

Universidad de las Ciencias Informáticas
Facultad 15



Trabajo de Diploma

Herramienta de control, planificación y evaluación de profesores para departamentos docentes.

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

Autor: Mabel Díaz Nápoles

Tutor: Ing. Yosvany D. Medina Hernández

La Habana
Junio de 2010



Quien tenga una computadora dispone de todos los conocimientos publicados. La privilegiada memoria de la máquina le pertenece también a él.

Fidel Castro Ruz

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 _____.

Mabel Díaz Nápoles

Ing. Yosvany D. Medina Hernández

Firma del Autor

Firma del Tutor

Agradecimientos

A nuestro Comandante en Jefe por haber soñado y realizado este gran proyecto.

A mis padres Teresita y Pedro, por estar siempre a mi lado y guiarme por el camino correcto. Los quiero mucho y este logro se lo debo a ustedes.

A mi abuela Teresa, por su amor y dedicación, eres la mejor del mundo.

A mi ti lindo, por estar a mi lado estos 5 años y demostrarme que el amor existe. Por todo el empeño que ha puesto en mi carrera, dándome siempre sus sabios consejos. Te quiero mucho, mucho, mucho, del tamaño del cielo.

A mi hermana Maypú, por su apoyo incondicional y por darme la sobrina mas linda del mundo. Las quiero mucho a las dos.

A mi tía Esperanza, por ser tan especial y brindarme siempre su cariño.

A Papón, por su cariño y dedicación.

A mi Abuelito Alfonso, por su ternura y comprensión.

A toda mi familia: Mis primas, hermanos, tíos, sobrinas; por estar siempre pendientes de mi carrera y ofrecerme su comprensión y afecto en todo momento.

A mis suegros Isel y Medina, Por todo el cariño que me brindaron desde el comienzo y por hacerme sentir como una hija más. Los quiero mucho.

A mis amigas de toda la vida: Lianet, Yudith, Yanela, Liyixy y Lisbet, gracias por su amistad incondicional, nunca me olvido de ustedes.

A mis vecinos, en especial a Haydee, Alicia, Niriam, Anakarla y Tania, por todo su apoyo y cariño. Ustedes también forman parte de mi familia.

A Yoinet, por ser un gran amigo, brindarme su ayuda en innumerables ocasiones y por los buenos momentos que hemos compartido juntos.

A Yoel, por ser el mejor compañero de estudios y por brindarme su amistad.

A todos los que han compartido conmigo en estos 5 años, en especial a Lismary y Yusbel.

Resumen

En la Universidad de las Ciencias Informáticas desde el curso 2006 – 2007 se implantó un sistema de estimulación salarial según el desempeño de cada recurso humano, el mismo está asociado a la evaluación que recibe el trabajador en un período de tiempo. Hoy en día, la obtención de esta valoración, se realiza de forma manual, lo que afecta la integración de la misma con el plan de resultados, así como el control y gestión de su cumplimiento. Las facultades y por ende, los departamentos, son áreas donde los profesores realizan disímiles actividades, que complejizan aún más el proceso. Con el desarrollo de esta investigación se obtuvo una herramienta que permite la gestión de ambos mecanismos de forma práctica e íntegra, permitiendo a los evaluadores tener toda la información necesaria y las tareas realizadas, a la hora de efectuar la evaluación.

Palabras claves: Evaluación de Desempeño, Plan de Resultados, Desempeño, Resultados.

Índice

Agradecimientos	4
Introducción	8
1. Capítulo 1: Fundamentación Teórica	11
1.1. Introducción	11
1.2. Sistemas para la gestión y planificación de tareas	11
1.2.1. Características de Dotproject	12
1.2.2. Características de Phprojekt.....	12
1.2.3. Redmine.....	13
1.2.4. Microsoft Project Professional 2010	14
1.3. Plan de resultado	14
1.4. Evaluación del Desempeño.....	15
1.4.1. Indicadores a cumplir para recibir el pago adicional	16
1.4.2. Herramientas existentes para la evaluación de desempeño	17
1.5. Framework para desarrollo web con PHP	18
1.5.1. Definición de Framework:.....	18
1.5.2. Zend Framework	19
1.5.3. Symfony	19
1.5.4. CakePHP	20
1.5.5. Codeigniter	21
1.5.6. Fundamentación del framework PHP.	23
1.6. Framework para interfaz de aplicaciones web.....	23
1.6.1. Librería JQuery.....	24
1.6.2. Librería Dojo Toolkit	25
1.6.3. Librería Interfaz de Usuario de Yahoo (YUI)	25
1.6.4. Librería ExtJS.....	26
1.6.5. Fundamentación de la librería JavaScript a utilizar.....	27
1.7. Metodologías de desarrollo del software	27
1.7.1. Programación Extrema (XP).....	28
1.7.2. Proceso Unificado del Software (RUP).....	29
1.8. Tecnologías de servidores	30
1.8.1. Sistemas Gestores de Base de Datos (SGBD).....	30
1.8.2. Servidor Web	31
1.9. Conclusiones parciales	31

2. Capítulo 2: Características del sistema	32
2.1. Introducción	32
2.2. Modelo de dominio	32
2.3. Levantamiento de Requisitos	33
2.3.1. Requerimientos Funcionales	34
2.3.2. Requerimientos No Funcionales.....	37
2.4. Descripción del Sistema propuesto	37
2.4.1. Descripción de los actores del sistema.....	39
2.4.2. Estructura lógica de la aplicación	40
2.4.3. Casos de Uso del Sistema	41
2.5. Conclusiones parciales	59
3. CAPÍTULO 3: Construcción de la Solución propuesta.....	61
3.1. Introducción	61
3.2. Modelo de análisis	61
3.2.1. Clases de análisis	61
3.2.2. Diagrama de clases de análisis	61
3.3. Modelo de diseño.....	63
3.3.1. Diagrama de clases del diseño.....	63
3.3.2. Descripción de las principales clases del sistema.	66
3.3.3. Diseño de la Base de Datos (BD).....	68
3.3.4. Descripción de las principales tablas.....	69
3.4. Implementación.....	71
3.4.1. Modelo de Implementación	71
3.4.2. Diagrama de despliegue.....	73
3.5. Conclusiones Parciales	74
Conclusiones Generales	75
Recomendaciones	76
Bibliografía.....	77
Glosario de Términos.....	79

Introducción

La Universidad de las Ciencias Informáticas (UCI) está integrada por varias facultades, donde cada una de ellas cuenta con una estructura administrativa formada por: un decano, 3 vicedecanos (vicedecano de formación, vicedecano de investigación y postgrado y vicedecano de extensión universitaria) y departamentos, que a su vez cuentan con una estructura organizativa interna integrada por un jefe de departamento y jefes de asignatura.

En cada uno de los departamentos los profesores elaboran sus planes de resultados en conjunto con la dirección del mismo, éstos deben tener en cuenta los intereses personales de sus recursos humanos, los objetivos y estrategias de la facultad, así como los del departamento, en específico, la asignatura y disciplina a la que pertenecen. El plan, aún carece de la calidad a la que aspira la universidad y resulta tedioso a la hora de ser elaborado por los profesores, así como a veces complejo y nuevo para ellos en el caso de los adiestrados e Instructores recién graduados.

Cada profesor es evaluado trimestralmente de acuerdo a su desempeño laboral. Dicha evaluación es gestionada por el jefe de departamento en dependencia del cumplimiento del plan de resultados previamente elaborado. Los tipos de calificación que se establecen para una evaluación del trabajo y desempeño de los profesores son: deficiente, adecuado y superior. Cada área con previa aprobación de la rectoría de la UCI, de común acuerdo con la organización sindical, define cuales son los indicadores y resultados para obtener una de estas calificaciones. (MIC 2009)

En la actualidad este proceso de evaluación se torna un tanto complejo y muchas veces trivial, debido a la gran cantidad de tareas y actividades realizadas por los profesores. Cada tarea tributa a un resultado sobre una esfera determinada: Docencia, Investigación y Postgrado, Producción y Extensión, y es reflejado en el plan de resultados de cada evaluado.

El proceso como entorno único es difícil de concertar, sin dejar de destacar que es complicado darle seguimiento al cumplimiento de los resultados de cada profesor, es válido señalar el apoyo de algunas herramientas del paquete Office, así como pequeños sistemas personalizados para diferentes áreas que garantizan un alivio en el trabajo,

estas aplicaciones presentan poca integración entre sí y provocan un ineficiente acceso a los datos requeridos trayendo como consecuencia pérdida de tiempo. Muchas veces todos estos mecanismos no se gestionan de la mejor forma, por las dificultades mencionadas, y esto provoca que los resultados no sean los más prácticos y esperados. Constituye una necesidad el desarrollo de un sistema que automatice e integre los procesos de evaluación y gestión del plan de resultados, además de que brinde una solución orientada a la optimización del trabajo dentro del departamento, mediante el control, seguimiento y evaluación del nivel de cumplimiento de las tareas de cada uno de los profesores, obteniendo de esta forma, una elevada colaboración y eficiencia.

Dadas las dificultades antes mencionadas se plantea como **Problema a resolver:**

¿Cómo gestionar e integrar de forma práctica los planes de resultados y la evaluación de desempeño de los profesores en los departamentos docentes?

El **objeto de estudio** del trabajo radica en los procesos que se realizan en los departamentos docentes relacionados con los recursos humanos. Y el **campo de acción** es los procesos de planificación y control de resultados de los profesores, así como la evaluación de los mismos en los departamentos docentes.

Por tanto el **Objetivo General** de este trabajo es implementar una herramienta que permita gestionar e integrar el plan de resultados con la evaluación de desempeño de los profesores.

Para dar cumplimiento al objetivo propuesto se concibieron las siguientes **tareas de la Investigación:**

- ▶ Elaboración del marco teórico para planificar, organizar y ejecutar la investigación.
- ▶ Investigación de los diferentes procesos de evaluación de desempeño y plan de resultados en cada facultad para conocer la realidad de las mismas en cuanto a la situación planteada.
- ▶ Realización del estado del arte de las aplicaciones de apoyo a la gestión de la evaluación de desempeño y de tareas, existentes en las facultades, en otras universidades y en el mundo, para conocer si existe alguna solución parcial o completa del problema.

- ▶ Identificación de requerimientos comunes a partir de las soluciones encontradas y que respondan al objetivo general, para tener un punto de partida hacia el desarrollo de la aplicación.
- ▶ Realización del análisis y diseño de la aplicación para poder obtener un conjunto de artefactos que faciliten la implementación
- ▶ Implementación de la aplicación teniendo en cuenta los requerimientos identificados para obtener una solución acorde a la problemática.

Hipótesis:

Si se implementa una herramienta para automatizar el plan de resultados en conjunto con la evaluación de desempeño de los profesores en los departamentos docentes, entonces se logrará la gestión e integración de estos procesos de forma práctica.

El contenido del presente trabajo se estructura en 3 capítulos:

Capítulo 1. Fundamentación Teórica: Se exponen los fundamentos generales que sirven de soporte teórico en la solución del problema. Se analizan las herramientas y lenguajes de programación idóneas para el desarrollo de la herramienta de evaluación de desempeño. Además se plantea la metodología a emplear en el desarrollo del mismo y se realiza el análisis de algunos sistemas similares a éste, tanto a nivel internacional como nacional.

Capítulo 2. Características del sistema: Se abordará sobre todo lo referente al sistema, realizando una propuesta que permitirá explicar y justificar la definición de los actores y casos de uso del mismo, así como una exposición de los requerimientos funcionales y no funcionales respectivamente.

Capítulo 3. Análisis, diseño e implementación del sistema: En este capítulo se modelan los principales artefactos que se obtienen como resultado de los flujos de trabajo de análisis, diseño e implementación. En la fase de análisis se muestran los diagramas de clases de análisis, y en la fase de diseño se obtienen los diagramas de clases de diseño. Además contiene los diagramas de despliegue y de componente, así como una breve descripción de estos conceptos.

1. Capítulo 1: Fundamentación Teórica

1.1. Introducción

Con el objetivo de encontrar una solución o los elementos necesarios para desarrollar la misma, en el presente capítulo se realiza un análisis detallado de los principales conceptos relacionados con los sistemas de gestión y planificación de tareas, el plan de resultados y la evaluación de desempeño, además de tendencias actuales y sistemas informáticos para este fin. Se fundamenta la selección de las herramientas y tecnologías que se utilizan para el modelado, diseño e implementación de la aplicación.

1.2. Sistemas para la gestión y planificación de tareas

Para la solución al problema planteado nos enfocaremos en los sistemas de gestión. A nivel mundial se conocen varios sistemas para la gestión y planificación, a continuación se analizan algunos de ellos, pero primeramente se citarán definiciones referentes al tema:

Según la definición del grupo BSI, que dentro de sus funcionalidades certifica los sistemas de gestión y productos relacionados, un sistema de gestión es:

Una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de una organización. Un sistema de gestión ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado. (GROUP 2010)

Otra definición sería:

Los sistemas de gestión hacen referencia a todos aquellos sistemas informáticos con los que se apoya una empresa para poder desarrollar correctamente las actividades de la misma, (GESTIÓN 2009)

Dotproject y Phprojekt ambos son open source o de código abierto y permiten modelar y adaptarse a las exigencias del cliente.

1.2.1. Características de Dotproject

- ▶ Permite la gestión y planificación de proyectos en entornos colaborativos: Basado en plataforma Web permite la participación online de los miembros de un proyecto.
- ▶ Permite la gestión de varios proyectos en el mismo ambiente y con distintas compañías.
- ▶ Esta característica lo hace apto para ser implementado en oficinas de proyectos, donde se gestionan varios proyectos para distintas empresas y con muchos trabajadores en proyectos distintos o compartidos.
- ▶ Permite la asignación de recursos (equipamientos, mobiliario...) a un proyecto o varios, así como la descomposición en tareas.
- ▶ Permite el almacenamiento de archivos (documentos o aplicaciones), versionados, por proyectos, tareas y categoría.
- ▶ Permite la gestión de tickets¹
- ▶ Permite la configuración de una agenda de contactos.
- ▶ Permite la visualización de informes y estadísticas sobre los proyectos registrados, como por ejemplo: Las horas asignadas (por usuario o proyecto) para un período de tiempo. Las horas asignadas y las realmente incurridas, para poder extraer porcentajes de trabajos realizados y porcentajes de eficiencia en base a tareas completadas. (DOTPROJECT 2009)

1.2.2. Características de Phprojekt

- ▶ Es un sistema modular para facilitar su ampliación.
- ▶ Gestión básica de proyectos, definiendo tareas y subtareas, con la posibilidad de implementar árboles de tiempo, limitaciones, estadísticas, etc.
- ▶ Accesos LDAP (Lightweight Directory Access Protocol) a usuarios y los contactos.
- ▶ Calendarios de usuarios y grupos para coordinar diferentes actividades.
- ▶ Soporta Time Card, de esta forma se podrá controlar la cantidad de horas de los integrantes y grupos, con posibilidad de ver estadísticas.
- ▶ El contacto vía eMail es fundamental, un forma de comunicación asíncrono y no presencial, es por ello que cuenta con un cliente de eMail integrado, en donde podremos redactar y recibir eMail.
- ▶ Gestiona Archivos, notas y marcadores.
- ▶ Gestión de encuestas o votaciones.(PHPROJEKT 2009)

Podemos apreciar, según las características de estas herramientas, que las mismas presentan gran complejidad operacional, por lo que se requiere de un alto dominio de su configuración, exigiendo así que los usuarios tengan un nivel de capacitación elevado. Además están enfocadas a la gestión de grandes proyectos y no a un ámbito docente. Presentan un componente sobre la administración de recursos humanos, pobre y poco personalizable por ejemplo, en ocasiones se necesita manejar atributos específicos de los profesores, como: Cargo, Grupo asignado, Asignatura que imparte, si es interno o externo, entre otros, que estos sistemas no permiten gestionar.

1.2.3. Redmine

Es una aplicación web destinada a soportar la gestión de proyectos de Software. Está desarrollada utilizando el framework Ruby on Rails. Es multiplataforma y soporta múltiples motores de bases de datos relacionales. Redmine soporta características tales como: soporte a múltiples proyectos, sistema de seguimiento a errores de software (bugs), publicación de noticias, documentos, foros, seguimiento al tiempo, entre otras. (REDMINE 2009)

Características de Redmine

- ▶ Soporte para manejar varios proyectos.
- ▶ Control de accesos basado en roles.
- ▶ Sistema de seguimiento (tracking).
- ▶ Calendario y diagrama de Gantt.
- ▶ Administrador de novedades, documentos y archivos.
- ▶ Notificaciones mediante RSS² y correo electrónico.
- ▶ Foros por proyecto.
- ▶ Seguimiento de tiempos.
- ▶ Integración con herramientas de gestión de versiones (SVN, CVS, Git, Mercurial, Bazaar y Darcs).
- ▶ Creación de peticiones vía correo.
- ▶ Soporte de autenticación LDAP.
- ▶ Aplicación Multilenguaje.
- ▶ Autoregistro por parte del usuario.
- ▶ Soporte de varias bases de datos.

Redmine, al igual que las dos herramientas anteriores, está enfocada al ámbito productivo y su configuración es netamente basada en proyectos o productos en desarrollo. No permite gestionar y personalizar un plan de trabajo a partir de resultados, solo trabaja con asignación de tareas y actividades independientes y por tanto no genera una evaluación de desempeño acorde a lo planificado.

1.2.4. Microsoft Project Professional 2010

Ofrece una forma potente de administrar una amplia gama de proyectos y de programas eficazmente. Mediante una experiencia novedosa e intuitiva, esta solución proporciona las herramientas de planificación, administración y colaboración empresarial, de personas y de equipos, necesaria para cumplir con los plazos de entrega cruciales o elegir los recursos adecuados para un equipo, entre otros objetivos. (MICROSOFT 2010)

Una de las desventajas en el uso de esta herramienta es que, contiene un número de características avanzadas que pueden causar problemas, a un usuario normal, o sea que no es una herramienta intuitiva que requiera poca capacitación o poca experiencia en el ámbito informático. Además es relativamente cara en comparación con otros productos libres. Está enfocada a la gestión de grandes proyectos, no contiene componentes destinados a la generación de la evaluación de desempeño y trabaja a partir de tareas las cuales no pueden ser asociadas a resultados o indicadores previamente planificados.

Estas aplicaciones a pesar de no tener en su totalidad las características que se desean, constituyen un punto de partida para la definición del diseño del sistema propuesto, ya que en ellas se identifican funcionalidades o requerimientos útiles, para el desarrollo de algunos componentes deseados, como sería lo referente a la planificación de actividades y tareas.

1.3. Plan de resultado

El Plan de resultados es una terminología manejada desde la universidad, que mundialmente se conoce como plan de trabajo y es por ello que se tratarán varios conceptos del segundo para conformar una idea general y ajustarla a nuestro dominio en particular.

Según el Centro Internacional de Investigaciones para el Desarrollo, el plan de trabajo no es más que:

Un documento que detalla los recursos y la metodología que se utilizarán en la realización de una evaluación. (IDRC 2007)

Plan de trabajo: Programación de las actividades a llevar a cabo durante la fase de desarrollo del estudio de evaluación, en la que se recogen las etapas a seguir, sus plazos y los productos a entregar como resultado de cada una de ellas. El Plan de trabajo surge de la negociación y acuerdo entre los gestores de la evaluación y los evaluadores. (NAVARRA 2008)

Teniendo en cuenta las definiciones antes mencionadas podemos decir que el plan de trabajo tiene grandes similitudes con el plan de resultados, que no es más que un artefacto que permite ordenar y sistematizar información relevante para realizar un trabajo. Se puede decir además que es un instrumento de planificación, que establece un cronograma, designa a los responsables y marca metas y objetivos.

Los resultados que aparecen incluidos dentro del plan de resultados responden a diferentes esferas y su cumplimiento puede evaluado por el jefe de departamento. El plan de resultados suele ser válido para un determinado período de tiempo. De esta manera, las acciones que propone deben desarrollarse en un cierto plazo y los objetivos tienen que ser cumplidos antes de una fecha límite. Al concluir un plan de resultados (que puede ser mensual, trimestral o anual), éste se reemplaza por uno nuevo.

1.4. Evaluación del Desempeño

Se puede hablar de evaluación del desempeño como un proceso sistemático y periódico de estimación cuantitativa y cualitativa del grado de eficacia con el que las personas llevan a cabo las actividades y responsabilidades de los puestos que desarrollan. (HAYA 2009)

El sistema de evaluación del desempeño facilita la toma de decisiones, no sólo administrativas que afectan a los trabajadores, sino también centradas en el progreso e investigación organizacional. Se trata de una serie de técnicas que ayudan a evaluar cómo el personal pone en práctica los conocimientos y experiencias adquiridas en su puesto laboral.

En la Universidad de las Ciencias Informáticas la evaluación de desempeño es un componente para el pago adicional, que se le da a cada trabajador según su evaluación trimestral. Este pago adicional según la Resolución 116 del 2009 constituye un pago en cuantía fija, adicional al salario del cargo que se desempeña, a partir de los indicadores establecidos. Tiene como objetivos:

- ▶ Captar especialistas con una elevada formación docente-productiva en las líneas de trabajo del centro.
- ▶ Incentivar la participación de los profesores en la actividad productiva y de especialistas de la producción en la actividad docente.
- ▶ Elevar los resultados individuales y colectivos de la gestión y los indicadores de eficiencia y disciplina en todas las áreas.
- ▶ Contribuir a garantizar la estabilidad de los trabajadores y la conformación de un claustro propio.
- ▶ Elevar los ingresos personales de los trabajadores como reconocimiento a la importancia económica social de la labor que desarrollan. (MIC 2009)

1.4.1. Indicadores a cumplir para recibir el pago adicional

- ▶ Tener no menos de 3 meses de permanencia en el centro.
- ▶ Obtener una evaluación de los resultados del trabajo y desempeño con calificación de superior o adecuado, en el período evaluado.
- ▶ No tener ausencias al trabajo y nomas de una llegada tarde en el mes que se evalúe. En el caso de trabajadores que hayan tenido ausencias, llegadas tarde o salidas antes de horario establecido, organizadas por situaciones consideradas como justificadas por la administración, en el período en que se evalúa el desempeño laboral, ello debe ser valorado a los efectos del derecho o no al pago adicional especial.
- ▶ No haber sido objeto de sanción. El trabajador que resulte sancionado pierde el derecho a 3 meses de estímulo o haber sido exonerado de la medida disciplinaria aplicada.
- ▶ Otros específicos que se decida incluir a propuesta de la dirección de cada área y previa aprobación de la rectoría de la UCI, de común acuerdo con la organización sindical, con el objetivo de alcanzar resultados superiores en el trabajo. (MIC 2009)

1.4.2. Herramientas existentes para la evaluación de desempeño

En el mundo de la informática se han desarrollado un gran número de sistemas capaces de facilitar las labores más comunes de la evaluación de desempeño. Entre estos se encuentran:

1.4.2.1. QTraining

Es una poderosa pero sencilla herramienta para evaluar el desempeño del personal, detectar necesidades de capacitación y administrar los cursos de capacitación de la empresa. Permite evaluar el impacto de las capacitaciones llevadas a cabo y relaciona los objetivos de desempeño con los cursos de capacitación. (WILSOFT 2008)

Esta herramienta está enfocada al entrenamiento, capacitación y superación profesional y de ahí se deriva la evaluación de desempeño que realiza. No tiene integración con sistemas de planificación de tareas y no es completamente personalizable como para gestionar la evaluación de desempeño en la universidad debido a las tantas particularidades que presenta la UCI. Las versiones existentes presentan sólo soporte para Windows. Es software propietario. Está orientada al sector empresarial privado y no para un ámbito docente.

- ❖ Los costos por licencia oscilan:
 - ▶ 790 EUR licencia principal
 - ▶ 190 EUR licencia usuario adicional
 - ▶ 2.990 EUR licencia empresarial
 - ▶ 7.990 EUR licencia corporativa

Por lo que se considera una herramienta con alto costo de adquisición.

1.4.2.2. eFeedback

Es un sistema desarrollado por el Instituto de Ingeniería del Conocimiento (IIC), para evaluar el desempeño profesional de los individuos mediante la medición de una serie de competencias, de forma objetiva y precisa. Para ello eFeedback permite desde la evaluación por el superior inmediato, hasta una medición 360°, en la que se toma como referencia al propio evaluado y a su equipo de trabajo habitual (Colaboradores y personas a su cargo). (EFEEDBACK 2008)

Esta herramienta solo puede ser accedida vía Internet, mediante un servicio on demand, por lo que no existe ninguna versión que pueda ser desplegada para su uso local. Su

enfoque está asociado a medir las competencias profesionales de la empresa y el contexto laboral en el que se desarrollan. No genera la evaluación de desempeño a partir de un plan de resultados.

1.4.2.3. GIRH (Gestión Integral de Recursos Humanos)

GIRH es un producto innovador, específicamente diseñado para empresas e instituciones, que desean sistematizar sus procesos en el área de Recursos Humanos. Permite realizar operaciones de cálculo de sueldo y jornales, además cuenta con un conjunto de módulos integrados para realizar evaluaciones, seguimiento estadístico y gestión de los diferentes aspectos de la administración de los recursos humanos de su empresa. (DATALOGIC 2009)

En este producto, la Evaluación de Desempeño no está concebida como una herramienta de "control", sino como una herramienta de gestión, que permite identificar las principales fortalezas y oportunidades de mejora de cada persona. No es completamente personalizable como para gestionar la evaluación de desempeño en la universidad debido a las tantas particularidades que presenta la UCI y está orientada al igual que las demás, al sector empresarial privado y no para un ámbito docente.

1.5. Framework para desarrollo web con PHP

1.5.1. Definición de Framework:

Realmente no existe una definición oficial de Framework, pero todos los autores coinciden en la utilización de un tema común: la reutilización.

Una definición dada por R. E. Johnson, and B. Foote en 1988 en su publicación "Designing Reusable Classes:

"Un Framework es un conjunto de clases que personifican un diseño abstracto para soluciones de una familia de problemas relacionados..." (JOHNSON 1988)

Otro punto de vista, según R. E. Johnson y V. F. Russo en "Reusing Object-Oriented Designs" incluyen: "Un Framework es el diseño de un conjunto de objetos que colaboran para llevar a cabo un conjunto de responsabilidades. De esta manera los Frameworks son diseños a escalas más grandes que las clases abstractas. Los Frameworks son una forma de reutilizar el diseño a un nivel superior."(JOHNSON 1991)

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

A continuación se caracterizan los framework más relevantes: Zend Framework, Symfony, CakePHP y Codeigniter, concluyendo con la selección de uno de ellos para el desarrollo del trabajo.

1.5.2. Zend Framework

Zend Framework (ZF) es un Frameworks de código abierto para desarrollar aplicaciones web y servicios web con PHP 5, está bajo la filosofía de programación orientado a objetos. La estructura de los componentes de ZF es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. (ZF 2010)

Algunas características de Zend Framework:

- ❖ Arquitectura de tres capas (MVC)
- ❖ Gran robustez y rendimiento
- ❖ Abstracción de base de datos fácil de usar
- ❖ Interfaz orientada a objetos
- ❖ Seguridad
- ❖ Administración de caché
- ❖ Herramientas y desarrollo rápido de aplicaciones (RAD)
- ❖ Programación orientada a objetos
- ❖ Soporte para internacionalización y localización de contenidos
- ❖ Soporte para envío de correo electrónico.
- ❖ Validación de Formularios
- ❖ Manejo propio de sesiones por usuarios.

1.5.3. Symfony

Symfony es un framework para PHP5 que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes y aburridos de las aplicaciones web, dejando que el programador se dedique a aportar valor, desarrollando las características únicas de cada proyecto. Está patentado bajo la licencia MIT3, orientado a la creación de aplicaciones web de un gran tamaño, es compatible con la mayoría de los gestores de

bases de datos más utilizados en el mundo, PostgreSQL, Oracle y SQL Server de Microsoft, entre otros en dependencia del tipo de abstracción de la Base de Datos que se utilice.(SYMFONY 2009)

Construido específicamente para sitios web empresariales, de necesidades complejas y para mejorar los procesos y las prácticas de desarrollo.

Algunas características de Symfony

- ❖ Su instalación es sencilla en cualquier plataforma (Windows, Unix) y su curva de aprendizaje es alta gracias al trabajo de su comunidad y de los desarrolladores. Permite la integración con otros Framework.
- ❖ Sencillo de usar en la mayoría de los casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ❖ Utiliza la arquitectura en tres capas (MVC)
- ❖ Código fácil de leer que incluye comentarios y que permite un mantenimiento muy sencillo.
- ❖ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros, además de la utilización de helpers que permiten una integración con código para HTML, Javascript, AJAX y JSON⁴.
- ❖ Excelente mecanismo de configuración mediante ficheros YML y XML.
- ❖ Administración de la caché.
- ❖ Manipula la gestión de credenciales.
- ❖ Independiente del gestor de base de datos.
- ❖ Mantiene la mayoría de mejores prácticas y patrones de diseño para la web.
- ❖ La capa de presentación utiliza plantillas que pueden ser creadas por diseñadores HTML sin ningún tipo de conocimiento del Frameworks.
- ❖ Los formularios incluyen validación automatizada, lo que posibilita insertar datos correctos en la base de datos.
- ❖ Tiene el CRUD⁵ de la base de datos incluido.

1.5.4. CakePHP

Según sus creadores “Cake es un rápido Framework de desarrollado para PHP que normalmente usa algunos de los más conocidos patrones de diseño como ActiveRecord, Mapeo de Datos, Controlador Frontal y MVC. Nuestro principal objetivo es suministrar la

estructura de un Framework que habilite a los usuarios de PHP a todos los niveles de aplicaciones web rápidas y robustas sin que pierda la flexibilidad". (CAKEPHP 2008)

CakePHP es un frameworks para la creación de aplicaciones web, escrito sobre PHP, facilita al usuario la interacción con la base de datos, pueden implementarse aplicaciones orientadas a objetos, es compatible con PHP 5, tiene un conjunto de plantillas fáciles de utilizar, está destinado a proyectos medianos, su flexibilidad, seguridad y rapidez lo hacen un framework líder en el mercado, tiene un CRUD de la base de datos integrado, utiliza los patrones de diseño MVC y ORM⁶.

Características que presenta el Cake son:

- ▶ URLs limpias y amigables, debido al despachador de peticiones con el que cuenta. Presenta sistemas de validación a los largo de todo el Framework
- ▶ Verificación de ingreso de datos permitidos (Data Sanitization): permite determinar qué datos pueden ser ingresados y darle el formato adecuado a aquellos que no cumplen las reglas de validación.
- ▶ Generación de plantillas de manera rápida y flexible: usando la sintaxis de PHP y con asistentes o helpers.
- ▶ Incorporación de asistentes de construcción de vistas: para la automatización de la generación de código en AJAX, Java Script, formularios HTML, entre otros. Componentes de seguridad, manejo de sesiones y de peticiones: que reúnen las mejoras prácticas estandarizadas por la industria del software.
- ▶ Listas de control de acceso flexibles: para gestionar el ingreso de usuarios a la aplicación construida con el Framework.
- ▶ Almacenamiento en caché de las vistas: para acelerar la descarga de las páginas web. Trabaja en cualquier subdirectorío de un servidor web: requiere poca o nula configuración del servidor Apache donde se instalará. (CAKEPHP 2008)

1.5.5. Codeigniter

Codeigniter es un framework open source que permite desarrollar aplicaciones web con PHP. Es pequeño, corre muy rápido, está bien documentado y es fácil de aprender (siempre que se posea conocimientos de PHP). Este marco de trabajo utiliza el patrón de diseño de software MVC(Modelo-Vista-Controlador), lo que facilita mucho estructurar la aplicación y su mantenimiento, además posee helpers, bibliotecas, etc. que facilitan y agilizan mucho el desarrollo. Trabajar con bases de datos es muy fácil, ya

que CodeIgniter brinda todo un conjunto de clases muy completas y además está el Scaffolding, que permite desde el navegador, crear y editar los datos de la DB. (CODEIGNITER 2009)

Algunas de las ventajas que posee codeigniter comparado con otros frameworks de php, es que no necesitas saber algún lenguaje de consola para pre-instalar componentes, la documentación es clara y simple de entender, no se necesitan conocimientos muy avanzados en php para entender las utilidades que ofrece.

Características generales de CodeIgniter

Algunos de los puntos más interesantes sobre este framework, sobre todo en comparación con otros productos similares, son los siguientes:

- ▶ **Versatilidad:** Quizás la característica principal de CodeIgniter, en comparación con otros frameworks PHP. CodeIgniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.
- ▶ **Facilidad de instalación:** No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas, como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- ▶ **Flexibilidad:** CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.
- ▶ **Ligereza:** El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.
- ▶ **Documentación tutorializada:** La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto nos facilita mucho la

referencia rápida, cuando ya sabemos acerca del framework y queremos consultar sobre una función o un método en concreto.

Ventajas de su uso:

- ▶ Usa muy pocos recursos y ofrece un rendimiento excepcional
- ▶ Utiliza el Patrón MVC. Esto permite escalabilidad y la modificación de una manera fácil al momento de arreglar/agregar o quitar
- ▶ Genera URL segmentadas aptas para los navegadores.
- ▶ Soporta desde PHP 4 hasta la versión Actual.
- ▶ Soporta las principales bases de datos Mysql, MySQLi, SQL Server, Postgres, Oracle, SQLite y ODBC.
- ▶ Operaciones Cacheadas Fácilmente.
- ▶ Los módulos solo se cargan cuando son necesarios
- ▶ Excelente documentación.

1.5.6. Fundamentación del framework PHP.

Teniendo en cuenta que la aplicación a desarrollar en el presente trabajo tiene un tamaño mediano, y que no es una herramienta de una gran cantidad de módulos, se decide la utilización de Codeigniter. Este framework, frente a cualquiera de los demás presentados, tiene fortalezas bien marcadas y ventajas que se tuvieron en cuenta por el tipo de desarrollo específico que queremos obtener, dentro de estas podemos mencionar su velocidad en cuanto a rendimiento, ya que la aplicación requiere de gran agilidad, la facilidad de aprendizaje del mismo mediante un código claro y legible, puesto que, existía poco tiempo para obtener una aplicación que cumpliera con los requerimientos propuestos y el tiempo de asimilación del framework debía ser acortado lo más posible, además su gran comunidad, soporte y documentación, permitiendo que la curva de aprendizaje fuese leve, posee gran flexibilidad, lo que hace que la arquitectura sea completamente personalizada por los desarrolladores, brindando así la posibilidad de utilizar buenas prácticas y patrones.

1.6. Framework para interfaz de aplicaciones web

Las aplicaciones web de hoy en día, necesitan de interfaces robustas, que posean gran interactividad, que sean sencillas, que los componentes visuales que contengan faciliten el trabajo a los usuarios y que las características de entorno de las mismas presenten

principios de la Web 2.0 y cumplan con los estándares de la W3C. Los frameworks de interfaz brindan estas posibilidades y otras ventajas como:

- ▶ Facilita y agiliza el proceso de desarrollo de aplicaciones Web.
- ▶ Permite reutilizar código ya existente y promueve buenas prácticas de desarrollo.
- ▶ Resuelve los problemas de compatibilidad entre los navegadores.

Seguidamente se abordarán aspectos de carácter teórico relacionados con las características de algunas de las librerías JavaScript para la programación web. Además se seleccionará la más adecuada para el desarrollo de la interfaz de la aplicación.

1.6.1. Librería JQuery

JQuery es una librería de JavaScript, rápida y concisa que simplifica el trabajo con documentos HTML. No es un extenso e inflamado framework que prometa lo mejor de AJAX, ni un conjunto de innecesarias y complicadas mejoras: JQuery ha sido diseñado para cambiar la forma de escribir JavaScript. Utiliza un interesante concepto para hacer código corto y simple, tiene manejadores de eventos, añade un conjunto de efectos creando un dinamismo visual a la presentación del sitio (JQUERY 2010).

Características

- ▶ JQuery Core: Principales funciones de JQuery.
- ▶ Selectores: Los selectores son una combinación de CSS 1-3, XPath, junto con funciones de código especiales que permiten que funcionen conjuntamente. De forma realmente sencilla podemos seleccionar diferentes elementos del árbol DOM mediante XPath.
- ▶ Atributos: Permiten seleccionar, cambiar, leer, añadir y borrar los atributos de los elementos DOM de forma sencilla.
- ▶ Traversing: Funciones de filtrado y de búsquedas.
- ▶ Manipulación: Funciones de inserción y modificación.
- ▶ CSS: Funciones para el cambio de estilo.
- ▶ Eventos: Control de eventos para elementos DOM con la sintaxis propia de CSS.
- ▶ Efectos: Sencillos efectos visuales.
- ▶ Utilidades: Otras funciones genéricas adicionales, permiten referenciar objetos sin problemas de espacios de nombres. (XPERIMENTOS 2007)

1.6.2. Librería Dojo Toolkit

Dojo Toolkit es una librería JavaScript de código abierto DHTML (Dynamic HTML) escrito en JavaScript. Proporciona un API ⁷ para el control y manipulación de historial, permite la manipulación de URL y marcadores/favoritos, el trabajo con widgets⁸ e incluye el ordenamiento de tablas, validación de formularios, menús y barra de menús, Google y Yahoo! Maps. Su idea es la de abstraer al desarrollador de las complejidades de DHTML y de las discrepancias existentes entre navegadores, que hacen, que el código JavaScript a utilizar sea diferente. Dojo Toolkit es hecho de un juego de bibliotecas de capas. También puede usar el API de bajo nivel y capas compatibles para escribir scripts portables y simplificar aquellos que son complejos. Proporciona un potente entorno de programación incorporando un sistema de control de eventos, un API E/S y un lenguaje estándar mejorado que grandemente mejoran y simplifican la vida de JavaScript para diseñadores.(DOJOTOOLKIT 2010)

Características

- ▶ Múltiples Puntos de Entrada: Es un concepto fundamental en el diseño de Dojo. Este término significa que los usuarios pueden empezar a utilizar Dojo al nivel que ellos deseen.
- ▶ Independencia del interpretador: Asegurar soporte para el mayor número de plataformas.
- ▶ Unifica varios estándares de codificación.
- ▶ Maneja incompatibilidades entre navegadores. (XPERIMENTOS 2007)

1.6.3. Librería Interfaz de Usuario de Yahoo (YUI)

Yahoo! User Interface (YUI) surge con la finalidad de crear Interfaces de Usuario. Cuenta con una serie de utilidades y controles desarrollados en JavaScript, para la construcción de aplicaciones web interactivas y opulentas, utilizando técnicas como DOM, DHTML y AJAX, también incluye diversos recursos basados en CSS. Todos los componentes de YUI son libres para el uso de todos los que accedan a ellos, esta librería esta patentada con la licencia BSD de software libre. (YAHOO 2008)

Características

Están disponibles dos tipos de componentes diferentes: Utilidades y Controles.

- ▶ Las Utilidades de YUI simplifican el desarrollo para la compatibilidad entre navegadores basados en técnicas DOM, DHTML y AJAX.

- ▶ Los Controles de YUI proporcionan elementos visuales altamente interactivos del diseño para sus aplicaciones web. Estos elementos se crean y se manejan íntegramente del lado del cliente (usuario) y nunca requieren de una recarga de página. (EGUÍLUZ PÉREZ 2005)

1.6.4. Librería ExtJS

La librería ExtJS, extiende de la librería YUI e integra AJAX, Prototype y Scriptaculous.

ExtJS es neutral al lenguaje que se use en el servidor. Siempre que el resultado se envíe a la página en el formato adecuado, no se preocupará de lo que pase en el servidor. Hay docenas de widgets a escoger en ExtJS, incluyendo composiciones automáticas de páginas, pestañas, menús, barras de herramientas, diálogos, vistas en árbol. Proporciona un selector de nodos DOM extremadamente poderoso llamado DomQuery que puede usarse como una librería independiente, pero en el contexto de ExtJS se usará para seleccionar elementos para poder interactuar con ellos a través de la interfaz Element, contiene muchos de los métodos y propiedades del DOM que con lo cual se proporciona una interfaz conveniente, unificada y multinavegador. (EXTJS 2009)

Características de ExtJS

- ▶ **Ext.Element:** Representa un elemento del árbol DOM. Muchas de las funciones de manipulación de los elementos, tienen un parámetro opcional que permite realizar el cambio mediante un efecto de animación. El parámetro de animación puede ser un dato booleano o un objeto que incluye las opciones de la animación.
- ▶ **Ext.BorderLayout:** Esta clase representa un diseño común para ser usado en aplicaciones de escritorio.
- ▶ **Ext.DomHelper:** Utilidades para trabajar plantillas o DOM. Soporta el uso de DOM o fragmentos de HTML de forma transparente.
- ▶ **Ext.TabPanel:** Un ligero contenedor de tabs.
- ▶ **Ext.UpdateManager:** Proporciona soporte para actualización AJAX de los objetos Element. (ZAMMETTI 2009)

Ventajas de EXT

- ▶ Código reutilizable
- ▶ Independiente o adaptable a frameworks diferentes (prototype, jquery, YUI)

- ▶ El API es homogeneizado independientemente del adaptador usado. Los controles siempre se verán igual.
- ▶ Soporte comercial
- ▶ Una extensa comunidad de usuarios

ExtJS tiene dos tipos de licencias GPL v3 y comercial. Para obtener soporte y contribuir a la continuidad del desarrollo del producto, se debe hacer comprando licencias comerciales de Ext. Esta opción asegura el derecho de distribuir tu aplicación en los términos de la licencia que elijas, en otro caso sería para contribuir a la comunidad Open Source con tu aplicación (por ejemplo con una licencia GPL v3). Esta opción asegura que todos los usuarios puedan obtener todo el código, lo puedan modificar y redistribuir.

1.6.5. Fundamentación de la librería JavaScript a utilizar

En esta sección se han abordado algunas de las librerías conocidas y más utilizadas, que de una forma u otra, incorporan JavaScript para realizar sus funcionalidades con mayor agilidad y eficacia. Se ha dado a conocer las características principales y fundamentales de cada una de ellas, para de este modo, seleccionar la que más se adecua a las necesidades requeridas, con el fin de darle una utilización eficiente y productiva para el sistema propuesto.

Se selecciona para el desarrollo de la mayoría de los componentes de la capa de presentación la librería ExtJS, por sus facilidades en cuanto a la reutilización de código, el alto nivel de integración con otras librerías, la diversidad de componentes existentes y la extensa comunidad de usuarios, la cual gana cada vez más experiencia y fuerza dentro de la universidad a través del gran número de proyectos que la utilizan. Además, cabe mencionar, que es una librería que cumple con todos los estándares web y se ajusta a las necesidades del cliente.

1.7. Metodologías de desarrollo del software

La Ingeniería del Software es una tecnología multicapa, en la que se pueden definir los métodos, el proceso y las herramientas. La Ingeniería del Software surge con el principal objetivo de crear y mantener aplicaciones de software, aplicando tecnologías y prácticas computacionales. Su constante desarrollo ha derivado la realización de varias tareas en

este campo, como son: análisis de requisitos, especificación, diseño y arquitectura, programación, prueba, documentación y mantenimiento.

Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. El proceso de desarrollo del software, define el conjunto de actividades precisas para convertir los requisitos de los usuarios en el conjunto seguro y resistente de artefactos que componen un producto de software. (JACOBSON 2000)

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Genéricamente es un conjunto de pasos que deben seguirse para obtener un producto de software. No existe una metodología de software universal, las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable. (ROGER S. PRESSMAN 2001)

Entre las metodologías de desarrollo de software más utilizadas a nivel mundial, en nuestro país, y específicamente en la UCI, se pueden encontrar:

- ▶ Extreme Programing (XP)
- ▶ Rational Unified Process (RUP)

Para obtener el producto final con la calidad requerida, se hace necesario definir la metodología de desarrollo de software a utilizar, por lo que sería preciso analizarlas, teniendo en cuenta sus características, ventajas, desventajas, y su adecuación al desarrollo del sistema.

1.7.1. Programación Extrema (XP)

La metodología XP, del inglés Extreme Programing, es una metodología ágil, que se centra en el aumento de las relaciones interpersonales para lograr un desempeño exitoso en el desarrollo del software, donde el trabajo en equipo es muy importante, preocupándose por la existencia de un ambiente de trabajo óptimo, y porque los desarrolladores realicen un buen aprendizaje. De todas las metodologías ágiles, ésta es la

que ha recibido más atención. Esto se debe en parte, al trabajo realizado por los precursores de la misma.

Las raíces de XP yacen en la comunidad de Smalltalk, y en particular la colaboración cercana de Kent Beck y Ward Cunningham a finales de los 1980s. Ambos refinaron sus prácticas en numerosos proyectos a principios de los 90s, extendiendo sus ideas de un desarrollo de software adaptable y orientado a la gente. (Beck 1999)

Una característica fundamental de XP, es que el código siempre se produce en parejas, parejas que van cambiando constantemente para lograr así que todo el equipo sepa y pueda modificar según necesidades, el código generado, esto logra en el equipo que los integrantes aprendan entre sí y compartan todo el código.

1.7.2. Proceso Unificado del Software (RUP)

El Proceso Unificado del Software (RUP por sus siglas en inglés), es una metodología de desarrollo de software, pesada y orientada a objetos, es un marco de trabajo genérico, que puede ser especializado para una gran variedad de software para distintas áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

Las actividades son agrupadas en grupos lógicos, donde se definen 9 flujos de trabajo fundamentales, de los cuales solo los primeros 6 son considerados flujos de ingeniería, y los 3 restantes y últimos como flujos de apoyo. Estos flujos son: Modelado de Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Configuración y Administración de Cambios, Administración de Proyecto y Entorno. Además, RUP define cuatro fases importantes en el desarrollo, las mismas son: Conceptualización (Concepción o Inicio), la cual describe el negocio y delimita el proyecto, se definen sus alcances con la identificación de los casos de uso del sistema; Elaboración, donde se define la arquitectura que tendrá el sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen; Construcción, donde se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario, se obtienen una o más entregas del producto que han pasado las pruebas y por último, pero no menos

importante, la fase de Transición, en la cual el release (versión) ya está listo para su instalación en condiciones reales. (Kruchten 2003)

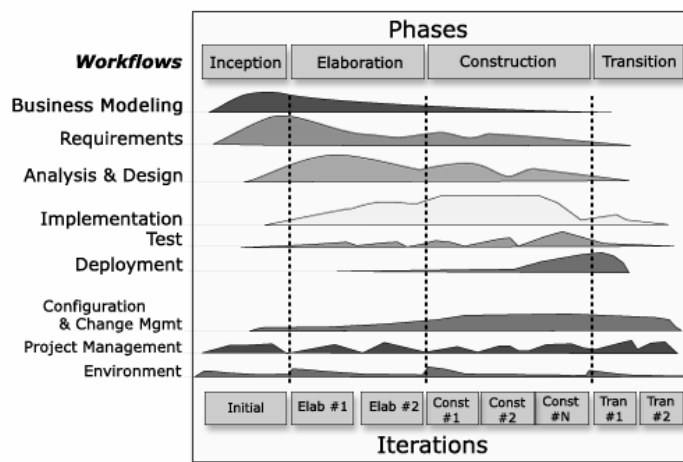


Figura 1. Representación del proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

La metodología RUP, fue la idónea para realizar el análisis y diseño de la aplicación, debido a la documentación que genera en cada iteración, además de que brinda una excelente organización y planificación, al dividir cada etapa por flujos de trabajo, donde los casos de uso y la arquitectura son los pilares fundamentales, realizando diferentes iteraciones que mejoran constantemente el producto, otro aspecto que se tuvo en cuenta, es el conocimiento adquirido de la metodología según el programa de estudios de la universidad.

1.8. Tecnologías de servidores

Dentro de las tecnologías para soportar la aplicación desde el servidor, se selecciona un Sistema Gestor de Base de datos y un Servidor Web, entiéndanse estos como:

1.8.1. Sistemas Gestores de Base de Datos (SGBD)

Los sistemas de gestión de bases de datos son un conjunto coordinado de programas, procedimientos, lenguajes, etc. Que suministra, tanto a los usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad. (DATE 2003)

1.8.2. Servidor Web

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). (MATEU 2004)

Se eligió MySQL como Gestor de Base de Datos, ya que presenta ventajas y características que permiten el buen desarrollo de la aplicación, en cuanto al manejo de la información, como por ejemplo: soporta gran cantidad de tipos de datos para las columnas. Gestiona usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos. Posee infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación. Resulta muy fácil su instalación y configuración. Además de que es el manejador de base de datos considerado como el más rápido de Internet.

Como servidor Web se optó por Apache 2, pues permite al administrador, debido a que es un programa modular, elegir qué funcionalidades se incluyen mediante la selección de un conjunto de módulos disponibles, es multiplataforma y muy popular, no sólo en la comunidad universitaria, sino también a nivel mundial, lo que hace que sea fácil conseguir ayuda y soporte para éste.

1.9. Conclusiones parciales

Después de haber realizado un análisis de la información recopilada sobre los diferentes Sistemas para la gestión y planificación de tareas, así como sistemas de gestión de la evaluación de desempeño que son utilizados en algunos países del mundo así como en nuestro país y específicamente en la Universidad, se puede concluir que ninguno de los estudiados se adecua a las características requeridas por los objetivos del sistema propuesto.

Por lo que se decide la creación de una Herramienta de control, planificación y evaluación de profesores, que permita un mejor desarrollo en la facultad y en específico dentro de los departamentos docentes, en cuanto a la gestión e integración del plan de resultados y la evaluación de desempeño.

Se detallaron tendencias y tecnologías actualmente usadas en el mundo del software y específicamente en el desarrollo web, donde se llegó a la conclusión de cuáles eran las herramientas más adecuadas para darle solución a la problemática planteada.

2. Capítulo 2: Características del sistema.

2.1. Introducción

El presente capítulo constituye el resultado del estudio de los procesos de evaluación de desempeño y creación del plan de resultados en los departamentos docentes de la Universidad de las Ciencias Informáticas, así como de la metodología RUP, que indica cómo modelarlos. Con el propósito de ofrecer una idea clara de todo lo que soporta el sistema que se modela. Para describir la solución se realiza el modelo de dominio, así como un glosario de términos para definir los diversos conceptos que serán utilizados en el mismo, Además se establecen las relaciones entre estos conceptos, en aras de lograr la utilización de un vocabulario común. A partir de este estudio, se definen los requisitos funcionales y no funcionales que deberá concebir la aplicación, Se describen las características que va a tener el sistema propuesto, así como los actores y casos de uso.

2.2. Modelo de dominio

¿Por qué modelo de dominio?

Teniendo en cuenta que la definición del proceso del negocio que tienen que ver con el objeto de estudio se hace difícil encontrarlo, se ve a simple vista, la necesidad de describir el funcionamiento de la herramienta mediante una serie de conceptos, entidades y sus relaciones, agrupándolos en un modelo del dominio con el fin de tener un fácil entendimiento del sistema. Se ha decidido realizar un modelo de dominio ya que es una alternativa apropiada dado el escenario del problema, además de que no se considera necesario un modelamiento completo del negocio, las fronteras del negocio no se pueden determinar con claridad y los procesos del negocio no están claramente definidos.

Un Modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema.

Con la aplicación del Modelo del dominio se detectaron las siguientes entidades y conceptos (objetos):

Evaluación: Es el resultado de todo el proceso, donde cada profesor va a contener una evaluación al final del trimestre.

Evaluador: Es el responsable de darle la evaluación a los profesores que le pertenecen, además de crear sus planes de resultados.

Plan de Resultados: Contiene los resultados que los profesores deben alcanzar en un período determinado.

Profesor: Es el responsable de cumplir con las tareas que se le asignen.

Tareas: Son todas aquellas actividades que se le asignan a los profesores y éstos a su vez deben de darle cumplimiento.

Resultados: Es el equivalente a un objetivo que debe cumplir el profesor.

Principales Eventos:

Contiene: Expresa que A está contenido lógicamente en B, Evaluador y Profesor contienen una evaluación. Las tareas contienen resultados y el plan de resultados contiene una evaluación.

Evalúa: Expresa que A evalúa a B, el Evaluador evalúa al Profesor.

Realiza: Expresa que A realiza lo que contiene B, el Profesor realizan Tareas y el Evaluador realiza una Evaluación y un Plan de resultados.

Representación del Modelo de Dominio

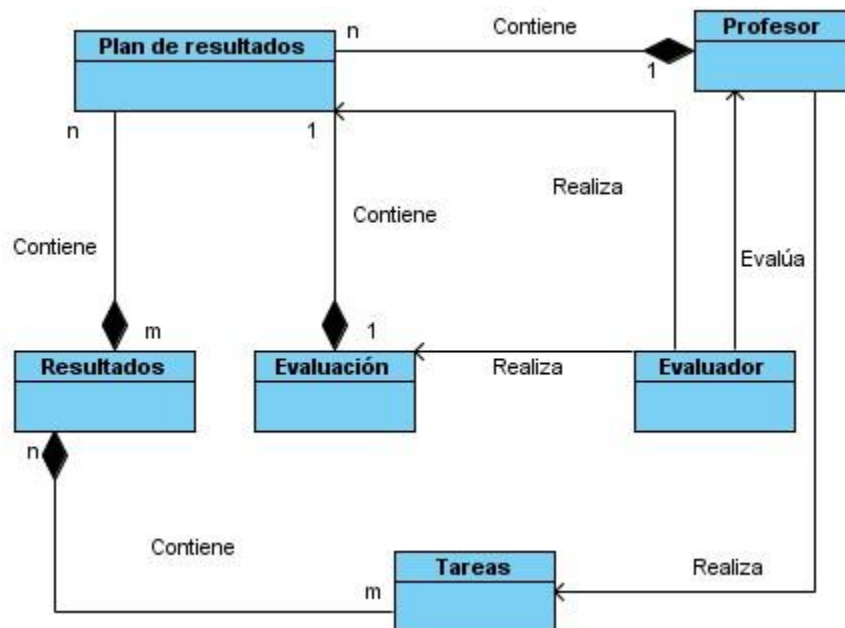


Figura 2. Modelo de Dominio

2.3. Levantamiento de Requisitos

El levantamiento de los requerimientos es uno de los flujos de trabajo que propone la metodología RUP para el desarrollo de un software. Los requisitos que aquí se exponen ayudarán a la comprensión del sistema que se modela, mostrando en qué basa su

funcionamiento y por qué, además se especifican las propiedades que debe tener el producto final, a través de los requisitos no funcionales.

2.3.1. Requerimientos Funcionales

1. El sistema debe validar los datos: el sistema validará los datos de entrada en todos los formularios.
2. Adicionar tarea: El sistema debe dar la posibilidad de adicionar una nueva tarea. De la misma se debe registrar Título, Lugar, Esfera, Estado, Fecha inicial, Hora inicial, Fecha final, Hora final, Tipo, Descripción y Resultados. Las tareas deben quedar registradas. Las tareas que son registradas como adicionales no forman parte del plan de trabajo.
3. Eliminar tarea: El sistema debe dar la posibilidad de eliminar una tarea seleccionada por el usuario.
4. Modificar tarea: El sistema debe dar la posibilidad de modificar una tarea seleccionada por el usuario.
5. Consultar tarea: El sistema debe permitir consultar las tareas del trimestre en todo momento.
6. Gestionar tareas de otros usuarios: Consiste en que el usuario con privilegios de jefe de área o jefe de departamento pueda Asignar, Adicionar, Eliminar y Modificar las tareas de otros usuarios.
 - 6.1. Mostrar el nombre del usuario al que se le realiza la gestión: cuando se gestionan tareas de otros recursos el sistema debe mostrar en todo momento el nombre que identifique el recurso al cual se le están gestionando las tareas.
 - 6.2. Registrar el propietario de las tareas: El sistema deberá registrar el id del usuario que esté adicionando las tareas, como propietario de la misma. En caso de que la tarea sea para el propio usuario que la adiciona, él mismo se convierte en propietario, en caso que la tarea vaya a ser asignada a otro recurso humano, el que la asigna sería el propietario y el usuario, el que la ejecuta.
 - 6.3. Las tareas solo pueden ser modificadas y eliminadas por sus propietarios: Todas las tareas podrán ser modificadas y eliminadas sólo por los propietarios de la misma, exceptuando al jefe de departamento, que puede acceder a todas las tareas.

7. Adicionar plan de resultados: El sistema debe dar la posibilidad de adicionar un nuevo plan de resultados. Del mismo se debe registrar Usuario, Período, Descripción, Resultados y Peso.
 - 7.1. Verificar estado del plan de resultados: Una vez que se registre un nuevo plan de resultados, el sistema debe verificar que el usuario al que se le va a adicionar el plan, no tenga ninguno activo.
8. Eliminar plan de resultados: El sistema debe dar la posibilidad de eliminar un plan de resultados seleccionado por el usuario.
9. Modificar plan de resultados: Solo podrá ser modificado siempre que el estado del mismo sea activo.
10. Consultar plan de resultados: El sistema debe dar la posibilidad de ver los detalles de un plan de resultados seleccionado por el usuario.
11. Adicionar evaluación de desempeño: El sistema permitirá registrar la evaluación de desempeño de un usuario, la misma será registrada por el jefe de departamento. De cada evaluación se registra nombre y apellidos, tipo de período, plan de resultados, tareas adicionales, tareas planificadas, cantidad de indicadores afectados, cantidad de incumplimientos, observaciones, evaluación por esferas y evaluación final.
12. Eliminar evaluación de desempeño: La evaluación de desempeño solo podrá ser eliminada mientras su estado sea activo.
13. Modificar evaluación de desempeño: La evaluación de desempeño solo podrá ser modificada mientras su estado sea activo.
14. Consultar evaluación de desempeño: El sistema permitirá mostrar en detalle, la evaluación de desempeño seleccionada.
15. Adicionar asignatura: El sistema permitirá la adición de asignaturas por parte del administrador.
16. Eliminar asignatura: El sistema permitirá eliminar una asignatura seleccionada por el administrador.
17. Modificar asignatura: El sistema permitirá modificar una asignatura seleccionada por el administrador.
18. Consultar asignatura: El sistema deberá mostrar las asignaturas existentes.
19. Adicionar esfera: El sistema permitirá la adición de esferas por parte del administrador.
20. Eliminar esfera: El sistema permitirá eliminar una esfera seleccionada por el administrador.

21. Modificar esfera: El sistema permitirá modificar una esfera seleccionada por el administrador.
22. Consultar esfera: El sistema deberá mostrar las esferas existentes.
23. Adicionar resultados: El sistema permitirá la adición de resultados.
24. Eliminar resultados: El sistema permitirá eliminar un resultado seleccionado.
25. Modificar resultados: El sistema permitirá modificar un resultado seleccionado.
26. Consultar resultados: El sistema deberá mostrar los resultados existentes.
27. Adicionar perfiles: El sistema permitirá la adición de perfiles.
28. Modificar perfiles: El sistema permitirá modificar el perfil de un usuario seleccionado.
29. Consultar perfiles: Cuando se muestren los perfiles, se mostrarán campos particulares para los profesores: grupo asignado y asignatura; además de otros campos comunes para todos los usuarios.
30. Adicionar roles: El sistema permitirá la adición de roles por parte del administrador.
31. Eliminar roles: El sistema permitirá eliminar un rol seleccionado por el administrador.
32. Modificar roles: El sistema permitirá modificar un rol seleccionado por el administrador.
33. Consultar roles: El sistema deberá mostrar los roles existentes.
34. Adicionar usuarios: El sistema permitirá la adición de usuarios por parte del administrador.
35. Eliminar usuarios: El sistema permitirá eliminar un usuario seleccionado por el administrador.
36. Modificar usuarios: El sistema permitirá modificar un usuario seleccionado por el administrador.
37. Consultar usuarios: El sistema deberá mostrar los usuarios existentes.
38. Autenticación de usuarios: El sistema permitirá el acceso a las diferentes opciones después de la autenticación, la cual será haciendo uso de LDAP. El sistema cargará los privilegios según el rol de cada usuario.
39. Uso de webservices uci: se hará uso de los webservices de la uci, para obtener información necesaria.
40. Cada usuario que acceda al sistema debe tener un rol.
41. Cada rol debe tener definido un conjunto de permisos en el sistema.
42. El sistema debe tener un registro auxiliar de todas las operaciones que se realizan en el mismo, por motivos de seguridad.

2.3.2. Requerimientos No Funcionales

Apariencia de Interfaz o Externa:

- ▶ El sistema contará con una interfaz sencilla y fácil de entender por los usuarios.

Usabilidad:

- ▶ El sistema debe permitir el acceso concurrente de un gran número de usuarios.
- ▶ El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Rendimiento:

- ▶ El sistema debe ser lo más eficiente posible para poder lograr un tiempo de respuesta adecuado.
- ▶ La velocidad de procesamiento de la información debe ser rápida.

Soporte:

- ▶ El sitio debe permitir posteriores modificaciones y actualizaciones a fin de alcanzar mayor funcionalidad o dado que cambien algunos elementos del negocio.

Portabilidad:

- ▶ Necesidad de que el sistema sea multiplataforma.

Seguridad:

- ▶ La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación.
- ▶ La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes por las personas autorizadas.
- ▶ Se deben crear grupos de usuarios que tendrán asignados permisos de acción sobre cada información manejada por el sistema para lo cual se requiere la autenticación del usuario.

2.4. Descripción del Sistema propuesto

El sistema debe cumplir con los requerimientos propuestos, con vistas a obtener un conjunto de funcionalidades que lo conviertan en la solución ideal.

Dentro de las funcionalidades, debe permitirle al jefe de departamento crear un plan donde se reflejarán los resultados que debe tener el profesor en un determinado período de tiempo, cuando cada resultado es adicionado al plan, se debe especificar el peso con que incidirá. Los usuarios podrán consultar dichos planes y en éstos se reflejará el

porcentaje de cumplimiento de los resultados hasta el momento, el cual es calculado de la siguiente forma:

$$\% \text{ cumplimiento Resultado X} = \frac{(\sum \text{tareas verificadas que contienen al resultado X}) \times 100}{\text{peso del resultado X}}$$

Este porcentaje, el sistema lo comparará con los indicadores establecidos en la configuración, donde se debe especificar con que porcentaje mínimo el resultado se considera adecuado, por debajo de éste se considera deficiente, y además con que porcentaje mínimo el resultado se considera superior. Esto se reflejará en el plan con una escala de colores:

Rojo Deficiente **Verde** Adecuado **Azul** Superior

Que le permitirá a los usuarios tener una guía de avance del plan y al evaluador tener criterio para dar la evaluación de desempeño final.

Los usuarios podrán gestionar sus propias tareas, éstas tributarán a un resultado específico dentro del plan y contendrán varios estados:

Estado	Descripción
Planificada	La tarea se encuentra planificada actualmente para su futura ejecución.
Completada	La tarea se considera completada después de su ejecución.
Verificada	La tarea fue verificada y aceptada por el evaluador una vez que fue completada.
Rechazada	La tarea no se considera válida o aceptada por el superior.
Incumplida	La tarea no fue cumplida según lo indicado.

Tabla #1. Descripción de los estados de las tareas.

Además un usuario, sólo podrá modificar las tareas que él mismo haya registrado, no así las tareas que le son asignadas por el jefe de departamento o los jefes de áreas, de las cuales sólo podrá cambiarle el estado a completada en caso de que sea planificada. Las tareas completadas de todos los usuarios estarán pendientes a ser verificadas por el evaluador. Estas tareas solo estarán disponibles hasta que plan de resultados sea cerrado.

La evaluación de desempeño será generada por el evaluador, en la misma se darán algunos datos que apoyarán la evaluación de cada esfera, como es la cantidad de resultados con adecuado, deficiente y superior, además de la cantidad de tareas adicionales que el usuario propuso y fueron aceptadas por el evaluador. El evaluador podrá introducir, una evaluación por cada esfera y la evaluación final, así como las observaciones que estime pertinentes realizar. Los usuarios podrán consultar sus evaluaciones en todo momento.

Los administradores podrán gestionar todo lo referente a asignaturas, roles, usuarios y esferas, los cuales son necesarios para el correcto funcionamiento del sistema, además de la configuración, en la que se guardará el área (facultades) donde radica el departamento y los porcentos que servirán de indicadores para determinar la evaluación a nivel de resultados.

El sistema permitirá llevar un perfil con información esencial de cada usuario, lo cual servirá para tomar decisiones que puedan afectar a los mismos, este perfil es gestionado por el jefe de departamento y puede ser consultado por todos.

2.4.1. Descripción de los actores del sistema

Los actores del sistema representan a cualquiera que deba interactuar con él (inclusive si fuera un sistema ya existente)

Los actores del sistema:

- ▶ No son parte de él.
- ▶ Pueden intercambiar información con él.
- ▶ Pueden ser un recipiente pasivo de información.
- ▶ Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

Actores	Justificación
Usuario	Es aquel que generaliza la funcionalidad de autenticación al sistema, tiene el rol básico dentro del mismo, por lo general son profesores o trabajadores del departamento, los cuales responden a un superior.
Jefe de área	Es aquel que gestiona los indicadores y las tareas de otros recursos humanos en el departamento, pero concernientes a su área.
Jefe de departamento	Es el encargado de gestionar las asignaturas del departamento, así como los planes de trabajo y la evaluación de desempeño de sus subordinados.

Administrador del sistema	Es aquel que gestiona todo lo referente a la administración. Gestiona roles, usuarios, permisos y esferas.
----------------------------------	--

Tabla #2. Descripción de los actores del sistema.

Para llevar a cabo el desarrollo de todo lo antes mencionado se generaron un conjunto de artefactos que se exponen a continuación.

2.4.2. Estructura lógica de la aplicación

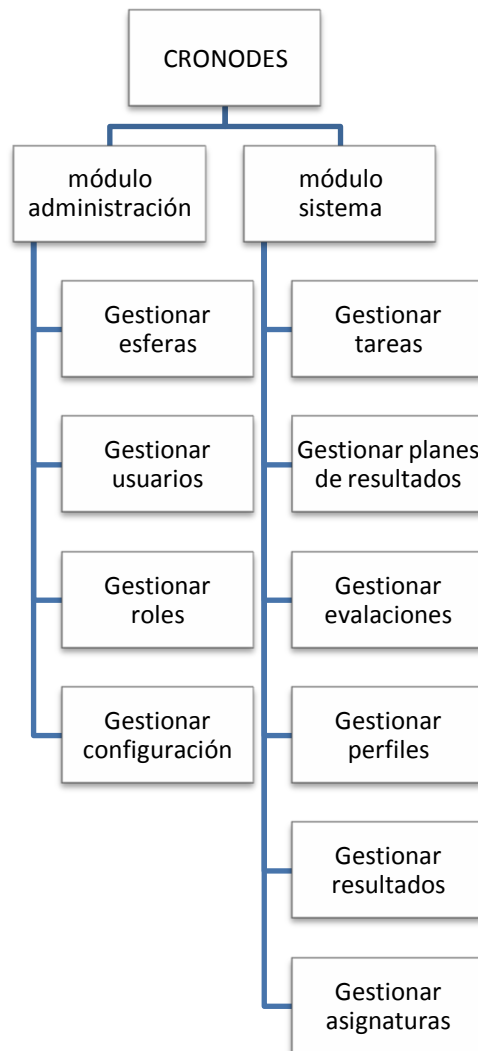


Figura 3. Estructura lógica de la aplicación

2.4.3. Casos de Uso del Sistema

Los Casos de uso del sistema son procesos que responden a las funcionalidades definidas en los requerimientos funcionales. Representan las funcionalidades que debe hacer el sistema.

2.4.3.1. Modelado del diagrama de CU del sistema

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

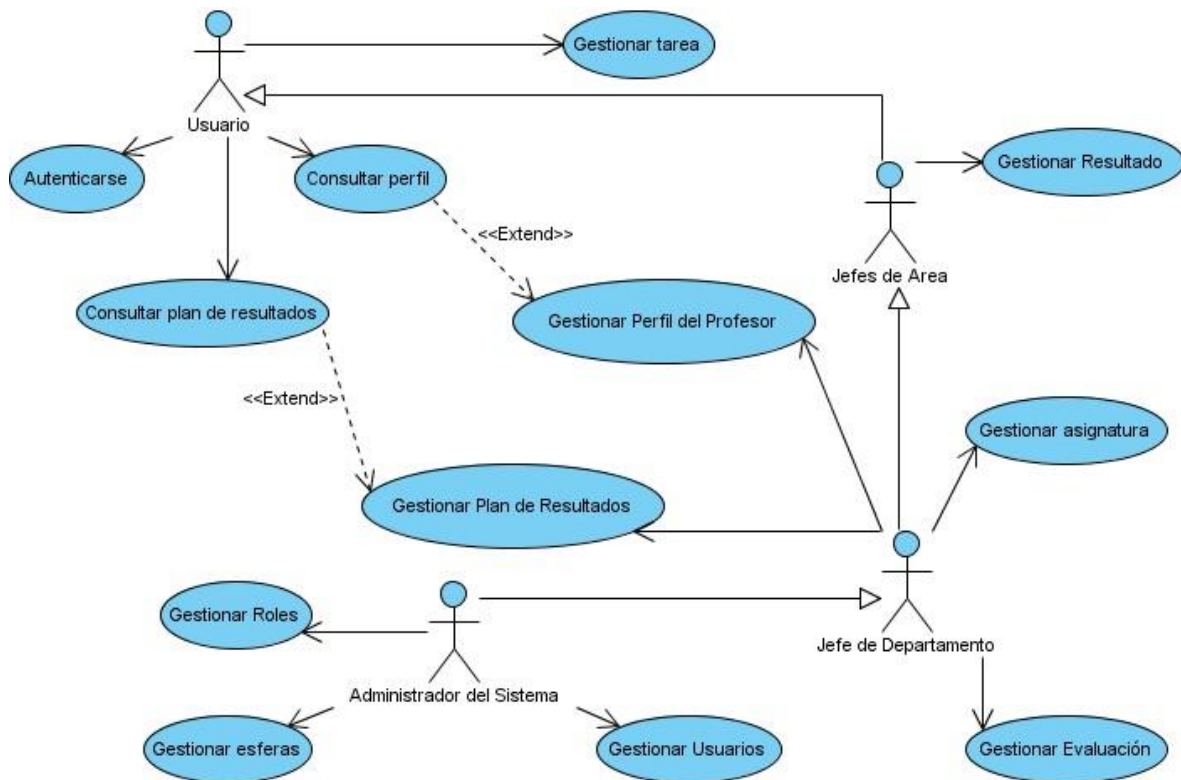


Figura 4. Diagrama de Casos de Uso (CU) del Sistema

2.4.3.2. Descripción expandida de los Casos de Uso del sistema

Se describen a continuación los Casos de uso más significativos del sistema propuesto.

Tabla 3. Caso de uso: Gestionar tareas

Caso de Uso:	Gestionar tareas
Actores:	Usuario, Jefe de área, Jefe de departamento, Administrador del Sistema.

Resumen:	El Caso de uso se inicia cuando el actor solicita Gestionar tareas, con el fin de agregar, eliminar o modificar una de ellas.
Precondiciones:	Los actores deben estar autenticados en el sistema
Referencias:	RF-2,RF-3,RF-4,RF-5, RF-6, RF-6.1, RF-6.2, RF-6.3
Prioridad:	Alta
Complejidad:	Media
Sección 1: Gestionar tareas	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción "Tareas"	1- El sistema muestra una lista de las tareas existentes por cada esfera.
	2- Muestra las opciones de eliminar, "ver detalles" y modificar correspondiente a cada tarea que se muestra en la lista, además de un vínculo "Adicionar tarea" para registrar una nueva.
Sección 1: Insertar tarea	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción "Adicionar Tarea"	2 – El sistema muestra un formulario con los campos a completar de la tarea: Datos: <div style="margin-left: 40px;"> Usuario(en dependencia del rol se podrá seleccionar el usuario al cual se le asignará la tarea) Título Lugar Esfera Estado </div>

	Fecha inicial
	Hora inicial
	Fecha final
	Hora final
	Tipo
	Descripción
	Resultados
3- Introduce los datos de la tarea y selecciona la opción aceptar.	
	4- El sistema valida los datos de entrada
	5- Registra los datos de la tarea
	6- Muestra el mensaje “Se ha registrado la tarea satisfactoriamente” y regresa a la interfaz “Adicionar tarea”.
Flujo Alterno	
4.a Si los datos no están correctos	
Acción del Actor	Respuesta del Sistema
	4. a.1 se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.
	4.a.2 Se regresa al paso 2
Sección 2: Eliminar tarea	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la tarea que desea eliminar	
2- Selecciona la opción “Eliminar”	
	3- Verifica que la tarea fue introducida por el mismo usuario.
	4- Muestra el mensaje “Tarea eliminada satisfactoriamente” y regresa a la interfaz “Gestionar tareas” con la lista

de las tareas actualizada.	
Flujo Alterno	
3.a Si la tarea fue asignada	
Acción del Actor	Respuesta del Sistema
	3. a.1 Se muestra un mensaje “No tiene privilegios para realizar esta operación” y regresa a la interfaz “Tareas”.
Sección 3: Modificar tarea	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la tarea que desea modificar	
2- Selecciona la opción “Modificar”	
	3- El sistema muestra un formulario con los datos de la tarea a modificar: Datos: Título Lugar Esfera Estado Fecha inicial Hora inicial Fecha final Hora final Tipo Descripción Resultados
4- Introduce los nuevos datos de la tarea y selecciona la opción aceptar.	
	5- El sistema valida los datos de entrada
	6- Registra los nuevos datos de la tarea
	7- Muestra el mensaje “Tarea modificada satisfactoriamente” y

	regresa a la interfaz “Detalles de la Tarea” con los datos de la tarea actualizados.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
5.a Si los datos no están correctos	
	5. a.1 Se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.
	5.a.2 Se regresa al paso 3.
Sección 3: Consultar tarea	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la tarea que desea ver en detalles	
2- Selecciona la opción “Ver detalles”	
	3- El sistema muestra todos los datos de la tarea Datos: Título Lugar Esfera Estado Fecha inicial Hora inicial Fecha final Hora final Tipo Descripción Resultados
Poscondiciones	Quedan insertados, modificados, eliminados o consultados los datos de una tarea.

Tabla 4. Caso de uso: Gestionar Plan de Resultados

Caso de Uso:	Gestionar Plan de Resultados
Actores:	Administrador de sistema, Jefe de Departamento, jefe de área, asistente, usuario.
Resumen:	El Caso de uso se inicia cuando el Jefe de Departamento solicita gestionar un Plan de Resultados, con el fin de agregar, eliminar o modificar el contenido del mismo.
Precondiciones:	El jefe de departamento debe estar autenticado en el sistema.
Referencias:	RF-7,RF-7.1,RF-8,RF-9
Prioridad:	Alta
Complejidad:	Media
Sección 1: Gestionar Plan de Resultados	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción "Plan de Resultados"	2- El sistema muestra un formulario para buscar la persona a la cual se le va a gestionar el plan de resultados.
	3- El sistema muestra una lista de los planes de resultados existentes para esa persona.
	4- Muestra las opciones de "eliminar", "ver detalles", "modificar" y "cerrar plan", correspondientes a cada plan de resultados que se muestra en la lista, además de un vínculo "Adicionar nuevo Plan de Resultados" para registrar uno nuevo.
Sección 1: Insertar Plan de Resultados	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción "Adicionar nuevo Plan	

de Resultados”	
	<p>2 – El sistema muestra un formulario con los campos a completar del plan de resultados:</p> <p>Datos:</p> <ul style="list-style-type: none"> Usuario Período Descripción Resultados Peso
3- Introduce los datos del plan y selecciona la opción aceptar.	
	4- El sistema valida los datos de entrada
	5- Registra los datos del plan de resultados.
	6- Muestra el mensaje “Se ha insertado el plan de resultados correctamente” y regresa a la interfaz “Nuevo Plan de Resultados”.
Flujo Alterno	
4.a Si los datos no están correctos	
Acción del Actor	Respuesta del Sistema
	4. a.1 se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.
	4.a.2 Se regresa al paso 2
Sección 2: Eliminar Plan de Resultados	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona el plan de resultados que desea	

eliminar	
2- Selecciona la opción "Eliminar"	
3- Muestra el mensaje "Plan de Resultados eliminado satisfactoriamente" y regresa a la interfaz "Listado de Plan de Resultados" con la lista de éstos actualizada.	
Sección 3: Modificar Plan de Resultados	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona el plan de resultados que desea modificar	
2- Selecciona la opción "Modificar"	
3- El sistema muestra un formulario con los datos del plan de resultados a modificar: Datos: Usuario Período Descripción Resultados Peso	
4- Introduce los nuevos datos del plan de resultados y selecciona la opción aceptar.	
5- El sistema valida los datos de entrada	
6- Registra los nuevos datos del plan de resultados.	
7- Muestra el mensaje "El Plan de resultados ha sido modificado satisfactoriamente" y regresa a la interfaz "Detalles del Plan de Resultados" con los datos del mismo	

actualizados.	
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
5.a Si los datos no están correctos	
	5. a.1 Se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.
	5.a.2 Se regresa al paso 3.
Poscondiciones	Queda creado, modificado o eliminado en la base de datos un plan de resultados.

Tabla 5. Caso de uso: Consultar Plan de Resultados

Caso de Uso:	Consultar Plan de Resultados
Actores:	Administrador de sistema, Jefe de Departamento, jefe de área, asistente, usuario.
Resumen:	El Caso de uso se inicia cuando el actor solicita consultar su Plan de Resultados.
Precondiciones:	El actor debe estar autenticado en el sistema.
Referencias:	RF-10
Prioridad:	Alta
Complejidad:	Media
Sección 1: Consultar Plan de Resultados	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona el plan de resultados que desea consultar	
2- Selecciona la opción “Ver Detalles”	
	2- El sistema muestra los datos referentes al plan de resultados.
Poscondiciones	Queda consultado por el actor un plan de resultados

Tabla 7. Caso de uso: Gestionar evaluación

Caso de Uso:	Gestionar evaluación
Actores:	Jefe de departamento
Resumen:	El Caso de uso se inicia cuando el jefe de departamento solicita gestionar la evaluación del profesor, con el fin de agregar, eliminar o modificar una de ellas.
Precondiciones:	El jefe de departamento debe estar autenticado en el sistema
Referencias:	RF-11,RF-12,RF-13,RF-14
Prioridad:	Alta
Complejidad:	Media
Sección 1: Gestionar evaluación	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción "Gestionar evaluación"	
	2- El sistema muestra un formulario para buscar la persona a la cual se le va a gestionar la evaluación.
	3- Muestra las opciones de eliminar y modificar correspondiente a cada evaluación que se muestra en la lista, además de un vinculo "ver" para mostrar todos los datos referentes a la misma y un botón "Adicionar" para registrar una nueva.
Sección 1: Insertar evaluación	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción "Adicionar"	
	2 – El sistema muestra el nombre completo de la persona, el tipo de período, el plan de trabajo que está activo, las tareas adicionales y las planificadas, estas últimas mediante un

	vínculo a su plan de resultados, además muestra la cantidad de resultados afectados y la cantidad de incumplimientos por esferas.
	3- Muestra un formulario con los campos a completar de la evaluación: Datos: Evaluación x Esferas Evaluación final Observaciones
4- Introduce los datos de la evaluación y selecciona la opción aceptar.	
	5- El sistema valida los datos de entrada
	6- Registra la evaluación
	7- Muestra el mensaje “Evaluación insertada satisfactoriamente” y regresa a la interfaz “Listado de Evaluaciones” con la lista de las evaluaciones actualizada.
Flujo Alterno	
5.a Si los datos no están correctos	
Acción del Actor	Respuesta del Sistema
	5. a.1 se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.
	5.a.2 Se regresa al paso 2
Sección 2: Eliminar evaluación	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la evaluación que desea eliminar	
2- Selecciona la opción “Eliminar”	

	3- Verifica que la evaluación fue introducida por el mismo usuario.
	4- Muestra el mensaje “Evaluación eliminada satisfactoriamente” y regresa a la interfaz “Gestionar evaluación” con la lista de las evaluaciones actualizada.
Flujo Alterno	
3.a Si la evaluación fue asignada por otro usuario	
Acción del Actor	Respuesta del Sistema
	3. a.1 Se muestra un mensaje “No tiene privilegios para realizar esta operación” y regresa a la interfaz “Gestionar evaluación”.
Sección 3: Modificar evaluación	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la evaluación que desea modificar	
2- Selecciona la opción “Modificar”	
	3 – El sistema muestra el nombre completo de la persona, el tipo de período, el plan de trabajo que está activo, las tareas adicionales y las planificadas, estas últimas mediante un vínculo a su plan de resultados, además muestra la cantidad de resultados afectados y la cantidad de incumplimientos por esferas.
	4- Muestra un formulario con los campos a modificar de la evaluación: Datos: Evaluación x Esferas Evaluación final Observaciones

5- Introduce los nuevos valores de la evaluación y selecciona la opción aceptar.	
6- El sistema valida los datos de entrada	
7- Registra los nuevos valores de la evaluación	
8- Muestra el mensaje “Evaluación modificada satisfactoriamente” y regresa a la interfaz “Gestionar evaluación” con la lista de las evaluaciones actualizada.	
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
6.a Si los datos no están correctos	
6. a.1 Se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.	
6.a.2 Se regresa al paso 3.	
Poscondiciones	Queda insertada, modificada o eliminada en la base de datos la evaluación de un profesor.

Tabla 6. Caso de uso: Gestionar Resultados

Caso de Uso:	Gestionar resultados
Actores:	Jefe de departamento, jefe de área
Resumen:	El Caso de uso se inicia cuando el actor solicita gestionar resultados, con el fin de agregar, eliminar o modificar uno de ellos.
Precondiciones:	Los actores deben estar autenticados en el sistema
Referencias:	RF-23,RF-24,RF-25,RF-26
Prioridad:	Alta
Complejidad:	Media
Sección 1: Gestionar resultados	
Flujo Básico	
Acción del Actor	Respuesta del Sistema

1- Selecciona la opción “Resultados”	
	2- Muestra una lista con los resultados existentes, así como las opciones de eliminar y modificar correspondiente a cada indicador que se muestra en la lista y un botón “Adicionar” para registrar uno nuevo.
Sección 1: Insertar resultado	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción “Adicionar”	
	2- Muestra un formulario con los campos a completar del resultado: Datos: Nombre Esfera a que pertenece Peso Fecha inicio
3- Introduce los datos del resultado y selecciona la opción aceptar.	
	4- El sistema valida los datos de entrada
	5- Registra el resultado
	6- Muestra el mensaje “Se ha adicionado correctamente” y regresa a la interfaz “Listado de Resultados” con la lista de los mismos actualizada.
Flujo Alterno	
4.a Si los datos no están correctos	
Acción del Actor	Respuesta del Sistema
	4. a.1 se muestra un mensaje “Datos incorrectos” y se señalan los campos

	que requieren revisión.
	4.a.2 Se regresa al paso 2
Sección 2: Eliminar resultado	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona el indicador que desea eliminar	
2- Selecciona la opción "Eliminar"	
	3- Verifica que el resultado fue introducido por el mismo usuario.
	4- Muestra el mensaje "Se ha eliminado satisfactoriamente" y regresa a la interfaz "Listado de Resultados" con la lista de los mismos actualizada.
Flujo Alterno	
3.a Si el resultado fue adicionado por otro usuario	
Acción del Actor	Respuesta del Sistema
	3. a.1 Se muestra un mensaje "No tiene privilegios para realizar esta operación" y regresa a la interfaz "Listado de Resultados".
Sección 3: Modificar resultados	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona el resultado que desea modificar	
2- Selecciona la opción "Modificar"	
	3- Muestra un formulario con los campos a modificar del resultado: Datos: Nombre Esfera a que pertenece Peso Fecha inicio

4- Introduce los nuevos valores del resultado y selecciona la opción aceptar.	
5- El sistema valida los datos de entrada	
6- Registra los nuevos valores del indicador.	
7- Muestra el mensaje “Se ha actualizado la información correctamente” y regresa a la interfaz “Listado de Resultados” con la lista de los indicadores actualizada.	
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
5.a Si los datos no están correctos	
5. a.1 Se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.	
5.a.2 Se regresa al paso 3.	
Poscondiciones	Queda insertado, modificado o eliminado un resultado en la base de datos.

Tabla 7. Caso de uso: Gestionar Usuario

Caso de Uso:	Gestionar Usuario
Actores:	Administrador del sistema
Resumen:	El Caso de uso se inicia cuando el actor solicita gestionar un Usuario, con el fin de agregar, eliminar o modificar uno de ellos.
Precondiciones:	El administrador del sistema debe estar autenticado en el sistema
Referencias:	RF-34,RF-35,RF-36,RF-37
Prioridad:	Alta
Complejidad:	Media
Sección 1: Gestionar usuario	
Flujo Básico	

Acción del Actor	Respuesta del Sistema
1- Selecciona la opción "Usuarios"	2- Muestra una lista con los usuarios existentes, así como las opciones de eliminar y modificar correspondiente a cada Usuario que se muestra en la lista y un botón "Adicionar" para insertar uno nuevo
Sección 1: Adicionar Usuario	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona la opción "Adicionar"	2- Muestra un listado con los recursos humanos de la facultad y un formulario con los campos requeridos para adicionar el usuario: Datos: Nombre Solapín Cargo Departamento Rol
3- Introduce los datos del usuario y selecciona la opción aceptar.	4- El sistema valida los datos de entrada
	5- Registra el usuario
	6- Muestra el mensaje "Usuario adicionado satisfactoriamente" y regresa a la interfaz "Adicionar usuario al Sistema".
Flujo Alterno	
4.a Si los datos no están correctos	

Acción del Actor	Respuesta del Sistema
	4. a.1 se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.
	4.a.2 Se regresa al paso 2
Sección 2: Eliminar usuario	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona el usuario que desea eliminar	
2- Selecciona la opción “Eliminar”	
	3- Verifica si el actor tiene permisos para eliminar el usuario
	4- Muestra el mensaje “Usuario eliminado satisfactoriamente” y regresa a la interfaz “Usuario” con la lista de los usuarios actualizada.
Flujo Alterno	
3.a Si el actor no tiene permisos	
Acción del Actor	Respuesta del Sistema
	3. a.1 Se muestra un mensaje “No tiene privilegios para realizar esta operación” y regresa a la interfaz “Usuario”.
Sección 3: Modificar usuario	
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1- Selecciona el usuario que desea modificar	
2- Selecciona la opción “Modificar”	
	3- Muestra un formulario con los campos a modificar del usuario: Datos: Nombre Solapín

	Cargo Departamento Rol
4- Introduce los nuevos valores del usuario y selecciona la opción aceptar.	
	5- El sistema valida los datos de entrada
	6- Registra los nuevos valores del usuario.
	7- Muestra el mensaje “Usuario modificado satisfactoriamente” y regresa a la interfaz “Usuario” con la lista de los usuarios actualizada.
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
5.a Si los datos no están correctos	
	5. a.1 Se muestra un mensaje “Datos incorrectos” y se señalan los campos que requieren revisión.
	5.a.2 Se regresa al paso 3.
Poscondiciones	Queda insertado, modificado o eliminado un usuario en la base de datos.

2.5. Conclusiones parciales

En este capítulo se realizó una amplia descripción del sistema propuesto, comenzando con la realización del modelo de dominio para una mejor comprensión del negocio. Se identificaron requerimientos funcionales y no funcionales a tener en cuenta para el desarrollo del sistema. Se obtuvo el diagrama de Casos de Uso del sistema y se describieron los mismos para su mejor entendimiento.

El estudio realizado sobre el negocio ha facilitado la comprensión de la dinámica y estructura de la organización para la que se realiza el trabajo, de esta manera se han

creado las condiciones para realizar el diseño del sistema respetando las especificaciones que nos plantea el estudio del negocio ya analizado.

3. CAPÍTULO 3: Construcción de la Solución propuesta

3.1. Introducción

El siguiente capítulo se encuentra dividido en dos partes fundamentales: en la primera se realiza el análisis y diseño del sistema, donde se obtienen los artefactos necesarios para cada caso de uso teniendo en cuenta los requisitos funcionales y no funcionales que se especificaron en el capítulo anterior. En la segunda parte se procede a la implementación de la aplicación, donde se mostrará el modelo de implementación mediante el diagrama de componentes, el cual se realiza según la información que se obtiene del diseño.

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.2. Modelo de análisis

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución.

3.2.1. Clases de análisis

Se 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.

3.2.2. Diagrama de clases de análisis

Un diagrama de clases de análisis es un artefacto en el que se representan los conceptos en un dominio del problema, representa las cosas del mundo real. El modelo de análisis puede considerarse como una primera aproximación al modelo de diseño.

A continuación se muestran los diagramas de clases del análisis de los casos de uso más significativos del sistema:

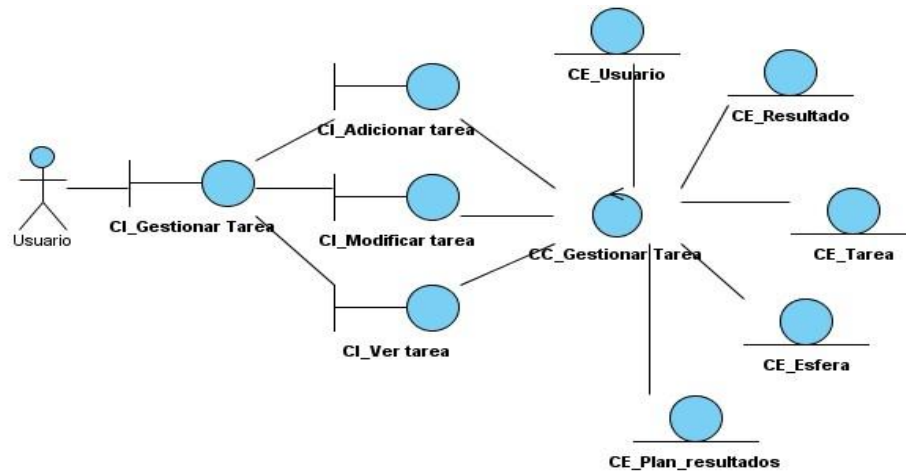


Figura 5. Diagrama de Clases del Análisis del CU Gestionar Tarea.

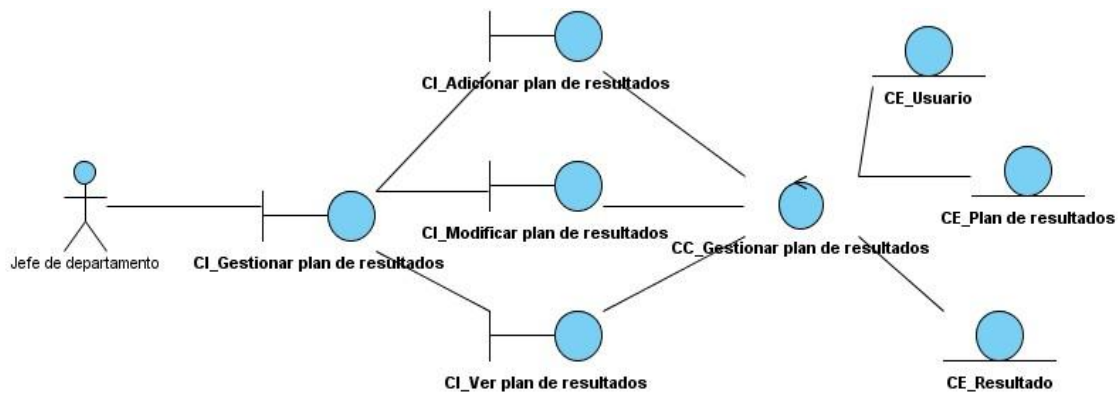


Figura 6. Diagrama de Clases del Análisis del CU Gestionar Plan de resultados.

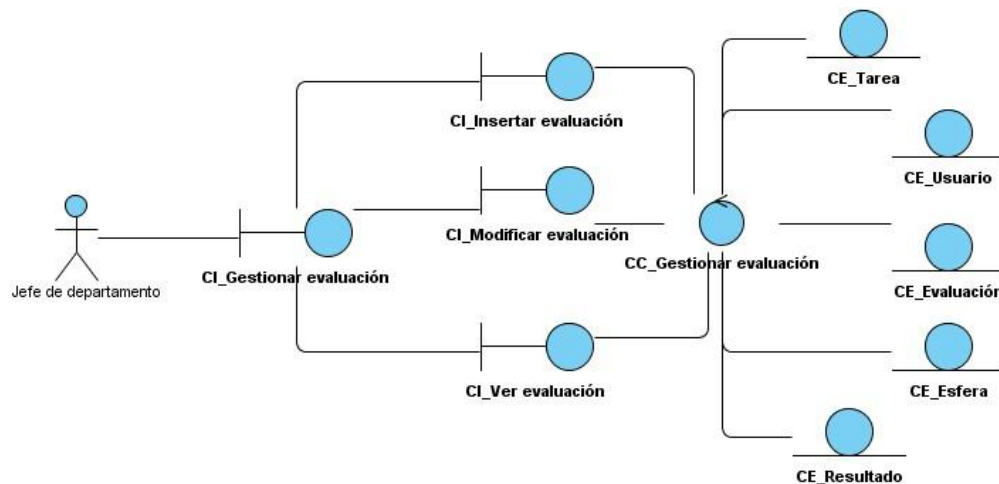


Figura 7. Diagrama de Clases del Análisis del CU Gestionar Evaluación.

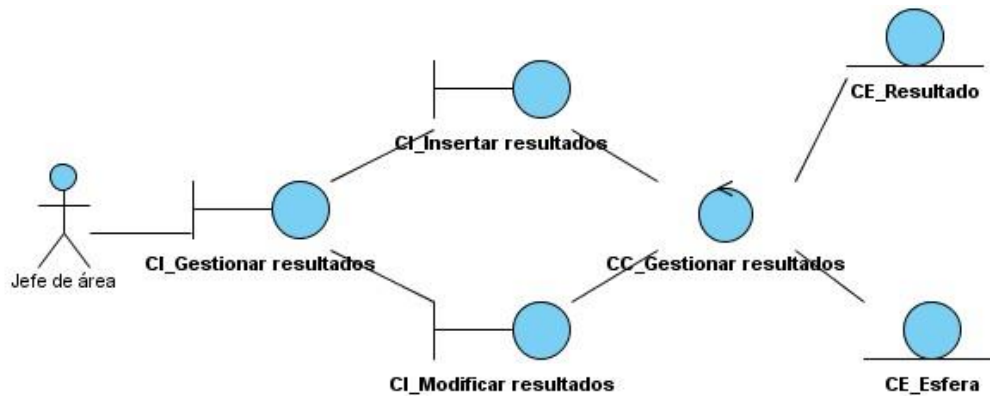


Figura 8. Diagrama de Clases del Análisis del CU Gestionar Resultados.

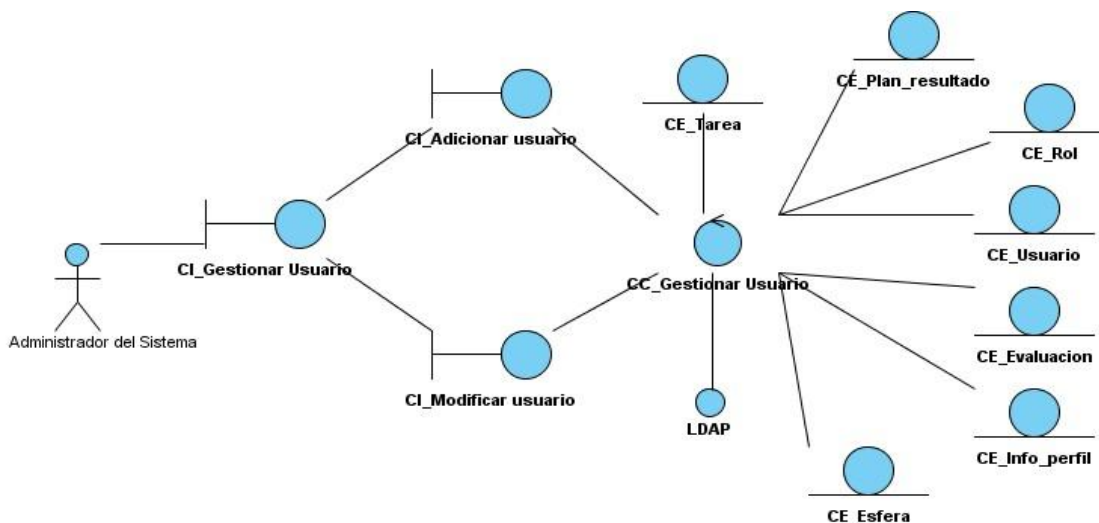


Figura 9. Diagrama de Clases del Análisis del CU Gestionar Usuario.

3.3. Modelo de diseño

Es un modelo de objetos que describe la realización de los casos de uso centrándose en el cumplimiento de los requisitos funcionales y no funcionales. Adquiere una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos y tecnologías de interfaz de usuario.

Además ayuda a descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por equipos de desarrollo.

3.3.1. Diagrama de clases del diseño

Un diagrama de clases contiene las clases de diseño que participan en la realización de un Caso de Uso. A continuación se muestran los diagramas de clases del diseño de una muestra de casos de uso que se decidió plasmar en el documento.

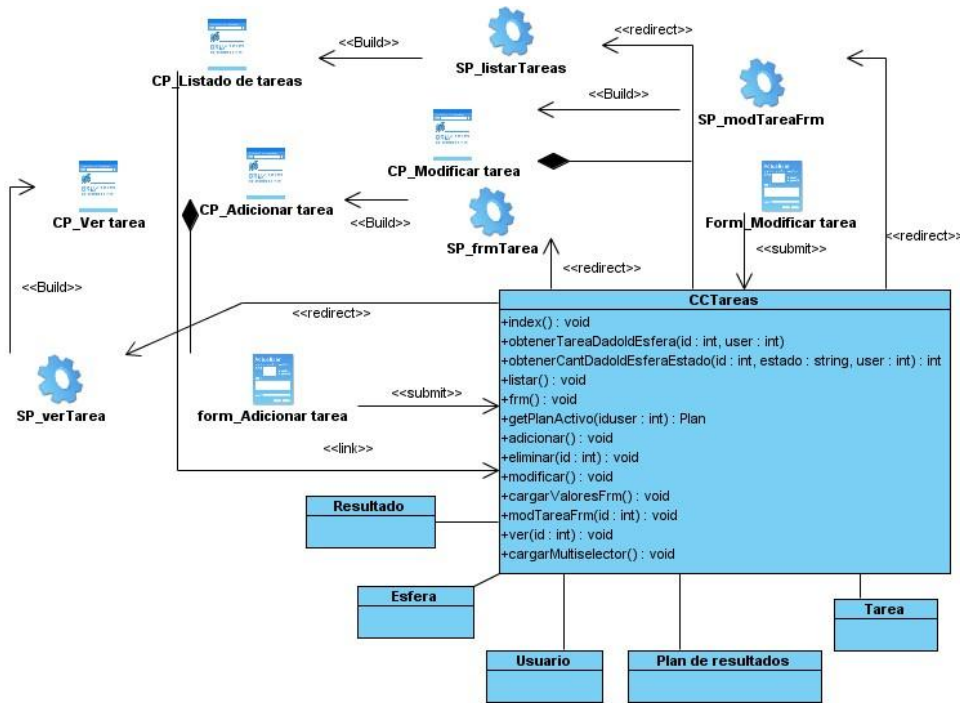


Figura 10. Diagrama de Clases del Diseño del CU Gestionar Tareas.

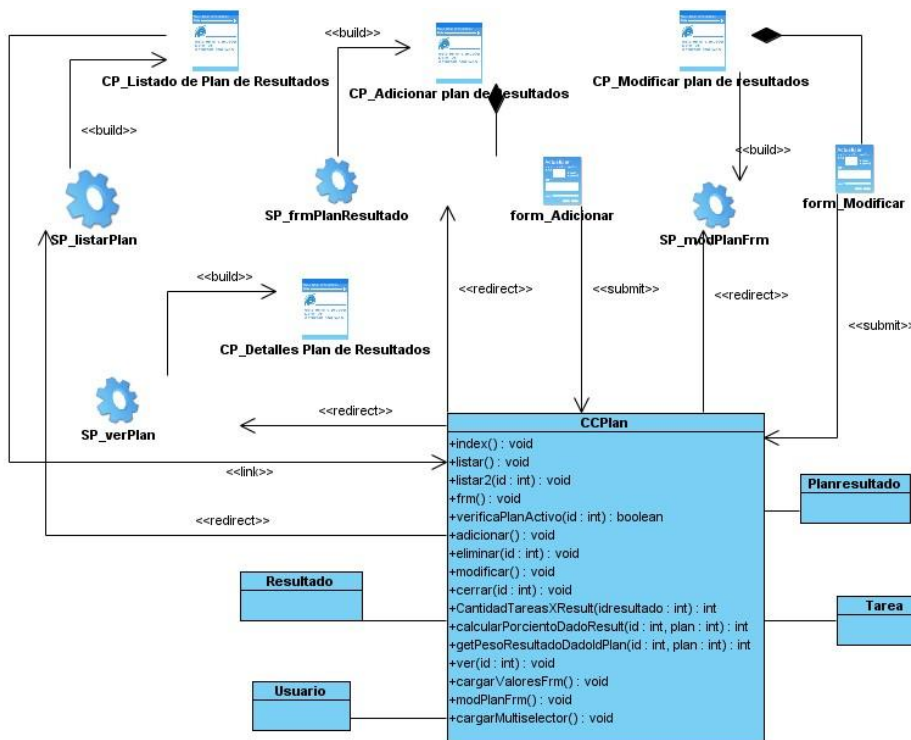


Figura 11. Diagrama de Clases del Diseño del CU Gestionar Plan de resultados.

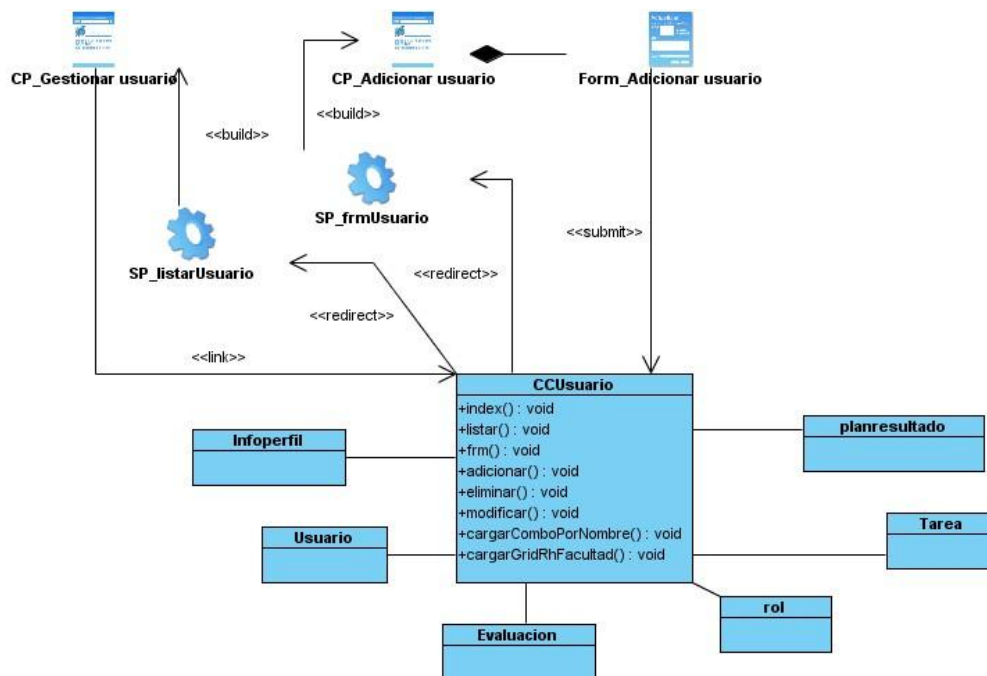


Figura 14. Diagrama de Clases del Diseño del CU Gestionar Usuario.

3.3.2. Descripción de las principales clases del sistema.

Nombre: CCTarea	
Tipo de clase: controladora	
Funcionalidades	
Nombre:	index()
Descripción:	Permite mostrar la información por defecto, cuando se accede a la controladora, en este caso se hace llamado a listar().
Nombre:	listar()
Descripción:	Conforma las estadísticas y las tareas del usuario autenticado, los cuales son pasados a la vista listarTareas.
Nombre:	obtenerTareaDadoldEsfera(\$id, \$user)
Descripción:	Retorna las tareas dado un id de esfera y un id de usuario determinado.
Nombre:	obtenerCantDadoldEsferaEstado(\$id, \$estado, \$user)
Descripción:	Retorna la cantidad de tareas dado un id de esfera, un estado y un id de usuario determinado.
Nombre:	frm()
Descripción:	Hace un llamado a la vista frmTarea, para mostrar el formulario de inserción de una nueva tarea.
Nombre:	getPlanActivo(\$idUser)
Descripción:	Retorna el plan activo dado el id de un usuario.
Nombre:	adicionar()
Descripción:	Permite adicionar una tarea al sistema.
Nombre:	modificar ()
Descripción:	Permite modificar una tarea seleccionada por el usuario.
Nombre:	eliminar(\$id)
Descripción:	Permite eliminar una tarea del sistema seleccionada por el usuario.
Nombre:	cargarValoresFrm()
Descripción:	Permite cargar los valores que se muestran a la hora de modificar una tarea.
Nombre:	modPerfilFrm(\$id)
Descripción:	Hace un llamado a la vista modTareaFrm para mostrar el formulario con los datos de la tarea a modificar.
Nombre:	ver(\$id)

Descripción:	Permite mostrar una tarea en detalle, dado el id de la misma, hace llamado a la vista verTarea.
Nombre:	cargarMultiselector()
Descripción:	Retorna un json con los resultados asociados a una esfera determinada, para ser adicionados a la tarea actual.

Tabla 8. Descripción de la Clase CCTarea.

Nombre: CCplan	
Tipo de clase: controladora	
Funcionalidades	
Nombre:	index()
Descripción:	Permite mostrar la información por defecto, cuando se accede a la controladora, en este caso se hace llamado a listar().
Nombre:	listar()
Descripción:	Lista los planes de resultados existente para un usuario determinado, hace llamado a la vista listarPlan.
Nombre:	frm()
Descripción:	Hace un llamado a la vista frmPlanResultado para mostrar el formulario de inserción de un nuevo plan.
Nombre:	verificaPlanActivo(\$iduser)
Descripción:	Verifica dado un usuario, si tiene un plan activo.
Nombre:	adicionar()
Descripción:	Permite adicionar un plan al sistema.
Nombre:	modificar ()
Descripción:	Permite modificar un plan seleccionado por el usuario.
Nombre:	eliminar(\$id)
Descripción:	Permite eliminar un plan seleccionado por el usuario.
Nombre:	cerrar(\$id)
Descripción:	Permite cerrar un plan seleccionado por el usuario.
Nombre:	CantidadTareasXResult(\$idresultado, \$periodo)
Descripción:	Permite obtener la cantidad de tareas que contiene un resultado dado el id del mismo y el período de evaluación.
Nombre:	calcularPorcentajeDadoResult (\$id, \$plan)
Descripción:	Permite calcular el porcentaje de cumplimiento de un resultado, dado el id del mismo y el plan.
Nombre:	getPesoResultadoDadoldPlan(\$id,\$plan)
Descripción:	Permite obtener el peso de un indicador dado el id y el plan al que pertenece.
Nombre:	ver(\$id)
Descripción:	Permite ver en detalles un plan seleccionado por el usuario.
Nombre:	cargarValoresFrm()
Descripción:	Permite cargar los valores que se muestran a la hora de modificar un plan.
Nombre:	modPlanFrm (\$id)
Descripción:	Hace un llamado a la vista modPlanFrm para mostrar el formulario con los datos del plan a modificar.
Nombre:	cargarMultiselector()
Descripción:	Retorna un json con los resultados asociados a una esfera determinada para ser adicionados al plan.

Tabla 9. Descripción de la Clase CCplan.

Nombre: CCeval	
Tipo de clase: controladora	
Funcionalidades	
Nombre:	index()
Descripción:	Permite mostrar la información por defecto, cuando se accede a la controladora, en este caso se hace llamado a listar().
Nombre:	listar()
Descripción:	Lista las evaluaciones de desempeño existente para un usuario determinado, hace llamado a la vista listarEval.
Nombre:	frm()
Descripción:	Hace un llamado a la vista frmEval para mostrar el formulario de inserción de una nueva evaluación.

Nombre:	adicionar()
Descripción:	Permite adicionar una evaluación al sistema.
Nombre:	modificar ()
Descripción:	Permite modificar una evaluación seleccionada por el usuario.
Nombre:	eliminar(\$id)
Descripción:	Permite eliminar una evaluación seleccionado por el usuario.
Nombre:	ver(\$id)
Descripción:	Permite ver en detalles una evaluación seleccionado por el usuario.
Nombre:	modPerfilFrm (\$id)
Descripción:	Hace un llamado a la vista modEvalFrm para mostrar el formulario con los datos de la evaluación a modificar.

Tabla 10. Descripción de la Clase CCEval.

Nombre: CCusuario	
Tipo de clase: controladora	
Funcionalidades	
Nombre:	index()
Descripción:	Permite mostrar la información por defecto, cuando se accede a la controladora, en este caso se hace llamado a listar().
Nombre:	listar()
Descripción:	Lista los usuarios existentes, hace llamado a la vista listarUsuario.
Nombre:	frm()
Descripción:	Hace un llamado a la vista frmUsuario para mostrar el formulario de inserción de un nuevo usuario. Además crea un perfil de dicho usuario.
Nombre:	modificar ()
Descripción:	Permite modificar el rol de un usuario seleccionado por el administrador.
Nombre:	eliminar(\$id)
Descripción:	Permite eliminar un usuario y consigo su perfil.
Nombre:	modUsuarioFrm (\$id)
Descripción:	Hace un llamado a la vista modUsuarioFrm para mostrar el formulario con los datos del perfil a modificar.

Tabla 11. Descripción de la Clase CCusuario.

3.3.3. Diseño de la Base de Datos (BD)

▣ **Modelo de datos.**

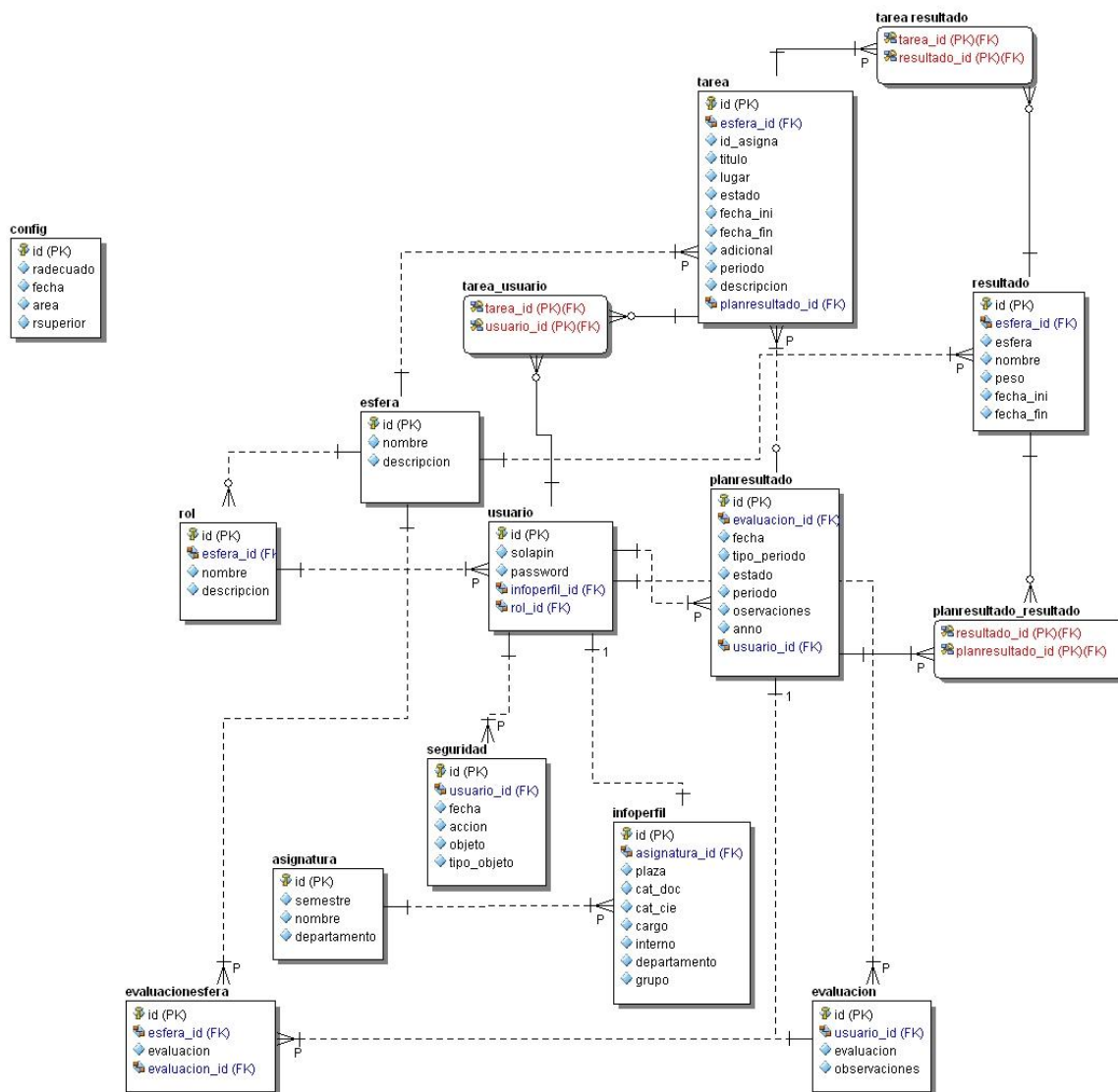


Figura 15. Modelo de Datos.

3.3.4. Descripción de las principales tablas.

Nombre: tarea		
Descripción: Contiene información referente a las tareas que le son asignadas a cada profesor dentro del departamento, éstas pueden ser adicionales o planificadas por el jefe de departamento.		
Atributo	Tipo	Descripción
Id_tarea.	INTEGER	Identificador en la tabla de de cada tarea asignada
Id_esfera	INTEGER	Identificador en la tabla de la esfera a la que corresponde la tarea.
Id_asigna	CHAR	Identificador en la tabla de cada tarea asignada
título	VARCHAR	Título que se le da a la tarea.
lugar	VARCHAR	Lugar donde se llevará a cabo la tarea
estado	VARCHAR	El estado en que se encuentra la tarea, el cual puede

		ser Rechazada, Incumplida, Planificada, Verificada o Completada.
fecha_ini	DATETIME	Fecha de inicio de la tarea
fecha_fin	DATETIME	Fecha de finalización de la tarea
adicional	VARCHAR	Si la tarea es Adicional(A) o pertenece al Plan (P)
descripción	VARCHAR	Descripción y características de la tarea en cuestión.
planresultado_id	INTEGER	Identificador en la tabla del plan de resultados al que pertenece la tarea.

Tabla 12. Descripción de la Tabla “tarea”.

Nombre: evaluación		
Descripción: Contiene información acerca de la evaluación asignada a cada profesores según su desempeño y cumplimiento del plan de resultados durante un período de tiempo.		
Atributo	Tipo	Descripción
id	INTEGER	Identificador en la tabla de cada evaluación.
usuario_id	INTEGER	Identificador en la tabla del usuario al cual corresponde la evaluación.
evaluacion	CHAR	Nombre de la evaluación que obtuvo el usuario. Ésta puede ser Adecuado, Deficiente o Superior.
observaciones	VARCHAR	Se describen las observaciones que pueda presentar la evaluación.

Tabla 13. Descripción de la Tabla “evaluación”.

Nombre: planresultado		
Descripción: Contiene información referente al plan de resultados de un profesor dentro del departamento en un período determinado.		
Atributo	Tipo	Descripción
id	INTEGER	Identificador en la tabla de cada plan de resultados.
fecha	DATE	Fecha en la que se creó el plan de resultados.
tipo_periodo	VARCHAR	El tipo de período a evaluar, que puede ser trimestral, anual o mensual
estado	VARCHAR	Estado en el que se encuentra el plan de resultados. Este puede ser Activo o Bloqueado.
periodo	VARCHAR	Período al que pertenece el plan.
observaciones	VARCHAR	Observaciones que desee especificar el creador del plan.
anno	VARCHAR	Año en el que se creó el plan
usuario_id	INTEGER	Identificador en la tabla, del usuario al que pertenece el plan de resultados.

Tabla 14. Descripción de la Tabla “planresultado”.

Nombre: usuario		
Descripción: Contiene información referente a los usuarios del sistema.		
Atributo	Tipo	Descripción
id	INTEGER	Identificador en la tabla de cada usuario.
solapin	VARCHAR	Numero del solapín del usuario.

infoperfil_id	INTEGER	Identificador del perfil de cada usuario.
rol_id	INTEGER	Identificador del rol que tiene usuario en el sistema.

Tabla 15. Descripción de la Tabla “usuario”.

3.4. Implementación

3.4.1. Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema.

Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos.(SCHMULLER 2000)

Para representar los diagramas del Modelo de Implementación se puede emplear el diagrama de Componentes.

3.4.1.1. 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. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.(SCHMULLER 2000)

A continuación se mostrarán los diagramas de componentes de los CU más significativos.

- ▢ Diagrama de componentes.

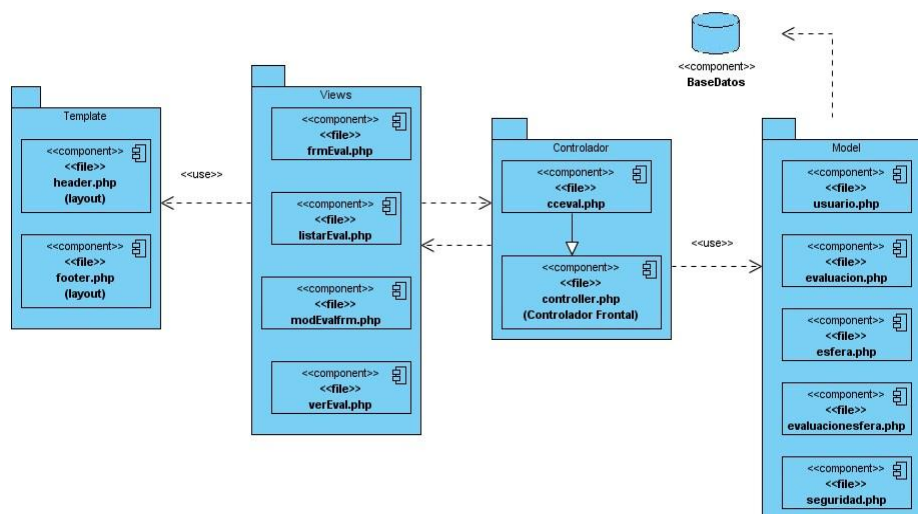


Figura 16. Diagrama de Componentes del CU Gestionar evaluación.

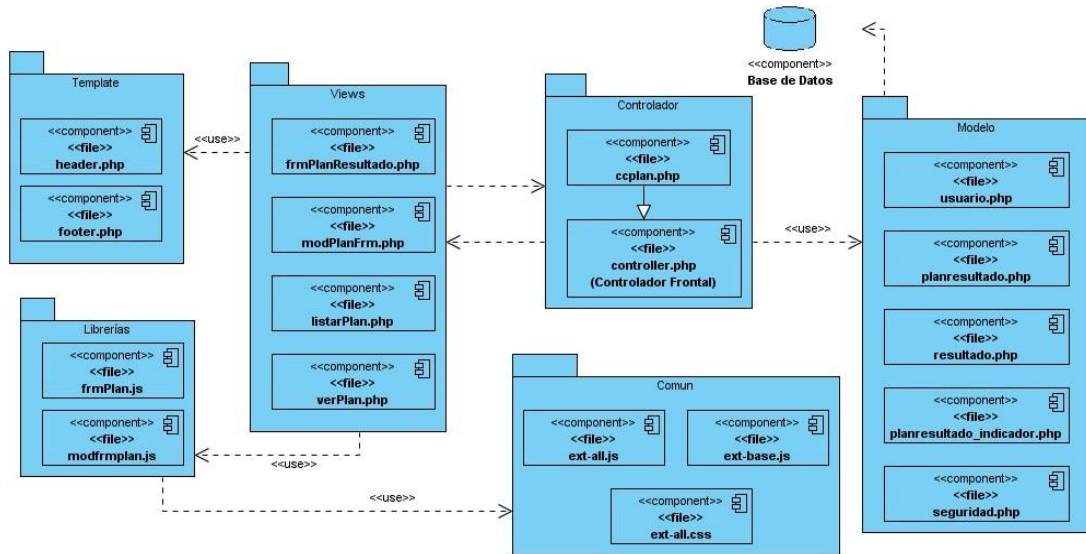


Figura 17. Diagrama de Componentes del CU Gestionar Plan de resultados.

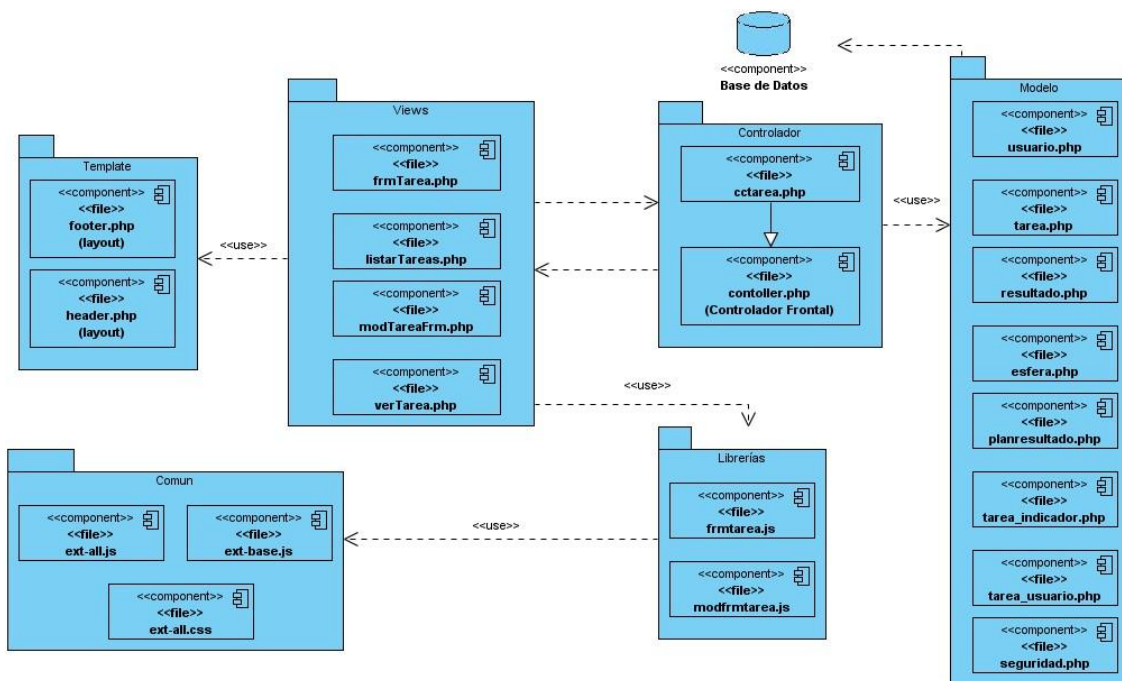


Figura 18. Diagrama de Componentes del CU Gestionar Tareas.

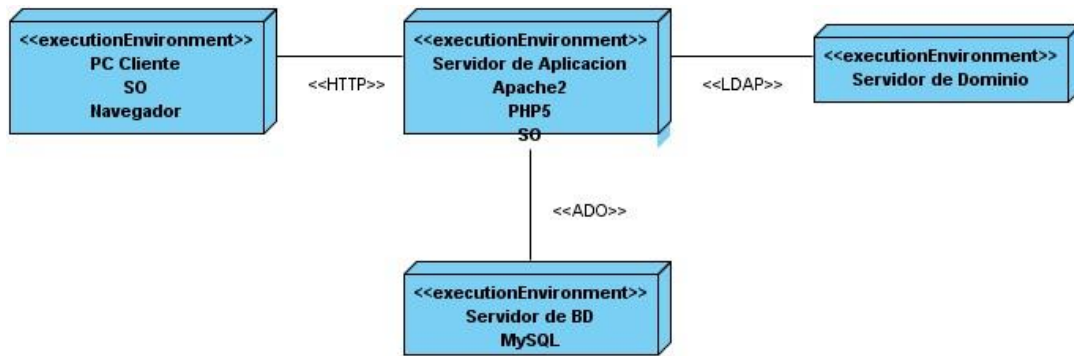


Figura 21. Diagrama de Despliegue

3.5. Conclusiones Parciales

Es en este capítulo donde se concreta todo el estudio y análisis realizado en los capítulos anteriores y se logra minimizar en gran medida la distancia existente entre la concepción de lo que se quería del sistema y la aplicación a desarrollar. Como resultado del estudio realizado correspondiente a las etapas de análisis, diseño e implementación del sistema, se modelaron los diagramas de clases para cada uno de los casos de uso del sistema. Se obtuvo el modelo de datos y se realizaron las descripciones de las tablas con sus atributos. Finalmente se muestran los diagramas de componentes para los casos de usos más significativos y el diagrama de despliegue.

Conclusiones Generales

El desarrollo de este trabajo ha dado como resultado una serie de contribuciones sobre el tema:

- ▶ Se realizó un estudio de los diferentes sistemas que existen a nivel mundial y en nuestro país en lo que a la gestión de la evaluación del desempeño se refiere, llegándose a la conclusión de que aunque existen varias herramientas, ninguna cumple con los requerimientos necesarios para la evaluación del desempeño de los trabajadores en los departamentos docentes de la universidad.
- ▶ Se realizó un estudio de las tecnologías más convenientes para llevar a cabo el desarrollo de la solución.
- ▶ Se realizó la modelación y generación de varios artefactos que permitieron la implementación y apoyarán el entendimiento de futuros mantenimientos, soporte y desarrollos.
- ▶ Como principal resultado se obtuvo una herramienta informática capaz de gestionar la información necesaria e integrar, los procesos de evaluación de desempeño y los planes de resultados en el departamento docente.

Recomendaciones

Los objetivos generales de este trabajo fueron alcanzados, pero durante su desarrollo, surgieron varios temas que son recomendables para futuras versiones de la herramienta:

- ▶ Notificaciones sobre tareas, planes y evaluaciones vía correo electrónico.
- ▶ Adicionar al mecanismo de evaluación, los controles a clases.
- ▶ Implementar un mecanismo que permita evaluar a los evaluadores.
- ▶ Implementar un conjunto de reportes según las necesidades que van surgiendo.

Bibliografía

1. DOJOTOOLKIT. Documentación Dojo JavaScript Toolkit (2010). 2010. [Disponible en: <http://dojotoolkit.org/docs>
2. GROUP, B. Sistemas de gestión, 2010. [Disponible en: <http://www.bsigroup.es/es/certificacion-y-auditoria/Sistemas-de-gestion/De-un-vistazo/que-son-sistemas-de-gestion/>]
3. JQUERY. Documentación JQuery JavaScript Library (2010). 2010. [Disponible en: http://docs.jquery.com/Main_Page
4. CAKEPHP. CakePHP:The rapid development php Framework, 2008. [Disponible en: <http://www.cakephp.org>
5. CODEIGNITER. 2009. [Disponible en: <http://codeigniter.com>
6. DATALOGIC. GIRH - Gestion Integral de Recursos Humanos, 2009. [Disponible en: http://www.datalogic.com.uy/index.php?option=com_content&task=view&id=25&Itemid=64
7. DATE, C. J. Introducción a los sistemas de Bases de Datos. Parte I, II y III, 2003.
8. DOJOTOOLKIT. Documentación Dojo JavaScript Toolkit (2010). 2010. [Disponible en: <http://dojotoolkit.org/docs>
9. DOTPROJECT. 2009. [Disponible en: <http://www.dotproject.net/>
10. EFEEDBACK. Sistema de Evaluación del Desempeño, 2008. [Disponible en: <http://www.iic.uam.es/flash/efeedbackiicflash.html>
11. EGUÍLUZ PÉREZ, J. CSS avanzado, 2005.
12. EXTJS. Ext JS, 2009. [Disponible en: <http://www.extjs.com>
13. GESTIÓN, A. Y. Sistemas de Gestión, 2009. [Disponible en: (<http://www.gestionyadministracion.com/empresas/sistemas-de-gestion.html>)
14. GROUP, B. Sistemas de gestión, 2010. [Disponible en: <http://www.bsigroup.es/es/certificacion-y-auditoria/Sistemas-de-gestion/De-un-vistazo/que-son-sistemas-de-gestion/>]
15. HAYA, H. R. U. C. Evaluacion del Desempeño, 2009. [Disponible en: www.carloshaya.net/recursoshumanos/files/manuales/epd.pdf
16. IDRC. Centro Internacional de Investigaciones para el Desarrollo, 2007. [Disponible en: http://www.idrc.ca/es/ev-30231-201-1-DO_TOPIC.html
17. JACOBSON, I. El Proceso Unificado de Desarrollo de Software. 2000. p.

18. JOHNSON , F. Designing Reusable Classes Journal of Object-Oriented Programming, 1988: 35.
19. JOHNSON, R. E., RUSSO, V. F Reusing Object-Oriented Designs University of Technical Report, 1991.
20. JQUERY. Documentación JQuery JavaScript Library (2010). 2010. [Disponible en: http://docs.jquery.com/Main_Page
21. MATEU, C. Desarrollo de aplicaciones web, 2004.
22. MIC, M. D. I. Y. L. C. Resolución 116, 2009.
23. MICROSOFT. Microsoft Project Professional 2010, 2010. [Disponible en: http://www.microsoft.com/spain/office/project2010/Project_Professional.html
24. NAVARRA. Glosario de Términos relacionados con el seguimiento y evaluación de las Políticas Públicas, 2008. [Disponible en: www.navarra.es/NR/rdonlyres/1BB115C8-F6D6-4FF7-A13E-14FD2F9564D6/111097/GlosariodeterminosevaluacionpoliticaspUBLICAS.doc
25. PHPROJEKT. 2009. [Disponible en: <http://www.phprojekt.com/features>
26. REDMINE. Redmine 2009. [Disponible en: <http://www.redmine.org/projects/redmine/wiki>
27. ROGER S. PRESSMAN, M.-H. Ingeniería del Software. Un enfoque práctico. 5ª edición. 2001. p.
28. SCHMULLER, J. Aprendiendo UML en 24 horas. 2000. p.
29. SYMFONY. ¿Qué es Symfony?, 2009. [Disponible en: <http://www.symfony.es/que-es-symfony>
30. WILSOFT. QTraining, 2008. [Disponible en: <http://www.wilsoft-la.com/qtraining.htm>
31. XPERIMENTOS. Los mejores 12 FrameWorks JavaScript, 2007. [Disponible en: <http://www.xperimentos.com/2007/09/04/los-mejores-12-frameworks-javascript>
32. YAHOO. Yahoo Developer Network, 2008. [Disponible en: <http://developer.yahoo.com/yui>
33. ZAMMETTI, F. W. Practical Ext JS Projects with Gears, 2009.
34. ZF, Z. F. What is Zend Framework?, 2010. [Disponible en: <http://framework.zend.com/wiki/pages/viewpage.action?pageId=10530>

Glosario de Términos

1 Tickets: Sistema que utiliza dotProject como helpDesk. Cada ticket representa un problema que se le ha presentado a un usuario del sistema. Tienen un status de procesamiento como: Abiertos, Procesando, Cerrado y Eliminado. También tienen una prioridad: Baja, Normal, Alta y la mas alta.

2 RSS: Es un formato de documento. Está basado en el XML conforme especificaciones publicadas por el World Wide Web Consortium (W3C). Constituyen una familia de documentos para actualizar las novedades y noticias de un sitio web.

3 MIT (Massachusetts Institute of Technology): Esta licencia permite reutilizar el Software así licenciado tanto para ser software libre como para ser software no libre.

4 JSON: (JavaScript Object Notation) Notación de Objetos JavaScript, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

5 CRUD: (Create, Retrieve, Update, Delete) Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un sistema de software.

6 ORM: (Object Relational Mapping) Mapeo de Objetos Relacional es una tecnica de programación para convertir tipos de datos incompatibles entre sistemas de bases de datos y lenguajes orientados a objetos.

7 API: Una interfaz de programación de aplicaciones o API (Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

8 Widgets: Un widget es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine. Entre sus objetivos están los de dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.