórica

1.6.6.2 Tecnologías elegidas para cada capa.

La aplicación utiliza una arquitectura multi-capa no-distribuida. La figura anterior muestra el particionamiento de las capas de la aplicación y las tecnologías elegidas para cada capa

A continuación se especifican los patrones utilizados.

Capa de Presentación

Java Server Faces.

JSF es un marco de trabajo de componentes de interface de usuario del lado del servidor para aplicaciones Web basadas en Java. JSF contiene un API para representar componentes UI y manejar sus estados, manejar sus eventos, la validación del lado del servidor, y la conversión de datos, definir la navegación entre páginas, soportar internacionalización y accesibilidad; y proporcionar extensibilidad para todas estas características. También contienen dos librerías de etiquetas JSP (Java Server Pages) personalizadas para expresar componentes UI dentro de una página JSP y para conectar componentes a objetos del lado del servidor.

Capa de la Lógica-de-Negocio

Spring Framework

Spring es un framework de Java que facilita la creación de la aplicación. Diseñado en módulos, con funcionalidades específicas y consistentes con otros módulos, facilita el desarrollo de funcionalidades específicas y hace que la curva de aprendizaje sea favorable para el desarrollador.

A pesar de que no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al considerársele una alternativa y sustituto del modelo de Enterprise JavaBean. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar. Elimina la necesidad de usar distintos y variados tipos de ficheros de configuración, mejora la práctica de programación, permite el uso o no de Enterprise JavaBeans, realizando el mismo tipo de funciones sin ellos. Los Enterprise JavaBeans (también conocidos por sus siglas EJB) son una de las Interfaces de Programación de Aplicaciones que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems. Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor.

Capa de Integración

Persistencia de Objetos Java utilizando Hibernate

Capítulo 1: Fundamentación Teórica

Para la mayoría de las aplicaciones, almacenar y recuperar información implica alguna forma de interacción con una base de datos relacional. Esto ha representado un problema fundamental para los desarrolladores porque algunas veces el diseño de datos relacionales y los ejemplares orientados a objetos comparten estructuras de relaciones muy diferentes dentro de sus respectivos entornos.

La tarea de persistir objetos Java en una base de datos relacional actualmente está siendo facilitada por un gran número de herramientas que permiten a los desarrolladores dirigir motores de persistencia para convertir objetos Java a columnas/registros de una base de datos y viceversa. Esta tarea implica serializar objetos Java estructurados en forma de árbol a una base de datos relacional estructurada de forma tabular y viceversa. Esencial para este esfuerzo es la necesidad de mapear los objetos Java a columnas y registros de la base de datos de una manera optimizada en velocidad y eficiencia.

Hibernate es un marco de trabajo Java que proporciona mecanismos de mapeo objeto/relacional para definir cómo se almacenan, eliminan, actualizan y recuperan los objetos Java. Además, ofrece servicios de consulta y recuperación que pueden optimizar los esfuerzos de desarrollo dentro de entornos SQL y JDBC. Por último, Hibernate reduce el esfuerzo necesario para convertir hojas de resultados de la base de datos relacional en gráficos de objetos Java. (15)

En el epígrafe 1.7.3 se aborda con mayor profundidad aspectos relacionados con esta herramienta.

1.6.7 Proceso Unificado de Desarrollo de Software (RUP).

Cada día la producción de software busca adecuarse más a las necesidades del usuario, esto trae como consecuencia que aumente en tamaño y complejidad. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo. Se hace necesario definir la metodología de ingeniería del software que guiará el proceso de automatización, se ha escogido el Proceso Unificado de Desarrollo de Software (RUP).

Es un proceso basado en componentes, que utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en que está dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, los cuales se muestran (en un anexo que vamos a poner), que parten de los casos de uso; centrado en la arquitectura y es iterativo e incremental. (16)

Capítulo 1: Fundamentación Teórica

1.7 Herramientas utilizadas.

Se mencionan las herramientas utilizadas en el desarrollo del sistema propuesto. Se describen sus principales características, ventajas y desventajas de su uso.

1.7.1 Entorno integrado de desarrollo Eclipse.

El entorno integrado de desarrollo Eclipse es una plataforma de software de Código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar un Entorno integrado de desarrollo (del Inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se embarca como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no.

1.7.2 My Eclipse.

MyEclipse es un plugin que se le añade a Eclipse para proporcionarle la posibilidad de realizar desarrollos, distribuciones e integraciones de aplicaciones Java.

MyEclipse es un excelente plugin para Eclipse, entre sus muchas características se encuentra la capacidad de generar código Java mediante XDoclet. Al utilizar XDoclet se reduce el tiempo de desarrollo porque automatiza la generación de los descriptores de despliegue y otro código de soporte.

1.7.3 Hibernate.

Hibernate es una herramienta de tipo ORM (Object-Relational Mapping), como se había mencionado en el epígrafe anterior.

ORM es un sistema que permite almacenar objetos de aplicaciones Java en tablas de sistemas de bases de datos relacionales usando metadatos que describen la relación entre los objetos y la base de datos, y lo hace de una manera transparente y autónoma.

El funcionamiento de Hibernate es excelente. Ha sido probado en numerosas situaciones, es un proyecto con más de cinco años de antigüedad, con una constante evolución. Es Código Abierto, es gratis, no requiere de un pago de licencia, no hay limitación al número de conexiones, y su comunidad es inmensa y

Capítulo 1: Fundamentación Teórica

siempre dispuesta a dar ayuda a quien lo necesite. Su documentación es excelente. Probablemente sea el proyecto de código abierto mejor documentado que haya.

El Fichero de Configuración de Mapeo de Hibernate

Las aplicaciones Hibernate hacen uso de ficheros de mapeo que contienen metadatos que definen los mapeos objeto/relacional para las clases Java. Un fichero de mapeo tiene el sufijo `.hbm.xml`. Dentro de cada fichero de configuración, se mapean a tablas de la base de datos las clases que se van a persistir y las propiedades se definen con mapeos de campo/columna y claves primarias.

1.7.4 Rational Rose.

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software(UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto.

Se decidió utilizar el Rational Rose Enterprise Edition 2003, para respaldar la documentación, como modelador visual de la notación UML (Unified Modeling Language) para la confección de los diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades.

1.7.5 Diseño de Interfaz. Macromedia Dreamweaver.

Se utilizó el editor de desarrollo Web Macromedia Dreamweaver MX, para el diseño preliminar del prototipo de interfaz de usuario. Dreamweaver 8 es un software fácil de usar que permite crear páginas web profesionales. Las funciones de edición visual de Dreamweaver 8 permiten agregar rápidamente diseño y funcionalidad a las páginas, sin la necesidad de programar manualmente el código HTML. Se puede crear tablas, editar marcos, trabajar con capas, insertar comportamientos JavaScript, etc., de una forma muy sencilla y visual. Además incluye un software de cliente FTP completo, permitiendo entre otras cosas trabajar con mapas visuales de los sitios web, actualizando el sitio web en el servidor sin salir del programa. (17)

1.8 Conclusiones

En el presente capítulo se mostró un estudio sobre el tema de la Gestión de los Recursos Humanos, de la gestión de Tiempo y de la importancia de Evaluar el trabajo. Además se evidencia la necesidad de implementar un software que permita controlar Gestión de Tiempo de los estudiantes vinculados a PROCYON. Para desarrollar el sistema se hace uso de la tecnología para la programación de páginas dinámicas en el lenguaje Java y con soporte de base de datos en PostgreSQL. El proceso de desarrollo es RUP, el cual está basado en la orientación a objetos y el moldeamiento visual usando UML, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios. Mediante este estudio también se pudo tener una visión del estado del arte del tema, a nivel internacional, nacional y en la Universidad.

Capítulo 2. Características del Sistema.

2.1 Introducción

En el presente capítulo, primeramente se muestra el resultado del estudio preliminar. Se realiza un análisis del objeto de estudio: problema, ubicación y características de la información que se maneja. Se presenta además la propuesta del sistema, especificando los requerimientos funcionales y no funcionales del mismo. Se realiza la definición de los casos de uso y de los actores que intervienen en los mismos y se muestran los diagramas de casos de uso y el modelo de objetos o diagrama de clases del negocio.

2.2 Objeto de estudio.

Registro y control del tiempo, las evaluaciones, el uso de los recursos de la red, y proyectos en los que han trabajado los estudiantes vinculados al proyecto PROCYON en la Universidad de las Ciencias Informáticas.

2.3 Problema y Situación Problemática.

A continuación se hace un estudio de los objetivos estratégicos de la organización y los principales procesos que lo soportan.

2.3.1 Objetivos estratégicos de la organización y procesos de negocio que los soportan.

PROCYON es una entidad concebida para la elaboración y posterior comercialización de aplicaciones y plataformas de software para telecomunicaciones, así como la prestación de servicios en dicha área. En el proyecto se ha identificado la necesidad de emplear estudiantes con habilidades actuales o potenciales en el Área de Software para Telecomunicaciones para dar cumplimiento a sus planes de producción y venta, por esta razón fueron establecidas fórmulas de cooperación con la Facultad 2 de la Universidad de Ciencias Informáticas. La Institución se compromete a cooperar en la transferencia a los estudiantes los conocimientos que el trabajo profesional exige, combinando los conocimientos teóricos con los de contenido práctico.

Capítulo 2: Características del Sistema

La relación entre PROCYON y la facultad 2, por medio del convenio de colaboración, tiene por objeto la realización en PROCYON de trabajos de elaboración de software por parte de los alumnos matriculados en la Facultad 2 de la Universidad de las Ciencias Informáticas. Los proyectos de trabajo se presentarán por PROCYON a la Facultad 2, los cuales deberán corresponder a los niveles de experticia y disponibilidad de tiempo de los estudiantes, así como a las necesidades de producción. Los proyectos de trabajo se conciliarán con los representantes de la Facultad 2 que ésta designe para ello. Los planes de trabajo y su cumplimiento en ningún caso podrán perjudicar las respectivas actividades, formas de trabajo, normal funcionamiento y objetivos fundamentales de cada una de las Partes.

2.3.2 Flujo actual de los procesos involucrados en el campo de acción. Causas que originan la situación problemática.

En PROCYON hasta el momento se utiliza el software de gestión de tiempo Time Tracker, el mismo permite tener un cierto control sobre el tiempo que los estudiantes dedican al proyecto, pero hasta cierto punto ya que los datos de las actividades que desarrollan en un momento determinado son introducidos a la aplicación por el propio estudiante, lo que puede ocasionar que los datos registrados no sean del todo creíbles o cien por ciento válidos. Esta herramienta permite que el trabajador tenga más conciencia de la importancia de su planificación laboral, así como del cumplimiento en tiempo de todas sus tareas y responsabilidades.

PROCYON necesita de un sistema que integre la información de tiempos de trabajo planificados y ejecutados por los estudiantes vinculados a PROCYON, así como tener un registro de los conocimientos adquiridos y en los proyectos en los que el estudiante trabaja y ha trabajado, además de tener la información de cómo el estudiante utiliza el tiempo y los servicios de la red mientras está en PROCYON.

Actualmente se cuenta con un sistema de Gestión de Tiempo de los estudiantes, un sistema de control de versiones (SVN Tortoise), un Proxy (Squid) y Dominio (Windows 2003 Server), pero se hace necesario acceder a la información que nos brindan todos estos servicios de la red para tenerla en cuenta a la hora de evaluar el trabajo de los estudiantes y el uso que hacen del tiempo y de los servicios de la red de PROCYON. En el proyecto no se cuenta con una herramienta automatizada para la administración de las evaluaciones de los estudiantes vinculados al proyecto, tal y como se establece en el Convenio de colaboración firmado entre la Facultad 2 y PROCYON. No se tiene un registro integral de las actividades realizadas por el estudiante, en curso, estado de cumplimiento y evaluaciones de las mismas. Los datos

Capítulo 2: Características del Sistema

relacionados con los estudiantes se encuentran dispersos, lo que supone un gasto de tiempo innecesario y un mayor esfuerzo por parte del personal encargado de la toma de decisiones en un momento dado.

2.3.4 Objeto de Automatización.

Describiremos los principales procesos que serán objetos de automatización en el campo de acción.

2.4.1 Procesos que serán objeto de automatización.

A continuación se realiza una breve descripción de los procesos que serán objeto de automatización.

Proceso de Asignación de un nuevo proyecto.

Este es el proceso más importante que se realiza en PROCYON. El director de PROCYON asigna un nuevo proyecto al líder del proyecto, este analiza la nueva propuesta y verifica si tiene estudiantes disponibles de los que se encuentran ya vinculados a PROCYON, para trabajar en dicho proyecto, si los tiene le asigna la nueva tarea, sino solicita los estudiantes a la dirección de la facultad 2. El líder del proyecto solicita a la facultad una cantera de estudiantes, el Vicedecano de producción recibe la solicitud y selecciona los estudiantes. Después de seleccionados los estudiantes envía al líder del proyecto el listado de estudiantes seleccionados. El líder del proyecto recibe la propuesta y realiza un examen y una encuesta a los mismos. Luego selecciona los aprobados y envía un listado al vicedecano de producción de la facultad y al Director de PROCYON.

Proceso de Firma del Convenio de colaboración de la facultad 2 de la Universidad de las Ciencias Informáticas y PROCYON.

La firma de este documento se realiza después de que el estudiante es seleccionado por el líder del proyecto para pertenecer al proyecto. Este convenio tiene como objetivo la realización en PROCYON de trabajos de elaboración de software por parte de los alumnos matriculados en la Facultad 2, de la Universidad de las Ciencias Informáticas. Si el estudiante está de acuerdo con lo establecido en el mismo, procede a su firma y es considerado un trabajador más del proyecto, sino está de acuerdo no puede pertenecer al proyecto.

2.4.2 Sistemas automatizados que existen en la empresa para la Gestión del Tiempo.

A continuación se realiza unos análisis de los sistemas automatizados existentes en PROCYON, vinculados con el campo de acción del proyecto.

Capítulo 2: Características del Sistema

2.4.2.1 Time Tracker.

Actualmente en PROCYON, para la gestión del tiempo de los trabajadores, se utiliza el software WR Time Tracker v0.5, sistema de monitoreo de proyectos basado en la Web, que permite hacer el seguimiento de los tiempos de trabajo dedicados a un proyecto, para cada uno de los colaboradores. Es fácil y rápido de usar. El software permite crear cuentas de usuarios, organizarlos en grupos de trabajo, crear y aportar todo tipo de modificaciones a múltiples proyectos, hacer el seguimiento en horas de trabajo de la dedicación de cada uno al proyecto, generar informes de trabajo y facturas, incluso enviar mensajes de correo electrónico.

2.5 Información que se maneja.

En PROCYON actualmente el documento que se maneja es el Convenio de Colaboración entre la Facultad 2 de la Universidad de las Ciencias Informáticas y PROCYON. Dicho convenio está concebido para la realización de trabajos de proyectos por alumnos de pregrado universitario. El presente Convenio tiene por objeto la realización en PROCYON de trabajos de elaboración de software por parte de los alumnos matriculados en la Facultad 2 de la Universidad de las Ciencias Informáticas.

2.6 Propuesta de sistema.

Partiendo de lo antes mencionado daremos una propuesta del sistema a desarrollar.

2.6.1 Descripción general de la propuesta de sistema.

El sistema garantiza en todo momento, ofrecer a los usuarios autorizados una información detallada sobre los proyectos en los que ha trabajado el estudiante, la planificación del trabajo de proyecto, y actividades realizadas por el estudiante, a través de la integración del sistema con otros recursos de la red de PROCYON, como el sistema de control de versiones TortoiseSVN, y el Proxy Squid. Además permite realizar evaluaciones periódicas a los estudiantes según su comportamiento y resultados en el trabajo, quedando registradas en la base de datos del sistema. El sistema propuesto permite gestionar toda la información relacionada con los datos del convenio de forma rápida, permite además, a partir de un conjunto de reportes, obtener el conocimiento necesario que ayuda a una mejor toma de decisiones en un momento dado.

2.6.2 Análisis comparativo de otras soluciones existentes con la propuesta.

A continuación se hace un análisis comparativo de las soluciones existente con la nueva propuesta.

Capítulo 2: Características del Sistema

2.6.2.1 Time Tracker v0.5.

Actualmente en PROCYON se utiliza el software WR Time Tracker v0.5, que permite hacer el seguimiento de los tiempos de trabajo dedicados a un proyecto, para cada uno de los colaboradores. El software permite crear cuentas de usuarios, organizarlos en grupos de trabajo, crear y aportar todo tipo de modificaciones a múltiples proyectos, hacer el seguimiento en horas de trabajo de la dedicación de cada uno al proyecto, generar informes de trabajo y facturas, incluso enviar mensajes de correo electrónico.

El uso del mismo tiene como desventaja que la información es introducida por cada colaborador, la cual puede cierta o no. El sistema implementado es capaz a partir de una integración de servicios, hacer reportes reales del tiempo de trabajo de los estudiantes, de la actividad en la red y en el repositorio. Además permite la gestión de toda la información referente a los estudiantes, organizarlos por grupos de trabajos, mostrando de cada estudiante los proyectos en los que ha trabajado.

2.7 Modelo del negocio.

Se describen los procesos de negocio propuestos y las mejoras que propone el negocio actual indicando cómo se solucionarían los problemas que originaron la situación problemática.

2.7.1 Descripción general de los procesos de negocio.

El negocio se desarrolla entre la Facultad 2 de la Universidad de las Ciencias Informáticas y el proyecto PROCYON, el director de PROCYON asigna un nuevo proyecto al líder del proyecto este verifica si tiene estudiantes para trabajar en el proyecto, si no cuenta con los estudiantes solicita a la facultad 2 una cantera para la posible selección de los mismos, la facultad 2 analiza los estudiantes que reúnen las condiciones para pertenecer al proyecto y envía los seleccionados. PROCYON realiza una serie de pruebas y entrevistas para definir quienes son los candidatos, después de la selección envía a la facultad el listado de los estudiantes aceptados.

2.7.2 Definición de los actores y trabajadores del negocio.

Se definen los actores del negocio y trabajadores del negocio, argumentando por qué lo son.

Capítulo 2: Características del Sistema

Tabla 2.6.2.1: Actores del negocio y su justificación

Actores del negocio	Justificación
Estudiante	Es el encargado de firmar el convenio después de ser seleccionado.
Director de PROCYON	Es el encargado de asignar proyectos a los líderes de Proyecto.

Tabla 2.6.2.2: Trabajadores del negocio y su justificación

Trabajadores del negocio	Justificación
Vicedecano de Producción	Es el encargado de seleccionar la cantera de estudiantes de la facultad.
Líder de proyecto	Es el encargado de realizar la solicitud de los Estudiantes. Evalúa los estudiantes seleccionados por el vicedecano de producción, de donde se obtienen los estudiantes aprobados para pertenecer al proyecto.

2.7.3 Diagrama de Casos de Uso del Negocio.

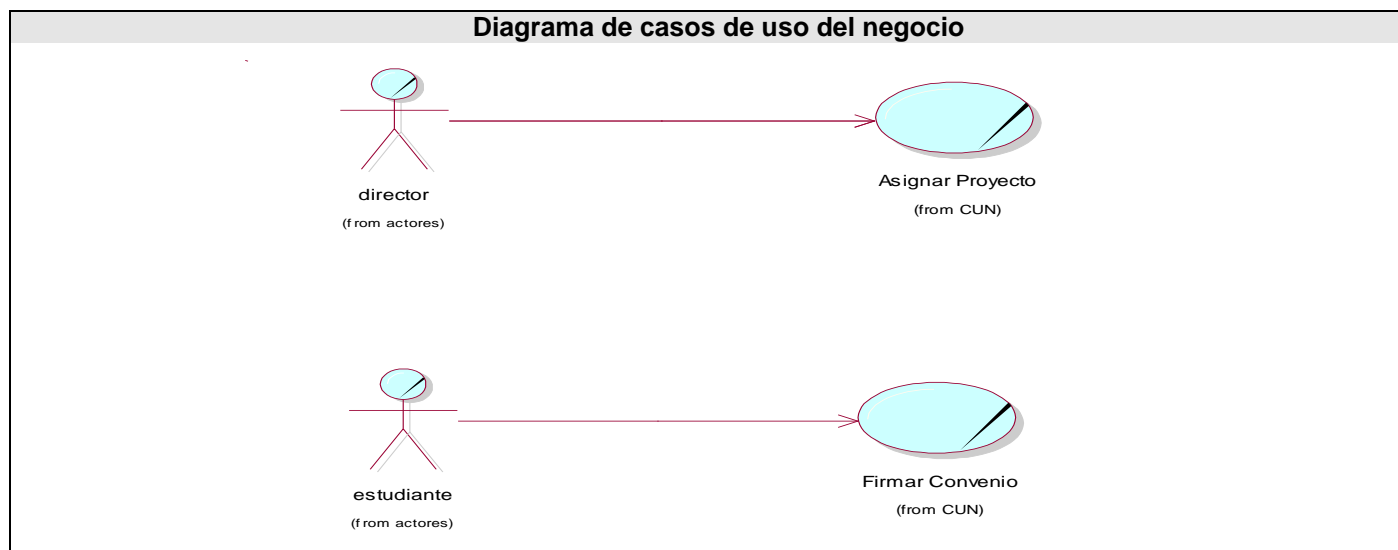


Fig. 2.1 Diagrama de Casos de Uso del Negocio.

Capítulo 2: Características del Sistema

2.7.4 Descripción textual del Caso de Uso de Negocio Asignar Proyecto. Ver [Anexo 1](#)

Se describe el caso de uso del negocio “Asignar Proyecto”, quedando especificados los actores y trabajadores que intervienen en el mismo.

2.7.5 Diagrama de Actividad del Caso de Uso de Negocio Asignar Proyecto. Ver [Anexo 2](#)

2.7.6 Descripción textual del Caso de Uso de Negocio Firmar Convenio. Ver [Anexo 3](#)

Se describe el caso de uso del negocio “Firmar Convenio”, quedando especificados los actores y trabajadores que intervienen en el mismo.

2.7.7 Diagrama de Actividad del Caso de Uso de Negocio Firmar Convenio. Ver [Anexo 4](#)

2.7.8 Modelo de objetos. Ver [Anexo 5](#)

2.8 Especificación de los requisitos de software.

A continuación se enuncian las dependencias y relaciones que presenta el sistema propuesto con otros sistemas existentes. Se especifican los requerimientos funcionales y no funcionales con los que el sistema debe cumplir. Se detallan los casos de uso del sistema y se presenta el diagrama de casos de uso del sistema.

2.8.1 Dependencias y relaciones con otros software.

El Sistema de Control del Tiempo de los estudiantes vinculados a PROCYON no presenta relación con otros software.

2.8.2 Requerimientos funcionales.

El sistema debe ser capaz de:

R 1- Autenticar usuario: Permitir a los usuarios acceder a la información que le corresponde.

R 1.1- Verificar usuario y contraseña con los usuarios del sistema.

R 1.2- Asignar privilegios.

R 2- Gestionar usuario.

R 2.1- Agregar usuario.

R 2.2- Eliminar usuario.

Capítulo 2: Características del Sistema

R 2.3- Mostrar Usuario.

R 2.4- Modificar usuario.

R 3- Controlar estudiantes vinculados a PROCYON.

R 3.1- Registrar nuevo estudiante.

R 3.2- Modificar datos del estudiante.

R 3.3- Eliminar estudiante.

R 3.4- Mostrar datos del estudiante.

R 4- Registrar actividad del estudiante en la red de PROCYON.

R4.1-Registrar acceso del estudiante al sistema de control de versiones de PROCYON (repositorio).

R 4.2-Mostrar la actividad del estudiante en el repositorio.

R 4.3- Registrar actividad del estudiante en Internet.

R4.4-Mostrar la actividad y tiempo que dedica el estudiante en la navegación en Internet.

R 5-Evaluar al estudiante según parámetros establecidos.

R 5.1-Registrar evaluación.

R 5.2-Modificar evaluación del estudiante.

R 5.3-Mostrar listado con las evaluaciones sistemáticas del estudiante.

R 6-Registrar el tiempo de trabajo del estudiante.

R 6.1- Registrar del estudiante las actividades y el tiempo destinado a las mismas.

R 7-Consultar tiempo de vinculación del estudiante a PROCYON.

R 7.1-Mostrar del estudiante el registro de las actividades y el tiempo que dedica a las mismas.

R 8-Registrar un nuevo proyecto productivo.

R 8.1-Registrar información del proyecto productivo.

R 8.2-Modificar datos del proyecto productivo.

R 8.3-Eliminar proyecto no productivo.

R 8.4-Mostrar los estudiantes que no están vinculados a proyectos productivo.

R 8.6-Mostrar datos de proyectos productivos.

2.8.3 Requerimientos no Funcionales.

Se especifican las propiedades o cualidades con que el sistema debe cumplir.

Capítulo 2: Características del Sistema

2.8.3.1 Requerimientos de Software.

El sistema se desarrollara con tecnología para el desarrollo de aplicaciones Web: Java. Se trabajará en el Entorno integrado de desarrollo Eclipse. La comunicación de las terminales clientes con el servidor será a través de cables UTP, a una velocidad constante de 1 GB. Se utilizará un servidor Web Apache Tomcat. El sistema utilizará gestor de Base PostgreSQL.

2.8.3.2 Requerimientos de Hardware.

Se debe disponer de un servidor que tenga como mínimo 512 MB de RAM, debido a que con esta memoria se permite que el servidor se desempeñe con un mejor rendimiento y eficiencia ante la demanda creciente, procesador Pentium IV en adelante.

2.8.3.3 Requerimientos de Portabilidad.

Al sistema se debe acceder desde cualquier plataforma, o sea debe garantizarse que sea multiplataforma.

2.8.3.4 Requerimientos de Apariencia o interfaz externa.

El sistema debe ser fácil de usar y su uso debe ser una ayuda al trabajo de los usuarios. La interfaz debe ser sencilla, sin grandes imágenes, manteniendo una misma línea de principio a fin.

2.8.3.5 Requerimientos de Usabilidad.

El sistema debe tener facilidad de uso para las personas que interactúan con el mismo.

2.8.3.6 Requerimientos de Rendimiento.

En general el sistema debe ser rápido y responder en un tiempo adecuado como para clasificarlo como interactivo.

2.8.3.7 Requerimientos de Seguridad y Privacidad.

Confidencialidad: La información manejada por el sistema estará protegida de acceso no autorizado y divulgación.

Integridad: la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.

Capítulo 2: Características del Sistema

Disponibilidad: a los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

2.9 Modelo de Casos de Uso del Sistema.

Se definen los casos de uso del sistema, los actores que interviene en los mismos y las descripciones de los casos de uso del sistema.

2.9.1 Definición de los actores del sistema a automatizar.

En este epígrafe se definen los actores del sistema y se justifica por qué lo son.

Tabla 2.7.1: Actores del sistema y su justificación

Actores del Sistema	Justificación
Usuario	Es un usuario que generaliza el rol de autenticación al sistema.
Administrador	Es el encargado de registrar un nuevo estudiante una vez que este es aceptado como trabajador de PROCYON. Además es el responsable de registrar el acceso de los estudiantes al sistema de control de versiones y de mantener un registro de las actividades que realiza el estudiante en la red de PROCYON.
Líder de Proyecto	Es el responsable de obtener la evaluación sistemática del estudiante, según su desempeño dentro de PROCYON. También es el encargado de administrar y actualizar la información referente al proyecto y asignar estudiantes al proyecto. Accede a la información correspondiente a la planificación que realiza el estudiante de las actividades y al registro del tiempo de vinculación a PROCYON del estudiante. Además debe registrar y actualizar la información referente a los proyectos en que ha trabajado el estudiante en PROCYON.
Estudiante	Es el responsable de la planificación del tiempo destinado al trabajo en PROCYON, así como de las actividades que realiza en el proyecto.

Capítulo 2: Características del Sistema

2.9.2 Diagrama de Casos de Uso del Sistema a automatizar.

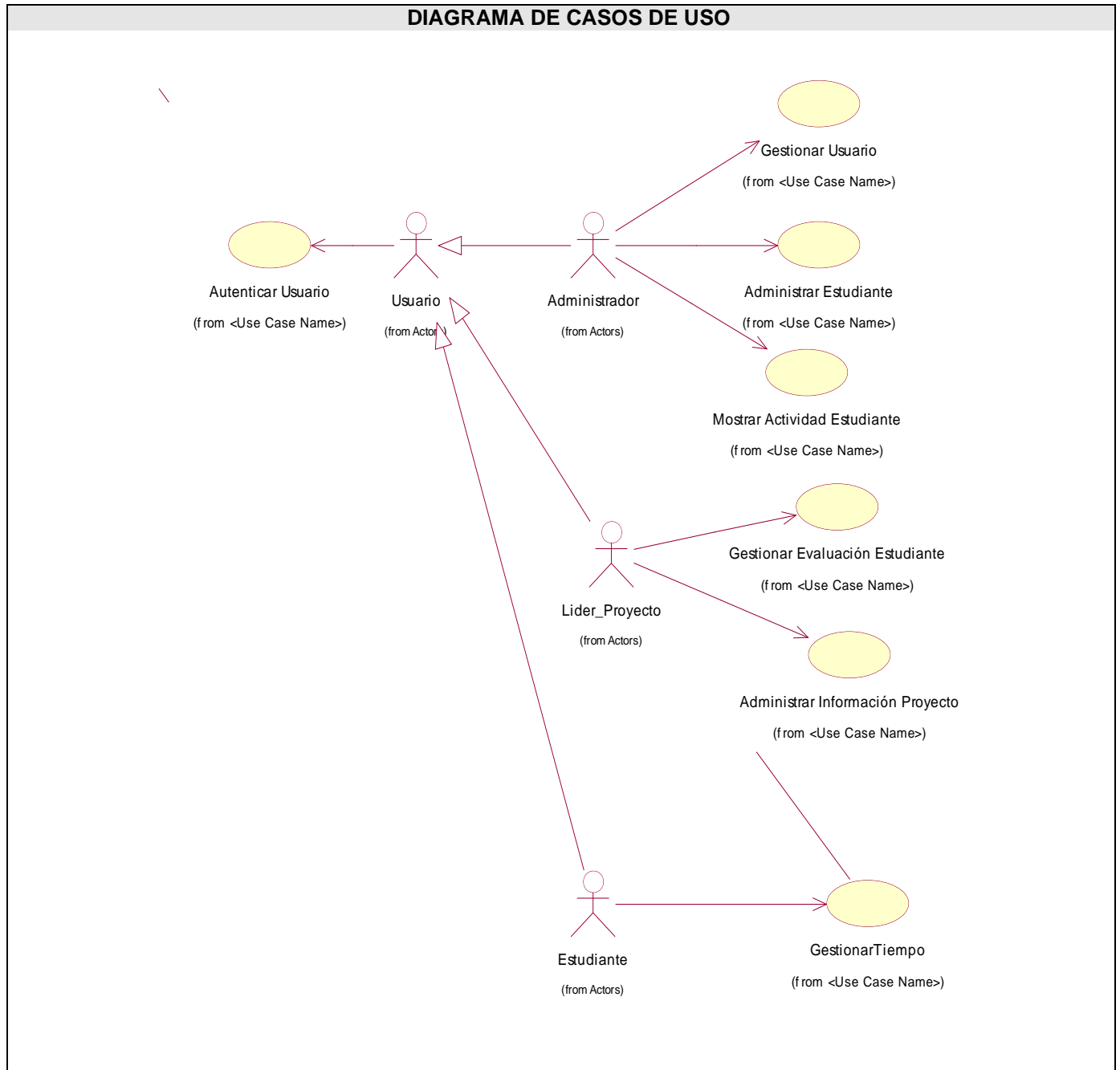


Fig. 2.2 Diagrama de Casos de Uso del Sistema a automatizar

Capítulo 2: Características del Sistema

2.9.3 Descripción de los Casos de uso del Sistema. Ver [Anexo 6](#).

2.10 Conclusiones.

En el presente capítulo se mostraron los resultados de la etapa de la captación de los requisitos y las características del sistema. Se definieron los actores Administrador, Líder de Proyecto, Estudiante, de forma que quedó claramente especificado qué agentes externos interactúan con el sistema.

Se definieron también los casos de uso a través de los cuales se logró una definición completa de la funcionalidad del sistema mediante la descripción de los procesos que automatizará, quedando definidos 7 casos de uso.

Capítulo 3. Análisis y Diseño del Sistema.

3.1 Introducción.

En el presente capítulo se presentan los resultados de la etapa de análisis y diseño del sistema, guiándose por la metodología RUP. Se realiza la definición de los diagramas de clases de análisis y diseño respectivamente, así como los diagramas de interacción del diseño. Se define el Modelo de Datos, quedando especificados el diagrama Entidad Relación de la BD. Finalmente se explica el tratamiento de errores, los principios generales en el diseño de la interfaz de usuario y la concepción del sistema de la ayuda y seguridad.

3.2 Modelo de Análisis.

El modelo de análisis contiene los resultados del análisis de los CU, instancias del artefacto: clases del análisis estas centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos.

A continuación se presentan los diagramas de clases del análisis, que estos están compuestos por la clases interfaces que modelan la interacción entre el sistema y sus actores, la clases controladoras que coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso y la clase entidad que modelan información que posee larga vida y que es a menudo persistente.

Capítulo 3: Análisis y Diseño del Sistema

3.2.1 Diagrama de Clases de Análisis. CU “Autenticar Usuario”.

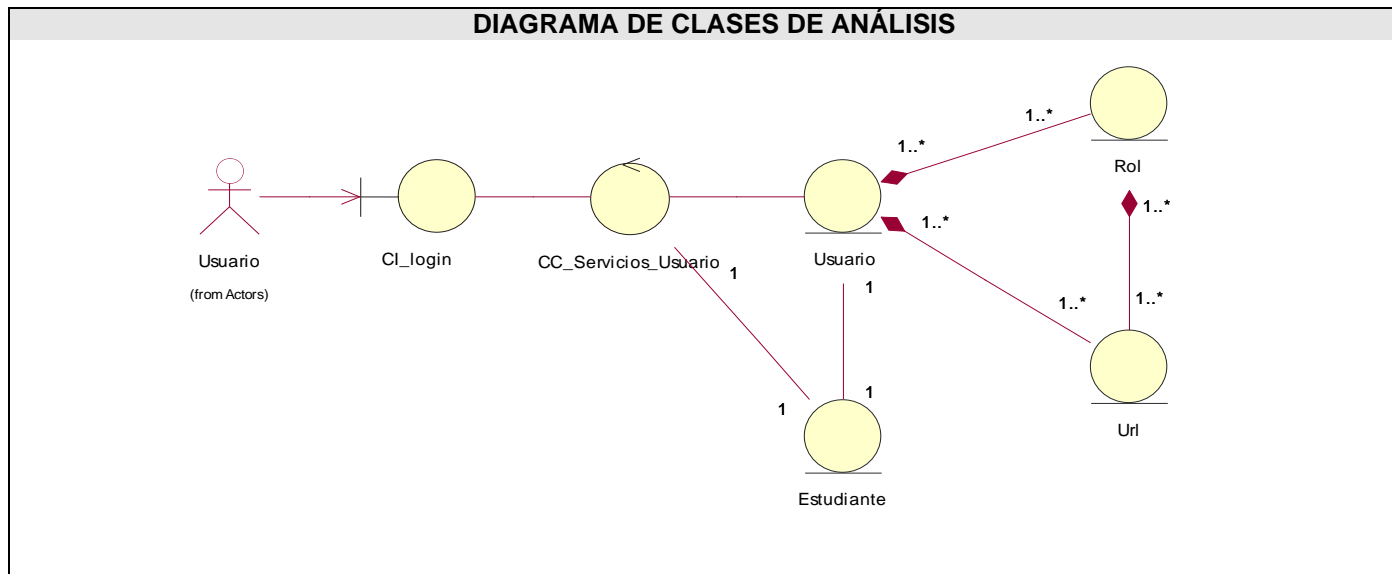


Fig. 3.1 Diagrama CA “Autenticar Usuario”

3.2.2 Diagrama de Clases de Análisis. CU “Gestionar Usuario”.

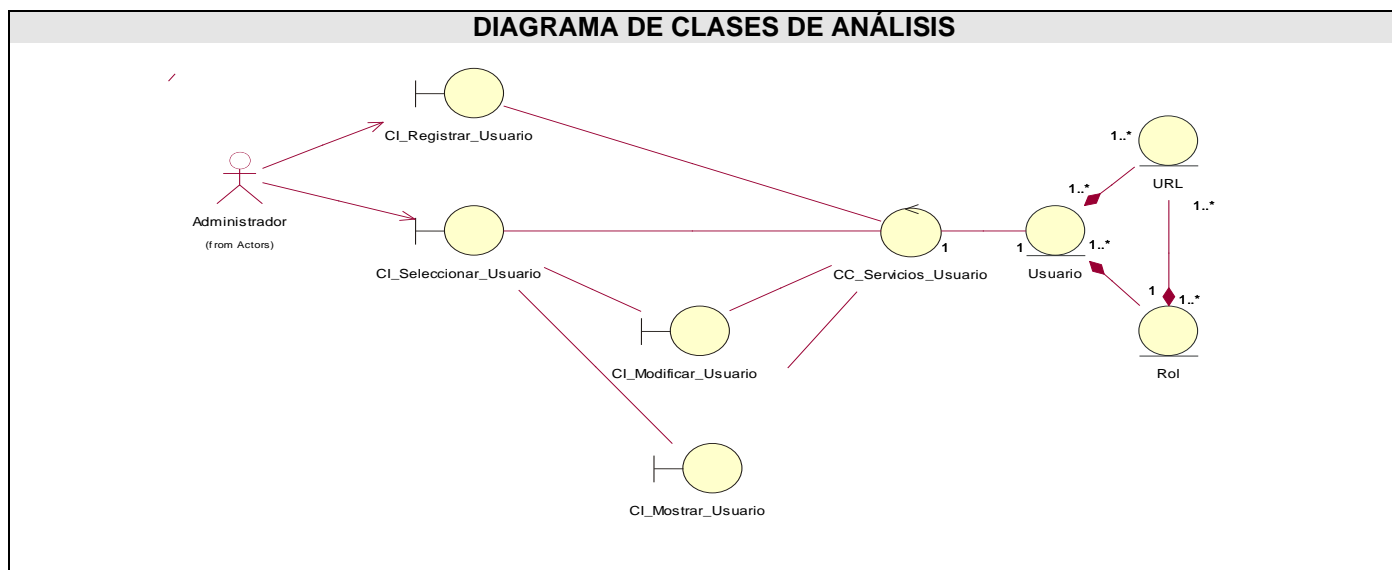


Fig. 3.2 Diagrama CA “Gestionar Usuario”

3.2.3 Diagrama de Clases de Análisis. CU "Administrar Estudiante".

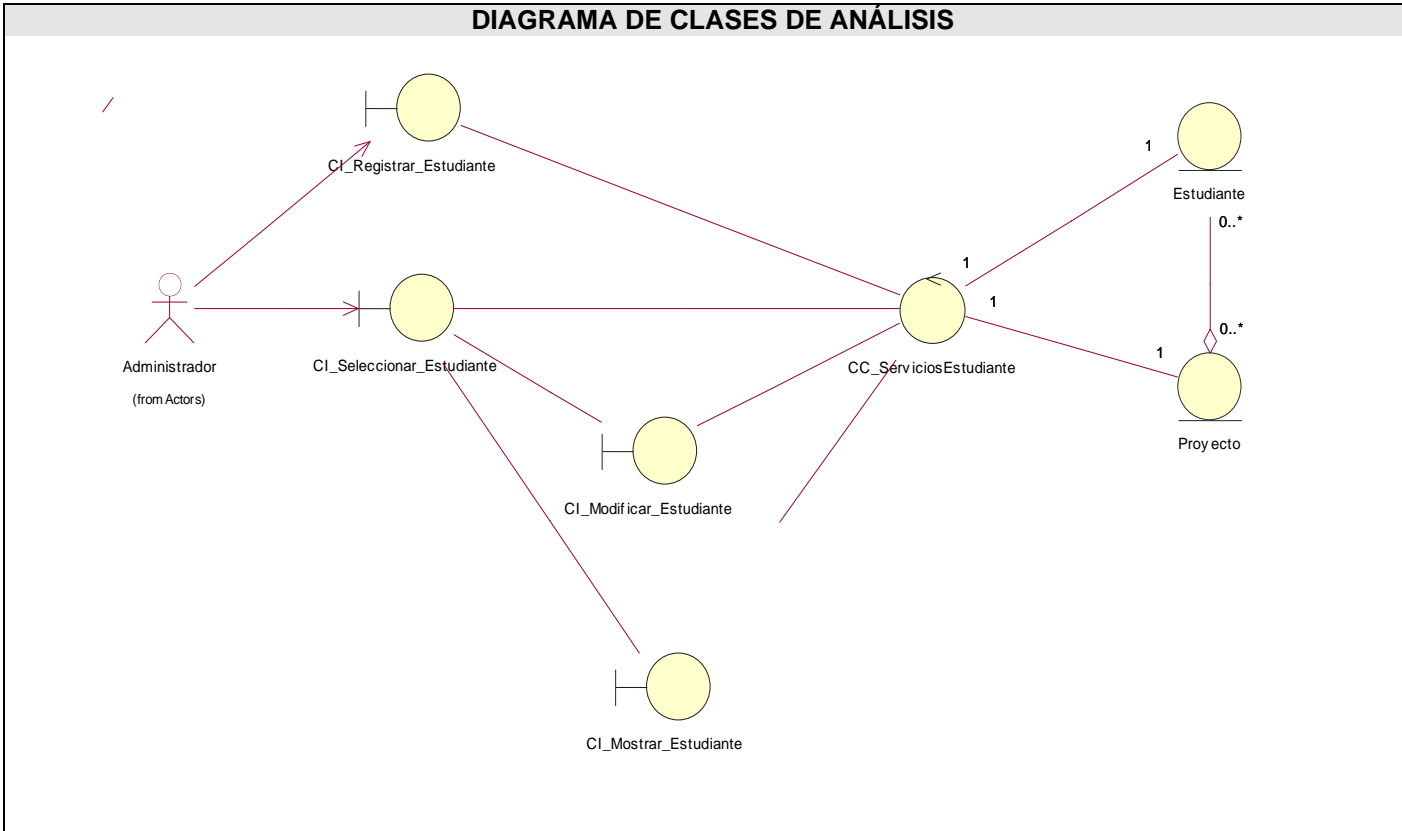


Fig. 3.3 Diagrama CA "Administrar Estudiante"

Capítulo 3: Análisis y Diseño del Sistema

3.2.5 Diagrama de Clases de Análisis. CU “Mostrar Actividad Estudiante”.

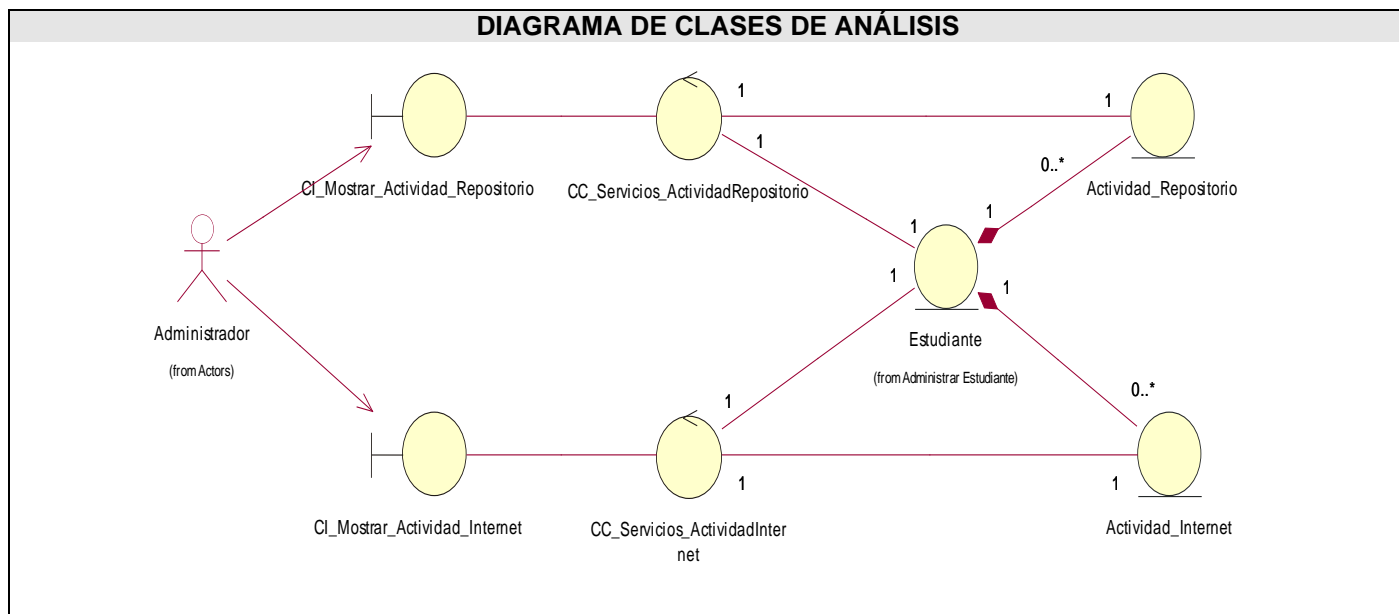


Fig. 3.5 Diagrama CA “Mostrar Actividad Estudiante”

3.2.6 Diagrama de Clases de Análisis. CU “Gestionar Tiempo”.

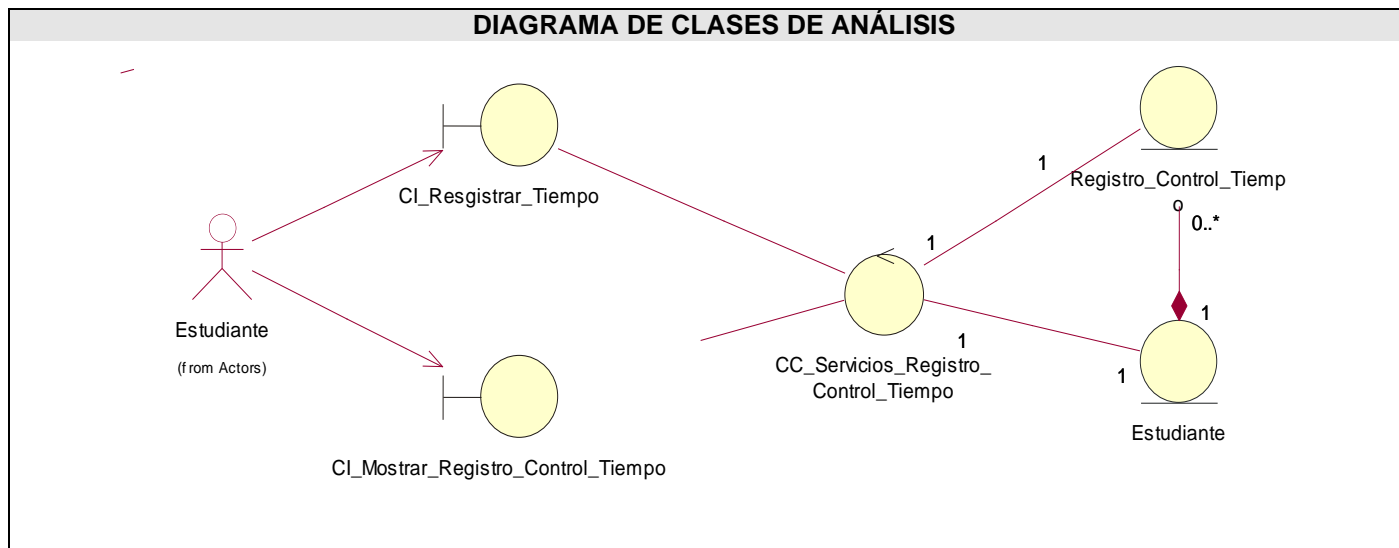


Fig. 3.6 Diagrama CA “Gestionar Tiempo”

3.2.4 Diagrama de Clases de Análisis. CU "Gestionar Evaluación Estudiante".

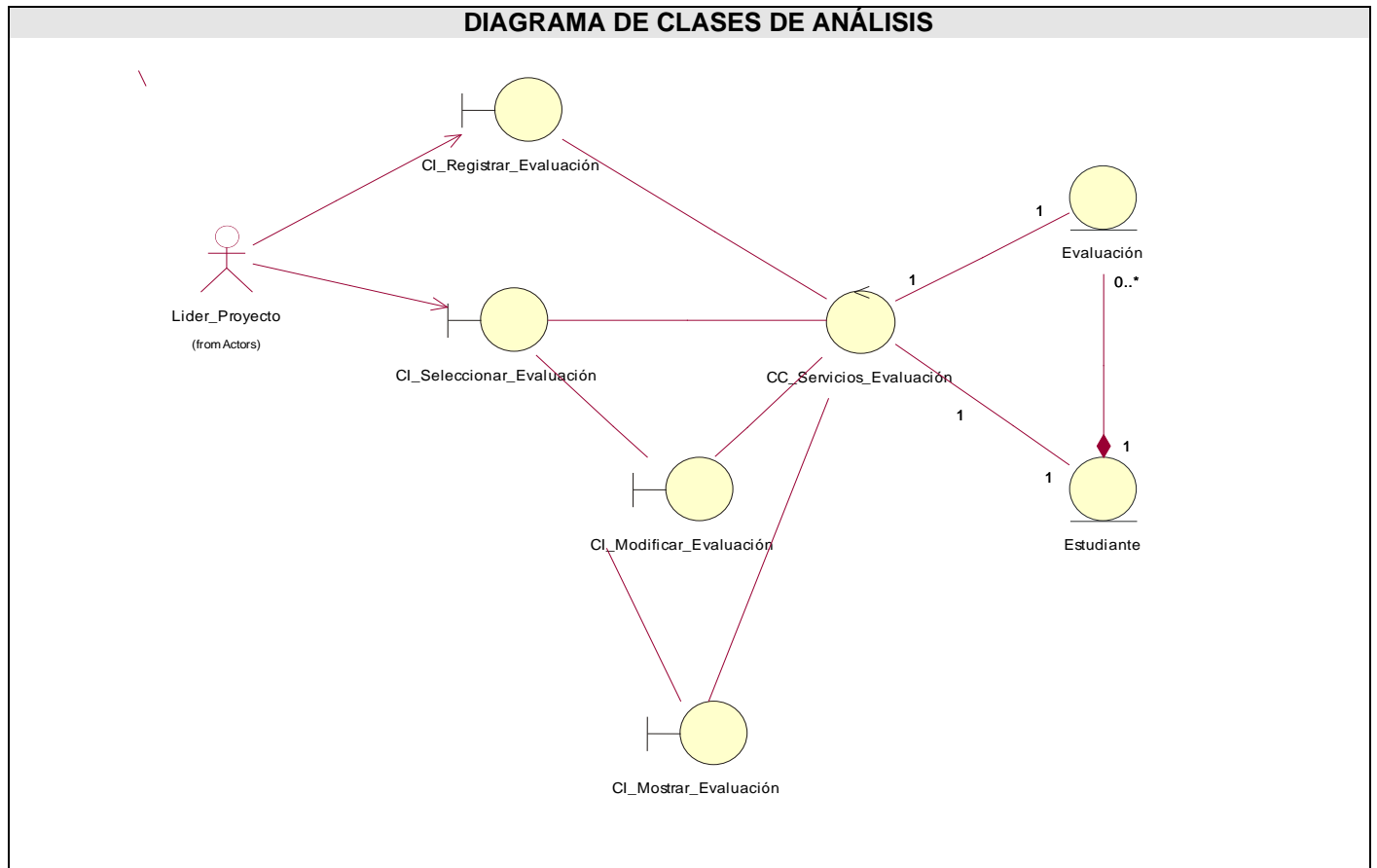


Fig. 3.4 Diagrama CA "Gestionar Evaluación Estudiante"

3.2.7 Diagrama de Clases de Análisis. CU “Administrar Información Proyecto”.

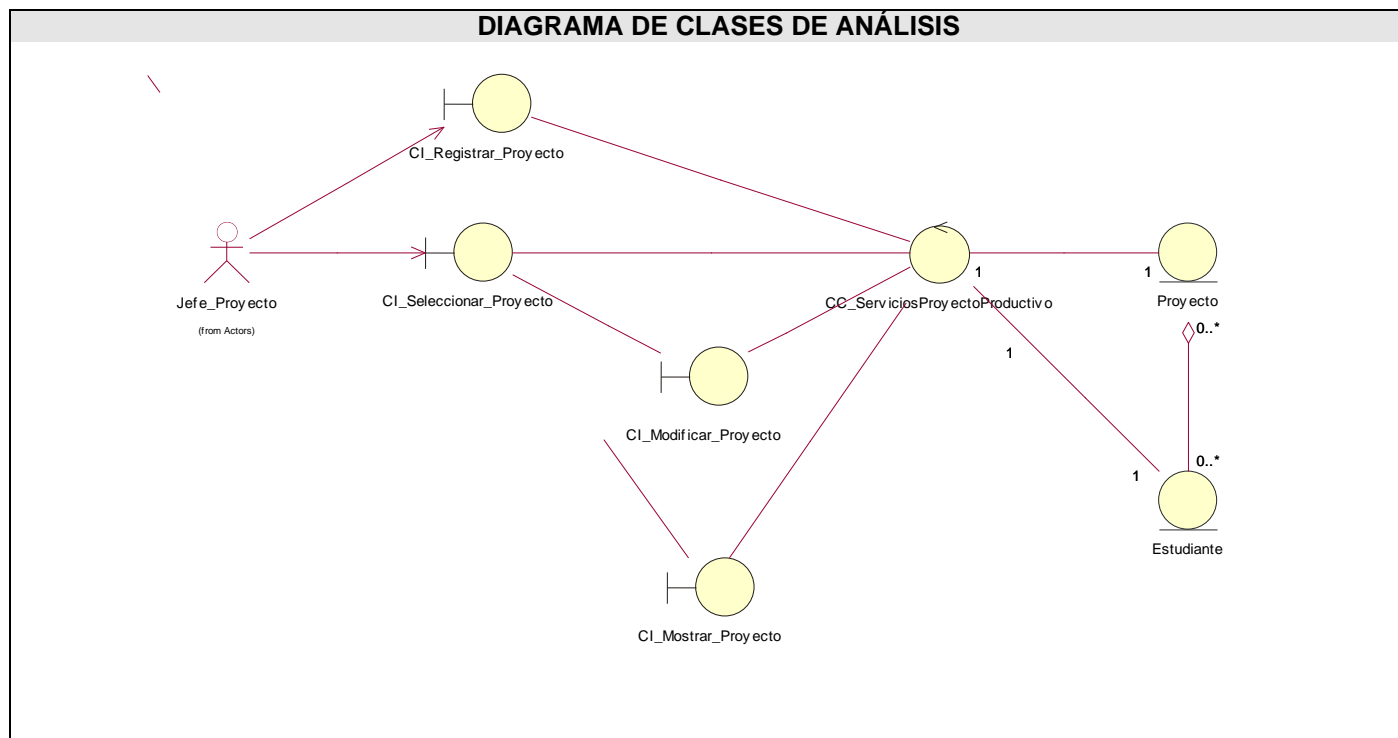


Fig. 3.7 Diagrama CA “Administrar información Proyecto”

3.3 Modelo de Diseño.

El modelo de diseño es una abstracción de la implementación del sistema. Es usado para concebir un documento del diseño del sistema de SW. Es abarcador, compuesto por artefactos que engloban todas las clases del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos.

A continuación se representan los diagramas de interacción (Secuencia) por cada realización de casos de uso y el diagrama de clases Web.

3.3.1 Diagramas de Interacción (Secuencia).

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Consta de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

Los diagramas de secuencia destaca la ordenación temporal de los mensajes. Se forma colocando en primer lugar los objetos que participan en la interacción en la parte superior del diagrama, a lo largo del eje X. Normalmente, se coloca a la izquierda el objeto que inicia la interacción, y los objetos subordinados a la

Capítulo 3: Análisis y Diseño del Sistema

derecha. A continuación, se colocan los mensajes que estos objetos envían y reciben a lo largo del eje Y, en orden de sucesión en el tiempo, desde arriba hasta abajo. Esto ofrece al lector una señal visual clara del flujo de control a lo largo del tiempo. **Ver Anexo 8.**

3.3.3 Diagramas de Clases Diseño Web.

3.3.3.1 Diagramas de Clases de Diseño Web. CU Autenticar Usuario.

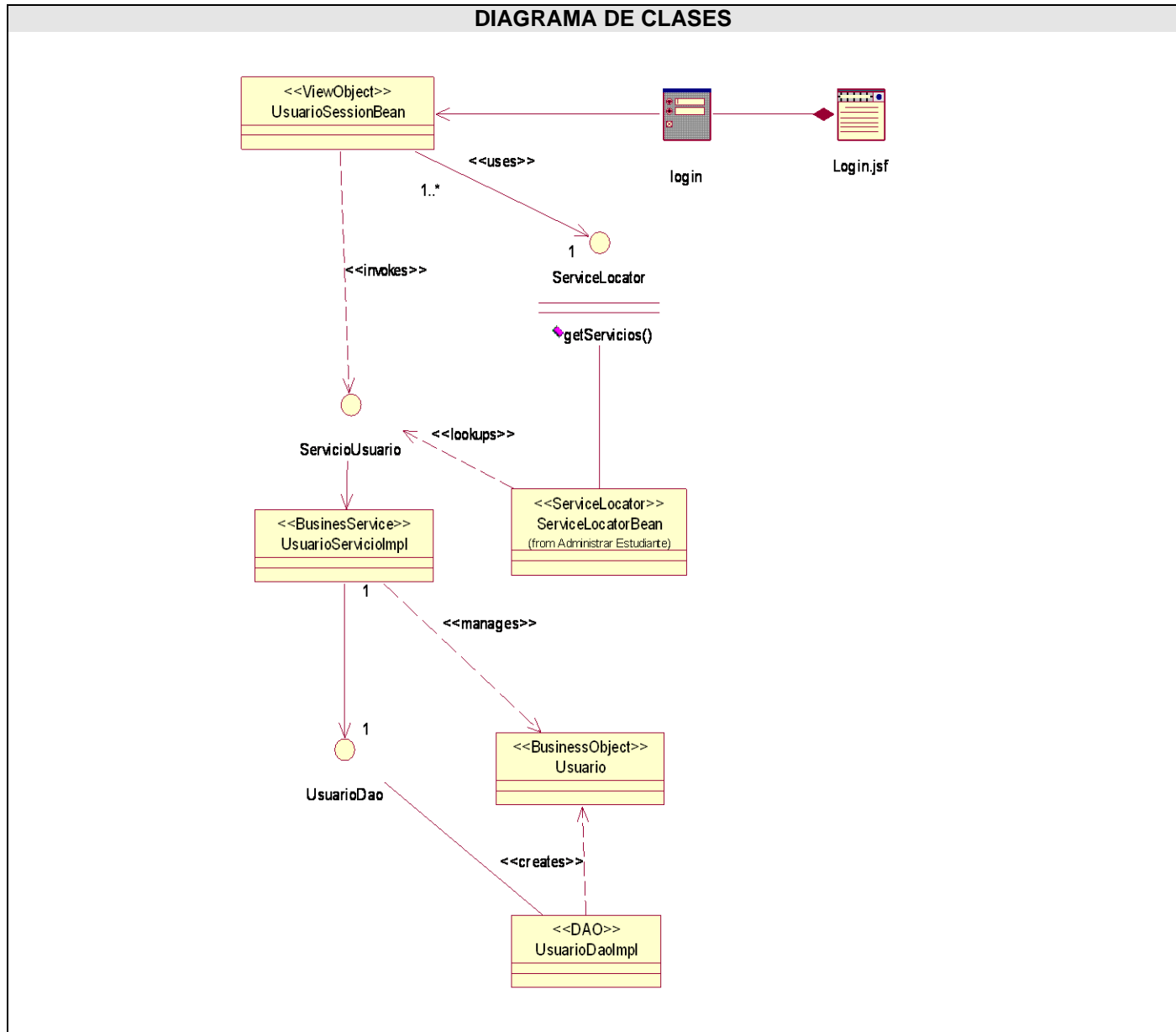


Fig. 3.24 Diagrama de CDW. CU "Administrar información Proyecto"

Capítulo 3: Análisis y Diseño del Sistema

3.2.3.2 Diagramas de Clases de Diseño Web. CU Administrar Estudiante.

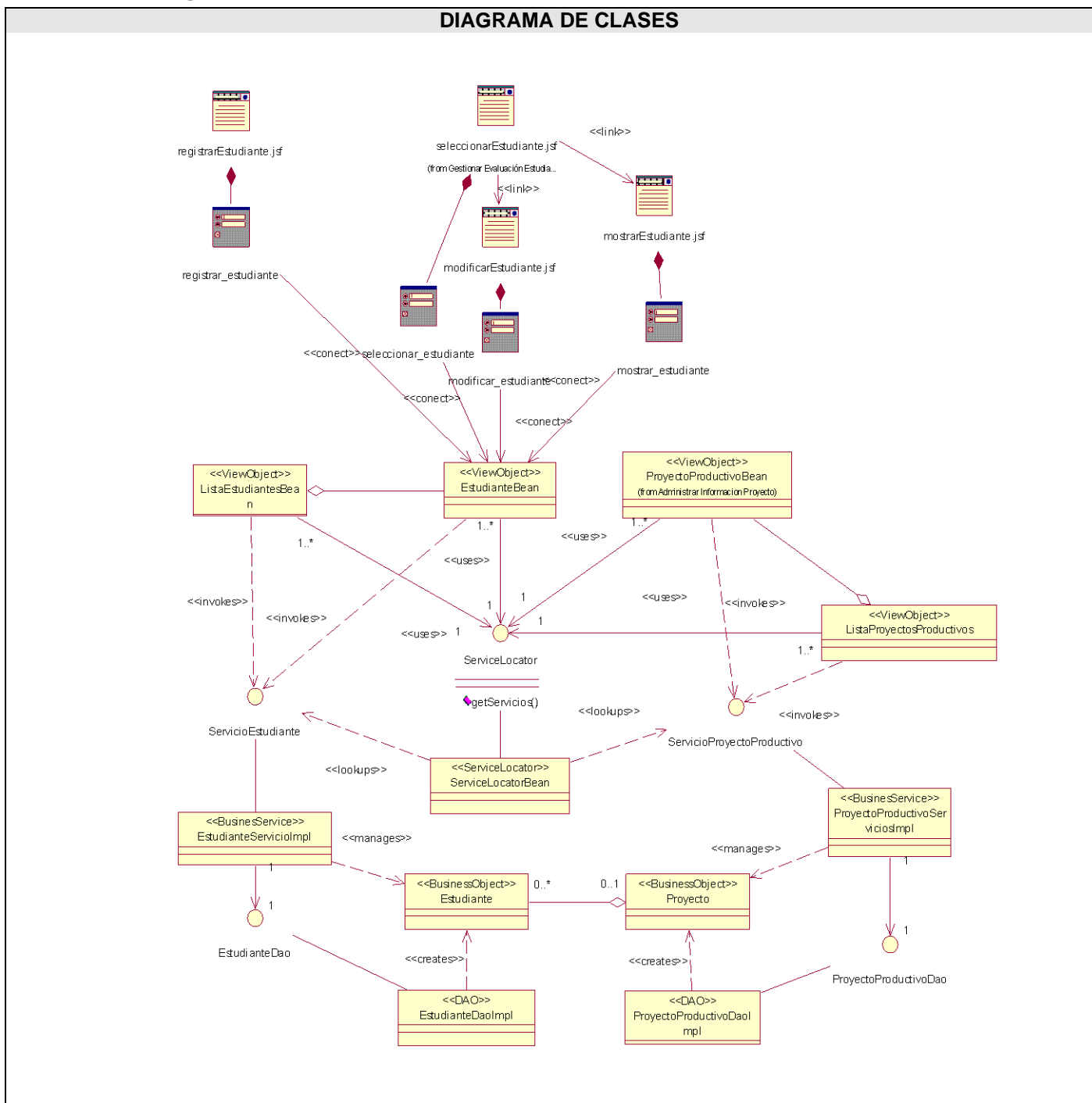


Fig. 3.25 Diagrama de CDW. CU "Administrar Estudiante"

3.2.3.3 Diagramas de Clases de Diseño Web. CU Gestionar Evaluación Estudiante.

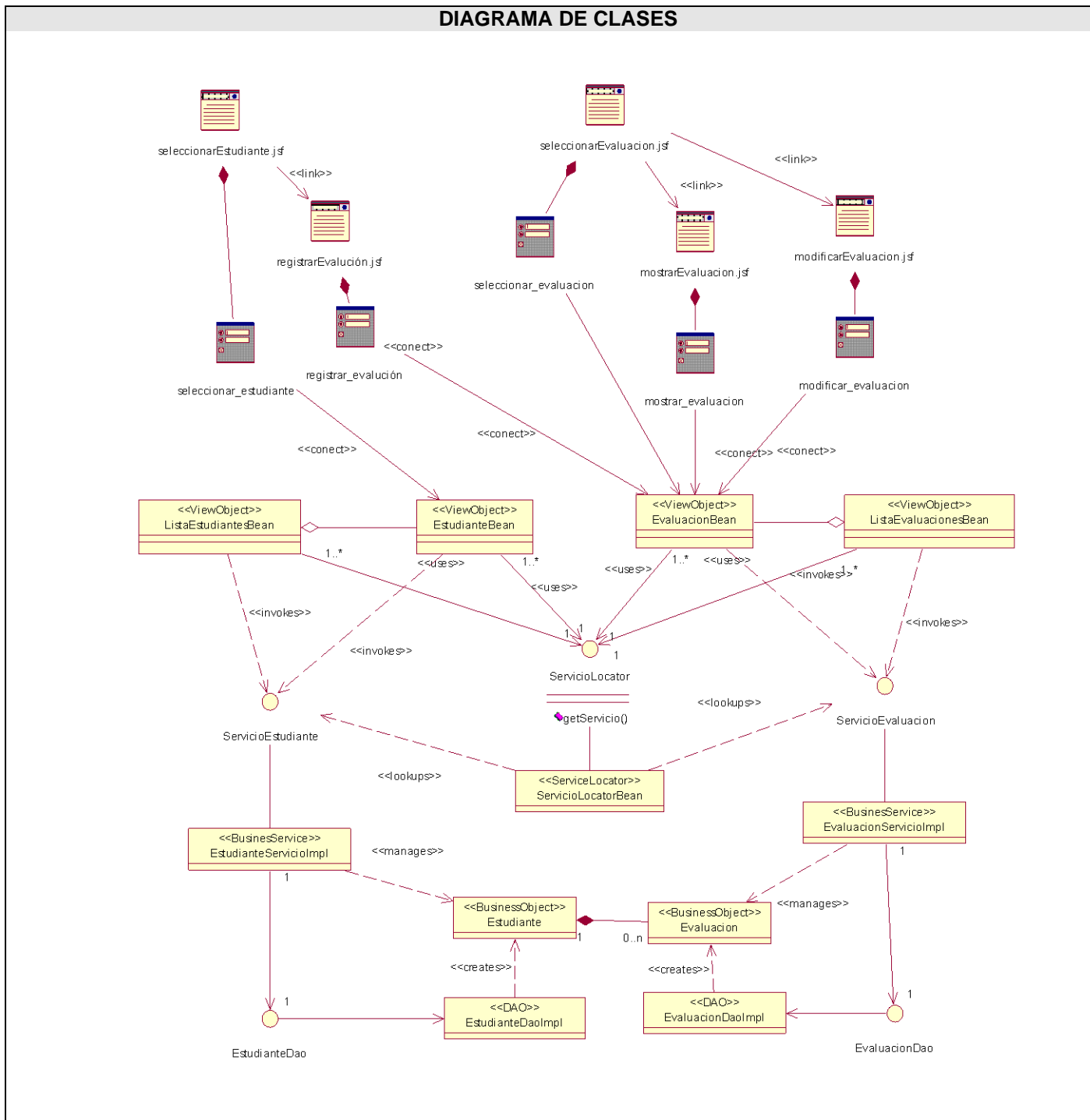


Fig. 3.26 Diagrama de CDW. CU "Gestionar Evaluación Estudiante"

3.2.3.4 Diagramas de Clases de Diseño. CU Mostrar Actividad Estudiante.

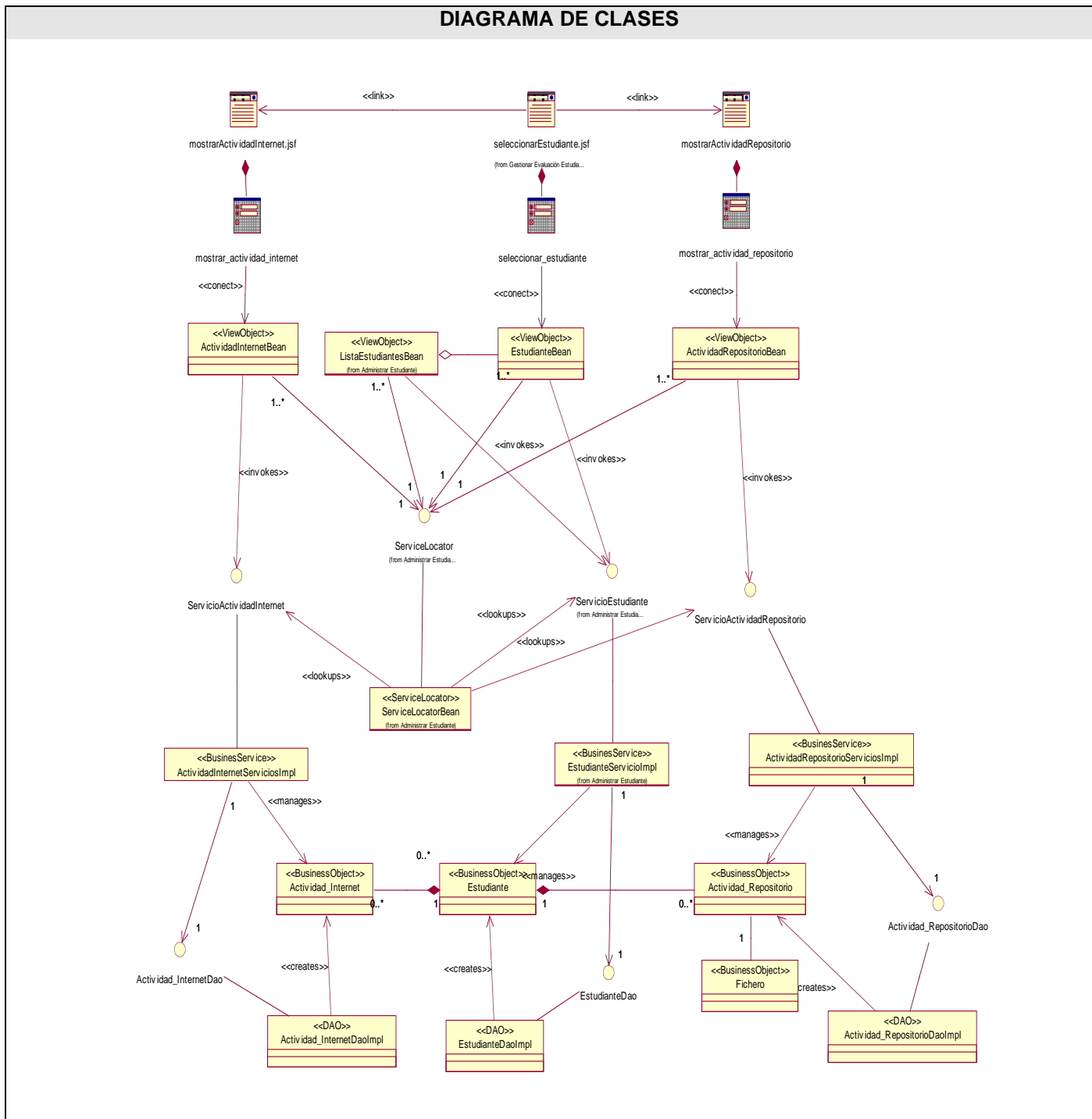


Fig. 3.27 Diagrama de CDW. CU "Mostrar Actividad Estudiante"

3.2.3.5 Diagramas de Clases de Diseño. CU Gestionar Tiempo.

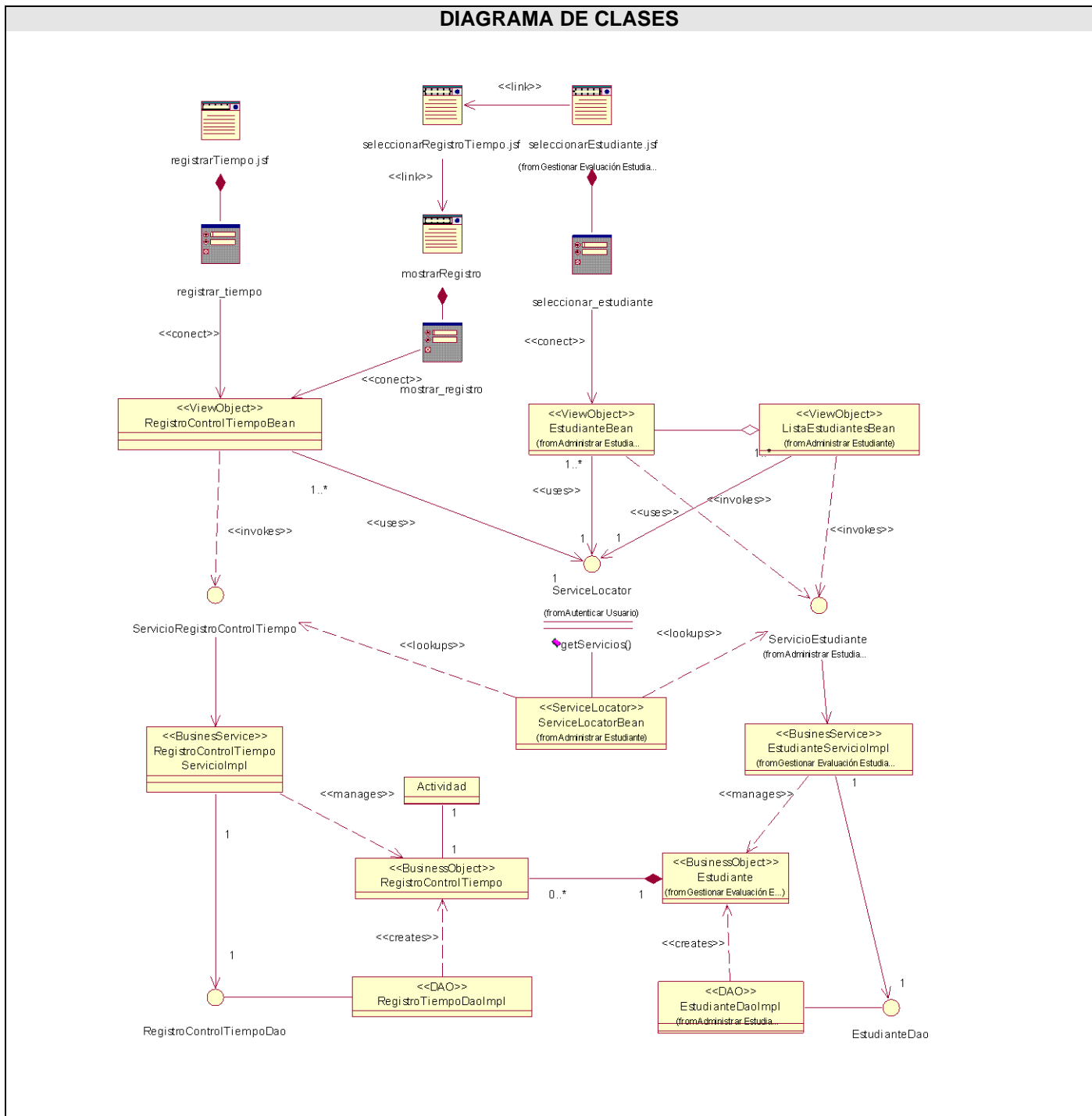


Fig. 3.28 Diagrama de CDW. CU "Gestionar Tiempo"

Capítulo 3: Análisis y Diseño del Sistema

3.2.3.6 Diagramas de Clases de Diseño. CU Administrar Información Proyecto.

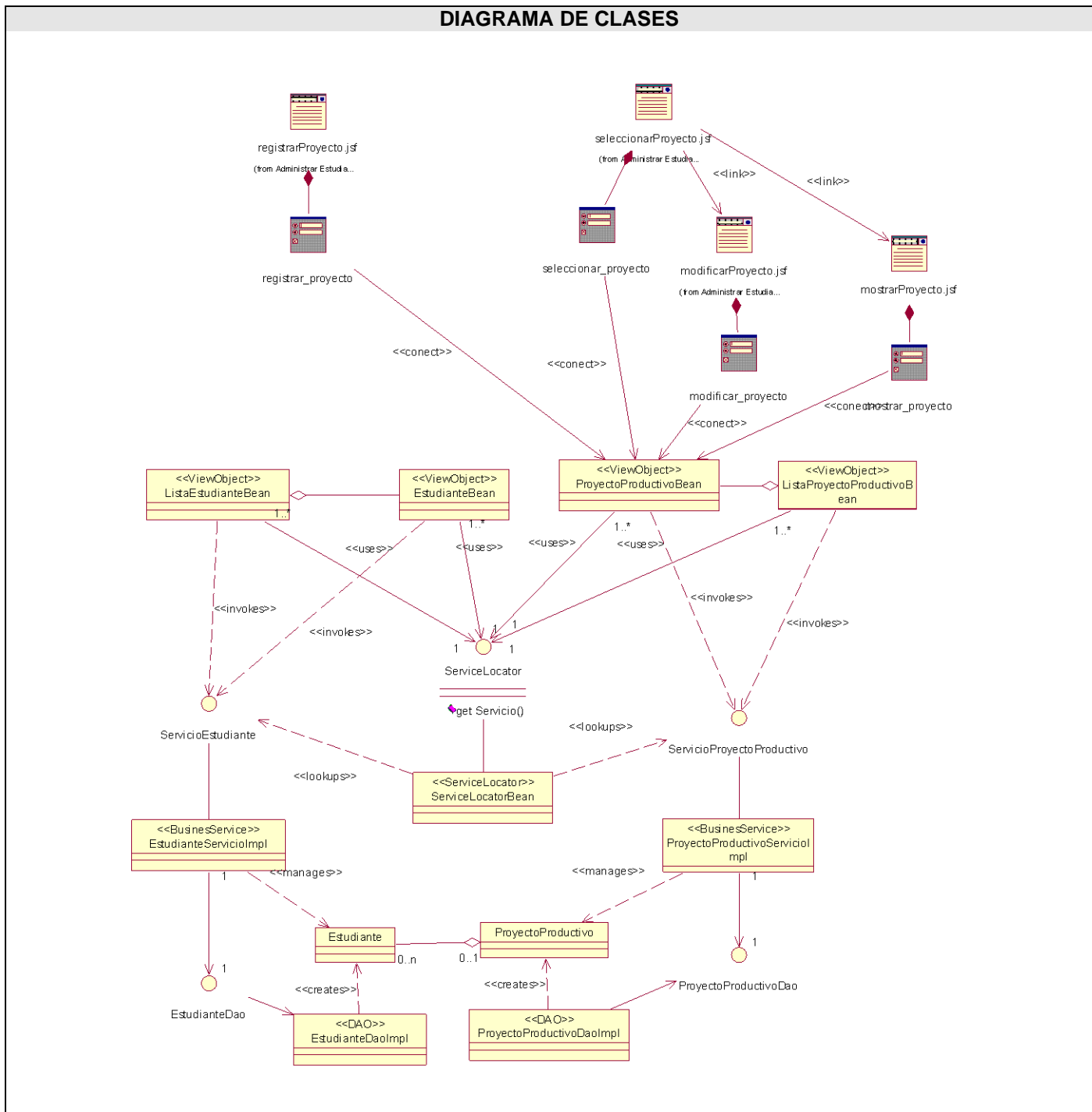


Fig. 3.29 Diagrama de CDW. CU "Administrar Información Proyecto"

Capítulo 3: Análisis y Diseño del Sistema

3.3 Diseño de la BD.

3.3.1 Modelo lógico de datos (diagrama de clases persistentes).

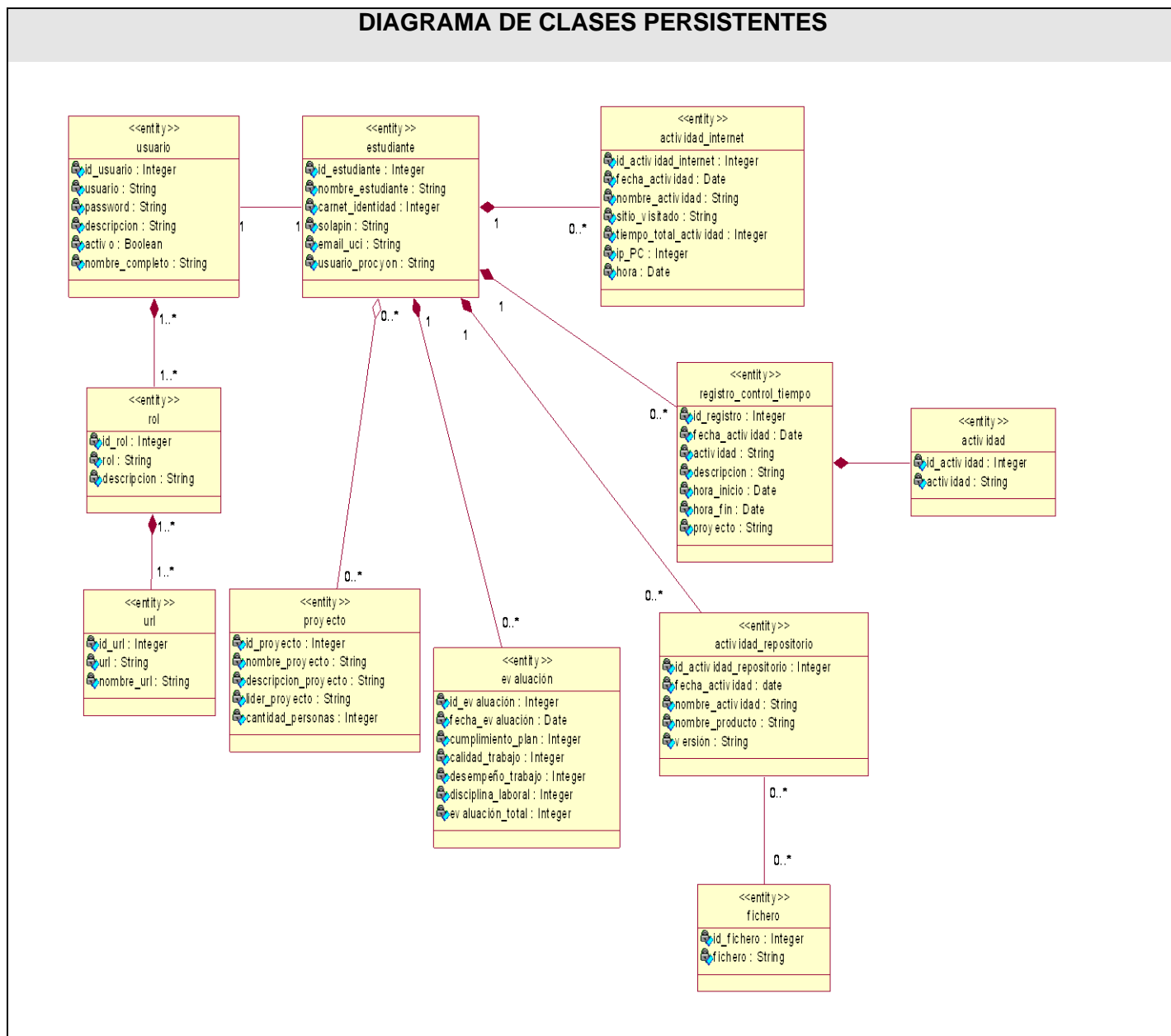


Fig. 3.31 Diagrama de clases persistentes.

Capítulo 3: Análisis y Diseño del Sistema

3.4 Definiciones de diseño de la interfaz de usuario.

La importancia del diseño de la aplicación se basa en que este será el que modele la interacción entre usuario y aplicación, y por tanto posibilitará o no el logro de los objetivos perseguidos por el usuario.

El sistema propuesto está desarrollado en un ambiente agradable y cómodo ya que uno de los principales objetivos de un sistema informático que automatice un proceso dado es que sea aceptado sin problemas por el usuario, o sea, que el cambio no implique rechazo; para esto, la navegación por el sitio es muy cómoda, de cada pantalla se puede acceder a toda la información que se necesite.

De esta forma, cada página cuenta con un menú de navegación a la izquierda, de forma tal que el usuario tenga acceso rápido y fácil a la opción que desee. Ver anexo 7, figura 7.1.

Predomina el azul, color corporativo, con el objetivo de contribuir a la uniformidad del sistema con el resto de los software utilizados en PROCYON, como por ejemplo el Time Tracker.

Los reportes que brinda el sistema mantienen una misma línea de diseño, que se caracteriza, por el título del informe, la información; se mantuvo además la misma estructura de representación de la información a fin de que quien la reciba no note la diferencia y por tanto no rechace el cambio. Ver anexo 7, figura 7.2.

3.5 Tratamiento de errores.

Una parte importante en el desarrollo de un determinado sistema lo constituye la validación de los datos que son entrados al mismo y el análisis de los errores que puedan ser introducidos por el usuario. Tener esto en cuenta y solucionarlo contribuye en gran medida a la calidad del producto terminado.

Al cometerse un error la aplicación debe detectarlo y tratar de corregirlos. El mejor remedio para los errores es la prevención, pero evidentemente los errores ocurrirán incluso con los usuarios de más habilidad y experiencia, por esto si no se pueden evitar los errores al menos hay que tratar de minimizar sus consecuencias y ayudar a los usuarios a recuperarse de estos.

Las entradas de datos incorrectas causan, al menos, pérdida de tiempo a los usuarios y pueden incluso causarles interrupciones a las aplicaciones Web.

Para evitar entradas incorrectas se trató en la mayoría de los casos que se usaran controles de selección tales como SelectOne, SelectMany, entre otros, de forma que el usuario no tenga que teclear la información, sino que la escoja de valores ya introducidos. Ver anexo 7, figuras 7.3.1, 7.3.2.

Capítulo 3: Análisis y Diseño del Sistema

Sin embargo, en todos los casos esto no es posible, ya que hay que obtener información de los usuarios. Con el objetivo de verificar estos datos, se siguió la filosofía de chequear los datos introducidos contra el valor, el rango o el formato que se espera recibir del usuario. Se usaron para esto los controles de validación que vienen incorporados en el Eclipse para aplicaciones JAVA. Ver anexo 7, figura 7.4.

Con el uso de los controles de validación se mejora la experiencia de los usuarios con el sitio Web reduciendo el tiempo de espera por los mensajes de error o la posibilidad de interpretaciones incorrectas o la falla del sitio Web debido a problemas con la entrada de datos de los usuarios.

Cuando se combinan con mensajes de errores comprensibles y útiles, los controles de validación pueden mejorar en gran medida la usabilidad de la aplicación Web y por tanto mejorar la percepción del cliente sobre la calidad del sitio.

Los controles de validación verifican que el usuario haya llenado correctamente los controles de entrada de datos, antes de que la petición sea procesada en el servidor, actuando como un filtro de datos antes de que la página sea procesada. De forma que el usuario recibe una retroalimentación inmediata de si el dato que entró es válido o no.

Este tipo de validación trabaja comparando la entrada del usuario contra un formato predeterminado de entrada, este formato puede incluir el número de caracteres, el uso de dígitos y/o letras, el rango de valores o una cadena específica de caracteres.

De esta forma, los datos entrados por los usuarios que no se correspondan con los formatos predeterminados provocan que el procesamiento de la página sea bloqueado hasta que el usuario rectifique los datos a fin de que tengan el formato apropiado y entonces la página será reenviada para ser procesada.

3.6 Seguridad.

Los datos que se manejan en la aplicación no poseen gran sensibilidad, no obstante se garantiza tener un sistema de seguridad y protección que garantice la calidad de los datos y el buen funcionamiento del sistema.

La aplicación cuenta con un subsistema de seguridad basado en la restricción de permisos al conjunto de usuarios posibles; por lo que solo podrán interactuar con el mismo aquellos usuarios autorizados previamente; la página inicial es una página de identificación, según el tipo de usuario podrá acceder a los datos que necesita.

Capítulo 3: Análisis y Diseño del Sistema

3.7 Conclusiones.

En el presente capítulo se mostraron los resultados de la etapa de análisis y diseño del sistema. Se desarrollaron los diagramas de interacción a través de los diagramas de secuencia.

Se obtuvo el diagrama de clases donde se representaron las clases de forma que quedó una representación de las clases, sus asociaciones, atributos y responsabilidades.

Como culminación al diseño se realizó la concepción del tratamiento de errores, los principios generales del diseño de interfaz de usuario y el sistema de seguridad.

Todos estos elementos obtenidos brindan una idea mucho más clara de las clases, subsistemas, entre otros, para lograr una mejor codificación.

Capítulo 4. Implementación.

4.1 Introducción

En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

4.2 Modelo de Implementación.

Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará a aplicación.

4.2.1 Diagrama de despliegue.

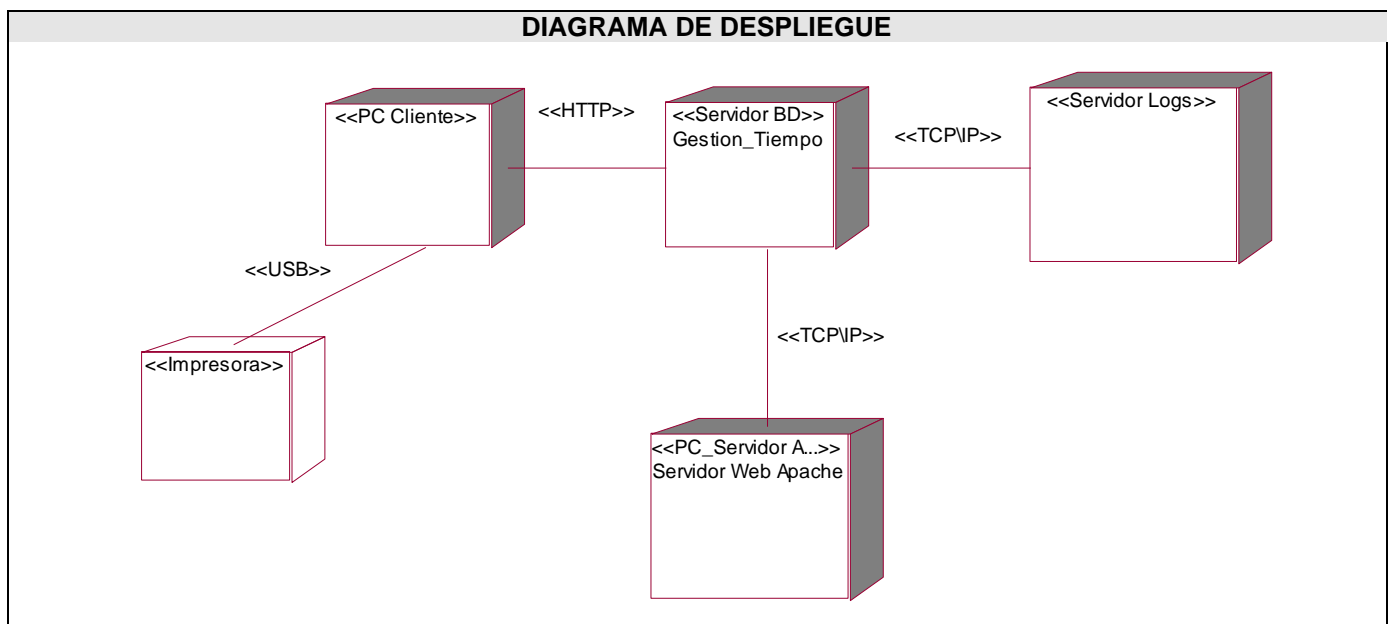


Fig. 4.1 Diagrama de despliegue.

4.2.2 Diagrama de componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones.

4.2.2.1 Diagramas de componentes. CU Administrar Información Proyecto

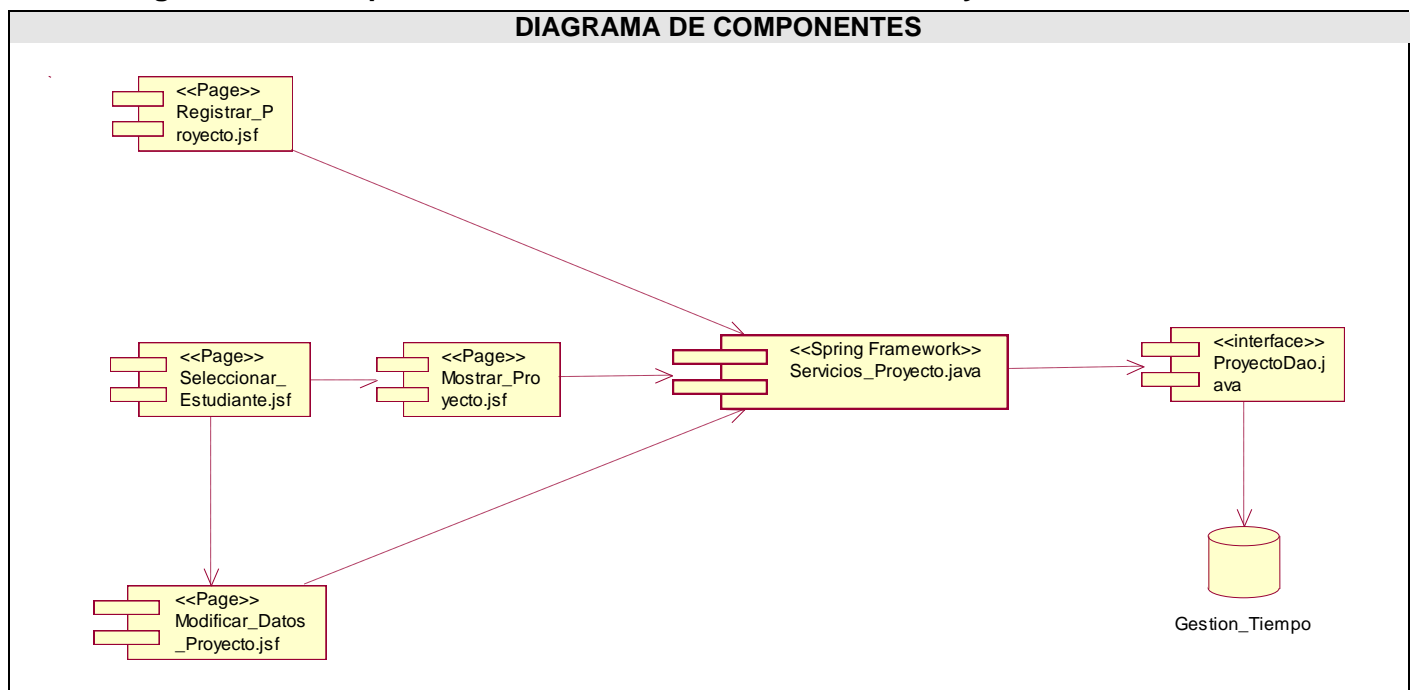


Fig. 4.2 Diagrama de componentes. CU "Administración Información Proyecto".

4.2.2.2 Diagramas de componentes. CU“Autenticar Usuario”.

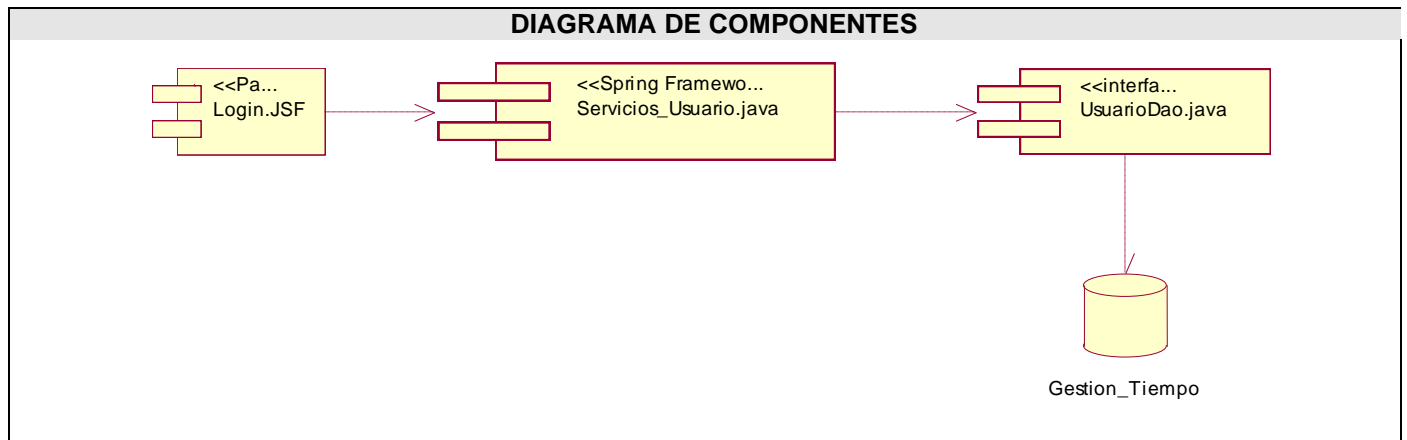


Fig. 4.2 Diagrama de componente. CU “Autenticar Usuario”.

4.2.2.3 Diagramas de componentes. CU “Administrar Estudiante”.

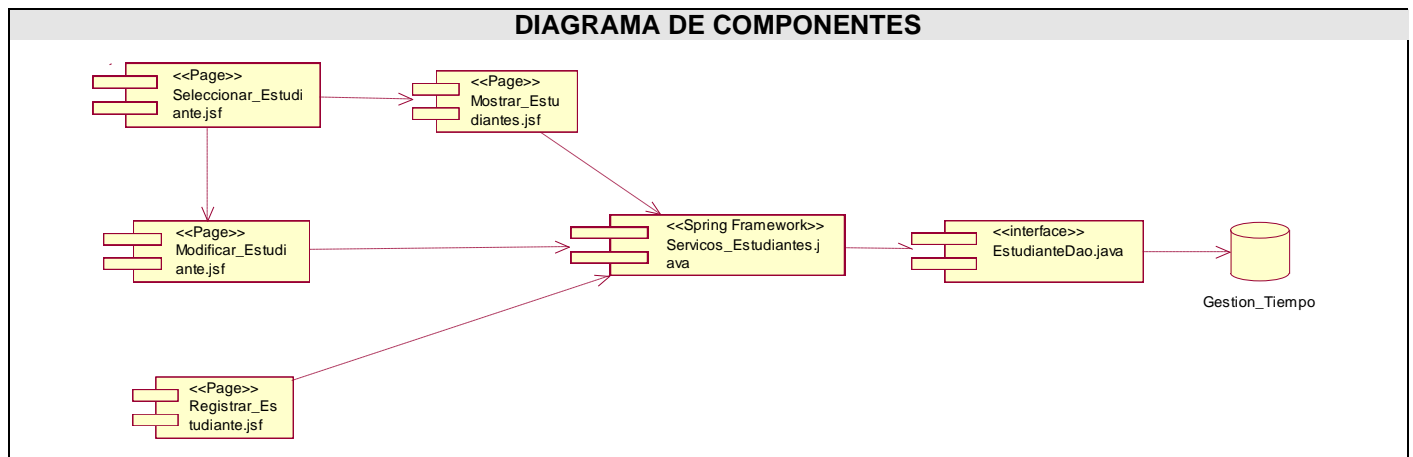


Fig. 4.3 Diagrama de componente. CU “Administrar Estudiante”

4.2.2.4 Diagramas de componente. CU “Gestionar Evaluación Estudiante”.

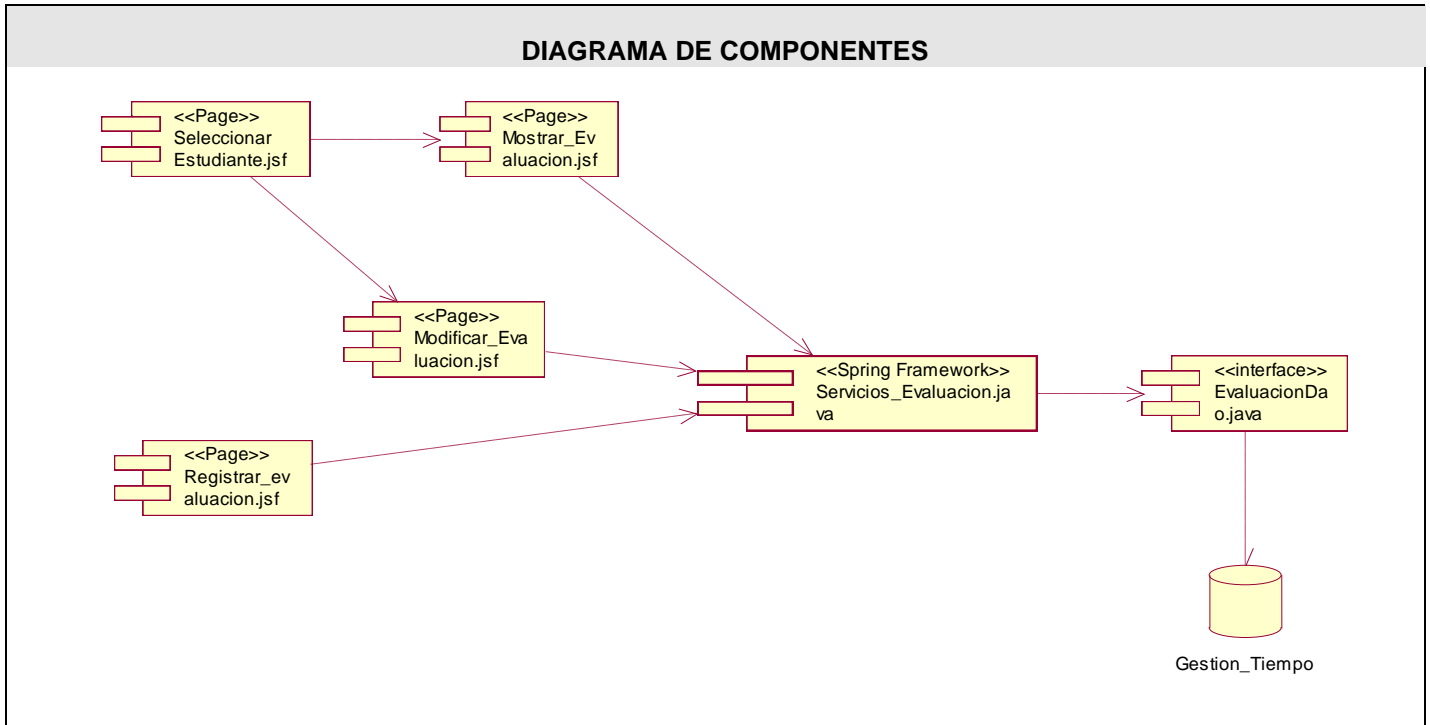


Fig. 4.4 Diagrama de componente. CU “Gestionar Evaluación Estudiante”

4.2.2.5 Diagrama de componente. CU “Mostrar Actividad Estudiante”

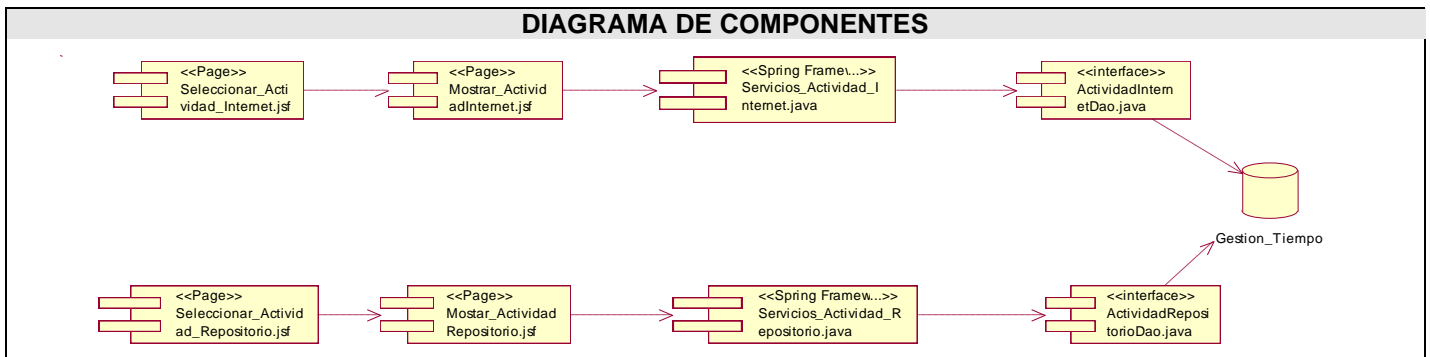


Fig. 4.5 Diagrama de componente. CU “Mostrar Actividad Estudiante”

4.2.2.6 Diagrama de componente. CU “Gestionar Tiempo”

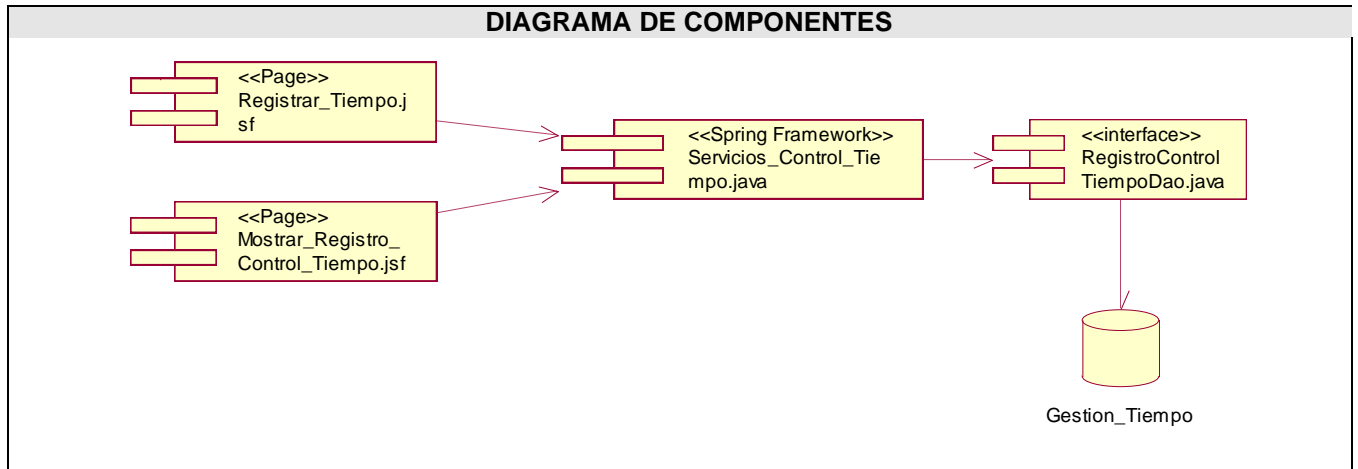


Fig. 4.6 Diagrama de componente. CU “Gestionar Tiempo”

4.2.2.7 Diagrama de componente. CU “Gestionar Usuario”

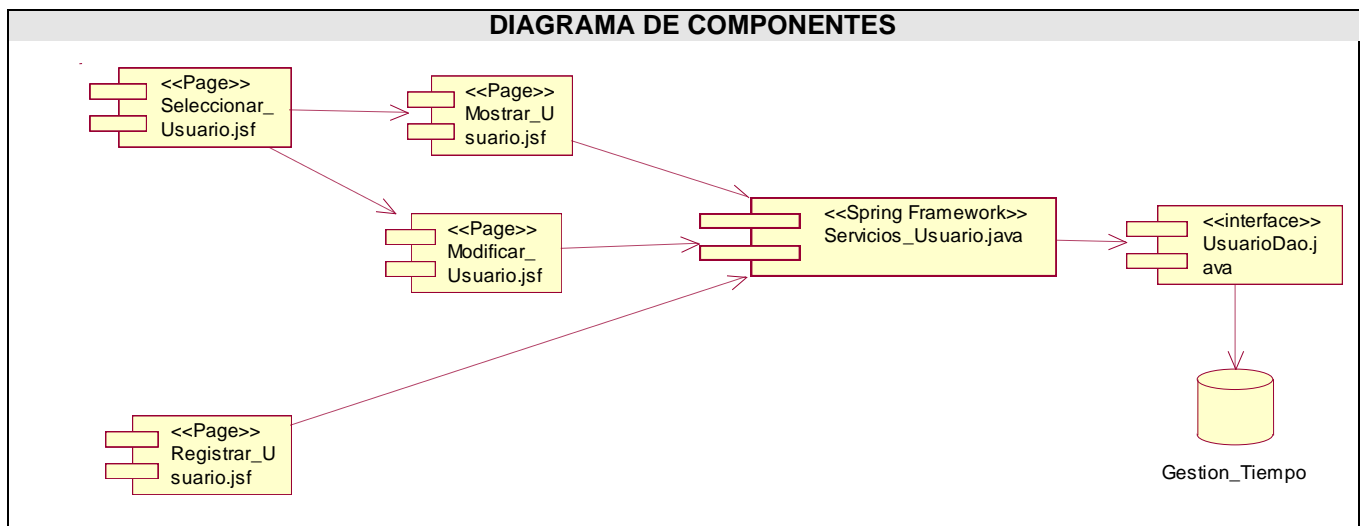


Fig. 4.7 Diagrama de componente. CU “Gestionar Usuario”

4.3 Conclusiones

El Diagrama de Componentes y el Diagrama de Despliegue dan una visión más clara de cómo se estructurará el sistema desde el punto de vista de componentes y nodos.

Todos estos elementos obtenidos brindan una idea mucho más clara de las clases, subsistemas, algoritmos, etc. para lograr una mejor codificación.

Capítulo 5. Estudio de Factibilidad.

5.1 Introducción.

Para la realización de un proyecto es de suma importancia el análisis del costo y los beneficios que reportará. Como resultado de este análisis se obtiene el tiempo de desarrollo en meses, costo y la cantidad de personas que se necesitan para desarrollar el proyecto. En este capítulo se describe la estimación de costos del sistema propuesto y sus beneficios.

5.2 Planificación basada en casos de uso.

La especificación de los requerimientos mediante Casos de Uso ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema. El método de Casos de Uso permite documentar los requerimientos de un sistema en términos de Actores y Casos de Uso. Un Actor típicamente representa a un usuario humano o a otro sistema que interactúa con el sistema bajo análisis. Un Caso de Uso representa un gránulo funcional del sistema bajo análisis, relatado como una secuencia de acciones que uno o más actores llevan a cabo en el sistema para obtener un resultado de valor significativo.

5.2.1 Paso 1. Cálculo de los Puntos de casos de uso Desajustados.

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

Capítulo 5 : Estudio de Factibilidad

Tabla 5.1 Factor de peso de los actores sin ajustar.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	3	9
Total			3	9

$$UAW = \sum \text{cant actores} * \text{peso}$$

$$UAW = 3 * 3$$

$$UAW = 9$$

Tabla 5.2 Factor de peso de los casos de uso sin ajustar.

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	1	5
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	2	20
Complejo	El caso de uso tiene más de 8 transacciones.	15	4	60
Total			7	85

$$UUCW = \sum \text{cantCU} * \text{Peso}$$

$$UUCW = 85$$

$$UUCP = UAW + UUCW$$

$$UUCP = 9 + 85$$

$$UUCP = 94$$

Capítulo 5 : Estudio de Factibilidad

5.2.1 Paso 2. Cálculo de los Puntos de casos de uso ajustados.

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de casos de uso ajustados.

UUCP: Puntos de casos de uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

El factor de complejidad técnica (TCF) se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada factor se cuantifica en un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Tabla 5.3 Factor de complejidad técnica.

Factor	Descripción	Peso	Valor asignado	Comentario	Total
T1	Sistema distribuido	2	0	El sistema es centralizado.	6
T2	Tiempo de respuesta	1	3	La velocidad es relativamente alta.	2
T3	Eficiencia del usuario final	1	1	Abundantes restricciones de eficiencia.	1
T4	Funcionamiento Interno complejo	1	1	No hay cálculos complejos	1
T5	El código debe ser reutilizable	1	4	El código es reutilizable.	1
T6	Facilidad de instalación	0,5	1	El sistema no necesita instalación sino	0,5

Capítulo 5 : Estudio de Factibilidad

				algunas configuraciones.	
T7	Facilidad de uso	0,5	4	Alta	1,5
T8	Portabilidad	2	4	El sistema es portable.	2
T9	Facilidad de cambio	1	2	Es flexible al cambio.	1
T10	Concurrencia	1	4	Hay Concurrencia.	3
T11	Incluye objetivos especiales de seguridad	1	1	Seguridad Normal	1
T12	Provee acceso directo a terceras partes	1	1	No provee conexiones con otros sistemas.	1
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	El sistema es muy fácil de usar.	1
Total					28.5

$TCF=0.6+0.01* \Sigma (\text{peso}*\text{valor asignado})$

$TCF=0.6+0.01*28$

$TCF=0.6+0.28$

$TCF=0.885$

El factor de ambiente (EF) está relacionado con las habilidades y entrenamiento del grupo de desarrollo que realiza el sistema. Cada factor se cuantifica con un valor desde 0 (aporte irrelevante) hasta 5 (aporte muy relevante).

Capítulo 5 : Estudio de Factibilidad

Tabla 5.4 Factor de ambiente.

Factor	Descripción	Peso	Valor asignado	Comentario	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	0	El grupo no está familiarizado.	0
E2	Experiencia en la aplicación	0,5	2	La mayoría del grupo ha trabajado con la aplicación.	1
E3	Experiencia en la orientación a objetivos.	1	3	La mayoría del grupo programa en Objetos.	4
E4	Capacidad del analista líder.	0,5	5	Tiene bastante Experiencia.	2
E5	Motivación.	1	5	El grupo está altamente motivado.	5
E6	Estabilidad de requerimientos	2	3	Hubo desestabilidad en los requerimientos.	10
E7	Personal Part–Time	-1	0	No hubo personal Part–Time.	0
E8	Dificultad del lenguaje de programación	-1	5	Se usará Java Hibernate.	-2
Total					12.5

$EF=1.4 - 0.03* \Sigma(\text{peso}*\text{valor asignado})$

$EF=1.4 - 0.03* \Sigma(\text{peso}*\text{valor asignado})$

$$EF=1.4 - 0.03*12.5$$

$$EF=1.4 - 0.375$$

$$EF=1.025$$

$$UCP=UUCP*TCF*EF$$

$$UCP=94*0.885*1.025$$

$$UCP=85.26975$$

5.2.3 Paso 3. Estimación de esfuerzo a través de los puntos de casos de uso.

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

$$CF = 20 \text{ Horas-Hombre / Punto de Casos de uso.}$$

$$E= UCP*CF$$

$$E=85.26975*20$$

$$E=2387.553\text{Horas-Hombre}$$

Capítulo 5 : Estudio de Factibilidad

5.2.4 Paso 4. Calcular esfuerzo de todo el proyecto.

Tabla 5.5 Esfuerzo del proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	20	1193.7765
Diseño	20	1193.7765
Implementación	40	2387.553
Pruebas	10	596.88825
Sobrecarga (otras actividades)	10	596.88825
Total	100	5968.8825

Si $E_T = 5968.8825$ horas-hombre y se estima que cada mes tiene como promedio 192 horas laborables, eso daría un $E_T = 31.0879297$ mes-hombre.

Esto quiere decir que 1 persona puede realizar el problema analizado en 31 meses y medio aproximadamente.

-Costo del Proyecto.

Se asume como salario promedio mensual \$100.00

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH =4 hombres

CHM = 4 * Salario Promedio

CHM = 400.00 \$/mes

Costo = CHM * E_T / CH

Costo = 400.00 * 5968.8825 / 4

Costo = 12435.119 \$

Tiempo = E_T / CH

Tiempo = 15.20416 / 4

Capítulo 5 : Estudio de Factibilidad

Tiempo = 7.77 ≈ 8 meses

De los resultados obtenidos se interpreta que con 4 hombres trabajando en el proyecto el mismo se desarrolla en 8 meses y su costo total se estima que sea 12435.119 \$.

5.3 Beneficios tangibles e intangibles.

El Sistema de Gestión de Tiempo de los Estudiantes vinculados al Proyecto PROCYON no es un producto con fines comerciales, su principal objetivo es resolver el problema existente en el proyecto de no contar con la información actualizada de los tiempos que el estudiante utiliza para trabajar en dicho proyecto, siendo esto de gran importancia a la hora de una toma de decisiones en un momento dado.

El beneficio fundamental del sistema es contar con una aplicación Web flexible, dinámica y de interfaz agradable que le permita registrar, actualizar y conocer de una forma más precisa y en el menor tiempo posible la información referente a los estudiantes.

Por tanto, los beneficios inmediatos son generalmente intangibles:

- Disminución del tiempo y esfuerzo que se invierte en esta tarea que se realiza, hasta ahora, de forma manual.
- Disminución de la acumulación de materiales impresos relacionados con los procesos de gestión de la información referente a los estudiantes.
- Fácil detección de errores.
- Fácil y rápido acceso y publicación de la información actualizada.
- Fácil procesamiento de la información y obtención, dinámica, de reportes de la situación de ejecución del presupuesto de los estudiantes en cualquier momento.

5.4 Análisis de costos y beneficios.

Llevar a cabo el desarrollo de un producto de software requiere gastos. Para poder justificar su desarrollo depende de los beneficios que reporta su implantación y utilización. Estos beneficios pueden ser económicos y de orden social, de gran importancia ambos. El sistema propuesto se utilizará en el proyecto PROCYON de la Universidad de las Ciencias Informáticas, para la gestión de tiempo de los estudiantes, garantizando una constante actualización de la información.

Capítulo 5 : Estudio de Factibilidad

La tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios, debido a la estructuración en capas de los procesos del negocio que se diseñaron.

Analizando el costo del proyecto, los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es realmente factible.

5.5 Conclusiones.

En este Capitulo se describió el estudio de factibilidad correspondiente al sistema propuesto, analizando el costo estimado y las mejoras que aportarán su implantación.

La herramienta propuesta reportará beneficios significativos e importantes para la gestión de tiempo de los estudiantes vinculados al proyecto PROCYON, contribuye a mejorar la estrategia de trabajo y así lograr la mejor productividad del proyecto. Además permite tener más organizada toda la información referente a los estudiantes. Por lo cual se concluye que resulta factible implementar la herramienta propuesta.

Conclusiones Generales

Se espera que el documento en forma general, permita la comprensión teórica de la situación problemática existente y su solución, así como el desarrollo de las diferentes etapas de la aplicación usando la metodología RUP.

El desarrollo de este trabajo de tesis está orientado a la concepción de una herramienta informática para la gestión del tiempo de los estudiantes vinculados a PROCYON. El valor fundamental de esta herramienta se expresa en la contribución a simplificar el trabajo y la demora que produce el procesamiento manual de la información y mejorar la gestión de tiempo de las actividades que se realizan durante el horario de trabajo.

Se alcanzó, satisfactoriamente, el objetivo propuesto: Implementar un sistema informático para la automatización de la gestión del tiempo de los estudiantes vinculados a PROCYON; reafirmando así la utilidad y validez de emplear las tecnologías informáticas para simplificar el trabajo del personal encargado de la tarea.

Se ha demostrado la eficacia de los lenguajes y tecnologías utilizadas para el desarrollo del sistema.

Se realizó una base de datos, donde se almacena toda la información necesaria que se genera de los procesos de gestión de los estudiantes, del tiempo, los proyectos y las actividades que el estudiante realiza en internet, así como la gestión de la evaluación, para de esta forma garantizar la veracidad y centralización de la misma.

Se realizó el análisis, diseño e implementación del sistema.

La solución propuesta ha sido acertada, los requerimientos soportan al sistema y los casos de uso satisfacen las necesidades funcionales.

Se han seguido los principios básicos de diseño descritos para el desarrollo del sistema.

Se logra una seguridad y protección de los datos consecuente con el nivel de seguridad requerido.

Recomendaciones.

A continuación se mencionan algunas recomendaciones con el objetivo de realizar un seguimiento y mejora al software.

- Diseñar un menú dinámico para un mejor funcionamiento del software.
- Añadir al software todo lo referente a la capacitación de los estudiantes.
- Implementar lo relacionado con la actividad del estudiante en el repositorio.
- Añadir al software para otras versiones la funcionalidad de Verificar el logueo y deslogueo del estudiante en el dominio.
- Utilizar el sistema en otros proyectos similares, realizando los cambios pertinentes.

Referencia Bibliográfica

1. mailxmail.com. *Gestión del tiempo - Encauzar reuniones*. [Online] http://www.mailxmail.com/curso/empresa/habilidades_directivas/capitulo14.htm.
2. **Maroto, Juan Carrión**. Gestión del Conocimiento.com. [En línea] <http://www.Introducción Conceptual a La Gestión del Conocimiento>.
3. **Monster.com**. Monster. Orientación Profesional.com. *Diez consejos para la gestión efectiva del tiempo*. [En línea] 2000. http://contenido.monster.es/trabajar/vida_lab/diez_consejos/
4. **Microsoft Corporation**. Microsoft.Centro de Ayuda para Empresas. *Ponga a prueba estas estrategias para ahorrar tiempo o programar sus reuniones*. [En línea] 2007. http://www.microsoft.com/spain/empresas/temas/optimize_tareas/reuniones.aspx.
5. **Palenzuela, Otniel Barrera and Pérez, Deiny Garcia**. *Monitoreando el Tráfico de Internet. Una solución al alcance de la mano*.
6. **Linalco Consulting SL. Especialistas en Linux y Software Libre**. Servidor Proxy-Cache Squid . [Online] 2007. <http://www.linalco.com/squid.html>.
7. **espanol.softpicks.net**. espanol.softpicks.net. *Internet Access Monitor para Proxy+ .* [Online] enero 12, 2004. http://espanol.softpicks.net/software/Internet-Access-Monitor-para-Proxy-_es-32682.htm.
8. **Microsoft**. Microsoft.com. [Online] Mayo 24 , 2007. <http://www.microsoft.com/spain/windowsserver2003/default.aspx>.
9. http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java. [Online] mayo 21, 2007. [Cited:]
10. **Wikipedia**. Wikipedia. *Programación orientada a objetos*. [Online] febrero 15, 2007. http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos.
11. **Wikipedia**. <http://es.wikipedia.org/wiki/PostgreSQL>. [Online] mayo 2007. [Cited: mayo 16, 2007.] <http://es.wikipedia.org/wiki/PostgreSQL>.
12. **Welicki, León**. MSDN. *Patrones y Antipatrones: una Introducción - Parte II*. [Online] 2007. http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.asp#M19.

13. **Shen, Derek Yang.** Integración de JSF, Spring e Hibernate para crear una Aplicación Web del Mundo Real. [Online] 2006 . sitio http://www.programacion.net/tutorial/jap_jsfwork/3/.
14. **Glögl, Michael and Palos, Juan Antonio(traductor).** Java en Castellano. *Persistencia de Objetos Java: El Camino hacia Hibernate*. [Online] 2006. <http://www.programacion.net/java/tutorial/hibernate/>.
15. **Martín, Cesar Crespo.** adictosaltrabajo.com. [Online] agosto 14, 2004. <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernate>
16. **Sanchez, María A. Mendoza.** informatizate. *Metodologías De Desarrollo De Software*. [Online] junio 7, 2004. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
17. **aulaClic.com.** aulaClic. Curso de Dreamweaver 8. [Online] agosto 2006. <http://www.aulacli.com/dreamweaver8/index.htm>.

Bibliografía.

aulaClic.com. aulaClic. Curso de Dreamweaver 8. [Online] agosto 2006. <http://www.aulacli.com/dreamweaver8/index.htm>.

espanol.softpicks.net. espanol.softpicks.net. *Internet Access Monitor para Proxy+* . [Online] enero 12, 2004. http://espanol.softpicks.net/software/Internet-Access-Monitor-para-Proxy-_es-32682.htm

Glögl, Michael and Palos, Juan Antonio(traductor). Java en Castellano. *Persistencia de Objetos Java: El Camino hacia Hibernate*. [Online] 2006. <http://www.programacion.net/java/tutorial/hibernate/>.

http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java. [Online] mayo 21, 2007.

Linalco Consulting SL. Especialistas en Linux y Software Libre. Servidor Proxy-Cache Squid . [Online] 2007. <http://www.linalco.com/squid.html>.

mailxmail.com. *Gestión del tiempo - Encauzar reuniones*. [Online] http://www.mailxmail.com/curso/empresa/habilidades_directivas/capitulo14.htm.

Martín, Cesar Crespo. adictosaltrabajo.com. [Online] agosto 14, 2004. <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=hibernate>

Maroto, Juan Carrión. Gestión del Conocimiento.com. [En línea] <http://www.Introducción Conceptual a La Gestión del Conocimiento>.

Microsoft. Microsoft.com. [Online] Mayo 24 , 2007.
<http://www.microsoft.com/spain/windowsserver2003/default.msp>

Microsoft Corporation. Microsoft.Centro de Ayuda para Empresas. *Ponga a prueba estas estrategias para ahorrar tiempo o programar sus reuniones.* [En línea] 2007.
http://www.microsoft.com/spain/empresas/temas/optimize_tareas/reuniones.msp

Monster.com. Monster. Orientación Profesional.com. *Diez consejos para la gestión efectiva del tiempo.* [En línea] 2000. http://contenido.monster.es/trabajar/vida_lab/diez_consejos/

Sanchez, María A. Mendoza. informatizate. *Metodologías De Desarrollo De Software.* [Online] junio 7, 2004.
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html

Shen, Derek Yang. Integración de JSF, Spring e Hibernate para crear una Aplicación Web del Mundo Real. [Online] 2006 . sitio http://www.programacion.net/tutorial/jap_jsfwork/3/.

Wikipedia. Wikipedia. *Programación orientada a objetos.* [Online] febrero 15, 2007.
http://es.wikipedia.org/wiki/Programaci3n_orientada_a_objetos.

Wikipedia. <http://es.wikipedia.org/wiki/PostgreSQL>. [Online] mayo 2007. [Cited: mayo 16, 2007.]
<http://es.wikipedia.org/wiki/PostgreSQL>.

Welicki, León. MSDN. *Patrones y Antipatrones: una Introducción - Parte II.* [Online] 2007.
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.asp#M19.

Anexos

Anexo1

Descripción del caso de uso Asignar Proyecto

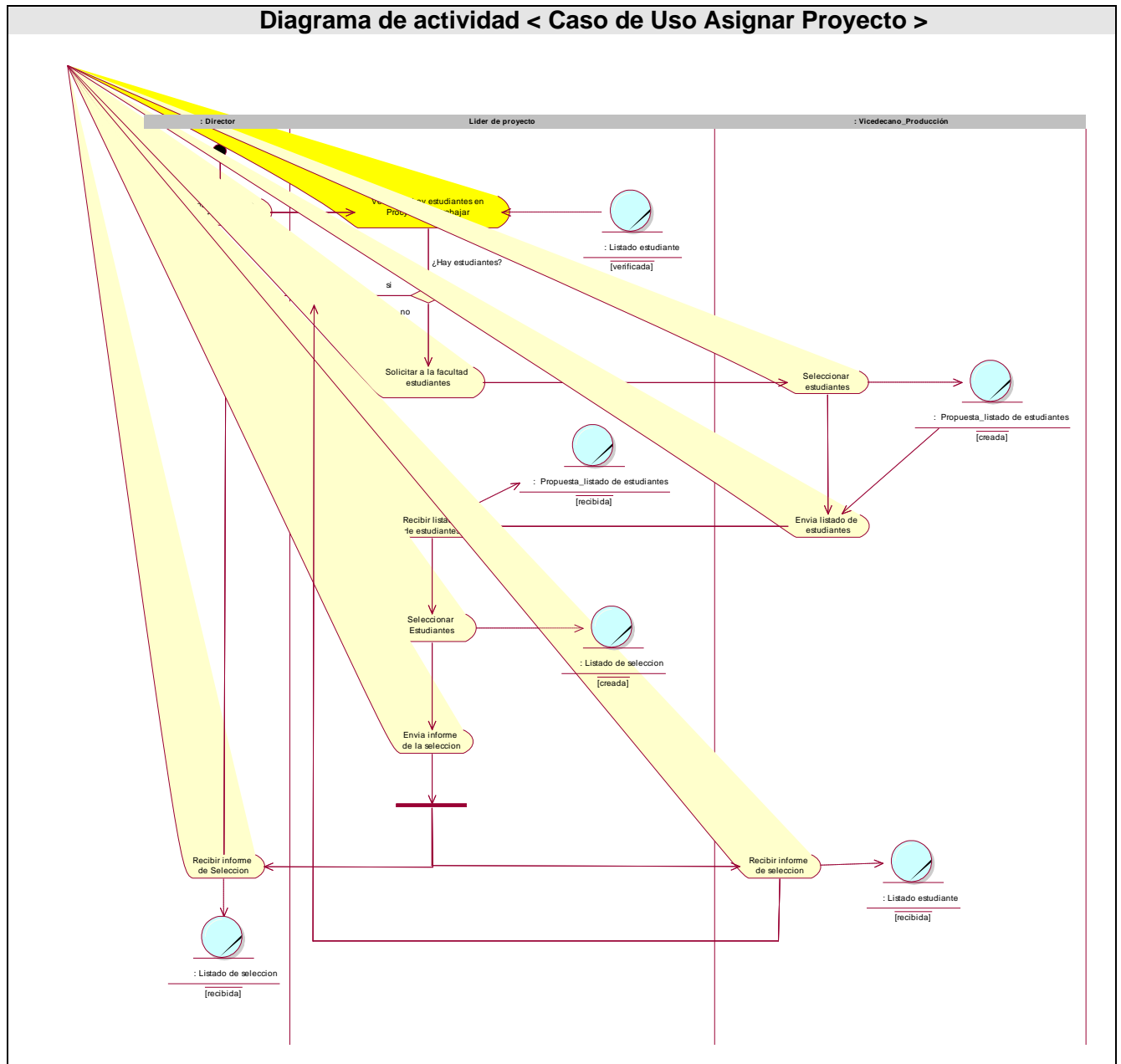
Tabla 1.1: Descripción del caso de uso Asignar Proyecto

Caso de Uso	Asignar proyecto
Actores	Director
Propósito	
Resumen	
El caso de uso se inicia cuando el director de PROCYON asigna un nuevo proyecto a un líder de proyecto. Este selecciona los estudiantes que van a trabajar en el proyecto y así finaliza el caso de uso.	
Casos de uso asociados	–
Curso normal de eventos	
Sección” Asignar proyecto”	
Acción del actor	Respuesta del negocio.
1. El director de PROCYON asigna un nuevo proyecto al líder de proyecto, en el cual se vinculan estudiantes que trabajan en PROCYON o nuevos estudiantes de la facultad.	1.1 El líder del proyecto verifica si hay estudiantes para trabajar en el proyecto. a. El líder de proyecto selecciona los estudiantes para trabajar.
Flujos alternos	
Acción del actor	Respuesta del negocio
	1.1 El líder de proyecto solicita un listado de estudiantes a la facultad.

	<p>1.2 El Vicedecano recibe la solicitud y selecciona una cantera de estudiantes y se los envía al líder del proyecto.</p> <p>1.3 Este recibe la información y realiza pruebas a los estudiantes y una encuesta.</p> <p>1.4 Selecciona a los aprobados y envía un listado al Vicedecano de producción y al director de PROCYON.</p>
<p>2. El director recibe el listado de los estudiantes seleccionados para trabajar en el proyecto.</p>	
<p>Prioridad</p>	
<p>Mejoras</p>	

Anexo 2

Diagramas de Actividad del caso de uso Asignar Proyecto.



Anexo 3

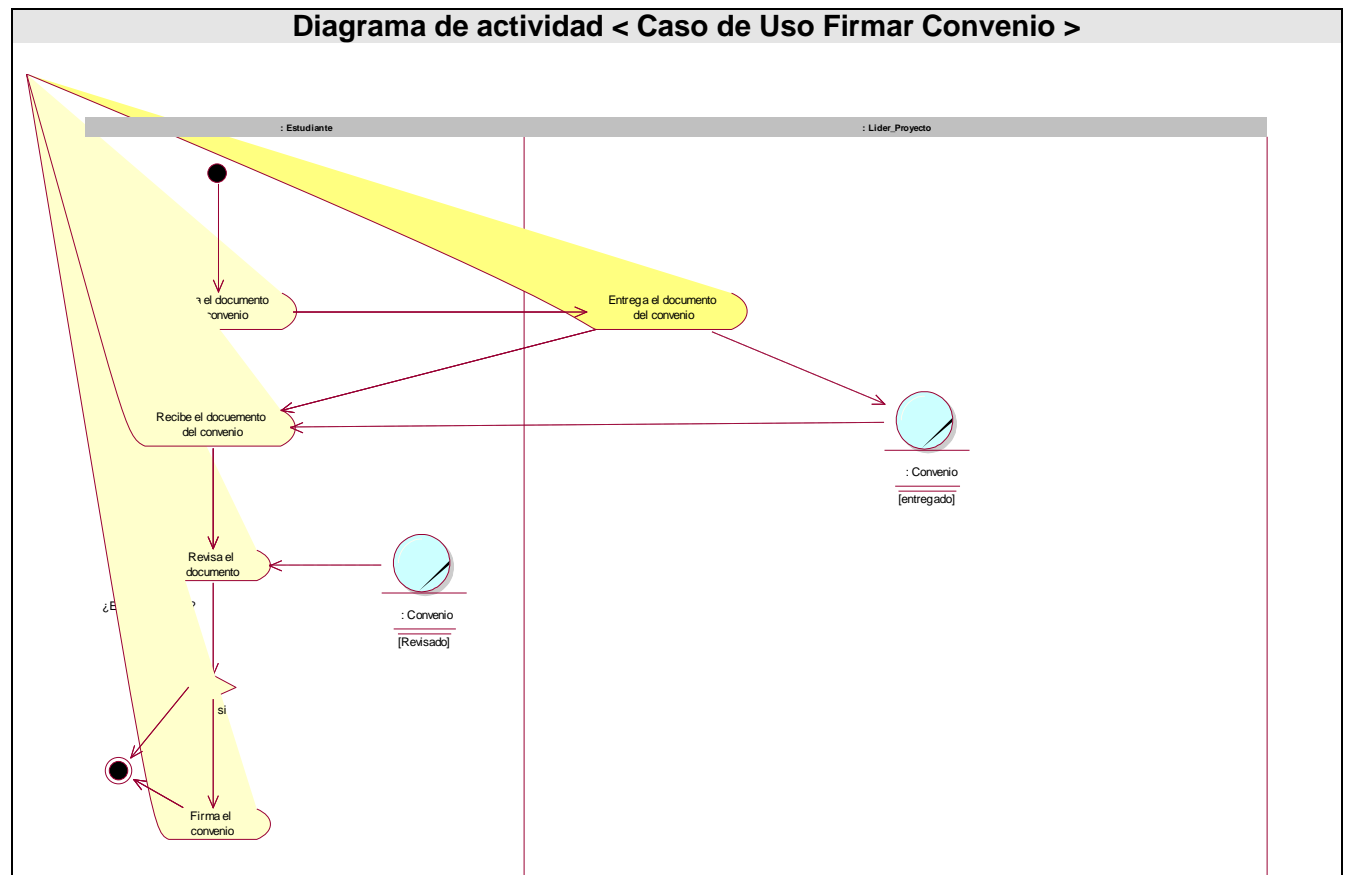
Descripción del caso de uso Firmar Convenio.

Tabla 3.1: Descripción del caso de uso Firmar Convenio

Caso de Uso	Firmar Convenio	
Actores	Estudiante	
Trabajadores	Líder del Proyecto	
Resumen		
El caso de uso se inicia cuando el estudiante una vez aceptado en el proyecto procede a firmar el convenio que respalda la relación entre el estudiante y PROCYON. Para ello el estudiante debe conocer las condiciones que recoge el mismo, si las acepta, firma dicho convenio.		
Precondiciones	El estudiante debe haber sido seleccionado para trabajar en el proyecto.	
Flujo normal de eventos		
Acción del actor	Respuesta del negocio.	
1. El estudiante solicita el documento del convenio.	1.1 El líder del proyecto le entrega el documento al estudiante.	
2. El estudiante lee el documento.		
3. Si está de acuerdo con lo que este establece procede a la firma del mismo.		
4. Entrega el documento al líder del proyecto.	4.1 Recibe el documento.	
Flujos alternos		
Acción del actor	Respuesta del negocio	
3. Si el estudiante no está de acuerdo con lo que establece el convenio lo rechaza y no lo firma.		
Poscondiciones		

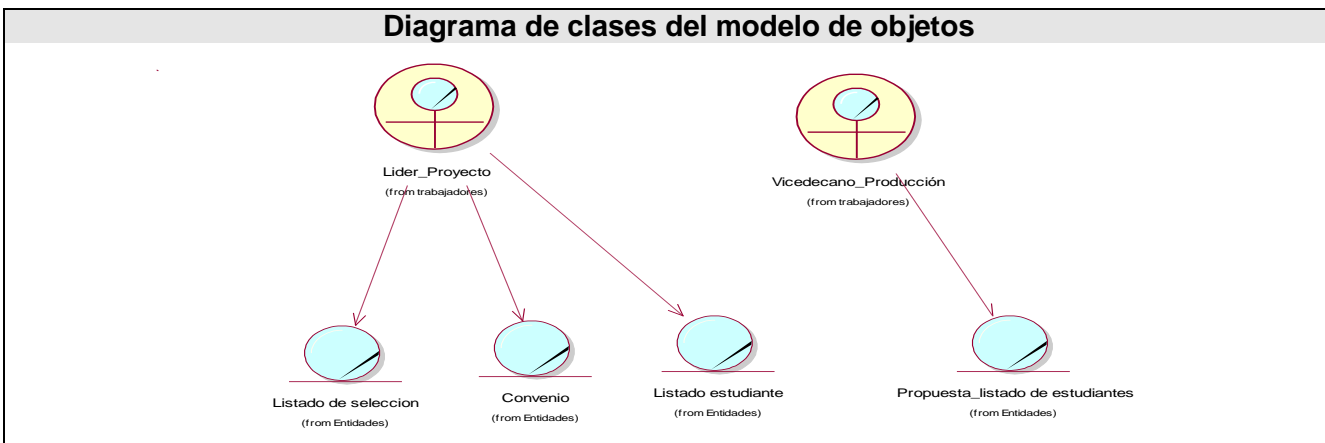
Anexo 4

Diagrama de Actividad del caso de uso Firmar Convenio.



Anexo 5

Diagrama de Clases del Modelo de Objetos



Anexo 6

Descripción de los casos de uso del sistema.

Tabla 6.1: Descripción del caso de uso Autenticar Usuario

Nombre del Caso de Uso	Autenticar Usuario	
Actores	Usuario del Sistema (inicia)	
Propósito	Permitir al usuario autenticarse para acceder al sistema.	
Resumen	El Caso de Uso se inicia cuando el usuario introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza el caso de uso dándole los permisos y habilitándole la entrada.	
Referencias	R1	
Precondiciones		
Poscondiciones	Se habilitan las funcionalidades según lo privilegios.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	




<p>1. El usuario introduce Usuario y Contraseña</p>	<p>1.1 El sistema valida que no existan campos vacíos. 1.2 Encripta la contraseña. 1.3 Busca el usuario. 1.4 Compara la contraseña con la almacenada en la base de datos del sistema. 1.5 Se le asignan los permisos.</p>
<p>Curso Alternativo</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
	<p>1.1 Se emite un mensaje para que llene los campos vacíos. 1.2 En caso del que el usuario no este activo envía un mensaje. 1.3 En caso de no existir se envía un mensaje de aviso. 1.4 En caso de no coincidir la contraseña con la almacenada se envía un mensaje de aviso.</p>
<p>Prioridad:</p>	<p>Crítico</p>
<p>Prototipo</p>	

Tabla 6.2: Descripción del caso de uso Gestionar Usuario.

Nombre del CU	Gestionar Usuario	
Actores	Administrador	
Propósito	Permitir registrar, modificar, eliminar o mostrar los usuarios existentes en el sistema.	
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción de registrar, modificar, eliminar o mostrar usuarios existentes.	
Referencias	R2	
Precondiciones	Administrador previamente autenticado en el sistema	
Poscondiciones	Se registra, modifica, elimina y se muestra un determinado usuario del sistema.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El administrador necesita registrar, modificar, eliminar o mostrar los usuarios del sistema.	<p>El sistema ejecuta alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> a) Si decide registrar un nuevo usuario, ir a la sección "Registrar Usuario" b) Si decide actualizar los datos de un usuario, ir a la sección "Modificar Usuario". c) Si decide mostrar los usuarios ir a la sección "Mostrar Usuario". d) Si se decide eliminar los usuarios ir a la sección "Eliminar Usuario". 	
Sección "Registrar Usuario"		
Acciones del Actor	Respuesta del Sistema	
2 El administrador inserta los datos del usuario para realizar su	2.1 El sistema valida que los campos del usuario estén llenos y los datos estén correctos.	

registro en la aplicación.	<p>2.2 El sistema valida que no esté registrado un usuario con esos datos en el sistema.</p> <p>2.3 El nuevo usuario se almacena en el sistema.</p> <p>2.4 Se muestra un mensaje informándosele al administrador que ya ha sido efectuado el registro del usuario y finaliza el caso de uso.</p>
Curso Alternativo	
	<p>2.1 Se emite un mensaje para que llene los campos obligatorios y garantice que los datos estén correctos.</p> <p>2.2 Si el usuario está registrado se muestra un mensaje informativo y finaliza el caso de uso.</p>
Prototipo	
Sección “Modificar Usuario”	
Acciones del Actor	Respuesta del Sistema
3. El administrador selecciona el usuario del cual desea modificar sus datos.	3.1 El sistema muestra los datos del usuario que se va a modificar.
4 El administrador realiza las modificaciones deseadas.	<p>4.1 Se valida que los campos obligatorios estén llenos y correctos.</p> <p>4.2 Se actualiza la Información y finaliza el caso de uso.</p>

Curso Alternativo	
	3.1 Si el usuario no ha sido registrado no se muestra en la lista de usuarios registrados.
	4.1 Se emite un mensaje para que llene los campos obligatorios y verifique que los datos estén correctos.
Prototipo	
Sección “Mostrar Usuario”	
Acciones del Actor	Respuesta del Sistema
5. El administrador selecciona la opción de mostrar usuarios registrados.	5.1 El sistema muestra la lista de usuarios almacenados en el sistema.
6. El administrador selecciona el usuario del cual desea conocer sus datos.	6.1 El sistema muestra el usuario seleccionado y finaliza el caso de uso.
Curso Alternativo	
	5.1 Si no existen usuarios registrados se muestra un mensaje informativo y finaliza el caso de uso.




Sección "Eliminar Usuario"	
Acciones del Actor	Respuesta del Sistema
7. El administrador inserta los datos del usuario para realizar su búsqueda en la aplicación.	<p>7.1 El sistema verifica que los campos necesarios estén llenos y correctos.</p> <p>7.2 El sistema verifica que el usuario seleccionado exista.</p> <p>7.3 El sistema elimina el usuario seleccionado y finaliza el caso de uso.</p>
Curso Alternativo	
	<p>7.1 Se emite un mensaje para que llene los campos obligatorios y verifique que los datos estén correctos.</p> <p>7.2 Si el usuario no está registrado se muestra un mensaje informativo y finaliza el caso de uso.</p>
Prioridad	Critico
Prototipo	

Tabla 6.3: Descripción del caso de uso Administrar Estudiante

Nombre del Caso de Uso	Administrar Estudiante
Actores	Administrador
Propósito	Permitir registrar, modificar, eliminar y mostrar datos acerca de los estudiantes.
Resumen	El caso de uso se inicia cuando el administrador del sistema selecciona la opción de registrar, modificar, eliminar o mostrar los datos de los estudiantes.
Referencias	R3
Precondiciones	Administrador del sistema previamente autenticado.
Poscondiciones	Se registra un estudiante, se actualizan o muestran los datos o se elimina los mismos.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El administrador del sistema necesita registrar, eliminar, mostrar y modificar los datos de un estudiante.	<p>El sistema ejecuta alguna de las siguientes acciones:</p> <ul style="list-style-type: none"> a) Si decide registrar un estudiante, ir a la sección "Registrar Estudiante" b) Si decide actualizar los datos de un estudiante, ir a la sección "Modificar Datos Estudiante". c) Si decide eliminar un estudiante, ir a la sección "Eliminar Estudiante".

	d) Si decide mostrar datos del estudiante, ir a la sección “Mostar Estudiante”
Sección “Registrar Estudiante”	
Acciones del Actor	Respuesta del Sistema
2. El Administrador del Sistema entra los datos del estudiante para realizar su registro en la aplicación.	<p>2.1 El sistema valida que los campos del Código, el Nombre, el Carné de Identidad, Solapín, email-uci, usuario_procyon, activo y roles estén correctos.</p> <p>2.2 El sistema valida que este estudiante no exista.</p> <p>2.3 El estudiante se almacena en el sistema.</p> <p>2.4 Se muestra un mensaje informándosele al administrador que ya ha sido efectuado el registro del estudiante y finaliza el caso de uso.</p>
Curso alternativo	
	<p>2.1 Se emite un mensaje para que llene los campos obligatorios y verifique los datos.</p> <p>2.2 Si el estudiante existe se muestra un mensaje informativo y finaliza el caso de uso.</p>

<p>Prototipo</p>	
<p>Sección “Modificar Datos Estudiante”</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
<p>3. El administrador del sistema selecciona el estudiante a modificar.</p>	<p>3.1 El sistema brinda la posibilidad de modificar los datos.</p>
<p>4. El administrador del sistema realiza las actualizaciones deseadas.</p>	<p>4.1 El sistema valida que los campos obligatorios estén llenos y correctos. 4.2 Se actualiza la Información y finaliza el caso de uso.</p>
<p>Curso Alternativo</p>	
	<p>4.1 Se emite un mensaje para que llene los campos obligatorios y verifique que los datos sean correctos.</p>

<p>Prototipo</p>	
<p>Sección “Eliminar Estudiante”</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
<p>5. El administrador del sistema selecciona el estudiante a eliminar.</p>	<p>5.1 El sistema pide confirmación. 5.2 El sistema elimina el Estudiante.</p>
<p>Sección “Mostar Estudiante”</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
<p>6. El administrador selecciona la opción de mostrar estudiantes del menú principal.</p>	<p>6.1 El sistema muestra el listado de estudiantes almacenados en el sistema.</p>
<p>7. El administrador selecciona el estudiante del cual desea conocer sus datos.</p>	<p>7.1 El sistema muestra los datos del estudiante seleccionado y finaliza el caso de uso.</p>
<p>Curso Alternativo</p>	

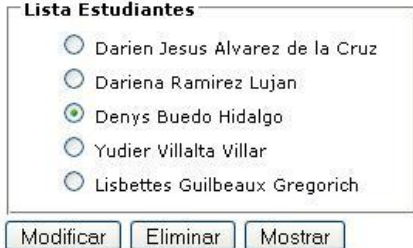
	6.1 Si no existen estudiantes registrados en el sistema se muestra un mensaje informativo y finaliza el caso de uso.
Prioridad	Crítico
Prototipo	

Tabla 6.4: Descripción del caso de uso Mostrar Actividad Estudiante

Nombre del Caso de Uso	Mostrar Actividad Estudiante
Actores	Administrador
Propósito	Permitir Mostar la actividad del estudiante en Internet y en el sistema de control de versiones.
Resumen	El caso de uso se inicia cuando el administrador selecciona la opción de Mostar la actividad del estudiante dentro de la red de PROCYON. Se puede mostrar la actividad del estudiante en el repositorio y en Internet desde la red de PROCYON.
Referencias	R4
Precondiciones	Administrador del sistema ya autenticado.
Poscondiciones	Se muestra la actividad del estudiante en el repositorio o en Internet mediante el servidor Proxy utilizado en PROCYON.

Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El administrador del sistema necesita mostrar las actividades del estudiante dentro de la red de PROCYON.	1.1 El sistema ejecuta alguna de las siguientes acciones: a) Si decide mostrar actividad del estudiante en el repositorio ir a la sección "Mostrar Actividad Repositorio". b) Si decide mostrar actividad del estudiante en Internet ir a la sección "Mostrar Actividad Internet".
Sección "Mostrar Actividad Repositorio"	
Acciones del Actor	Respuesta del Sistema
2 El administrador selecciona la opción de mostrar la actividad en el repositorio de un estudiante.	2.1 El sistema muestra el listado de los estudiantes registrados en el sistema.
3 El administrador selecciona el estudiante del cual desea conocer su actividad en el repositorio.	3.1 El sistema muestra un listado de las actividades del estudiante en el repositorio con sus respectivas fechas de realización.
4 El administrador selecciona la fecha de la actividad de la cual desea conocer sus datos.	4.1 El sistema muestra los datos correspondientes a la actividad en el repositorio del estudiante según la fecha seleccionada y finaliza el caso de uso.
Curso Alternativo	

	<p>2.1 Si no hay estudiantes registrados en el sistema se muestra un mensaje informativo y finaliza el caso de uso.</p> <p>3.1 Si el estudiante no tiene almacenada en el sistema ninguna actividad en el repositorio se emite un mensaje informativo y finaliza el caso de uso.</p>
Sección “Mostrar Actividad Internet”	
Acciones del Actor	Respuesta del Sistema
5 El administrador selecciona la opción de mostrar la actividad en internet de un estudiante.	5.1 El sistema muestra el listado de los estudiantes registrados en el sistema.
6 El administrador selecciona el estudiante del cual desea conocer su actividad en internet dentro de la red de PROCYON.	6.1 El sistema muestra un listado de las actividades del estudiante en Internet con sus respectivas fechas de realización.
7. El administrador selecciona la fecha de la actividad de la cual desea conocer sus datos.	7.1 El sistema muestra los datos correspondientes a la actividad en Internet del estudiante según la fecha seleccionada y finaliza el caso de uso.
Curso Alternativo	


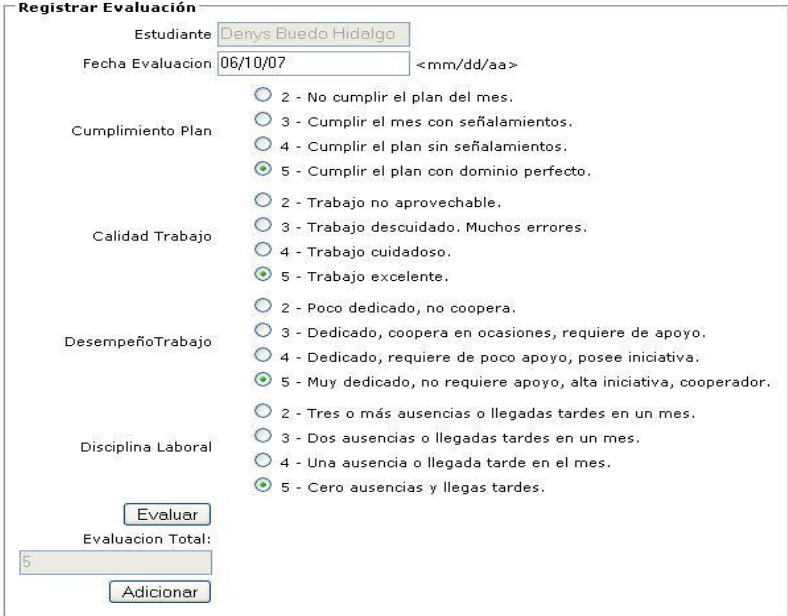

	<p>5.1 Si no hay estudiantes registrados en el sistema se muestra un mensaje informativo y finaliza el caso de uso.</p> <p>6.1 Si el estudiante no tiene almacenada en el sistema ninguna actividad en Internet se emite un mensaje informativo y finaliza el caso de uso.</p>
Prioridad:	Crítico
Prototipo	

Tabla 6.5: Descripción del caso de uso Gestionar Evaluación Estudiante

Nombre del CU	Gestionar Evaluación Estudiante
Actores	Líder _Proyecto
Propósito	Permitir registrar, modificar o mostrar la evaluación del estudiante según el mes seleccionado.
Resumen	El caso de uso se inicia cuando el Jefe de proyecto selecciona la opción de registrar, modificar o mostrar la evaluación sistemática del estudiante, según el mes seleccionado.
Referencias	R5
Precondiciones	Líder de proyecto previamente autenticado en el sistema

Poscondiciones	Se registra, modifica, elimina o muestra la evaluación del estudiante según el mes seleccionado.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Líder de proyecto necesita registrar, modificar y mostrar la evaluación de un estudiante, según el mes seleccionado.	<p>El sistema ejecuta alguna de las siguientes acciones:</p> <p>a) Si decide registrar una nueva evaluación del estudiante, ir a la sección "Registrar Evaluación"</p> <p>b) Si decide actualizar los datos de una evaluación del estudiante, ir a la sección "Modificar Evaluación".</p> <p>c) Si decide mostrar las evaluaciones del estudiante ir a la sección "Mostrar Evaluación"</p>
Sección "Registrar Evaluación"	
Acciones del Actor	Respuesta del Sistema
2. El Jefe de proyecto inserta los datos de la evaluación del estudiante para realizar su registro en la aplicación.	<p>2.1 El sistema valida que los campos de la evaluación del estudiante estén llenos y los datos estén correctos.</p> <p>2.2 El sistema valida que la evaluación del mes seleccionado no exista.</p> <p>2.3 La evaluación del estudiante se almacena en el sistema.</p> <p>2.4 Se muestra un mensaje informándosele Líder del proyecto que ya ha sido efectuado el registro de la evaluación del estudiante y finaliza el caso de uso.</p>
Curso Alternativo	
	2.1 Se emite un mensaje para que llene los campos obligatorios y


	<p>garantice que los datos estén correctos.</p> <p>2.2 Si la evaluación del mes seleccionado está registrada se muestra un mensaje informativo y finaliza el caso de uso.</p>
<p>Prototipo</p>	
<p>Sección “Modificar Evaluación”</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
<p>3. El Líder de proyecto selecciona el estudiante y el mes del cual desea modificar su evaluación.</p>	<p>3.1 El sistema brinda la posibilidad de modificar los datos de la evaluación según el mes seleccionado.</p>
<p>4. El Líder de proyecto realiza las modificaciones deseadas.</p>	<p>4.1 Se valida que los campos obligatorios estén llenos y correctos.</p> <p>4.2 Se actualiza la Información y finaliza el caso de uso.</p>
<p style="text-align: center;">Curso Alternativo</p>	

	4.1 Se emite un mensaje para que llene los campos obligatorios y garantice que los datos estén correctos.
Prototipo	
Sección “Mostrar Evaluación”	
Acciones del Actor	Respuesta del Sistema
5. El Líder de proyecto selecciona la opción de mostrar las evaluaciones de un determinado estudiante.	5.1 El sistema muestra el listado de los estudiantes registrados en el sistema.
6. El Líder de Proyecto selecciona el estudiante del cual desea conocer su evaluación.	6.1 El sistema muestra un listado de las evaluaciones que tiene registradas el estudiante, con sus respectivas fechas de registro.
7. El Líder de proyecto selecciona la fecha correspondiente a la evaluación que le interesa.	7.1 El sistema muestra los datos de la evaluación del estudiante según la fecha seleccionado y finaliza el caso de uso.

Curso Alternativo	
	6.1 Si el estudiante no tiene evaluaciones registradas se emite un mensaje informativo y finaliza el caso de uso.
Prioridad	Secundario
Prototipo	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Mostrar Evaluación Estudiante</p> <p style="text-align: center;">Estudiante : Denys Buedo Hidalgo</p> <p style="text-align: center;">Fecha Evaluacion : 2007-10-10</p> <p style="text-align: center;">Cumplimiento Plan : 3</p> <p style="text-align: center;">Calidad Trabajo : 4</p> <p style="text-align: center;">Desempeño Trabajo : 3</p> <p style="text-align: center;">Disciplina Laboral : 5</p> <p style="text-align: center;">Evaluacion Total : 4</p> </div>

Tabla 6.6: Descripción del caso de uso Gestionar Tiempo


Nombre del CU	Gestionar Tiempo
Actores	Estudiante (inicia). Registra su tiempo y las actividades que realiza. Líder de Proyecto. Puede ver los registros de tiempo de los estudiantes.
Propósito	Permitir registrar y mostrar el tiempo que dedica un estudiante a realizar las actividades en PROCYON.
Resumen	El caso de uso se inicia cuando el Estudiante selecciona del menú principal la opción de Registrar o Mostrar Tiempo, donde el estudiante registra o muestra las actividades y el tiempo que dedica a las mismas.
Referencias	R6, R7
Precondiciones	Estudiante previamente autenticado en el sistema
Poscondiciones	Se registran o se muestran las actividades y el tiempo de trabajo de un determinado estudiante.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El estudiante necesita registrar, o mostrar las actividades y el tiempo que	El sistema ejecuta alguna de las siguientes acciones: a) Si decide insertar un nuevo registro de tiempo, ir a la sección "Registrar Tiempo"

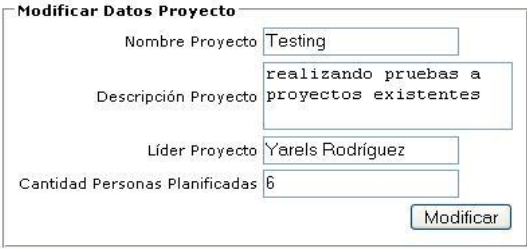
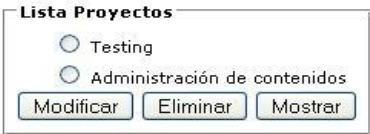
<p>dedica a las mismas en el marco laboral, según la fecha seleccionada.</p>	<p>b) Si decide mostrar algún registro de tiempo ir a la sección “Mostrar Registro Tiempo”</p>
<p>Sección “Registrar Tiempo”</p>	
<p>2. El Estudiante introduce los datos relacionados con la actividad que realizó y el tiempo que dedicó a la actividad.</p>	<p>2.1 El sistema verifica que los campos de Proyecto Asociado, Actividad Realizada, Fecha de la Actividad, Hora de Inicio, Hora de Fin, Descripción estén llenos y correctos.</p> <p>2.2 El sistema almacena la información correspondiente a la actividad planificada y finaliza el caso de uso.</p>
<p>Curso Alternativo</p>	
<p>.</p>	<p>2.1 Se emite un mensaje para que llene los campos obligatorios y verifique que los datos estén correctos.</p> <p>.</p>
<p>Prototipo</p>	
<p>Sección “Mostrar Registro Tiempo”</p>	
<p>3. El Estudiante o el Jefe de Proyecto, selecciona la opción de mostrar los registros de tiempos de un estudiante.</p>	<p>3.1 El sistema muestra un listado con los registros de tiempo que posee el estudiante y las fechas correspondientes.</p>
<p>4. El Estudiante o el Jefe de Proyecto, selecciona la fecha del registro para conocer toda la información al respecto.</p>	<p>4.1 Se muestra la información correspondiente al registro del tiempo dedicado a las actividades y finaliza el caso de uso.</p>

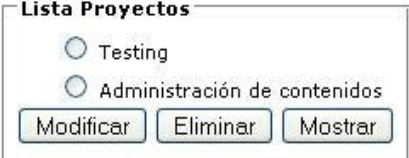
Curso Alternativo	
	3.1 Se emite un mensaje informando que el estudiante no posee ningún registro de tiempo y finaliza el caso de uso.
Prioridad	Crítico
Prototipo	

Tabla 6.7: Descripción del caso de uso Administrar Información Proyecto

Nombre del CU	Administrar Información Proyecto
Actores	Líder de Proyecto.
Propósito	Permitir registrar, eliminar, modificar y mostrar datos acerca de un determinado proyecto.
Resumen	El caso de uso se inicia cuando el Líder de Proyecto selecciona del menú principal, la opción de registrar, modificar, eliminar o mostrar un determinado proyecto.
Referencias	R8
Precondiciones	Líder de Proyecto previamente autenticado en el sistema.
Poscondiciones	Se registra, modifica, muestra o elimina algún proyecto.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Líder de Proyecto necesita registrar, eliminar, modificar o mostrar los datos de algún proyecto.	1.1. El sistema ejecuta alguna de las siguientes acciones: <ul style="list-style-type: none"> a) Si decide registrar un nuevo proyecto, ir a la sección "Registrar Proyecto" b) Si decide actualizar los datos de un determinado proyecto, ir a la sección "Modificar Proyecto". <ul style="list-style-type: none"> c) Si decide eliminar los datos de un proyecto, ir a la sección "Eliminar Proyecto". d) Si decide listar los datos de un determinado proyecto ir a la sección "Mostrar Proyecto"

Sección “Registrar Proyecto”	
2. El Líder de Proyecto inserta los datos del proyecto para realizar su registro en la aplicación.	<p>2.1 El sistema valida que los campos del nombre_proyecto, descripción_proyecto, líder_proyecto, cantidad_personas estén llenos.</p> <p>2.2 El nuevo proyecto se almacena en el sistema.</p> <p>2.3 Se muestra un mensaje informándosele al Líder de Proyecto que ya ha sido efectuado el registro del nuevo proyecto y finaliza el caso de uso.</p>
Curso Alterno	
	2.1 Se emite un mensaje para que llene los campos obligatorios y verifique que los datos estén correctos.
Prototipo	
Sección “Modificar Proyecto”	
Acciones del Actor	Respuesta del Sistema
3. El Líder de Proyecto selecciona el proyecto del cual desea modificar algún dato.	3.1 El sistema brinda la posibilidad de modificar los datos del proyecto seleccionado.
4. El Líder de Proyecto realiza las modificaciones deseadas.	<p>4.1 Se verifica que los campos obligatorios estén llenos y correctos.</p> <p>4.2 Se modifica la información y finaliza el caso de uso.</p>
Curso Alterno	
	4.1 Se emite un mensaje para que llene los campos obligatorios y verifique que los datos estén correctos.

<p>Prototipo</p>	
<p>Sección “Eliminar Proyecto”</p>	
<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
<p>5. El Jefe de Proyecto selecciona el proyecto que desea eliminar.</p>	<p>5.1 Es sistema pide confirmación. 5.2 El sistema elimina el proyecto seleccionado.</p>
<p>Prototipo</p>	
<p>Sección “Mostrar Proyecto”</p>	
<p>Acciones del Actor</p>	<p>Respuesta del Sistema</p>
<p>6. El Jefe de Proyecto selecciona la opción de mostrar los proyectos existentes, del menú principal.</p>	<p>6.1 El muestra la lista de proyectos almacenados en el sistema.</p>
<p>7. El administrador selecciona el proyecto del cual desea conocer sus datos.</p>	<p>7.1 El sistema muestra los datos del proyecto seleccionado y finaliza el caso de uso.</p>
<p>Curso Alternativo</p>	

	6.1 Si no existen proyectos registrados se muestra un mensaje informativo y finaliza el caso de uso.
Prioridad	Secundario
Prototipo	

Anexo 7
Fig. 7.1 Prototipo funcional del sistema.

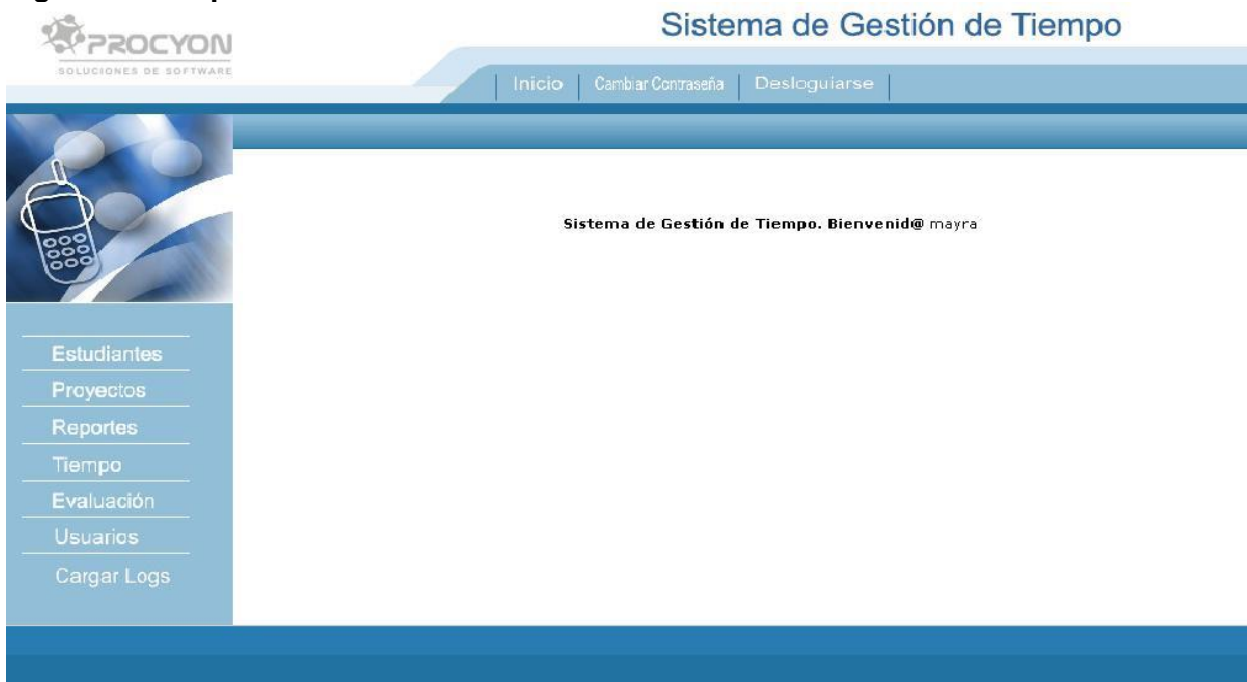


Fig. 7.2 Reportes del sistema.

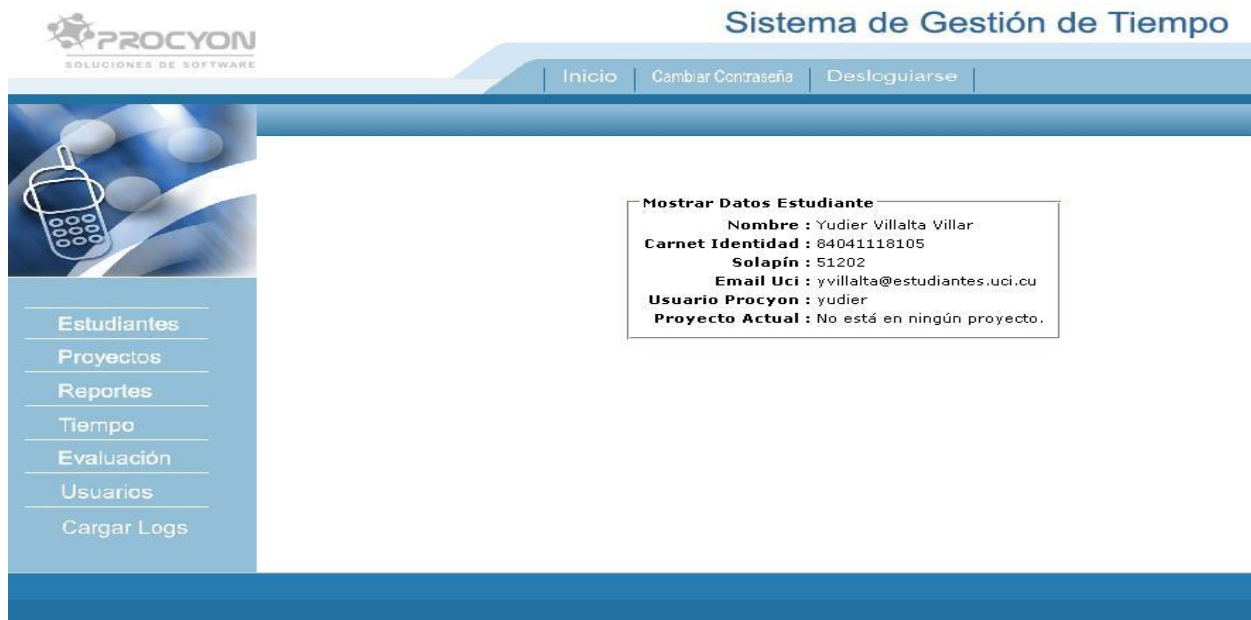


Fig. 7.3 Componentes utilizados.

Fig. 7.3.1 Componente Select One.

Lista Proyectos

Testing

Administración de contenidos

Fig. 7.3.2 Componente Select Many. (Atributo “roles”)

Registrar Nuevo Estudiante

Nombre:

Carnet Identidad:

Solapín:

Email Uci:

Proyecto a que Pertenece: Ninguno

Usuario Procyon:

Descripción:

Activo

Roles: Estudiante
 Lider de Proyecto

Fig. 7.4 Mensajes de errores.

Sistema de Gestión de Tiempo

Inicio | Cambiar Contraseña | Desloguarse



- Estudiantes
- Proyectos
- Reportes
- Tiempo
- Evaluación
- Usuarios
- Cargar Logs

Registrar Nuevo Estudiante

Nombre:

Carnet Identidad: El valor no alcanza el mínimo permitido.

Solapín: El valor no alcanza el mínimo permitido.

Email Uci:

Proyecto a que Pertenece: Ninguno

Usuario Procyon: Ya existe un estudiante con ese usuario

Descripción:

Activo

Roles: Estudiante
 Lider de Proyecto

Anexo 8

Glosario de términos y siglas.

- **ACL:** Access Control List. Permite controlar el flujo del tráfico en equipos de redes, tales como routers y switches. Su principal objetivo es filtrar tráfico, permitiendo o denegando el tráfico de red de acuerdo a alguna condición.
- **Apache Software Foundation:** Organización no lucrativa (en concreto una, fundación) creada para dar soporte a los proyectos de software bajo la denominación *Apache*, incluyendo el popular servidor HTTP Apache.
- **Applet:** Componente de software que corre en el contexto de otro programa, por ejemplo un navegador web.
- **Active Directory:** Nombre utilizado por Microsoft para referirse a su implementación de seguridad en una red distribuida de computadores.
- **API:** Application Programming Interface - Interfaz de Programación de Aplicaciones- Conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.
- **BSD:** Berkeley Software Distribution.
- **Código abierto:** Término con el que se conoce al software distribuido y desarrollado libremente.
- **Compilación:** Proceso por el cual se traducen programas en código fuente a programas en código objeto. El programa que realiza esta traducción se llama compilador.
- **CGLIB:** Librería de generación de código Java muy poderosa y de alto rendimiento.
- **EJB:** Enterprise JavaBeans. Son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems. Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJBs.
- **Framework:** Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Glosario de Términos y Siglas

- **FTP:** File Transfer Protocol. Protocolo de transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor, de manera que desde un equipo cliente nos podemos conectar a un servidor para descargar ficheros desde él o para enviarle nuestros propios archivos independientemente del sistema operativo utilizado en cada equipo.
- **GOPHER:** Servicio de Internet consistente en el acceso a la información a través de menús. La información se organiza de forma arborescente: sólo los nodos contienen menús de acceso a otros menús o a hojas, mientras que las hojas contienen simplemente información textual. En cierto modo es un predecesor de la Web, aunque sólo se permiten enlaces desde nodos-menús hasta otros nodos-menús o a hojas, y las hojas no tienen ningún tipo de hiperenlaces.
- **HTML:** HyperText Markup Language. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986.
- **HTTP:** HyperText Transfer Protocol. Protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido.
- **HQL:** Hibernate Query Language. Es una extensión orientada a objetos de SQL. HQL permite acceder a los datos de varias formas, incluyendo consultas orientadas a objetos
- **IDE:** Entorno integrado de desarrollo.
- **ISAServer:** Microsoft Internet Security and Acceleration (ISA) Server 2006 es la solución ideal para la optimización y el correcto aprovechamiento de los recursos de Tecnología de Información (TI) en sus oficinas sucursales.
- **ICID:** Instituto Central de Investigación Digital.
- **ISAWEB:** Software desarrollado en el ICID que facilita el seguimiento al acceso a Internet mediante la lectura de logs.
- **J2EE:** Arquitectura por si misma que involucra otras arquitecturas, incluyendo servlets, JavaServer Pages, Enterprise JavaBeans, y más, merece su propio conjunto de patrones específicos para diferentes aplicaciones empresariales.
- **JDBC:** Java Database Connectivity. API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema de operación donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

Glosario de Términos y Siglas

- **Java EE:** Java Platform Enterprise Edition Plataforma de programación parte de la Plataforma Java, para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de n niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una *especificación*.
- **Java Community Process:** Proceso de la Comunidad Java. Es un proceso formalizado el cual permite a las partes interesadas a involucrarse en la definición de futuras versiones y características de la plataforma Java.
- **Logs:** Registro de actividad de un sistema, que generalmente se guarda en un fichero de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema.
- **Log de visitas:** Archivo creado por el servidor donde se registran las acciones que los usuarios generan en la web. El formato de los logs varía dependiendo del servidor Web.
- **LDAP:** Lightweight Directory Access Protocol. Protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas.
- **Lenguajes de Alto Nivel:** Expresan los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de a la capacidad ejecutora de las máquinas.
- **Navegador web** Aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet.
- **Plugin:** Aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica.
- **Protocolo:** Conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red.
- **PostgreSQL:** es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.
- **Patrones de arquitectura:** Expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Ayudan además a especificar la estructura fundamental de una aplicación.

Glosario de Términos y Siglas

- **Red de Computadoras:** Conjunto de computadoras y/o dispositivos conectados por enlaces de un medio físico (medios guiados) ó inalámbricos (medios no guiados) y que comparten información (archivos), recursos (CD-ROM, impresoras, etc.) y servicios (e-mail, chat, juegos), etc.
- **Spring Framework:** También conocido simplemente como Spring es un Framework Open Source de desarrollo de aplicaciones para la plataforma Java. .
- **Software:** Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.
- **SSL:** Secure Socket Layer. El protocolo SSL es un sistema diseñado y propuesto por Netscape Communications Corporation. Se encuentra en la pila OSI entre los niveles de TCP/IP y de los protocolos HTTP, FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA.
- **SQL:** Structured Query Language. Es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla.
- **Servlets:** Objetos que corren dentro del contexto de un contenedor de servlets (ej.: Tomcat) y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones.
- **UML:** Unified Modeling Language. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.
- **URL:** Uniform Resource Locator (localizador uniforme de recurso). Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
- **UNIX:** Sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.
- **WAIS:** Wide Area Information Servers es un sistema de búsqueda de texto distribuido que usa el protocolo standart cliente servidor ANSI Z39.50 para buscar bases de datos indexadas en ordenadores remotos. WAIS permite a los usuarios descubrir la información y resuelve el acceso a la información en la red sin tener en cuenta su ubicación física.