

Universidad de las Ciencias Informáticas



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

Título: “Herramienta para la Gestión del Rendimiento del Personal en el proyecto ERP-Cuba.”

Autores: Elier Rodríguez González

Maikel Pérez Torres

Tutores: Ing. Pedro Manuel Nogales Cobas

Ing. Cealys Alvarez Trujillo

Ciudad de La Habana, Cuba

Junio 2009

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

Aristóteles

DEDICATORIA

A mi familia quien me ha dado todo el amor del mundo y lo necesario para alcanzar este logro.

Elier

A mi familia por estar siempre a mi lado.

A la memoria de mi abuelo Manolito.

Maikel

AGRADECIMIENTOS

Agradezco en primer lugar a mis padres por todo el apoyo que me brindaron para salir adelante ante cualquier dificultad, a mi hermano Elieser y Jesus a todos mis sobrinos y en especial a mis abuelos maternos y paternos. Gracias por ser la familia que todos quisieran tener. A mi novia por su comprensión, a todos mis amigos que de una forma u otra me apoyaron y dieron su aporte a este trabajo. Por todo esto muchas gracias.

Elier.

Zuiero agradecer en primer lugar a mis padres Marlenis y Ernesto por apoyarme siempre y confiar en mí. A toda mi familia y en especial a mi abuela Noemis a mis tios Yennis, Manolin y Cecilia a mis primos Eduardito y Lázaro, los quiero mucho. A mi novia Tita por darme su amor. A Miguel Valera por cuidar bien a mi madre. A todos mis amigos por su ayuda y por aguantarme. A Martha y Andres por ser mis padres en la Habana. Muchas gracias a todos.

Maikel

AGRADECIMIENTOS

Zueremos agradecer a los tutores Pedro Manuel Nogales y Cealys Alvarez Trujillo, a nuestros amigos Reinier Blanco Zambrano, Rosalba Carralero Medina, Yuniel Rodríguez Bello, José Carlos Guevara, Diouis Proenza Labadie por ayudarnos en la realización de nuestra tesis, sin ustedes no habiéramos podido terminarla como queríamos.

A todos muchas gracias.

Maikel y Elier

RESUMEN

El país avanza con amplios pasos dentro del proceso de utilización ordenada y masiva de las tecnologías de la información y las comunicaciones. Factor decisivo para el desarrollo de las empresas, economía y de la sociedad cubana. Proceso que busca lograr más eficacia y eficiencia, satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad.

La Universidad de las Ciencias Informáticas se ha propuesto desarrollar un producto ERP usando tecnologías de software libre como una estrategia para desarrollar la economía del país. El proyecto cuenta con una infraestructura productiva compleja con un total de más de 400 trabajadores entre estudiantes, profesores y especialistas.

El deficiente control de las evaluaciones del desempeño del personal en el proyecto ERP-Cuba ha impulsado al desarrollo de una herramienta que evalúe el rendimiento del personal de una forma más eficiente. Esta herramienta es un sistema de información gerencial que integra y maneja actividades de control y evaluación del personal ayudando así a lograr una mejor planificación de las tareas y a la toma de decisiones estratégicas. Este trabajo abarca la implementación del Sistema de Gestión del Rendimiento del Personal (SisGER) que permitirá una eficiencia y confiabilidad en la gestión de los procesos de evaluación del personal en el proyecto ERP-Cuba.

ÍNDICE

Introducción	1
1 CAPITULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción	5
1.2 Gestión de capital humano.....	5
1.3 Gestión del Rendimiento del Personal	6
1.4 Sistema de Gestión del Rendimiento del Personal existentes	6
1.4.1 En el Mundo	6
1.4.2 En Cuba	7
1.5 Metodologías, Lenguajes y Herramientas de Desarrollo.....	8
1.5.1 Metodología de desarrollo utilizada	8
1.5.2 Lenguajes del lado del cliente.....	8
1.5.3 Lenguajes del lado del servidor	10
1.5.4 Fundamentación de la selección del lenguaje a utilizar	12
1.5.5 Herramientas para el desarrollo de la solución informática.....	13
1.5.6 Marco de trabajo.....	15
1.5.7 Base de Datos.....	17
1.6 Navegadores.....	19
1.7 Conclusiones.....	20
2 CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	21
2.1 Introducción	21
2.2 Arquitectura Base	21
2.2.1 Patrones arquitectónicos utilizados.....	21
2.3 Modelo de negocio.....	23
2.3.1 Modelo de negocio basado en proceso	23
2.3.2 Procesos que tienen lugar en el Sistema de Gestión del Rendimiento del Personal.....	24
2.4 Requisitos funcionales	31

2.5	Requisitos no funcionales	36
2.5.1	Especificaciones de requisitos.....	38
2.6	Prototipos de interfaces	49
2.7	Conclusiones	58
3	CAPÍTULO 3: DISEÑO DEL SISTEMA	59
3.1	Introducción	59
3.2	Diagramas de Clases del diseño	59
3.2.1	Herramientas consultadas para realizar el Diseño.....	60
3.2.2	Descripción de las clases	64
3.3	Diagrama Entidad Relación de la BD	75
3.4	Conclusiones	80
4	CAPITULO 4: IMPLEMENTACIÓN Y PRUEBA	81
4.1	Introducción	81
4.2	Diagrama de despliegue	81
4.3	Diagrama de componentes	81
4.3.1	Descripción y presentación del Diagrama de componentes.....	82
4.4	Tipos de prueba	84
4.4.1	Pruebas de caja negra.....	84
4.5	Validación del Modelo de diseño propuesto	87
4.6	Conclusiones	94
5	CONCLUSIONES GENERALES	95
6	RECOMENDACIONES	96
7	REFERENCIAS BIBLIOGRÁFICAS	97
8	BIBLIOGRAFÍA	98
9	ANEXOS	100
	Anexo 1: Diagrama de clase del diseño	100

Anexo 2: Diagrama de Entidad de Relación de la Base de Datos	106
Anexo3: Prototipo de interfaz del sistema	107
10 GLOSARIO DE TÉRMINOS	111

ÍNDICE DE TABLAS

Tabla 2.1: R2.1.1 Adicionar Criterio	39
Tabla 2.2: R2.1.2 Eliminar Criterio.....	40
Tabla 2.3: R2.1.3 Modificar Criterio	40
Tabla 2.4: R2.2 Asignar criterios al trimestre	41
Tabla 2.5: R2.3.1 Gestionar observación.....	42
Tabla 2.6: R3.1.1 Adicionar criterio	43
Tabla 2.7: R3.1.2 Eliminar criterio	43
Tabla 2.8: R3.1.3 Modificar criterio.....	44
Tabla 2.9: R3.3.1 Gestionar Acciones Formativas.....	45
Tabla 2.10: R4.1.1 Adicionar criterio	46
Tabla 2.11: R4.1.2 Eliminar criterio	46
Tabla 2.12: R4.1.3 Modificar criterio.....	47
Tabla 2.13: R4.4.1 Adicionar tarea	48
Tabla 2.14: R4.4.2 Eliminar tarea	48
Tabla 2.15: R4.4.3 Modificar tarea.....	49
Tabla 3.1: Descripción de clase. asignacionCriterioActions	65
Tabla 3.2: Descripción de clase. criterioActions	66
Tabla 3.3: Descripción de clase. evaluacionActions.....	67
Tabla 3.4: Descripción de clase. evaluacionEstudianteActions.....	69
Tabla 3.5: Descripción de clase. evaluacionPersonalActions	70
Tabla 3.6: Descripción de clase. evaluacionProfesorActions	72
Tabla 3.7: Descripción de clase. rolActions	73
Tabla 3.8: Descripción de clase. tareaActions	74
Tabla 3.9: Descripción de clase. usuarioActions.....	75
Tabla 4.1: Prueba Caja Negra. Evaluación según el Rol del Proyecto.....	85
Tabla 4.2: Requisito a probar: Evaluación del profesor	87
Tabla 4.3: Tamaño operacional de clase (TOC).....	88
Tabla 4.4: Relaciones entre clases (RC)	89
Tabla 4.5: Umbrales.....	93
Tabla 4.6: Resumen de los resultados.....	93

ÍNDICE DE FIGURAS

Figura 1: Esquema de la evolución de HTML y XHTML	9
Figura 2: Esquema de la arquitectura MVC	22
Figura 3: Proceso de negocio. Evaluación según rol del proyecto.....	25
Figura 4: Proceso de negocio. Evaluación Personal	26
Figura 5: Proceso de negocio. Evaluación Estudiante	27
Figura 6: Proceso de negocio. Evaluación Profesor	28
Figura 7: Proceso de negocio. Gestionar Tareas.....	29
Figura 8: Proceso de negocio. Gestionar Incidencias.....	30
Figura 9: Proceso de negocio. Presentar Información	31
Figura 10: Prototipo de interfaz. Gestionar usuario	50
Figura 11: Prototipo de interfaz. Cambiar contraseña.....	50
Figura 12: Prototipo de interfaz. Gestionar Rol.....	51
Figura 13: Prototipo de interfaz. Gestionar rol del proyecto.....	52
Figura 14: Prototipo de interfaz. Gestionar trimestre	53
Figura 15: Prototipo de interfaz. Gestionar incidencias	54
Figura 16: Prototipo de interfaz. Gestionar criterios	55
Figura 17: Prototipo de interfaz. Asignar criterios al trimestre	56
Figura 18: Prototipo de interfaz. Gestionar tareas	57
Figura 19: Diagrama de clases. Evaluación según Rol.....	61
Figura 20: Diagrama de clases. Evaluación del profesor.....	62
Figura 21: Diagrama de clases. Evaluación del estudiante	63
Figura 22: Diagrama de clases. Evaluación del personal.....	64
Figura 23: Diagrama de entidad relación Evaluación según el Rol del Proyecto.	76
Figura 24: Diagrama de entidad relación Evaluación del profesor.....	77
Figura 25: Diagrama de entidad relación Evaluación del estudiante.	78
Figura 26: Diagrama de entidad relación Evaluación del personal.	79
Figura 27: Diagrama de despliegue.	81
Figura 28: Diagrama de componente.	82
Figura 29: Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.	89
Figura 30: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.	90
Figura 31: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.	90
Figura 32: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación.	90
Figura 33: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.	91
Figura 34: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.	91
Figura 35: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.	92

Figura 36: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento.	92
Figura 37: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.	92
Figura 38: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización. ..	93

Introducción

La gestión empresarial es una herramienta fundamental para el inicio de cualquier empresa sin importar de qué tipo se trate, por ello es importante que se considere como un órgano social que se encarga de lograr la productividad en los recursos. La misma persigue lograr sus objetivos mediante las personas de las que dispone, es decir, mediante el capital humano con los cuales la misma cuenta. Es uno de los elementos más importantes a considerar cuando se trata de llevar a cabo un negocio, pues de la misma dependerá el éxito. Esta comprende las actividades de planificación, organización, dirección y control que serán aplicadas sobre el uso de todos los recursos de los que la empresa disponga.

La evaluación es un proceso para medir el rendimiento laboral del trabajador, con el objeto de llegar a la toma de decisiones objetivas sobre el capital humano. Las organizaciones modernas utilizan la evaluación del trabajador para determinar incrementos de sueldos, necesidades de capacitación y desarrollo, así como ofrecer la documentación para apoyar las acciones de rotación del personal. En consecuencia, una evaluación bien manejada sirve como instrumento de supervisión y desarrollo de personal. Es una herramienta de dirección, imprescindible en la actividad administrativa, ayuda a determinar políticas de capital humano adecuadas a las necesidades de la organización. La evaluación es el proceso por el cual objetivamente se valora cuantitativamente y cualitativamente al trabajador en el puesto que desempeña y se obtiene con la suma de factores determinantes; positivos o negativos de su rendimiento laboral.

La evaluación del capital humano, es un proceso destinado a determinar y comunicar a los colaboradores, la forma en que están desempeñando su trabajo y en principio, a elaborar planes de mejora. Cuando se realiza adecuadamente la evaluación de personal no solo hacen saber a los colaboradores cual es su nivel de cumplimiento, sino que influyen en su nivel futuro de esfuerzo y en el desempeño correcto de sus tareas.

En los últimos años, con la generalización de la informática en las actividades empresariales, han surgido distintas herramientas de software que intentan llevar a cabo la gestión de los flujos de la información a lo largo de la empresa. En un entorno de creciente dinamismo y ante las necesidades de hacer frente a la gestión de una ingente cantidad de información al menor coste posible, los esfuerzos de las empresas

desarrolladoras de aplicaciones orientadas a cubrir dichas necesidades, se han concentrado en la conjugación de las ventajas de los tipos de software, obteniendo como resultado las llamadas aplicaciones ERP (Enterprise Resource Planning) o de Planificación de Recursos Empresariales.

En la Universidad de las Ciencias Informáticas (UCI) se ha propuesto desarrollar un producto ERP nacional en software libre como estrategia para afrontar los desafíos en el nuevo milenio para la economía del país. Para un buen desarrollo de este producto, por su gran cantidad de trabajadores y estudiantes, es necesario tener el control del rendimiento del personal, por lo que surge la siguiente **Situación Problémica**:

Debido a que es un proyecto de gran magnitud y se involucra a personal de diferentes cualidades se crearon líneas de trabajo para hacer más factible la realización del producto final. El gran número de involucrados tiene muy poca experiencia en los roles que ocupan, por lo que se les ha dedicado bastante tiempo en la formación de las competencias para lograr la efectividad en sus puestos de trabajo y contribución a la producción.

La evaluación del rendimiento del personal debe ser llevada con detalles y cuidadosamente. Este proceso ha sido de pobre desarrollo y poco auge por lo que la evaluación del rendimiento del personal no está exenta de problemas, requiere altos costos de tiempo y de recursos así como gran esfuerzo por parte de la dirección pues la gran cantidad de datos que genera el proceso fluye muy lentamente, además la información no está disponible en todo momento y con la objetividad y veracidad que se requiere, haciendo que el proceso de evaluación no sea suficientemente eficiente.

Por tanto el **problema científico** es el siguiente:

¿Cómo hacer más eficiente la gestión de la evaluación del rendimiento del personal en el proyecto ERP Cuba?

El **objeto de estudio** lo constituyen los sistemas para la Evaluación del Rendimiento del Personal.

Objetivo general:

Desarrollar una herramienta que haga más eficiente la Gestión de la Evaluación del Rendimiento del Personal en el Proyecto ERP-Cuba.

Objetivos específicos:

- Realizar un estudio del estado del arte sobre la existencia de herramientas para la Evaluación del Rendimiento Laboral.
- Identificar los procesos para la Evaluación del Rendimiento del Personal en el Proyecto ERP-Cuba a informatizar.
- Desarrollar la herramienta para la Gestión de la Evaluación del Rendimiento del Personal en el Proyecto ERP-Cuba.
- Realizar pruebas de software a la herramienta para la Gestión de la Evaluación del Rendimiento del Personal en el Proyecto ERP-Cuba.

El **campo de acción** se enmarca en los procesos para la informatización de la Gestión de la Evaluación del Rendimiento del Personal en el Proyecto ERP-Cuba.

Idea a defender:

El desarrollo de una herramienta para la Gestión de la Evaluación del Rendimiento del Personal en el Proyecto ERP-Cuba garantizará una mayor eficiencia en dicho proceso.

Tareas de la investigación:

- Estudiar sobre la existencia de otras aplicaciones o soluciones similares asociadas a la Evaluación del Rendimiento Laboral.
- Modelar y diseñar la propuesta de la herramienta a utilizar en el proceso de Evaluación del Rendimiento.
- Implementar la herramienta asociada a la Evaluación del Rendimiento Laboral.

- Probar la herramienta de Evaluación del Rendimiento Laboral.

MÉTODOS

Métodos teóricos: Analítico–sintético: Se consultará la bibliografía necesaria para dar cumplimiento a las tareas de la investigación y se resumirán los principales aspectos de cada una de ellas. Estas operaciones no son independientes, pues el análisis de un objeto se realiza a partir de la relación que existe entre los elementos que lo conforman y a su vez, la síntesis se produce sobre la base de los resultados previos del análisis.

Histórico – lógico: Como está vinculado al conocimiento de las distintas etapas de los objetos en su sucesión cronológica, posibilitará conocer los antecedentes de las herramientas para la evaluación del personal. Mediante este método, se analizará la trayectoria concreta de la teoría, su condicionamiento a los diferentes períodos de la historia y al basarse en el desarrollo histórico, pondrá de manifiesto la lógica interna de su desarrollo, su teoría y encontrará el conocimiento más profundo de esta, su esencia.

Método de modelación: Permitirá la creación de modelos, es decir representar lo que se quiere estudiar de forma más simple, explicando lo que pasa de una manera lógica.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se describe el marco teórico referencial del trabajo, realizando un análisis del estado del arte de los sistemas de gestión del rendimiento del personal; el estado actual de los sistemas con que cuenta el proyecto que pueden tributar información relevante a nuestro sistema, así como un análisis de los diferentes procesos objetos de informatización, imprescindibles para entender la lógica del negocio. Además se ofrece una breve panorámica de las características fundamentales de este tipo de sistemas y se realiza un estudio de las tecnologías en las que se apoyará la realización de este software en función de un análisis de las tendencias actuales, apostando por supuesto por el Software Libre.

1.2 Gestión de capital humano

En estos tiempos el conocimiento es la base fundamental para el logro de los objetivos, pues el hombre con su talento puesto al servicio de su organización juega un papel estratégico fundamental. Por esta causa se afianza aún más la idea de que el capital humano es el factor fundamental del éxito teniendo en cuenta el aporte o no de sus conocimientos y habilidades para crear. Una gestión eficaz y efectiva de los mismos constituye hoy la prioridad fundamental para lograr el nivel de competitividad que se requiere.

Recurso humano: Es el conjunto de capital humano que esta bajo el control de la empresa en una relación directa de empleo, en este caso personas, para resolver una necesidad o llevar a cabo cualquier actividad en una empresa.

Capital humano: Conjunto de conocimientos, habilidades y aptitudes inherentes a los individuos que forman la organización.

Haciendo una valoración de los conceptos anteriores se puede concluir que el término Gestión del Capital Humano, esta referido a la definición de estrategias, políticas, procedimientos de ejecución de las prácticas de recursos humanos, indicadores de efectividad y principios que tengan como objetivo lograr la identificación plena de los individuos con la organización, logrando establecer nexos favorables entre ellos siguiendo siempre un enfoque participativo y de completa comunicación (Eudaldo Morales, 2002).

1.3 Gestión del Rendimiento del Personal

Varias empresas en países como Estados Unidos, Europa y recientemente en América Latina, han incorporado la Gestión del Rendimiento del Personal (GRP) como una herramienta para mejorar la productividad y mantener un clima positivo en las relaciones con sus colaboradores. La justificación de estos esfuerzos se encuentra en el intento de mejorar los niveles de productividad y competitividad mediante la movilización del conocimiento y en la capacidad de aprender de la organización.

1.4 Sistema de Gestión del Rendimiento del Personal existentes

1.4.1 En el Mundo

En Bolivia, la Institución Nacional de Formación Profesional, el **Instituto Nacional de Formación y Capacitación Laboral (INFOCAL)**, realizó la actualización del "Clasificador Nacional de Ocupaciones (CNO)" teniendo en cuenta la necesidad de contar con un mapa orientador sobre las distintas ocupaciones existentes en el país, sus diferentes niveles de competencia y funciones. Con la utilización del CNO se logrará un instrumento orientador en la actividad formadora, que servirá para lo referente a la ampliación de instrumentos de identificación de competencias, al insumo para la elaboración o la modificación de los currículos instruccionales de la institución y finalmente contribuir con ello a la modernización del país mediante la formación de capital humano altamente competitivos.

El **Servicio Nacional de Aprendizaje Industrial (SENAI) de Brasil** está desarrollando su modelo de formación basada en competencia laboral. Al efecto ha definido un proyecto estratégico nacional encaminado a dos grandes resultados:

La elaboración de perfiles y programas de formación por competencias.

La puesta en marcha de un proceso de reconocimiento de las competencias adquiridas por la experiencia laboral.

En **Colombia** está presente la formación por competencias en el Servicio Nacional de Aprendizaje (SENA) y la Red Nacional de Instituciones de Formación para el Trabajo. El SENA intenta crear un lenguaje

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

común para el manejo de diferentes temas entre los cuales está el diseño de las normas de competencia laboral, los programas de formación y el reconocimiento de competencias.

El **Instituto Técnico de Capacitación y Productividad (INTECAP) de Guatemala** ha incorporado el enfoque de competencia laboral en su gestión institucional. Es así como ha conformado el modelo "Norte" por: Normalización Técnica de Competencias. El modelo está siendo aplicado tanto en el diseño de planes y material didáctico, evaluación y certificación de las competencias, como en los procesos de asesoría en la Gestión de Recursos Humanos.

El **Instituto Salvadoreño de Formación Profesional (INSAFORP)**, ha organizado un proceso de desarrollo organizacional que concentra sus esfuerzos en la Formación Basada en Competencia Laboral y en el diseño y promoción de un Sistema Nacional de Formación y Certificación.

1.4.2 En Cuba

Hasta ahora no se contaba con una herramienta que evaluara el rendimiento laboral. En la Universidad de Ciencias Informáticas (UCI) también se está desarrollando la GRP en los proyectos productivos basado en competencias laborales. La Facultad 3 contó con dos trabajos de tesis relacionados con este importante tema los cuales son: **“Algoritmos para la asignación de estudiantes a proyectos productivos”**, elaborado por Yuniet Rodríguez Suárez y **“Procedimiento para la conformación de equipos de desarrollo de software para Facultad 3”**, elaborado por Heidi Pérez y Margarita de la Cruz.

El proyecto APS de la Facultad 7 también realiza la GRH en base a competencias, a través de la creación de modelos de competencias para cada uno de los roles que requiere el proyecto. Con estos modelos de competencias se pretende aumentar el rendimiento de cada miembro del proyecto y de los futuros integrantes.

En la Facultad 1 se propuso un sistema denominado Centro de Validación y Certificación de Competencias Profesionales UCI (*CERTIMAX*), con el fin de automatizar las tareas referentes a la certificación de competencias laborales desde la producción, garantizando la estandarización de una metodología diferente para la creación de equipos de proyectos a partir de las competencias avaladas y los créditos alcanzados por los estudiantes y profesores del centro.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La Facultad 4 sugirió un *Proceso de Selección de Roles y Personal con sus Niveles de Competencia para Proyectos Multimedia*. El objetivo de este trabajo fue proponer un proceso estándar de selección de roles y personal en los proyectos multimedia de la UCI

A pesar de todos estos intentos por desarrollar software similares se propone la herramienta asociada a la estrategia definida en la tesis "Estrategia para la Evaluación del Rendimiento del Personal en el Proyecto ERP-Cuba" de los autores Javier Morales Ramos y Adrian Pérez Mejias.

1.5 Metodologías, Lenguajes y Herramientas de Desarrollo

A continuación se ilustran la metodología, los lenguajes de programación y herramientas que se tuvieron en cuenta a la hora de seleccionar los más adecuados para la implementación de la solución.

1.5.1 Metodología de desarrollo utilizada

Como metodología de desarrollo se decidió utilizar el Modelo de Desarrollo definido por el proyecto ERP-Cuba, debido a que si se decide realizar una nueva versión los desarrolladores del proyecto puedan ser capaces de entender el funcionamiento del sistema.(Modelo de desarrollo de ERP)

1.5.2 Lenguajes del lado del cliente

Los lenguajes del lado del cliente son los encargados de dibujar, maquetar, estilizar y aportar dinamismo a la aplicación en los navegadores. Los lenguajes del lado del cliente abordados en este epígrafe son precisamente los utilizados, por ser estándares internacionales para el desarrollo Web. Los homólogos por los cuales se hubiera hecho un proceso de selección quedan exentos por su casi nulo desarrollo e incompetividad frente a estos señalados.

JavaScript: Es el lenguaje de programación del lado del cliente más utilizado por su compatibilidad con la mayoría de los navegadores modernos (Eguíluz Pérez, 2007). Con él se pueden generar páginas dinámicas en función de las preferencias del usuario, validar los datos introducidos en un formulario o modificar dinámicamente el contenido de la página. Es multiplataforma e interpretado, es decir, no requiere compilación y tiene una sintaxis semejante a los lenguajes Java y C. Este lenguaje es muy fácil de aprender e ideal para agregar ciertas funciones rápidas a una página Web.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

XHTML: El lenguaje XHTML (del inglés *eXtensible HyperText Markup Language*, Lenguaje Extensible de Marcado de Hipertexto) es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. Técnicamente, HTML es descendiente directo del lenguaje SGML, mientras que XHTML lo es del XML (que a su vez, también es descendiente de SGML).

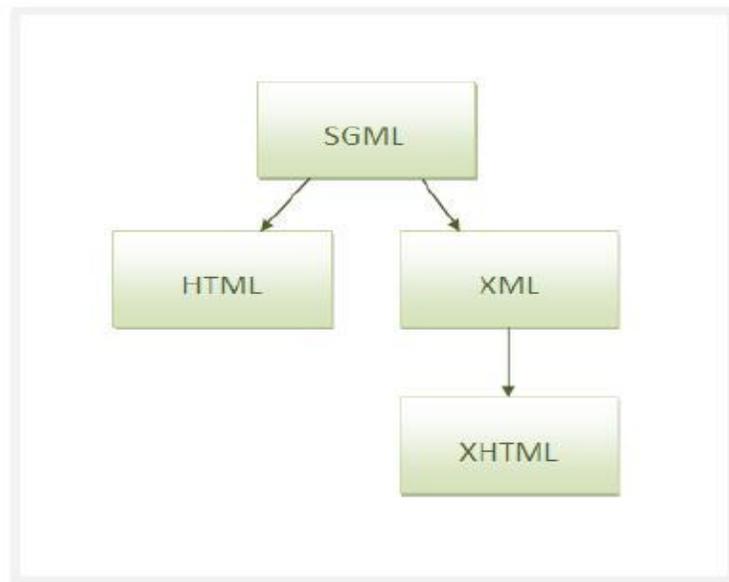


Figura 1: Esquema de la evolución de HTML y XHTML

Las páginas y documentos creados con XHTML son muy similares a las páginas y documentos HTML. Las discusiones sobre si HTML es mejor que XHTML o viceversa son recurrentes en el ámbito de la creación de contenidos web, aunque no existe una conclusión ampliamente aceptada.

Actualmente, entre HTML 4.01 y XHTML 1.0, los mejores desarrolladores siempre escogen XHTML. En un futuro cercano, si los diseñadores deben escoger entre HTML 5 y XHTML 1.1 o XHTML 2.0, quizás la elección sea diferente.

CSS: (del inglés *Cascading Style Sheets*, Hojas de Estilo en Cascada) es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (Eguíluz Pérez, 2007).

1.5.3 Lenguajes del lado del servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web y le envía una respuesta al cliente a través de una página. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, tratamientos de la información, entre otras funciones.

Se presentan a continuación tres candidatos que tienen gran aceptación en el desarrollo Web mundial.

Perl: Es un lenguaje de programación muy utilizado para construir aplicaciones para la Web. Tiene gran popularidad por ser un intérprete que se distribuye de forma gratuita. Es fácil de usar, soporta tanto la programación orientada a objetos como la programación estructurada, posee también una enorme colección de módulos disponibles. Aunque inicialmente fue diseñado para un entorno UNIX, en la actualidad hay versiones para casi todos los sistemas operativos: DOS, WINDOWS NT, MacOS y otros (Perl, 2008).

PHP: (del inglés Hypertext Preprocessor) es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor (Achour, y otros, 2007). Permite la conexión a gran cantidad de servidores de bases de datos lo que constituye una de sus características más destacadas y potentes. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, es rápido, posee una gran cantidad de funciones y mucha documentación. Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5.2.

Ventajas de PHP:

- Es multiplataforma.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos. Siempre se podrá disponer de objetos de colección de bases de datos para situaciones que lo requieran. Destaca su conectividad con My SQL.
- Permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código.
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es software libre.
- Rapidez. PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD. Otra alternativa es configurarlo como módulo CGI.
- Una gran variedad de módulos cuando un programador PHP necesite una interface para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, Flash, Cybercash, calendarios, XML, IMAP, POP, etc.
- Permite las técnicas de Programación Orientada a Objetos.
- Posee una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Java: Es un lenguaje de desarrollo de propósito general y como tal es válido para realizar todo tipo de aplicaciones profesionales. Incluye una combinación de características que lo hacen único y está siendo

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

adoptado por multitud de fabricantes como herramienta básica para el desarrollo de aplicaciones comerciales de gran repercusión. Puede considerarse como una evolución de C++, su sintaxis es parecida a la de este y al lenguaje C, aunque no acarrea los inconvenientes de los mismos ya que fue diseñado “partiendo desde cero”, es decir, no necesita ser compatible con versiones anteriores de ningún lenguaje y elimina herramientas de bajo nivel como punteros. Un lenguaje orientado a objetos puro, creado para trabajar con ellos desde cero. Es compilado e interpretado. Del código fuente se pasa a una representación denominada *bytecode* (compilación) lo que posteriormente es interpretado por la Máquina

Virtual Java (JVM) que es la encargada de ejecutar el programa. El *bytecode* hace que Java sea independiente de la plataforma hardware y del sistema operativo, o sea, cualquier código fuente de Java puede ser compilado en cualquier máquina y sistema operativo que tenga un compilador Java, y que cualquier fichero de *bytecode* resultado de compilar un fichero fuente puede ser ejecutado en cualquier máquina y sistema operativo que tenga una Máquina Virtual de Java (Molina, 2006).

1.5.4 Fundamentación de la selección del lenguaje a utilizar

Analizadas las características principales de los lenguajes de programación del lado del servidor descritos en el epígrafe anterior se selecciona a continuación el más indicado para realizar la aplicación. Todos los lenguajes analizados están soportados por múltiples plataformas, a nivel mundial los más utilizados son PHP y JAVA, sobre los que se pueden encontrar una gran cantidad de información, tutoriales, artículos y códigos de ejemplo. Recordar que sólo se están teniendo en cuenta las tecnologías y/o lenguajes libres. El lenguaje PHP es recomendado para aplicaciones que necesiten ser implementadas con rapidez, por la sencillez con que se programa del lado del servidor. En la Universidad de las Ciencias Informáticas el lenguaje PHP es utilizado por gran cantidad de desarrolladores y precisamente los proyectos de producción en particular el ERP-Cuba, poseen una gran experiencia sobre este lenguaje. Por lo que se propone el lenguaje PHP para realizar el software. Se propone además el uso de JavaScript, XHTML y CSS, teniendo en cuenta las ventajas que ofrecen para la programación del lado del cliente.

1.5.5 Herramientas para el desarrollo de la solución informática

A continuación se tratarán una serie de herramientas necesarias para la implementación de la aplicación. Para una mejor comprensión, cada uno de los sub-epígrafes que lo componen trata de una herramienta y de las diversas opciones a escoger, y se define entonces un sub-epígrafe para la opción seleccionada.

1.5.5.1 Entornos integrados de desarrollo

Entre los entornos de desarrollo libres más acreditados se encuentran NetBeans, MonoDevelop, Eclipse, entre otros. Se trabajó con Eclipse por ser el único con soporte para PHP, además de ser muy usado en la universidad, tener un ambiente amigable y ser muy potente.

Eclipse: Es una plataforma que proporciona un marco de trabajo o framework para desarrollar aplicaciones. Esto significa que distintos fabricantes pueden desarrollar e integrar sus herramientas con el WorkBench existente. El WorkBench es un conjunto de frameworks de Java y herramientas de desarrollo equipadas para constructores de herramientas. Eclipse se considera como uno de los IDE más poderosos para el desarrollo de aplicaciones en Java, provee soporte mediante plugins para trabajar con entornos J2EE y J2SE utilizados en el desarrollo de este trabajo (Eclipse, 2000).

La filosofía de plugins permite a Eclipse que muchas personas o empresas creen dichos complementos para desarrollar en esta plataforma en diferentes lenguajes, en este caso se utilizará el plugin PHPEclipse.

Dentro de las funcionalidades del Eclipse, la adición de plugins puede facilitar la automatización de diferentes tareas, es el caso de un plugin que permite la administrar el control de versiones de un proyecto, seguidamente se abordará más sobre el tema.

SubEclipse: Esta herramienta no es más que un Plug-in que integra el Eclipse con el SVN. Haciendo uso de este software se podrá facilitar el control de versiones desde el IDE Eclipse de forma tal que se pueden hacer operaciones de *commit*, *update*, *delete*, entre otras. Esta herramienta es libre y es ampliamente utilizada por los desarrolladores de software que utilizan el Eclipse como IDE a nivel mundial.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Visual Paradigm: Es una herramienta que soporta aplicaciones Web, es muy fácil de instalar y actualizar. Permite la generación de código para varios lenguajes. Es una herramienta de código abierto y presenta un entorno de creación de diagramas. Su diseño está centrado en casos de uso y enfocado al negocio generando un software de mayor calidad, presenta capacidades de ingeniería directa e inversa y disponibilidad en múltiples plataformas.

Está diseñada para desarrollar software con programación orientada a objetos. Busca reducir la duración del ciclo de desarrollo, brindando ayuda a arquitectos, analistas, diseñadores y desarrolladores; permite el uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

Una de las características más importantes de su uso es que brinda la posibilidad de sincronización del modelo de diseño y el código en todo el ciclo de desarrollo una vez que se integra con el Eclipse, permitiendo la facilidad de programar directamente sobre el código fuente generado y a su vez actualizar el diseño con cambios que se realicen en la programación.

Por todas las características anteriormente planteadas se escogió para la modelación del proyecto la herramienta Visual Paradigm.

Zend Studio para Eclipse.

Zend Studio para Eclipse es la última versión hasta el 2008 del ya conocido entorno de programación integrado Zend. Esta renovada versión de Zend agrupa las cualidades de sus versiones anteriores, como son:

- Integración del uso y completado de código personalizado de Zend Framework y vista de la lista de las funciones del framework desde la Visualización de Funciones PHP
- Aumentar la productividad con: Soporte PHP 5 completo, Analizador de Código, carpeta de Código, completado de Código, coloreado de Sintaxis, Administrador de Proyecto, Editor de Código, Depurador de gráficos y asistentes.
- Documentación del código de forma más sencilla, aplicaciones, y proyectos con PHPDocumentor, la herramienta de documentación estándar para PHP.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Asegurar la protección máxima de ubicaciones de proyectos con depuradores remotos seguros.

Zend Studio para Eclipse también brinda nuevas facilidades que lo hacen ser superior a las versiones anteriores:

- Dispone de un entorno mucho más flexible y profesional para controlar todo el ciclo de vida de un desarrollo.
- Capacidades de refactorización del código fuente: permite adecuar el comportamiento externo de una función o clase sin cambiar el funcionamiento interno.
- Dispone de un buen debugger local con la conexión a los servidores de desarrollo.

1.5.6 Marco de trabajo

Un framework o marco de trabajo es una estructura de soporte definida en la cual un proyecto de software puede ser organizado y desarrollado. Típicamente incluye soporte de programas, bibliotecas, lenguajes de script y otros software utilizados que ayudan a unir y desarrollar los diferentes componentes de un proyecto. Provee una estructura y una metodología de trabajo que extiende o utiliza las aplicaciones del dominio. Los objetivos principales que se persiguen con su uso son: reutilizar código ya existente, promover buenas prácticas de desarrollo y acelerar el proceso de construcción de un software (Symfony Framework, 2007).

A continuación se describe el framework seleccionado para la propuesta de solución.

Symfony Framework: Es un *framework* libre escrito en PHP5. Basado en las mejores prácticas de desarrollo web, usado en software comercial y educativo, Symfony fomenta la velocidad y eficiencia en la creación y mantenimiento de aplicaciones web, reemplazando tareas de codificación repetitivas por poder, control y placer.

Symfony es una alternativa para proyectos en PHP con características como:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Sistema de Plantillas sencillo.
- Administración de Cache.
- Scaffolding Avanzado.
- Modelo de Objetos y Separación MVC.
- Soporte para AJAX.
- Generación de Formularios.
- Componentes Gráficos.
- Seguridad.

El número de prerrequisitos para instalar y configurar es muy pequeño, apenas Unix o Windows con un servidor Web y PHP5 instalado. Symfony es compatible con motores de base de datos como MySQL, PostgreSQL y Oracle.

¿Qué es Symfony?

Symfony es un framework que ayuda a reducir el tiempo de desarrollo de una aplicación web sin producir efectos sobre los programadores. Está basado en los siguientes conceptos:

- Compatible con muchas plataformas.
- Fácil de instalar y configurar.
- Fácil de aprender.
- Simple en la mayor parte de casos pero flexible para adaptarse a casos más complejos.
- Soporta muchas características de Aplicaciones Web Actuales.
- Soporta las prácticas y patrones de programación más productivos y eficientes.

- Produce aplicaciones fáciles de mantener.
- Está basado en Software Libre.

1.5.7 Base de Datos

Un Sistema de Gestión de Bases de Datos (SGBD) consiste en un conjunto de programas, procedimientos y lenguajes que proporcionan las herramientas necesarias para trabajar con una base de datos, incorporar una serie de funciones que permitan definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos (Wikipedia, Colaboradores de , 2008).

En la actualidad existen numerosos sistemas gestores de bases de datos, de ellos se propuso estudiar las características de MySQL y PostgreSQL que además de sus potencialidades, cumplen con la condición de ser software libres.

MySQL posee un conjunto de utilidades y ventajas que lo hacen ser el más utilizado por los desarrolladores. Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL (MySQL AB, 2008).

De la misma forma se puede hacer referencia al PostgreSQL como otro de los sistemas gestores de bases de. PostgreSQL ofrece muchas ventajas respecto a otros sistemas de bases de datos:

- Instalación Ilimitada: Es frecuente que las bases de datos propietarias sean instaladas en más servidores de lo que permite la licencia. PostgreSQL al no tener costo asociado a la licencia del software permite que no se incurra en violaciones de este tipo además de brindar varias ventajas adicionales:
 - ✓ Modelos de negocios más rentables con instalaciones a gran escala.
 - ✓ No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.
- ✓ El software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos propietarios, conservando todas las características, estabilidad y rendimiento.
- Estabilidad y confiabilidad legendarias: En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.
- Extensible: El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.
- Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.
- Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.
- Herramientas gráficas de diseño y administración de bases de datos: Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos.

Al haberse estudiado ambos sistemas de gestión de bases de datos, se selecciona PostgreSQL, por su gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Además, PostgreSQL es muy utilizado en la universidad, donde existen muchos proyectos productivos que lo utilizan y posee gran cantidad de documentación.

1.6 Navegadores

Un navegador Web es una aplicación que le permite a un usuario recuperar y visualizar documentos de hipertexto, que pueden estar escritos en HTML desde servidores Web de todo el mundo a través de Internet. Actualmente los navegadores actuales permiten mostrar o ejecutar gráficos, secuencias de videos, sonido, animaciones así como otros programas además de texto e hipervínculos. La comunicación entre un servidor Web y el navegador se realizan mediante el protocolo HTTP, aunque también soportan FTP y HTTPS (Network-Press.Org, 2003).

Existen disímiles navegadores entre los que se destacan: Internet Explorer (IE), Mozilla Firefox, Opera, Safari, Koqueror, Netscape Navigator, entre otros muchos. En Cuba y más específicamente dentro de la universidad los más usados son: Internet Explorer y el Mozilla Firefox, este último poco a poco se va imponiendo sobre el conocido Internet Explorer debido a las grandes ventajas que presenta. Estudiar los navegadores Web a la hora de crear una aplicación Web es muy necesario porque la aplicación debe ser compatible con la mayor cantidad de navegadores disponibles para pueda ser accesada por cualquier tipo de usuario.

El sistema cumplirá con los estándares Web de desarrollo definidos por la W3C, por tanto todos los navegadores que sean capaces de cumplirlos podrán utilizar la aplicación sin ninguna dificultad. Para Mozilla Firefox, que a pesar de que no cumple algunos estándares es muy utilizado, la aplicación tiene reglas especiales que verifican si se está ejecutando en este, y adaptarla consecuentemente a este navegador.

1.7 Conclusiones

En este capítulo se hizo un análisis de diferentes sistemas que funcionan en distintos países, para tomar sus mejores prácticas y experiencias, además se estudiaron varias herramientas de código libre, obteniéndose PHP como lenguaje del lado del servidor con Zend Studio para Eclipse y como framework Symfony. PostgreSQL como gestor de base de datos, que se ejecutará sobre un servidor Apache además el Modelo de Desarrollo definido por el proyecto ERP-Cuba como metodología de desarrollo de software y como herramienta CASE para la modelación de los procesos el Visual Paradigm.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se comienza a tener la concepción práctica sobre lo que se pretende que haga el Sistema de Gestión del Rendimiento del Personal, teniendo en cuenta las necesidades y características del proyecto ERP-Cuba. Se explican algunos de los procesos críticos del negocio que son necesarias para entender la propuesta de solución, llegando a describir los procesos de negocio, además se definen los requisitos tanto funcionales como no funcionales. Por otra parte, se tratarán los artefactos definidos en la metodología de desarrollo definida en el proyecto ERP-Cuba referentes al flujo de trabajo análisis y diseños.

2.2 Arquitectura Base

En el campo del software, la arquitectura nos identifica los elementos más importantes de un sistema así como sus relaciones. Nos da una visión global del sistema.

¿Por qué es esto importante? Porque necesitamos arquitectura para entender el sistema, organizar su desarrollo, plantear la reutilización del software y hacerlo evolucionar.

Cómo determinar los elementos que definen una arquitectura es difícil y muy importante.

Generalmente las metodologías de desarrollo indican principios para identificar y diseñar una arquitectura, aunque por ahora la ayuda real que ofrecen es muy limitada al basarse en principios muy genéricos.

A continuación se ilustra la arquitectura base definida para el Sistema de Gestión del Rendimiento del Personal.

2.2.1 Patrones arquitectónicos utilizados

Patrones arquitectónicos sobre aspectos fundamentales de la estructura de un sistema software. Especifican un conjunto predefinido de subsistemas con sus responsabilidades y una serie de recomendaciones para organizar los distintos componentes (Buschamann, 1996).

CARACTERÍSTICAS DEL SISTEMA

Siguiendo la filosofía del modelo actual de desarrollo del software, para la realización del sistema se propone organizar los elementos de la aplicación en componentes independientes buscando alcanzar una mayor efectividad a la hora de administrarlos.

El patrón arquitectónico Modelo Vista Controlador (MVC) permite hacer la separación de las capas de interfaz, modelo y lógica de control de esta. La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario (The original MVC reports, 2007).

La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, simplemente es necesario conocer la API (Interfaz de Aplicación) que existe entre niveles. La división en capas reduce la complejidad, facilita la reutilización y acelera el proceso de ensamblar o desensamblar alguna capa, o sustituirla por otra distinta (pero con la misma responsabilidad) (Symfony Framework, 2007).

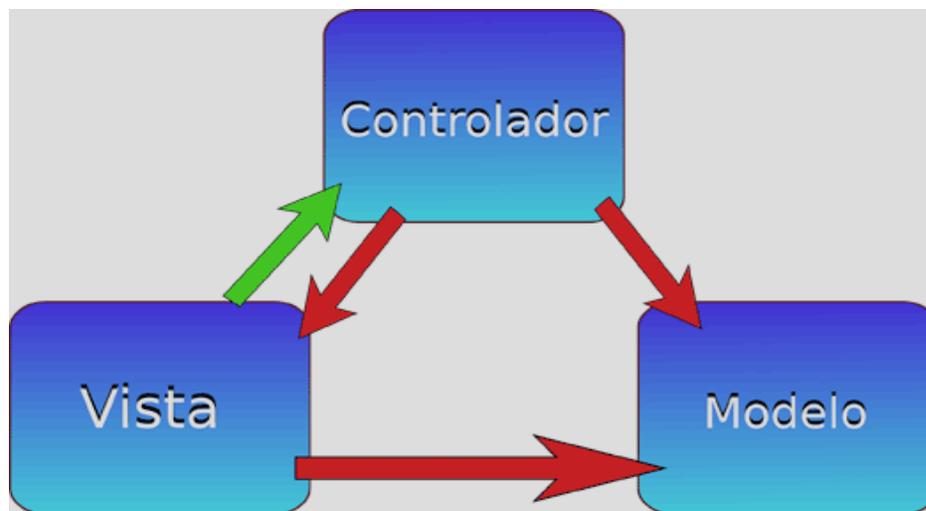


Figura 2: Esquema de la arquitectura MVC

Ventajas de esta arquitectura

El objetivo de este patrón es el realizar y mantener la separación entre la lógica de nuestra aplicación, los datos y la presentación. Esta separación tiene algunas ventajas importantes, como:

- Se puede identificar más fácilmente en qué capa se está produciendo un problema con sólo saber su naturaleza.
- Se pueden crear varias presentaciones sin necesidad de escribir varias veces la misma lógica de aplicación.
- Cada parte funciona independiente y cualquier cambio centraliza el efecto sobre las demás, así que se puede asegurar que una modificación en un componente realizará bien las tareas en cualquier parte de la aplicación.

2.3 Modelo de negocio

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

Una técnica para la especificación de los requisitos más importantes del sistema, que da soporte al negocio, es el modelo del negocio, con lo cual se refuerza la idea de que sea el propio negocio lo que determine los requisitos.

2.3.1 Modelo de negocio basado en proceso

Para conseguir sus objetivos, una empresa organiza su actividad por medio de un conjunto de procesos de negocio. Cada uno de ellos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes (por ejemplo, trabajadores o departamentos) participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan las políticas y la estructura de la

información de la empresa. Por tanto, la finalidad del modelado del negocio es describir cada proceso, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio. El primer paso del modelado del negocio consiste en capturar los procesos de negocio de la organización bajo estudio. La definición del conjunto de procesos del negocio es una tarea crucial, ya que define los límites del proceso de modelado posterior (Molina, y otros, 2000).

2.3.2 Procesos que tienen lugar en el Sistema de Gestión del Rendimiento del Personal

En la presente investigación se definieron seis procesos principales para el Sistema de Gestión del Rendimiento del Personal, éstos son:

- Gestionar Evaluación según el Rol.
- Gestionar Evaluación Personal.
- Gestionar Evaluación del Estudiante.
- Gestionar Evaluación del Profesor.
- Gestionar Incidencias
- Gestionar Tareas

A continuación se explica brevemente en qué consiste cada proceso así como los diagramas donde se muestra la secuencia de pasos para llevarlos a cabo. Una descripción exhaustiva se puede encontrar en los documentos de descripción de procesos que se adjunta a la investigación, como parte de los anexos.

Proceso de Negocio: Evaluar según rol del proyecto

En este proceso se evaluará al personal según el rol que desempeñe en el proyecto teniendo en cuenta los criterios a evaluar en el trimestre, arrojando como resultado la si el personal necesita de acciones formativas que pueden ser cursos optativos, documentación de algún tema, y también si es una persona relevante, destacado o insatisfactorio.

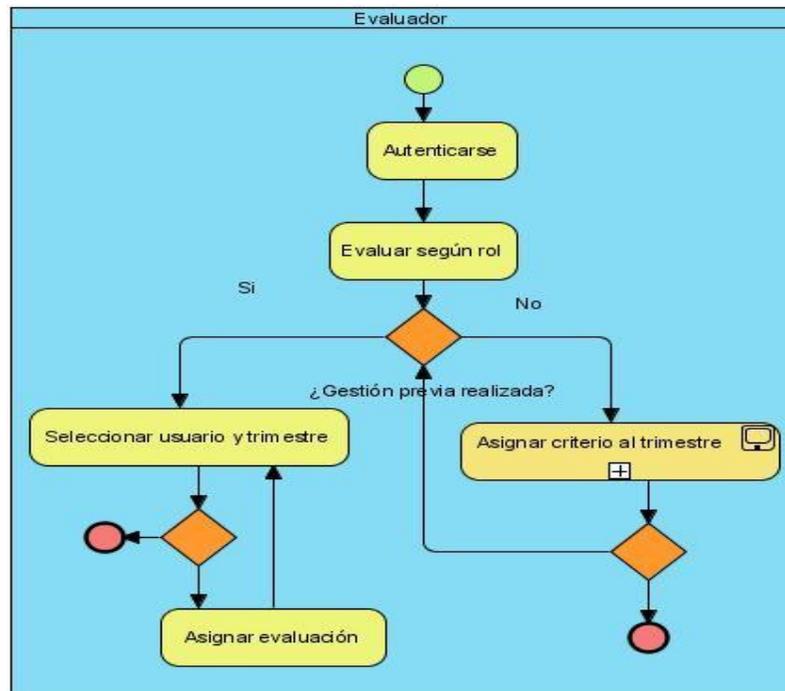


Figura 3: Proceso de negocio. Evaluación según rol del proyecto

Proceso de Negocio: Evaluar personal

En este proceso se evaluará a todo el personal teniendo en cuenta los criterios a evaluar en el trimestre, dando como resultado si la persona es relevante, destacada o insatisfactoria.

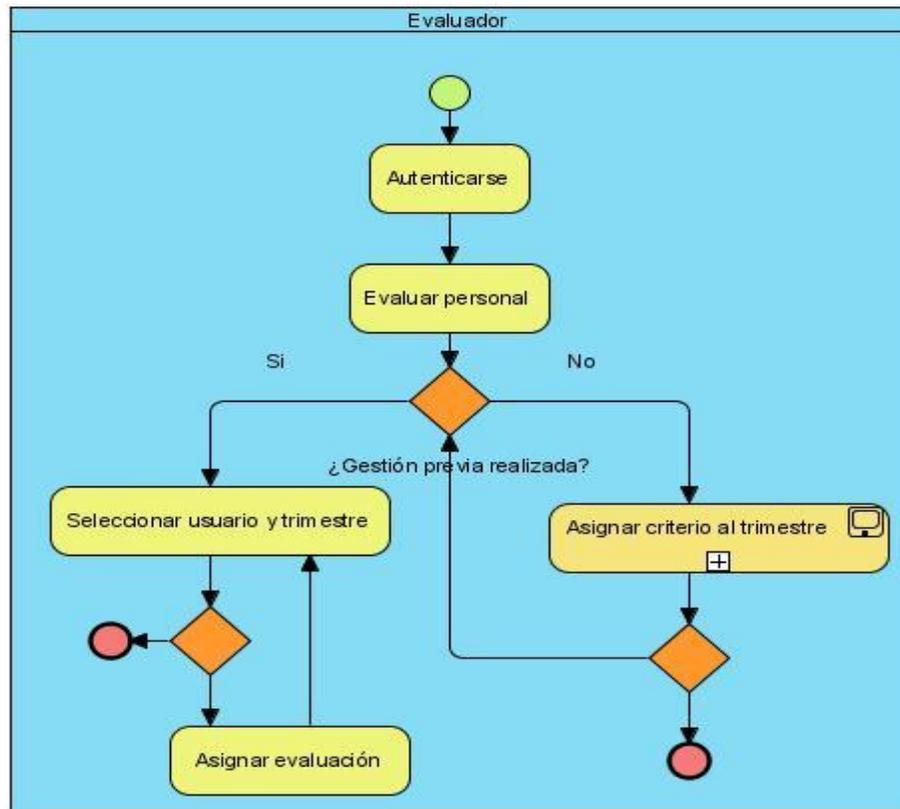


Figura 4: Proceso de negocio. Evaluación Personal

Proceso de Negocio: Evaluar estudiante

En este proceso se evaluará al estudiante teniendo en cuenta el cumplimiento de sus tareas asignadas y los criterios a evaluar en el trimestre deseado dando como resultado el rendimiento del estudiante en su puesto de trabajo.

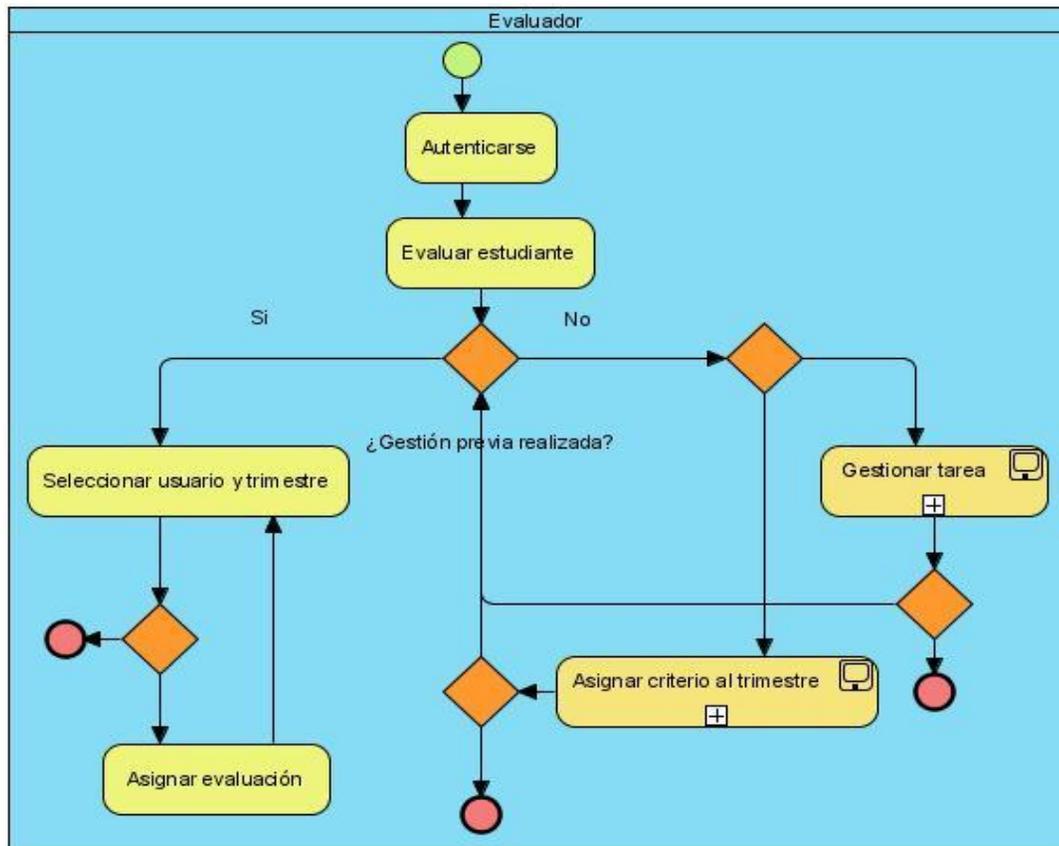


Figura 5: Proceso de negocio. Evaluación Estudiante

Proceso de Negocio: Evaluar profesor

En este proceso es de los más importantes ya que se evaluará al profesor teniendo en cuenta los criterios a evaluar en el trimestre, las tareas asignadas dando como resultado la retribución salarial ya sea básico, adecuado y superior.

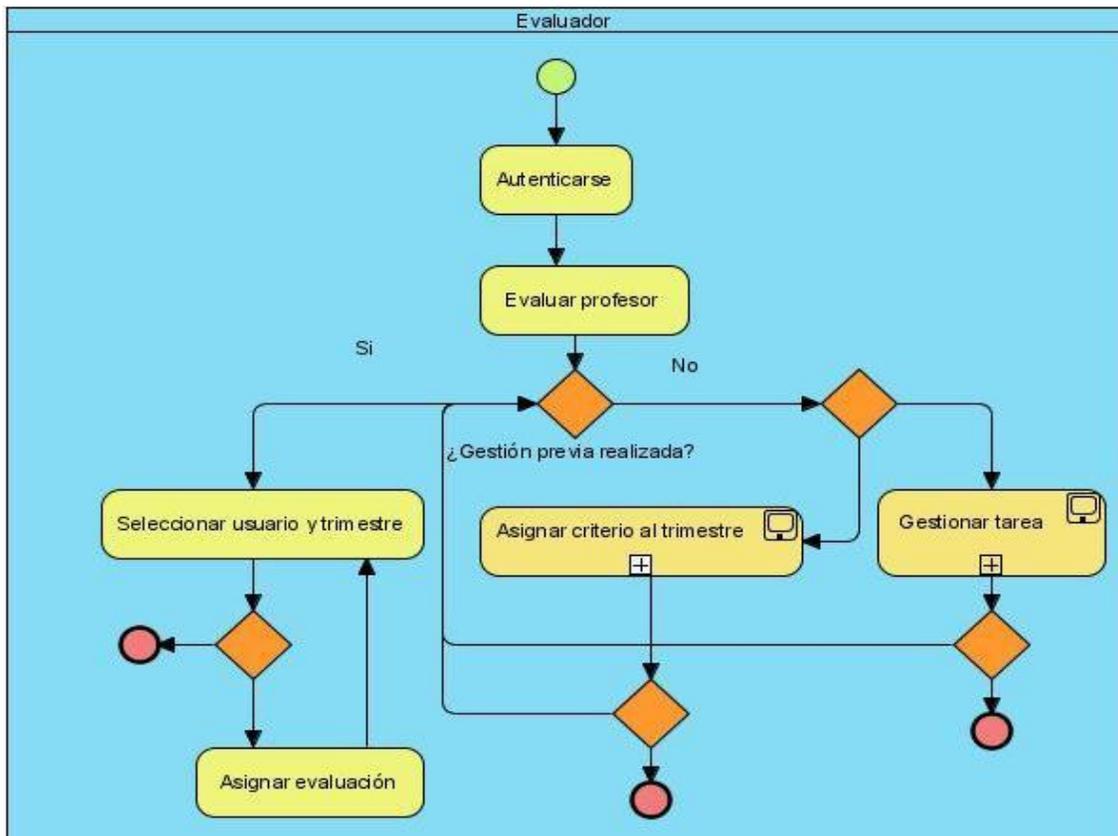


Figura 6: Proceso de negocio. Evaluación Profesor

Proceso de Negocio: Gestionar tarea

En este proceso se le asignará a cada persona sus tareas a cumplir por trimestre donde esas tareas pueden ser distintas dependiendo si es un estudiante o un profesor.

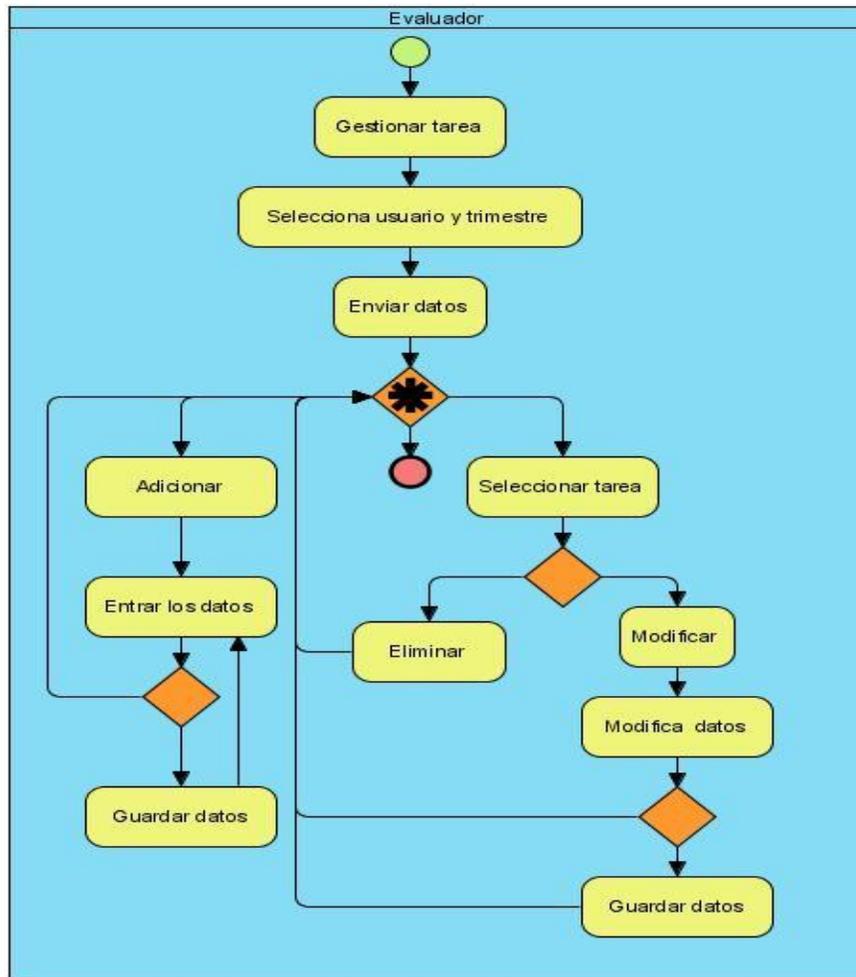


Figura 7: Proceso de negocio. Gestionar Tareas

Proceso de Negocio: Gestionar Incidencia

En este proceso se gestionan las incidencias de un usuario en el trimestre que se desea evaluar. Estas incidencias influyen de forma negativa en la evaluación trimestral del usuario.

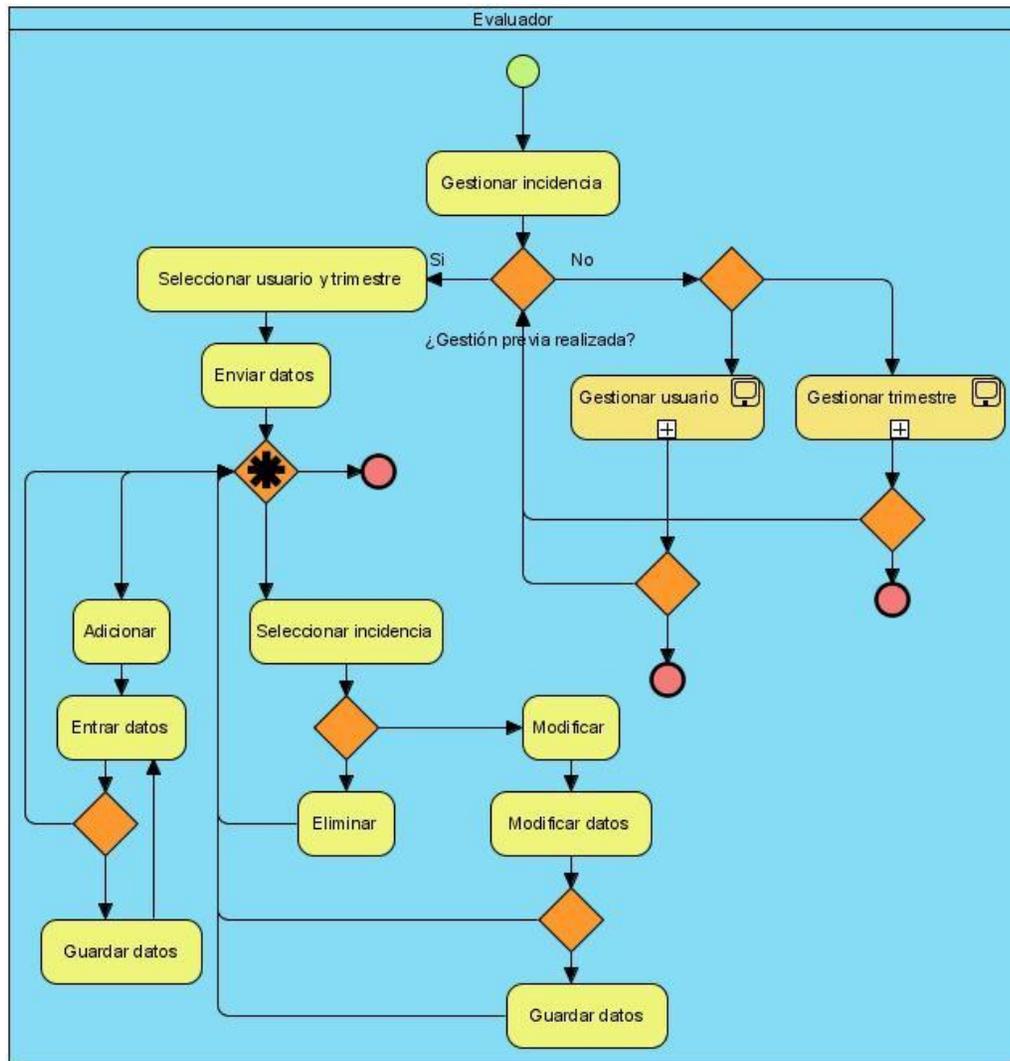


Figura 8: Proceso de negocio. Gestionar Incidencias

Proceso de Negocio: Presentar Información

En este proceso el usuario podrá ver sus incidencias y evaluaciones en el trimestre que desee, además podrá acceder a la opción de cambiar contraseña. La creación del menú se hace teniendo en cuenta si el usuario es estudiante o profesor, de manera que un estudiante no podrá ver la ficha de evaluación como profesor, así mismo un profesor no podrá ver la ficha de evaluación como estudiante.

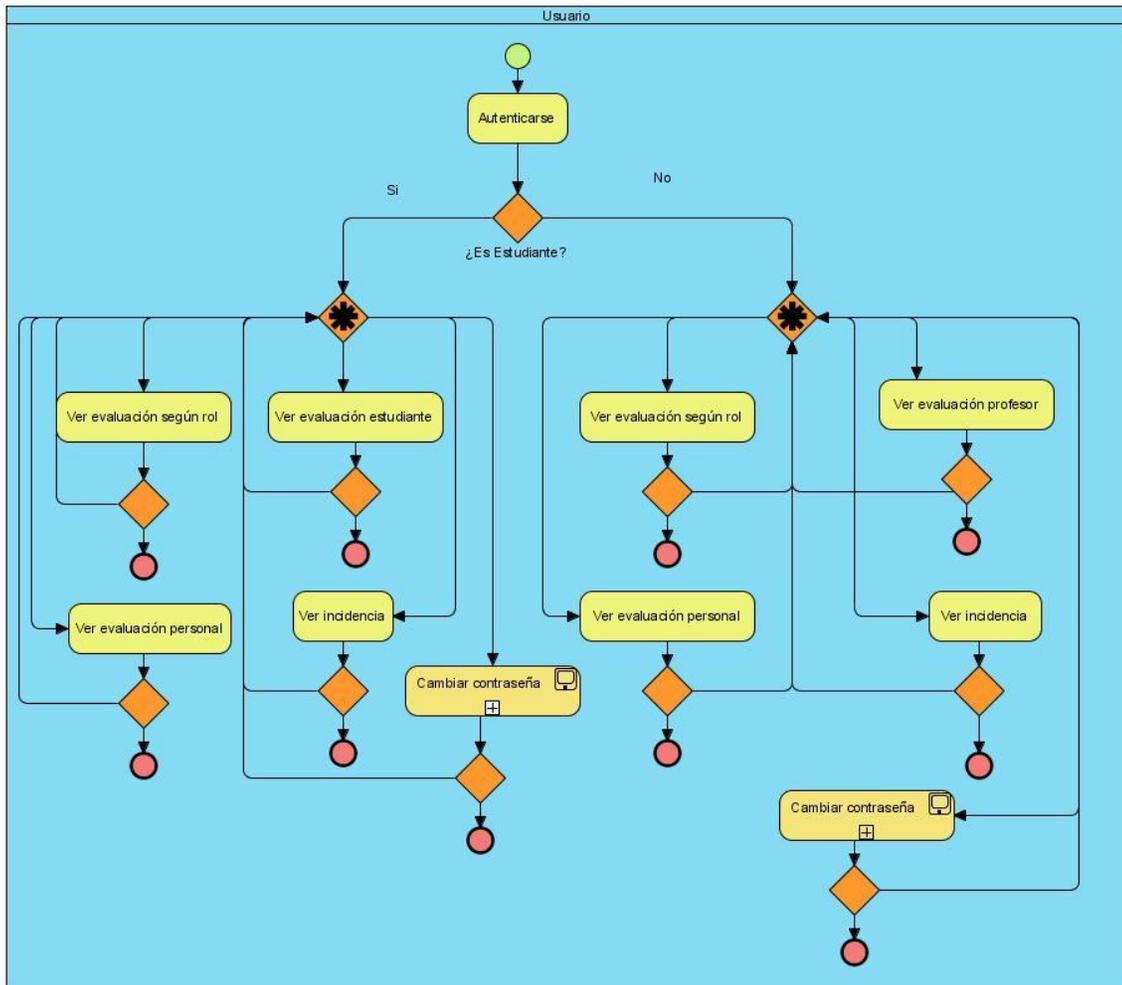


Figura 9: Proceso de negocio. Presentar Información

2.4 Requisitos funcionales

El levantamiento de requisitos es el proceso mediante el cual se establecen los servicios que el sistema debe brindar y las restricciones que debe cumplir. Para su realización se consultó la tesis “Estrategia para la gestión de la evaluación del rendimiento del personal en el proyecto ERP-Cuba”. A continuación se enumera los requisitos funcionales del sistema.

R1 Administración

R1.1 Gestionar Rol

R1.1.1 Adicionar Rol

R1.1.2 Eliminar Rol

R1.1.3 Modificar Rol

R1.2 Gestionar Rol del Proyecto

R1.2.1 Adicionar Rol del Proyecto

R1.2.2 Eliminar Rol del Proyecto

R1.2.3 Modificar Rol del Proyecto

R1.3 Gestionar Usuario

R1.3.1 Adicionar Usuario

R1.3.2 Eliminar Usuario

R1.3.3 Modificar Usuario

R1.4 Gestionar Trimestre

R1.4.1 Adicionar Trimestre

R1.4.2 Eliminar Trimestre

R1.4.3 Modificar Trimestre

R1.6 Cambiar contraseña

R1.6.1 Modificar contraseña

R2 Evaluación según el Rol

R2.1 Gestionar Criterio

R2.1.1 Adicionar Criterio

R2.1.2 Eliminar Criterio

R2.1.3 Modificar Criterio

R2.2 Asignar Criterio al Trimestre

R.2.2.1 Adicionar criterio al trimestre

R.2.2.2 Eliminar criterio al trimestre

R2.3 Asignar Evaluación

R2.3.1 Gestionar observación

R2.3.1.1 Adicionar observación

R2.3.1.2 Eliminar observación

R2.3.1.3 Modificar observación

R3 Evaluación Personal

R3.1 Gestionar Criterio

R3.1.1 Adicionar criterio

R3.1.2 Eliminar criterio

R3.1.3 Modificar criterio

R3.2 Asignar criterio al trimestre

R.3.2.1 Adicionar criterio

R.3.2.2 Eliminar criterio

R3.3 Asignar Evaluación

R3.3.1 Gestionar Acciones Formativas

R3.3.1.1 Adicionar acción formativa

R3.3.1.2 Eliminar acción formativa

R3.3.1.3 Modificar acción formativa

R3.3.2 Gestionar observación

R3.3.2.1 Adicionar observación

R3.3.2.2 Eliminar observación

R3.3.2.3 Modificar observación

R4 Evaluación estudiante

R4.1 Gestionar criterio

R4.1.1 Adicionar criterio

R4.1.2 Eliminar criterio

R4.1.3 Modificar criterio

R4.2 Asignar criterio al trimestre

R4.2.1 Adicionar criterio

R4.2.2 Eliminar criterio

R4.2.3 Modificar criterio

R4.3 Asignar Evaluación

R4.3.1 Gestionar observación

R4.3.1.1 Adicionar observación

R4.3.1.2 Eliminar observación

R4.3.1.3 Modificar observación

R4.4 Gestionar tarea

R4.4.1 Adicionar tarea

R4.4.2 Eliminar tarea

R4.4.3 Modificar tarea

R5 Evaluación profesor

R5.1 Gestionar criterio

R5.1.1 Adicionar criterio

R5.1.2 Eliminar criterio

R5.1.3 Modificar criterio

R5.2 Asignar criterio al trimestre

R5.2.1 Adicionar criterio

R5.2.2 Eliminar criterio

R5.2.3 Modificar criterio

R5.3 Asignar Evaluación

R5.3.1 Gestionar observación

R5.3.1.1 Adicionar observación

R5.3.1.2 Eliminar observación

R5.3.1.3 Modificar observación

R5.4 Gestionar tarea

R5.4.1 Adicionar tarea

R5.4.2 Eliminar tarea

R5.4.3 Modificar tarea

R6 Asignar incidencia

R6.1 Adicionar incidencia

R6.2 Eliminar incidencia

R6.3 Modificar incidencia

2.5 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto usable, rápido o confiable. Los requisitos no funcionales aunque no aportan funcionalidades propiamente dichas dentro de la aplicación, son de vital importancia para una puesta en marcha exitosa del software, y para lograr que este responda a las expectativas del usuario.

A continuación se exponen los requisitos no funcionales de nuestro Sistema.

Requerimientos de software

CARACTERÍSTICAS DEL SISTEMA

- Sistema Operativo Microsoft Windows XP; Sistema Operativo Linux con ambiente gráfico KDE o GNOME.

Requerimientos de apariencia e interfaz externa

- La aplicación debe poseer una interfaz amigable al usuario, basada en paneles con formularios, con una navegabilidad intuitiva para el usuario.

Restricciones de diseño e implementación

- La aplicación será implementada en Framework Symfony con ExtJs para las interfaces de usuario.
- EL lenguaje de programación utilizado será PHP.
- Para el análisis y el diseño de la aplicación se utilizó el modelo de desarrollo definido en el proyecto ERP-Cuba, usando el lenguaje de modelación BPMN y como herramienta de modelado Visual Paradigm.

Rendimiento del sistema

- La aplicación debe requerir un consumo mínimo de recursos.
- Debe tener tiempos de respuesta rápidos garantizando de esta forma la agilidad del sistema.

Portabilidad

- Una de las mayores ventajas que tendrá es su portabilidad ya que podrá correr en cualquier plataforma, Windows o Linux.

Usabilidad

- La aplicación estará disponible las 24 horas del día.
- Las interfaces orquestadas por la aplicación podrán ser usadas por cualquier persona que posea conocimientos básicos en el manejo de la computadora.

Seguridad

- Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.
- Proteger la información manejada por el sistema de accesos no autorizados.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.
- Verificación sobre acciones irreversibles (eliminaciones).

Hardware

- Capacidad de Disco Duro superior a 15 Giga bytes.
- Se requiere un mínimo de 512 MB de RAM y 2.0 GHz de velocidad de procesamiento.

2.5.1 Especificaciones de requisitos

En las especificaciones de requisitos se registran las características y condiciones definidas con que debe cumplir cada requisito funcional. A continuación se expondrán las descripciones principales, brindando una idea más profunda de su realización. Todas se pueden encontrar en los Documentos de Especificación de Requisitos que se adjuntan a la investigación, constituyendo artefactos generados durante el proceso de captura y descripción de los mismos.

R2 Evaluación según Rol

R2.1 Gestionar Criterio

R2.1.1 Adicionar Criterio

Conceptos tratados	Conceptos	Atributos
	Gestionar criterio	Criterio, Rol del Proyecto

CARACTERÍSTICAS DEL SISTEMA

Precondiciones	Precondiciones	Pre-requisito
	Deben estar agregado los roles del proyecto	Gestionar Rol del Proyecto
Descripción	Se adicionan todos los criterios posibles a evaluar.	
Validaciones	No puede ser vacío. El sistema deberá dar la posibilidad de adicionar tantos criterios como desee.	
Post-condiciones	Se han adicionado todos los criterios a evaluar	
Post-requisito	No procede.	

Tabla 2.1: R2.1.1 Adicionar Criterio

R2.1.2 Eliminar Criterio

Conceptos tratados	Conceptos	Atributos
	Gestionar criterio	
Precondiciones	Precondiciones	Pre-requisito
	Deben estar agregado los criterios	
Descripción	Se pueden eliminar los criterios seleccionados.	
Validaciones	El sistema deberá dar la posibilidad de eliminar tantos criterios como desee.	
Post-condiciones	Se han eliminado los criterios de evaluación.	

CARACTERÍSTICAS DEL SISTEMA

Post-requisito	No procede.
-----------------------	-------------

Tabla 2.2: R2.1.2 Eliminar Criterio

R2.1.3 Modificar Criterio

Conceptos tratados	Conceptos	Atributos
	Gestionar criterio	Criterio, Rol del Proyecto
Precondiciones	Precondiciones	Pre-requisito
	Deben estar agregado los criterios	
Descripción	Se Modifican todos los criterios posibles a evaluar.	
Validaciones	No puede ser vacío El sistema deberá dar la posibilidad de modificar tantos criterios como desee.	
Post-condiciones	Se han modificados todos los criterios a evaluar	
Post-requisito	No procede.	

Tabla 2.3: R2.1.3 Modificar Criterio

R2.2 Asignar criterios al trimestre

Conceptos tratados	Conceptos	Atributos
	Asignar criterios al trimestre	Criterio, Trimestre

CARACTERÍSTICAS DEL SISTEMA

Precondiciones	Precondiciones	Pre-requisito
	Deben estar agregado los criterios a evaluar Deben estar agregados los trimestres	Gestionar criterio Gestionar Trimestre
Descripción	Se seleccionan entre todos los criterios los que se necesitan para evaluar en ese trimestre.	
Validaciones	No puede ser vacío. El sistema debe permitir la selección o eliminación de tantos criterios como desee.	
Post-condiciones	Se han asignados los criterios a evaluar en ese trimestre.	
Post-requisito	No procede.	

Tabla 2.4: R2.2 Asignar criterios al trimestre

R2.3 Asignar Evaluación

R2.3.1 Gestionar observación.

Conceptos tratados	Conceptos	Atributos
	Asignar evaluación	Usuario, Trimestre, Observación
Precondiciones	Precondiciones	Pre-requisito

CARACTERÍSTICAS DEL SISTEMA

	<p>Deberán estar adicionados los usuarios</p> <p>Deberá estar adicionado el trimestre</p>	<p>Gestionar usuario</p> <p>Gestionar Trimestre</p>
Descripción	En la realización de la evaluación se le adicionará, modificará o eliminará una observación al usuario.	
Validaciones	El sistema debe permitir adicionar, modificar o eliminar la observación realizada al usuario.	
Post-condiciones	Se han asignados una observación al usuario dependiendo de su evaluación obtenida.	
Post-requisito	No procede.	

Tabla 2.5: R2.3.1 Gestionar observación.

R3 Evaluación Personal

R3.1 Gestionar criterio

R3.1.1 Adicionar criterio

Conceptos tratados	Conceptos	Atributos
	Gestionar criterio	Criterio
Precondiciones	Precondiciones	Pre-requisito
Descripción	Se adicionan todos los criterios posibles a evaluar.	

CARACTERÍSTICAS DEL SISTEMA

Validaciones	No puede ser vacío El sistema deberá dar la posibilidad de adicionar tantos criterios como desee.
Post-condiciones	Se han adicionado todos los criterios a evaluar
Post-requisito	No procede.

Tabla 2.6: R3.1.1 Adicionar criterio

R3.1.2 Eliminar criterio

Conceptos tratados	Conceptos	Atributos
	Gestionar criterio	
Precondiciones	Precondiciones	Pre-requisito
	Deberán estar adicionados los criterios	
Descripción	Se eliminan todos los criterios que se deseen.	
Validaciones	El sistema deberá dar la posibilidad de eliminar tantos criterios como desee.	
Post-condiciones	Se han eliminado los criterios deseados.	
Post-requisito	No procede.	

Tabla 2.7: R3.1.2 Eliminar criterio

R3.1.3 Modificar criterio

Conceptos tratados	Conceptos	Atributos

CARACTERÍSTICAS DEL SISTEMA

	Gestionar criterio	
Precondiciones	Precondiciones	Pre-requisito
	Deberán estar adicionado los criterios	
Descripción	Se modificarán todos los criterios que se deseen.	
Validaciones	El sistema deberá dar la posibilidad de modificar tantos criterios como se desee.	
Post-condiciones	Se han eliminado los criterios deseados.	
Post-requisito	No procede.	

Tabla 2.8: R3.1.3 Modificar criterio

R3.3 Asignar evaluación

R3.3.1 Gestionar Acciones Formativas

Conceptos tratados	Conceptos	Atributos
	Evaluación personal	Usuario, Trimestre, Acción formativa.
Precondiciones	Precondiciones	Pre-requisito
	Deberán estar adicionados los usuarios Deberá estar adicionado el	Gestionar usuario Gestionar Trimestre

CARACTERÍSTICAS DEL SISTEMA

	trimestre	
Descripción	En la realización de la evaluación se le adicionará, modificará o eliminará las acciones formativas referidas al usuario evaluado.	
Validaciones	El sistema debe permitir adicionar, modificar o eliminar las acciones formativas referidas al usuario.	
Post-condiciones	Se han asignado acciones formativas que el usuario deberá realizar. Estas serán dependiendo de su evaluación obtenida.	
Post-requisito	No procede.	

Tabla 2.9: R3.3.1 Gestionar Acciones Formativas

R4 Evaluación Estudiante

R4.1 Gestionar criterio

R4.1.1 Adicionar criterio

Conceptos tratados	Conceptos	Atributos
		Gestionar criterio
Precondiciones	Precondiciones	Pre-requisito
Descripción	Se adicionan todos los criterios posibles a evaluar.	
Validaciones	No puede ser vacío El sistema deberá dar la posibilidad de adicionar tantos criterios como se desee.	

CARACTERÍSTICAS DEL SISTEMA

Post-condiciones	Se han adicionado todos los criterios a evaluar
Post-requisito	No procede.

Tabla 2.10: R4.1.1 Adicionar criterio

R4.1.2 Eliminar criterio

Conceptos tratados	Conceptos	Atributos
		Gestionar criterio
Precondiciones	Precondiciones	Pre-requisito
	Deberán estar adicionados los criterios	
Descripción	Se eliminan todos los criterios que se deseen.	
Validaciones	El sistema deberá dar la posibilidad de eliminar tantos criterios como se desee.	
Post-condiciones	Se han eliminado los criterios deseados.	
Post-requisito	No procede.	

Tabla 2.11: R4.1.2 Eliminar criterio

R4.1.3 Modificar criterio

Conceptos tratados	Conceptos	Atributos
		Gestionar criterio
Precondiciones	Precondiciones	Pre-requisito

CARACTERÍSTICAS DEL SISTEMA

	Deberán estar adicionados los criterios	
Descripción	Se modificarán todos los criterios que se deseen.	
Validaciones	El sistema deberá dar la posibilidad de modificar tantos criterios como se desee.	
Post-condiciones	Se han eliminado los criterios deseados.	
Post-requisito	No procede.	

Tabla 2.12: R4.1.3 Modificar criterio

R4.4 Gestionar tarea

R4.4.1 Adicionar tarea

Conceptos tratados	Conceptos	Atributos
	Gestionar tarea	Usuario, Trimestre, Tarea, Esfera
Precondiciones	Precondiciones	Pre-requisito
	Deberán estar adicionados los usuarios Deberá estar adicionado el trimestre	Gestionar usuario Gestionar trimestre
Descripción	Se adicionarán todas las tareas que se deseen al estudiante.	

CARACTERÍSTICAS DEL SISTEMA

Validaciones	El sistema deberá dar la posibilidad de adicionar tantas tareas como desee.
Post-condiciones	Se han eliminado todos los criterios que se desearon.
Post-requisito	No procede.

Tabla 2.13: R4.4.1 Adicionar tarea

R4.4.2 Eliminar tarea

Conceptos tratados	Conceptos	Atributos
		Gestionar tarea
Precondiciones	Precondiciones	Pre-requisito
	Deberán estar adicionados las tareas	
Descripción	Se eliminarán todas las tareas que se deseen al estudiante.	
Validaciones	El sistema deberá dar la posibilidad de eliminar tantas tareas como desee.	
Post-condiciones	Se han eliminado todos los criterios que se desearon.	
Post-requisito	No procede.	

Tabla 2.14: R4.4.2 Eliminar tarea

R4.4.3 Modificar tarea

Conceptos tratados	Conceptos	Atributos
		Gestionar tarea

CARACTERÍSTICAS DEL SISTEMA

Precondiciones	Precondiciones	Pre-requisito
	Deberán estar adicionado las tareas	
Descripción	Se modificarán todas las tareas que se deseen al estudiante.	
Validaciones	El sistema deberá dar la posibilidad de modificar tantas tareas como desee.	
Post-condiciones	Se han modificado todos los criterios que se desearon.	
Post-requisito	No procede.	

Tabla 2.15: R4.4.3 Modificar tarea

2.6 Prototipos de interfaces

Prototipo de interfaz del requisito: R1 Administración

En esta interfaz se mostrará en forma de árbol todos los Usuarios creados, permitiendo así, las opciones de agregar un nuevo usuario, modificar o eliminar alguno existente; será necesario seleccionarlo previamente.

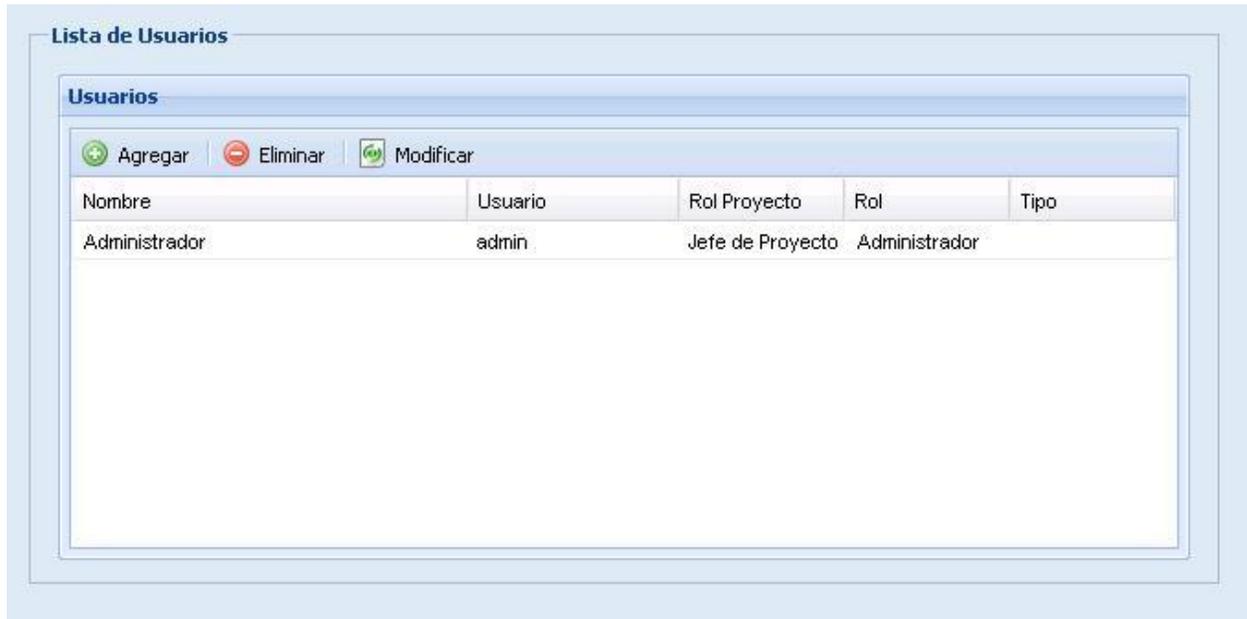


Figura 10: Prototipo de interfaz. Gestionar usuario

Prototipo de interfaz del requisito: R1 Administración

En esta interfaz se mostrará un formulario, permitiendo la opción de modificar la contraseña existente.

Cambiar Contraseña

Contraseña Anterior:

Contraseña Nueva:

Confirmar Contraseña:

Detailed description: This is a software interface prototype for changing a password. It has a light blue border and a title 'Cambiar Contraseña'. It contains three text input fields labeled 'Contraseña Anterior', 'Contraseña Nueva', and 'Confirmar Contraseña'. Below the fields is a button with a green checkmark icon and the text 'Aceptar'.

Figura 11: Prototipo de interfaz. Cambiar contraseña

Prototipo de interfaz del requisito: R1 Administración

En esta interfaz se mostrará en forma de árbol todos los Roles creados, permitiendo así, las opciones de agregar un nuevo Rol, modificar o eliminar alguno existente; será necesario seleccionarlo previamente.



Figura 12: Prototipo de interfaz. Gestionar Rol

Prototipo de interfaz del requisito: R1 Administración

En esta interfaz se mostrará en forma de árbol todos los Roles del Proyecto creados, permitiendo así, las opciones de agregar un nuevo Rol del Proyecto, modificar o eliminar alguno existente; será necesario seleccionarlo previamente.



Figura 13: Prototipo de interfaz. Gestionar rol del proyecto

Prototipo de interfaz del requisito: R1 Administración

En esta interfaz se mostrará en forma de árbol todos los Trimestre creados, permitiendo así, las opciones de agregar un nuevo Trimestre, modificar o eliminar alguno existente; será necesario seleccionarlo previamente.



Figura 14: Prototipo de interfaz. Gestionar trimestre

Prototipo de interfaz del requisito: R1 Administración

En esta interfaz se mostrará en forma de árbol todas las Incidencias creadas, permitiendo así, las opciones de agregar una nueva Incidencia, modificar o eliminar alguna existente; será necesario seleccionarla previamente.

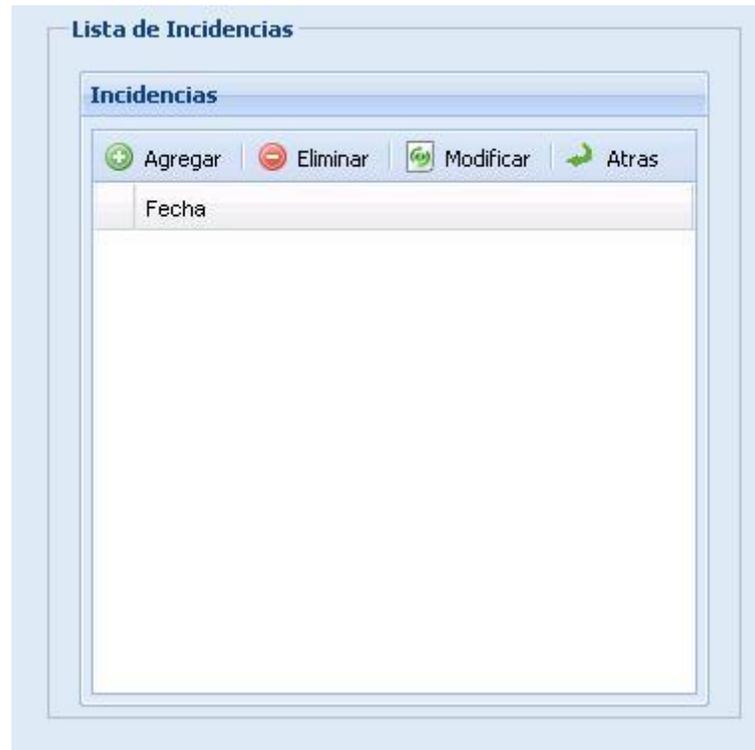


Figura 15: Prototipo de interfaz. Gestionar incidencias

Prototipo de interfaz del requisito: R2 Evaluación según Rol

En esta interfaz se mostrará en el primer y segundo tabulador en forma de árbol todas las Competencias y Responsabilidades creadas, permitiendo así, las opciones de agregar una nueva Competencia o Responsabilidad, modificar o eliminar alguna existente; será necesario seleccionarla previamente.

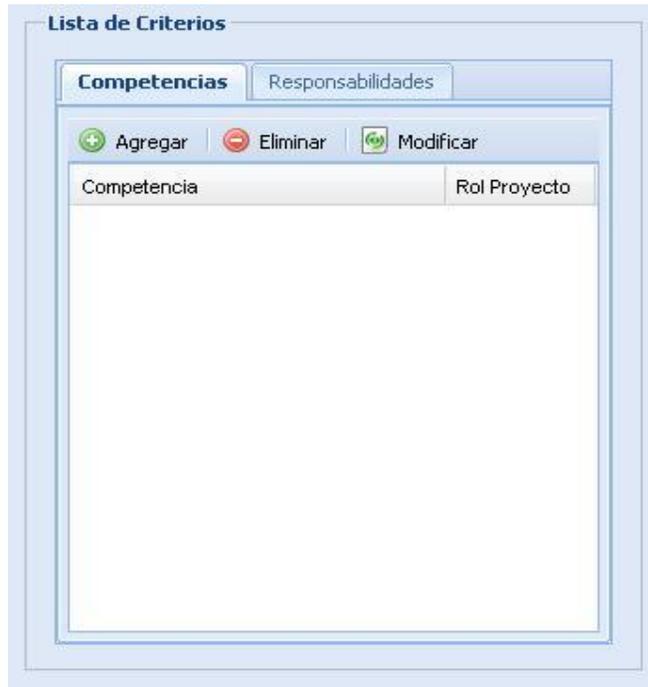


Figura 16: Prototipo de interfaz. Gestionar criterios

Prototipo de interfaz del requisito: R2 Evaluación según Rol

En esta interfaz se mostrará en la parte izquierda en dos tabuladores en forma de árbol las Competencias y Responsabilidades creadas previamente, permitiendo las opciones de agregar una nueva. En parte derecha en dos tabuladores en forma de árbol las Competencias y Responsabilidades adicionadas a evaluar en el trimestre dando la opción de eliminar alguna existente para lo cual será necesario seleccionarla previamente.



Figura 17: Prototipo de interfaz. Asignar criterios al trimestre

Prototipo de interfaz del requisito: R4 Evaluación Estudiante

En esta interfaz se mostrará en forma de árbol todas las tareas por Usuario creadas, permitiendo así, las opciones de agregar una nueva Tarea, modificar o eliminar alguna existente; será necesario seleccionarla previamente.

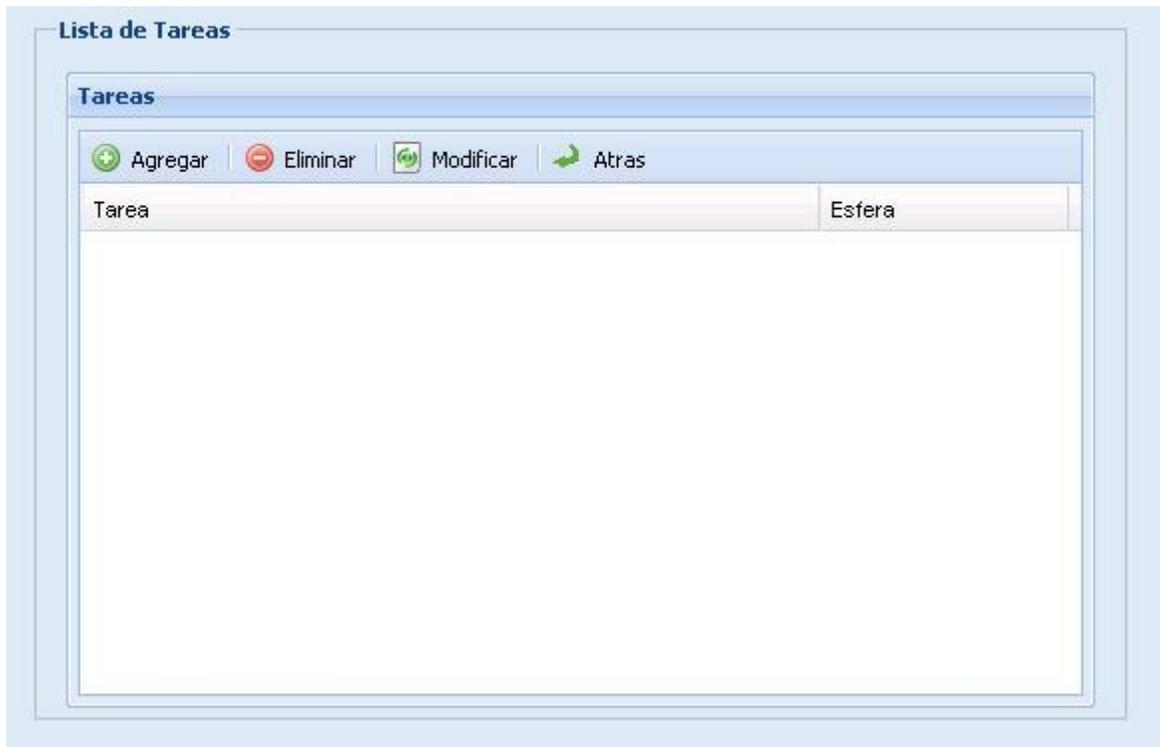


Figura 18: Prototipo de interfaz. Gestionar tareas

2.7 Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución a partir de un listado con las propiedades que debe cumplir el sistema y las funcionalidades que debe realizar, representándolas mediante el modelado de los procesos del negocio y finalmente describiendo paso a paso los requisitos funcionales que intervienen en el sistema. Ahora se puede comenzar a construir la aplicación, tratando de que se cumplan todos los requisitos y las funciones que se han considerado necesarias para este capítulo.

CAPÍTULO 3: DISEÑO DEL SISTEMA

3.1 Introducción

En este capítulo se realiza el diagrama de clases del diseño para el sistema, dando respuesta a la solución que se propone y se define el diagrama de despliegue con el objetivo de mostrar la distribución física de los nodos de cómputo que necesita la aplicación.

Para realizar el diseño de la solución propuesta, se tuvo en cuenta las herramientas a emplearse en la implementación y se realizó un estudio sobre las mismas, por lo que las clases del diseño web se modelaron basadas en el patrón Modelo-Vista-Controlador (MVC), utilizando mecanismos de diseño de clases generadas estándares para todas las vistas creadas con ExtJs Framework. Además se incluyó un paquete llamado Propel del cual heredaran las clases generadas en base de datos o mapeadas, incluidas en el domain.

3.2 Diagramas de Clases del diseño

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de software y de las interfaces, en ellas se puede encontrar las siguientes informaciones:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de los atributos.
- Navegabilidad.
- Dependencias.

3.2.1 Herramientas consultadas para realizar el Diseño

Para realizar el modelado del diseño de clase se utilizaron diferentes herramientas. A continuación se dará una breve introducción de estas:

ExtJs Framework: es un framework para Java Script muy utilizado en el desarrollo de aplicaciones Web con AJAX. Tiene una librería inmensa que permite configurar las interfaces Web de manera semejante a aplicaciones de escritorio. Contiene la mayoría de los controles de los formularios Web incluyendo Grids para mostrar datos y elementos semejantes a la programación de escritorio como los formularios, paneles, barras de herramientas, menús y muchos otros. Dentro de su librería de componentes incluye elementos para el manejo de datos, lectura de XML, lectura de datos JSON e implementaciones basadas en AJAX. Presenta el uso de Java Script con una programación orientada a objetos (Diseño Avanzado de Aplicaciones Web., 2009).

Propel: es un potente y completo sistema ORM (Object Relational Mapper) para PHP 5.2+ . Entre muchas otras cosas tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (favorablemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. (Diseño Avanzado de Aplicaciones Web., 2009).

A continuación se muestran los diagramas de clases del diseño correspondientes a los procesos del negocio críticos del sistema:

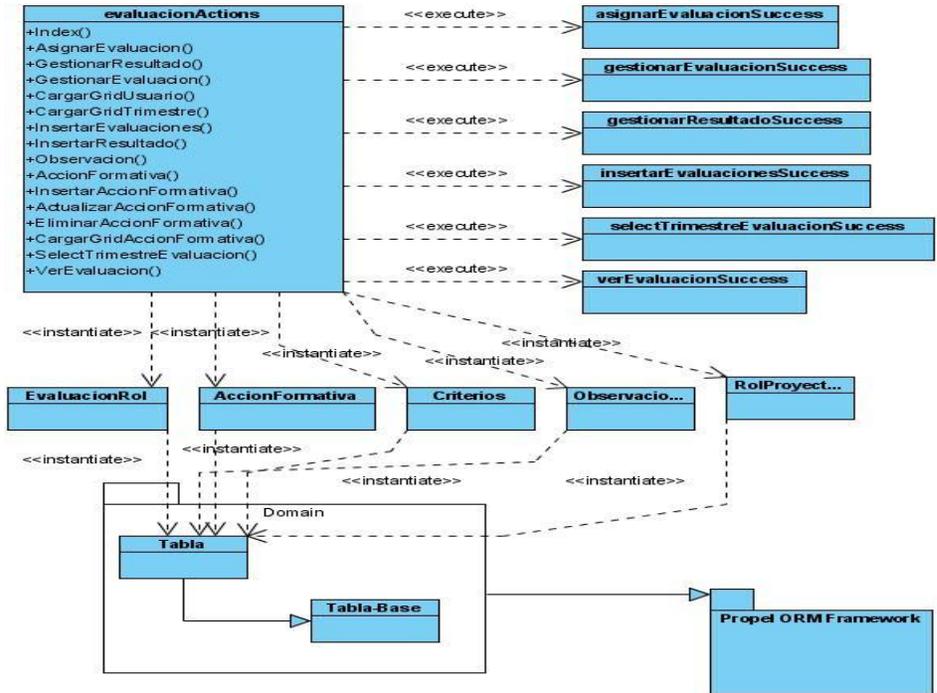


Figura 19: Diagrama de clases. Evaluación según Rol

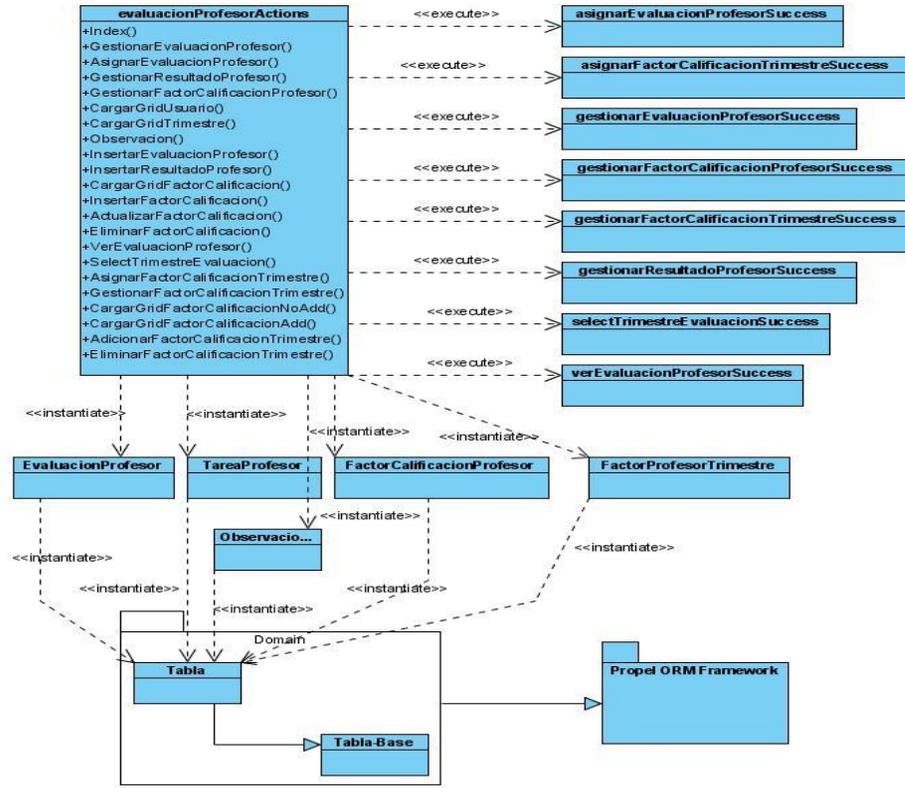


Figura 20: Diagrama de clases. Evaluación del profesor

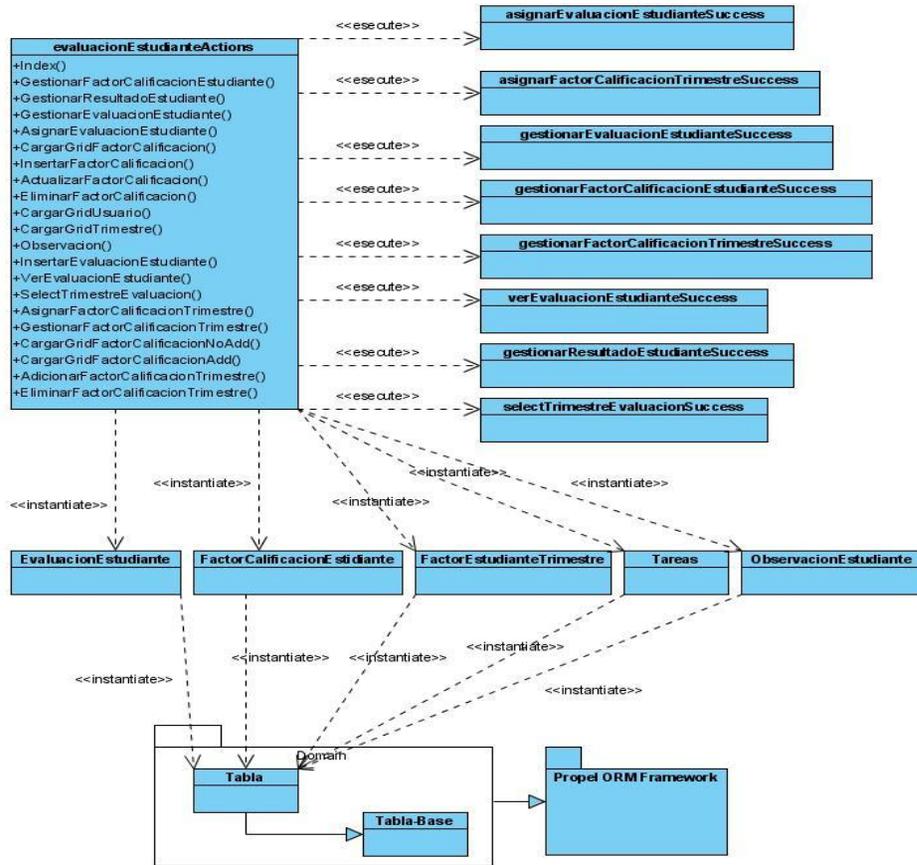


Figura 21: Diagrama de clases. Evaluación del estudiante

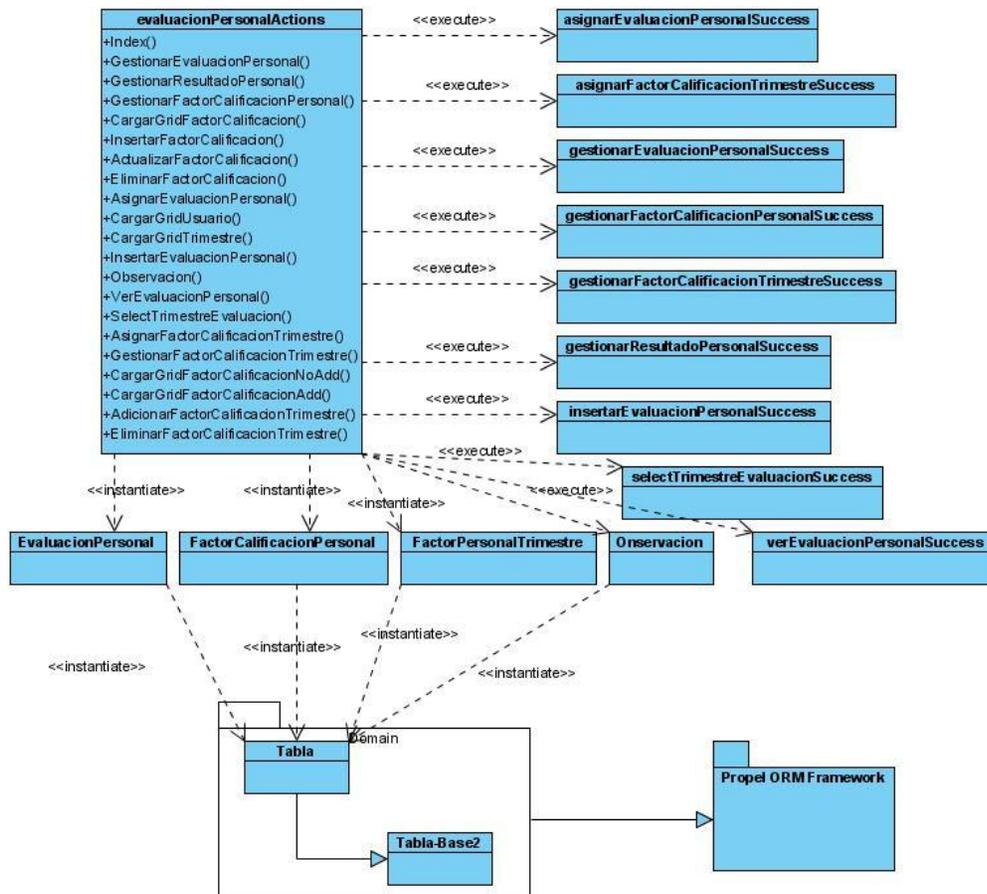


Figura 22: Diagrama de clases. Evaluación del personal

3.2.2 Descripción de las clases

A continuación se realizan las descripciones de las clases. En ellas se muestran datos como sus funciones con una breve descripción de sus responsabilidades, sus atributos y además el tipo de clase, permitiendo así comprender las funcionalidades de las clases en el sistema.

Nombre: asignacionCriterioActions	
Tipo de Clase: Controladora	
Atributo	Tipo

Para cada responsabilidad:	
Nombre:	Descripción:
executeAsignarCriterioTrimestreRol()	Carga el área de selección del trimestre y el rol.
executeGestionarCriterioTrimestreRol()	Carga el área de gestión de criterios por Trimestres.
executeCargarGridTrimestre()	Carga el grid de Trimestre.
executeCargarGridRolProyecto()	Cargar el grid de Rol del Proyecto.
executeCargarGridCompetencia()	Carga el grid de Competencia.
executeCargarGridResponsabilidad()	Carga el grid de Responsabilidad.
executeCargarGridCompetenciaAdd()	Carga el grid de las Competencias agregadas.
executeCargarGridResponsabilidadAdd()	Carga el grid de las Responsabilidades agregadas.
executeAdicionarCriterioEva()	Adiciona los criterios de Evaluación por Trimestre.
executeEliminarCriterioEva()	Elimina los criterios de Evaluación por Trimestre.

Tabla 3.1: Descripción de clase. asignacionCriterioActions

Nombre: criterioActions	
Tipo de Clase: Controladora	
Atributo	Tipo

Para cada responsabilidad:	
Nombre:	Descripción:
executeGestionarCriterio()	Carga el área para adicionar, eliminar y modificar los criterios.
executeEliminarCriterio()	Elimina el criterio seleccionado.
executeCargarCombRolProyecto()	Carga el combobox con los datos del Rol del Proyecto.
executeCargarGridCompetencia()	Carga el grid con las Competencias.
executeInsertarCompetencia()	Adiciona una Competencia.
executeActualizarCompetencia()	Actualiza los datos de una Competencia.
executeCargarGridResponsabilidad()	Carga el grid con las Responsabilidades.
executeInsertarResponsabilidad()	Adiciona una Responsabilidad.
executeActualizarResponsabilidad()	Modifica los datos de una Responsabilidad.

Tabla 3.2: Descripción de clase. criterioActions

Nombre: evaluacionActions	
Tipo de Clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
executeAsignarEvaluacion()	Carga el área de selección del Usuario y el Trimestre.

executeGestionarEvaluacion()	Carga el área donde se realiza la evaluación.
executeCargarGridUsuario()	Carga el grid con los datos de los Usuarios.
executeCargarGridTrimestre()	Carga el grid con los datos de los Trimestres.
executeInsertarEvaluaciones()	Realiza las evaluaciones previas.
executeObservacion()	Adiciona una Observación al Usuario.
executeAccionFormativa()	Carga el área para adicionar Acciones Formativas.
executeInsertarAccionFormativa()	Adiciona Acciones Formativas al Usuario.
executeActualizarAccionFormativa()	Modifica las Acciones Formativas.
executeEliminarAccionFormativa()	Elimina las Acciones Formativas.
executeCargarGridAccionFormativa()	Carga el grid con los datos de las Acciones Formativas.
executeSelectTrimestreEvaluacion()	Carga el área de selección del Trimestre que desee ver su evaluación.
executeVerEvaluacion()	Carga el área para ver la Evaluación.

Tabla 3.3: Descripción de clase. evaluacionActions

Nombre: evaluacionEstudianteActions	
Tipo de Clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:

executeGestionarFactorCalificacionEstudiante()	Carga el área para adicionar, eliminar o modificar los criterios a evaluar.
executeGestionarEvaluacionEstudiante()	Carga el área donde se realiza la evaluación.
executeAsignarEvaluacionEstudiante()	Carga el área para seleccionar el usuario y el trimestre a evaluar.
executeCargarGridFactorCalificacion()	Carga el grid con los datos de los criterios a evaluar.
executeInsertarFactorCalificacion()	Adiciona los criterios por trimestre.
executeActualizarFactorCalificacion()	Modifica los criterios por trimestre.
executeEliminarFactorCalificacion()	Elimina los criterios por trimestre.
executeCargarGridUsuario()	Carga el grid con los datos de los Usuarios.
executeCargarGridTrimestre()	Carga el grid con los datos de los trimestres.
executeObservacion()	Adiciona, Modifica o Elimina la Observación realizada a un usuario.
executeInsertarEvaluacionEstudiante()	Realiza la evaluación previa.
executeVerEvaluacionEstudiante()	Carga el área para ver la Evaluación.
executeSelectTrimestreEvaluacion()	Carga el área de selección del Trimestre que desee ver su evaluación.
executeAsignarFactorCalificacionTrimestre()	Carga el área para seleccionar el trimestre
executeGestionarFactorCalificacionTrimestre()	Asigna los criterios que se evaluarán en el trimestre seleccionado.
executeCargarGridFactorCalificacionNoAdd()	Carga el grid con los datos de los criterios no agregados.

executeCargarGridFactorCalificacionAdd()	Carga el grid con los datos de los criterios agregados.
executeAdicionarFactorCalificacionTrimestre()	Adiciona los criterios al trimestre.
executeEliminarFactorCalificacionTrimestre()	Elimina los criterios del trimestre.

Tabla 3.4: Descripción de clase. evaluacionEstudianteActions

Nombre: evaluacionPersonalActions	
Tipo de Clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
executeGestionarFactorCalificacionPersonal()	Carga el área para adicionar, eliminar o modificar los criterios a evaluar.
executeGestionarEvaluacionPersonal()	Carga el área donde se realiza la evaluación.
executeAsignarEvaluacionPersonal()	Carga el área para seleccionar el usuario y el trimestre a evaluar.
executeCargarGridFactorCalificacion()	Carga el grid con los datos de los criterios a evaluar.
executeInsertarFactorCalificacion()	Adiciona los criterios por trimestre.
executeActualizarFactorCalificacion()	Modifica los criterios por trimestre.
executeEliminarFactorCalificacion()	Elimina los criterios por trimestre.
executeCargarGridUsuario()	Carga el grid con los datos de los Usuarios.
executeCargarGridTrimestre()	Carga el grid con los datos de los trimestres.

executeObservacion()	Adiciona, Modifica o Elimina la Observación realizada a un usuario.
executeInsertarEvaluacionPersonal()	Realiza la evaluación previa.
executeVerEvaluacionPersonal()	Carga el área para ver la Evaluación.
executeSelectTrimestreEvaluacion()	Carga el área de selección del Trimestre que desee ver su evaluación.
executeAsignarFactorCalificacionTrimestre()	Carga el área para seleccionar el trimestre
executeGestionarFactorCalificacionTrimestre()	Asigna los criterios que se evaluarán en el trimestre seleccionado.
executeCargarGridFactorCalificacionNoAdd()	Carga el grid con los datos de los criterios no agregados.
executeCargarGridFactorCalificacionAdd()	Carga el grid con los datos de los criterios agregados.
executeAdicionarFactorCalificacionTrimestre()	Adiciona los criterios al trimestre.
executeEliminarFactorCalificacionTrimestre()	Elimina los criterios del trimestre.

Tabla 3.5: Descripción de clase. evaluacionPersonalActions

Nombre: evaluacionProfesorActions	
Tipo de Clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
executeGestionarFactorCalificacionProfesor()	Carga el área para adicionar, eliminar o

	modificar los criterios a evaluar.
executeGestionarEvaluacionProfesor()	Carga el área donde se realiza la evaluación.
executeAsignarEvaluacionProfesor()	Carga el área para seleccionar el usuario y el trimestre a evaluar.
executeCargarGridFactorCalificacion()	Carga el grid con los datos de los criterios a evaluar.
executeInsertarFactorCalificacion()	Adiciona los criterios por trimestre.
executeActualizarFactorCalificacion()	Modifica los criterios por trimestre.
executeEliminarFactorCalificacion()	Elimina los criterios por trimestre.
executeCargarGridUsuario()	Carga el grid con los datos de los Usuarios.
executeCargarGridTrimestre()	Carga el grid con los datos de los trimestres.
executeObservacion()	Adiciona, Modifica o Elimina la Observación realizada a un usuario.
executeInsertarEvaluacionProfesor()	Realiza la evaluación previa.
executeVerEvaluacionProfesor()	Carga el área para ver la Evaluación.
executeSelectTrimestreEvaluacion()	Carga el área de selección del Trimestre que desee ver su evaluación.
executeAsignarFactorCalificacionTrimestre()	Carga el área para seleccionar el trimestre
executeGestionarFactorCalificacionTrimestre()	Asigna los criterios que se evaluarán en el trimestre seleccionado.
executeCargarGridFactorCalificacionNoAdd()	Carga el grid con los datos de los criterios no agregados.
executeCargarGridFactorCalificacionAdd()	Carga el grid con los datos de los criterios agregados.

executeAdicionarFactorCalificacionTrimestre()	Adiciona los criterios al trimestre.
executeEliminarFactorCalificacionTrimestre()	Elimina los criterios del trimestre.

Tabla 3.6: Descripción de clase. evaluacionProfesorActions

Nombre: rolActions	
Tipo de Clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
executeGestionarRol()	Carga el área para adicionar, eliminar y modificar los Roles del Sistema.
executeInsertarRol()	Adiciona el Rol del Sistema.
executeEliminarRol()	Elimina el Rol del Sistema seleccionado
executeActualizarRol()	Actualiza los datos del Rol del Sistema.
executeCargarGridRol()	Carga el grid con los datos de los roles del Sistema
executeGestionarRolProyecto()	Carga el área para adicionar, eliminar y modificar los Roles del Proyecto.
executeInsertarRol Proyecto ()	Adiciona el Rol del Proyecto.
executeEliminarRol Proyecto()	Elimina el Rol del Proyecto seleccionado
executeActualizarRol Proyecto ()	Actualiza los datos del Rol del Proyecto.
executeCargarGridRol Proyecto ()	Carga el grid con los datos de los roles del Proyecto.

Tabla 3.7: Descripción de clase. rolActions

Nombre: tareaActions	
Tipo de Clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
executeAsignarTareaProfesor()	Carga el área para seleccionar el profesor y el trimestre.
executeCargarGridProfesor()	Carga el grid con los datos del profesor
executeCargarGridTrimestre()	Carga el grid con los datos del trimestre
executeCargarCombEsferas()	Carga un combobox con datos del profesor
executeGestionarTareaProfesor()	Carga el área para adicionar, modificar o eliminar tareas del profesor.
executeInsertarTareaProfesor()	Adiciona una tarea al profesor
executeEliminarTareaProfesor()	Elimina una tarea al profesor.
executeActualizarTareaProfesor()	Modifica una tarea al profesor
executeCargarGridTareaProfesor()	Carga el grid con las tareas del profesor.
executeAsignarTareaEstudiante()	Carga el área para seleccionar el estudiante y el trimestre.
executeGestionarTareaEstudiante()	Carga el área para adicionar, modificar o eliminar tareas del estudiante.
executeInsertarTareaEstudiante()	Adiciona una tarea al estudiante

executeActualizarTareaEstudiante()	Modifica una tarea al estudiante
executeEliminarTareaEstudiante()	Elimina una tarea al estudiante.
executeCargarGridTareaEstudiante()	Carga el grid con las tareas del estudiante.
executeCargarGridEstudiante()	Carga el grid con los datos del estudiante

Tabla 3.8: Descripción de clase. tareaActions

Nombre: usuarioActions	
Tipo de Clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Descripción:
executeGestionarUsuario()	Carga el área para adicionar, modificar o eliminar un usuario.
executeInsertarUsuario()	Adiciona un usuario
executeEliminarUsuario()	Elimina un usuario.
executeActualizarUsuario()	Modifica los datos del usuario
executeCargarGridUsuario()	Carga el grid con los datos de los usuarios.
executeCargarCombRoles()	Carga el grid con los datos de los roles del sistema.
executeCargarCombRolesProyecto()	Carga el grid con los datos de los roles del proyecto.
executeCambiarPassword()	Carga el área para cambiar la contraseña del usuario

executeCambiarPass()	Actualiza la contraseña del usuario
executeCargarGridTrimestre()	Carga el grid con los datos de los Trimestres
executeGestionarIncidencia()	Carga el área para adicionar, modificar o eliminar las incidencias.
executeAsignarIncidencia()	Carga el área para seleccionar el usuario y el trimestre.
executeCargarGridIncidencia()	Carga el grid con los datos de las incidencias del usuario seleccionado
executeInsertarIncidencia()	Adiciona una Incidencia
executeEliminarIncidencia()	Elimina una Incidencia
executeActualizarIncidencia()	Modifica una Incidencia

Tabla 3.9: Descripción de clase. usuarioActions

3.3 Diagrama Entidad Relación de la BD

Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de un esquema gráfico empleando los terminología de entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver con el diagramado y se distinguen de otros por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades esta representada por la relación del modelo.

A continuación se mostrarán los diagramas de entidad de relación de los principales módulos del sistema.

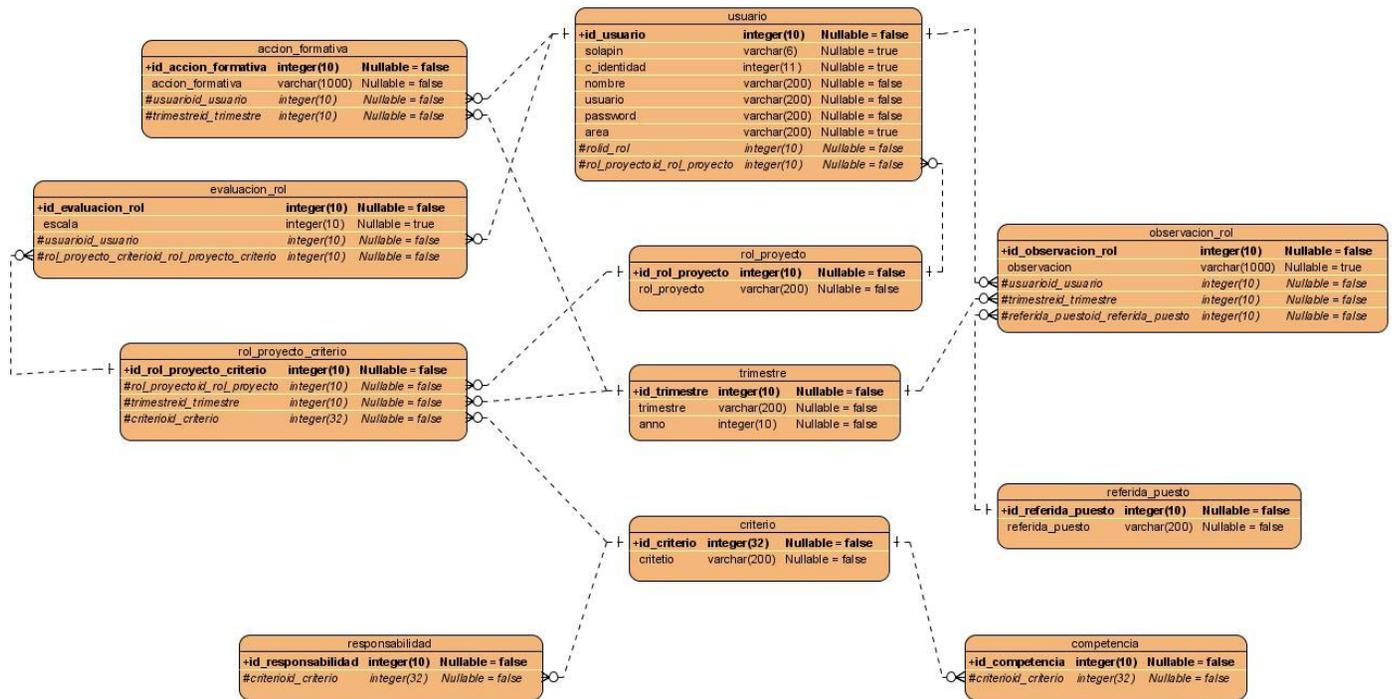


Figura 23: Diagrama de entidad relación Evaluación según el Rol del Proyecto.

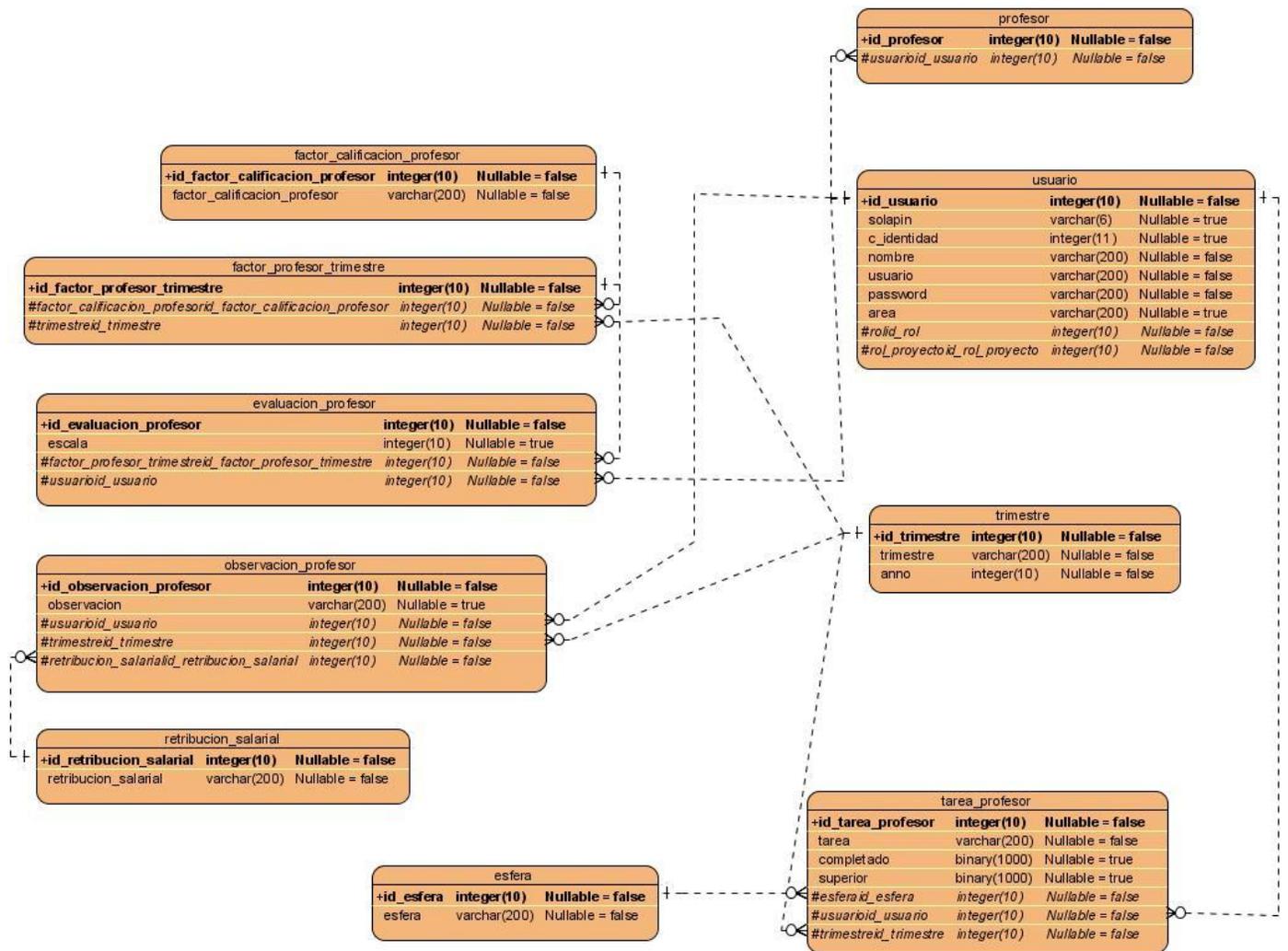


Figura 24: Diagrama de entidad relación Evaluación del profesor.

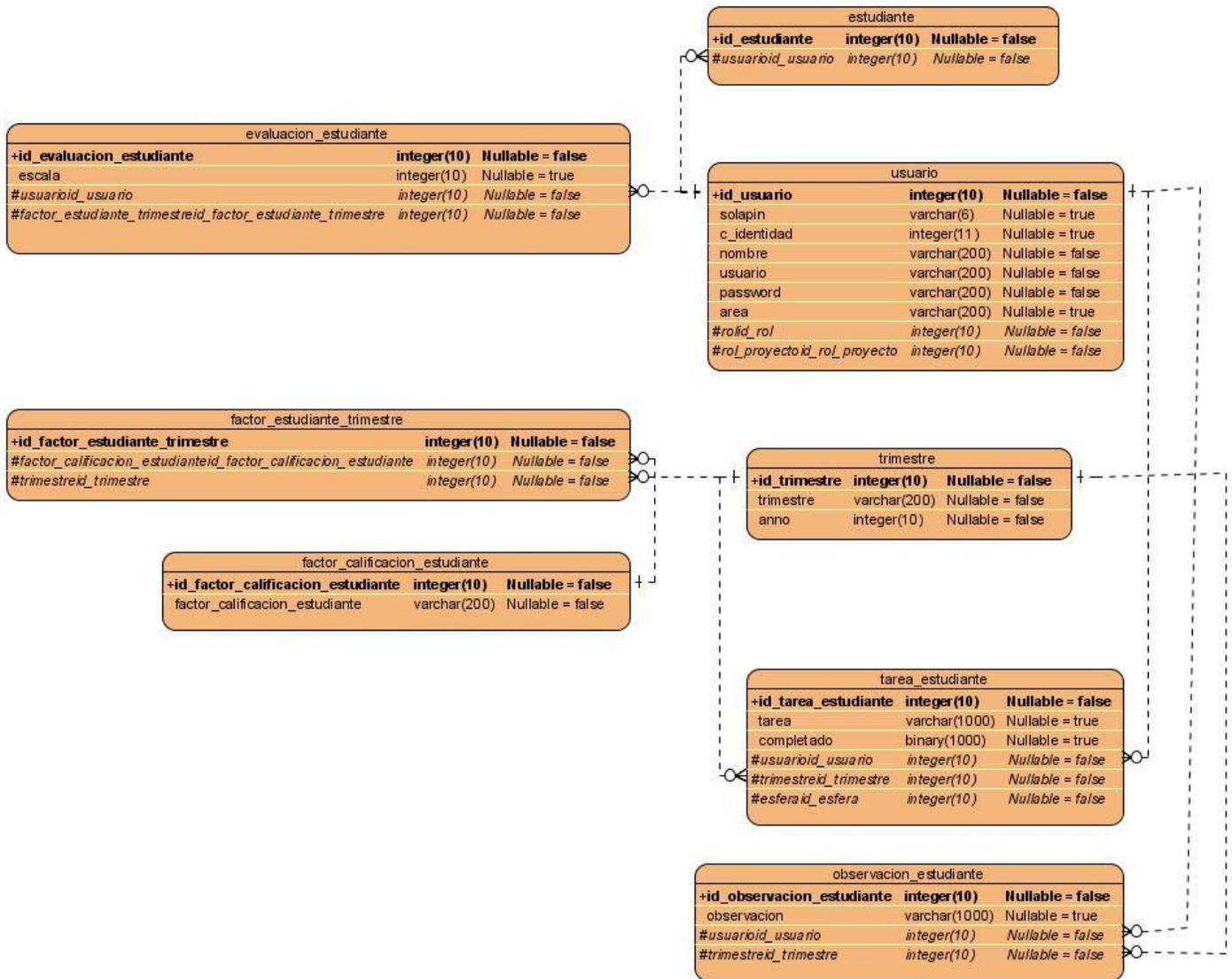


Figura 25: Diagrama de entidad relación Evaluación del estudiante.

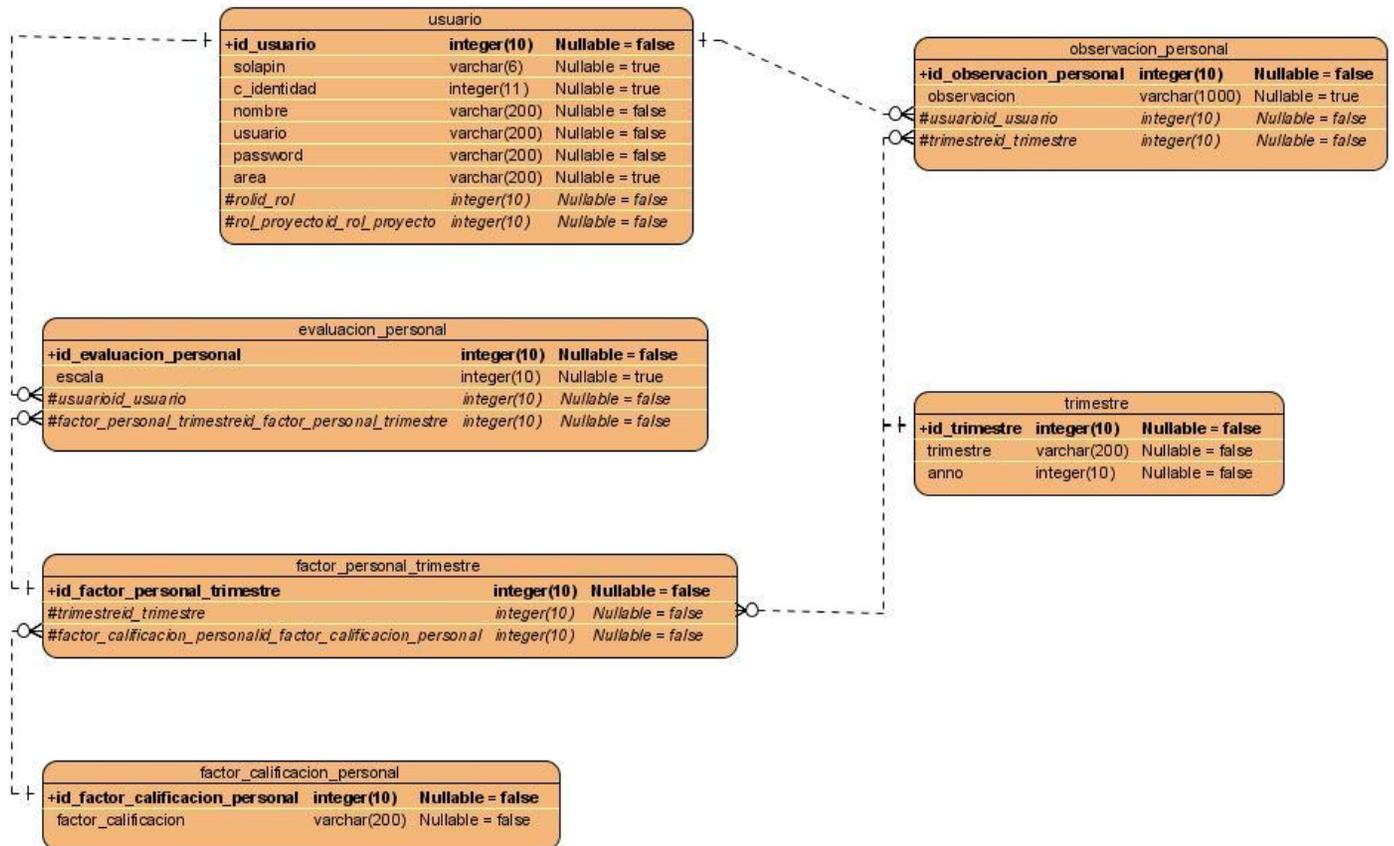


Figura 26: Diagrama de entidad relación Evaluación del personal.

3.4 Conclusiones

En este capítulo, se ha expuesto todo lo referente al diseño, las herramientas que fueron estudiadas y utilizadas para el mismo y el patrón empleado, en este caso el Modelo-Vista-Controlador, que fue el que sirvió de basamento para la realización del diseño como tal. Se incluyen los diagramas de diseño de las clases web de algunos componentes y el diagrama de componentes.

CAPITULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

En el presente capítulo se establecen criterios asociados a la validación del sistema propuesto, partiendo del estudio y puesta en práctica el método de Caja Negra. Para lograr este propósito se formularon los casos de pruebas afines con los procesos de negocio más significativos para la arquitectura, teniendo en cuenta para su construcción la Técnica de Partición de Equivalencia. Además, se tienen el Diagrama de Componentes y de Despliegue del sistema. En el flujo de trabajo de la prueba se verifica el resultado de la implementación, probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros.

4.2 Diagrama de despliegue



Figura 27: Diagrama de despliegue.

4.3 Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre los componentes del software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

A continuación se muestra el diagrama de componente del sistema:

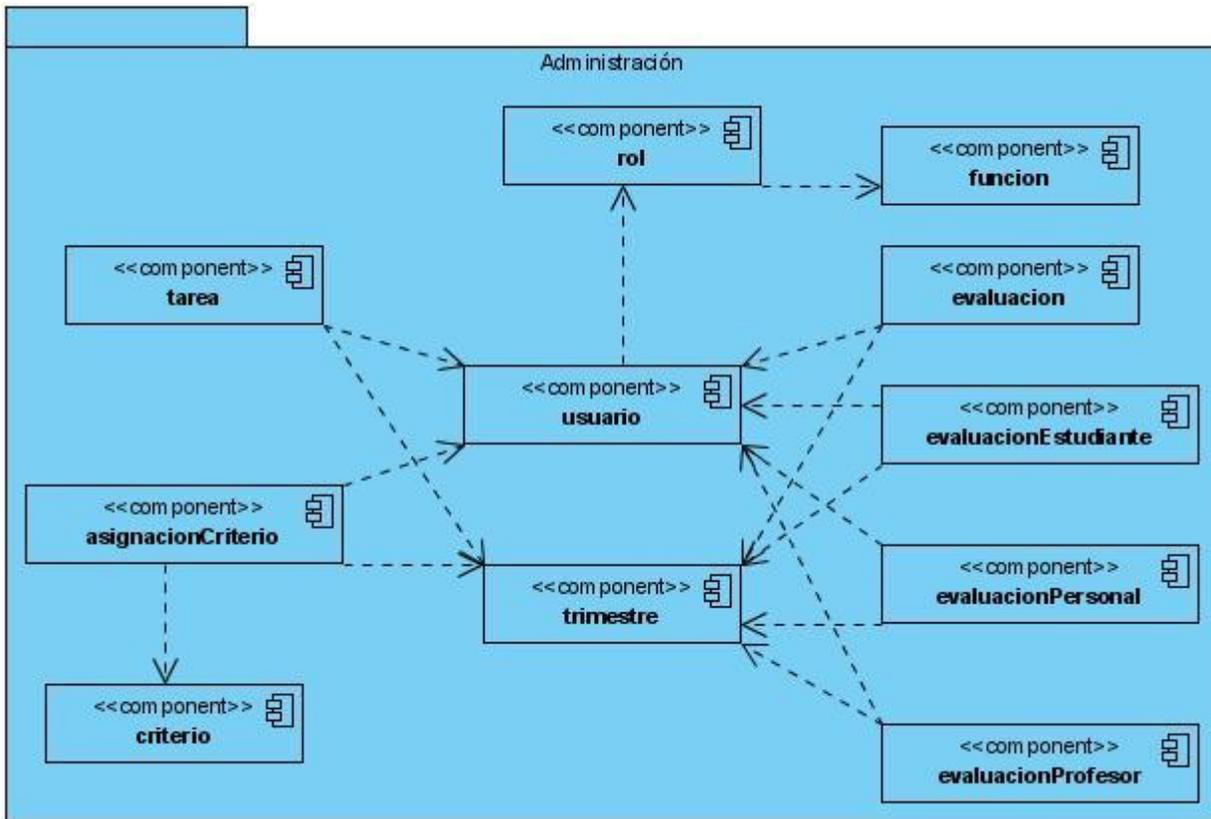


Figura 28: Diagrama de componente.

4.3.1 Descripción y presentación del Diagrama de componentes

El Sistema de Gestión del Rendimiento del Personal está formado por un conjunto de componentes que encapsulan funcionalidades. Los mismos se interrelacionan entre sí para dar respuesta a las necesidades de lograr agrupar todas las características y especificidades en un único sistema y que permita tener un óptimo control del personal en el proyecto ERP-Cuba.

Componentes definidos:

Evaluación: En este componente se gestionan todas las evaluaciones del personal según el rol que desempeñe en el proyecto dando un resultado final teniendo en cuenta los criterios evaluados.

Evaluación personal: En este componente se gestionan todas las evaluaciones de forma personal teniendo en cuenta los criterios evaluados en el trimestre.

Evaluación estudiante: En este componente se gestionan todas las evaluaciones de los estudiantes teniendo en cuenta los criterios evaluados en el trimestre y las tareas cumplidas arrojando un resultado de su rendimiento en su puesto de trabajo.

Evaluación profesor: En este componente se gestionan todas las evaluaciones de los profesores teniendo en cuenta los criterios evaluados en el trimestre y las tareas cumplidas arrojando un resultado de su rendimiento en el proyecto.

Criterio: En este componente se gestionan los criterios a evaluar.

Asignación criterio: En este componente se asignan los criterios a evaluar en cada trimestre.

Usuario: En este componente se gestionan los usuarios de los cuales se va a llevar el control de las evaluaciones.

Rol: En este componente se gestiona el nomenclador Rol que define los privilegios de los usuarios en el sistema y el nomenclador Rol del Proyecto define el rol que desempeña el usuario en el proyecto.

Trimestre: En este componente se gestiona los trimestres que se evalúan.

Tarea: En este componente se gestionan las tareas que se evalúan a los estudiantes y a los profesores para cada trimestre y que se tienen en cuenta en las evaluaciones.

Función: En este componente se gestiona el menú del sistema.

4.4 Tipos de prueba

Con la realización de las pruebas se pretende demostrar que hay errores, en ningún momento es posible asegurar lo contrario. El principal objetivo al diseñar las pruebas es tratar de ejecutar todos los flujos posibles, para descubrir la mayor cantidad de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

- Pruebas de caja negra: Pruebas que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se analiza el código).
- Pruebas de caja blanca: Pruebas que se llevan a cabo sobre el código del software. Es considerada como uno de los tipos de pruebas más importantes que se le aplican a los programas informáticos, logrando como resultado que disminuya en un gran por ciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

4.4.1 Pruebas de caja negra

Para la aplicación de este tipo de prueba se usará el requisito Evaluación según el Rol del Proyecto y Evaluación del Profesor. El objetivo de estos requisitos es realizar las evaluaciones del rendimiento del personal según los criterios y tareas que fueron definidos con antelación.

Requisito a probar: Evaluación según el Rol del Proyecto

Nombre del requisito	Descripción General	Escenarios de Prueba	Flujos del Escenario
Evaluación según el Rol del Proyecto.	Se selecciona el usuario y el trimestre en el que se desea evaluar. Luego se presiona el botón Aceptar ,	Evaluar al usuario según su rol arrojando un resultado de su rendimiento ya sea Insatisfactorio, Destacado o	-Se selecciona el usuario y el trimestre. Se presiona el botón Aceptar . -Se realizan las

	<p>inmediatamente en dependencia del rol del proyecto del usuario se mostrara una interfaz con los datos del usuario, los criterios que se van a evaluar en ese trimestre. En la interfaz se podrán gestionar las acciones formativas teniendo en cuenta los resultados obtenidos, además el evaluador podrá hacer una observación dependiendo del resultado obtenido.</p>	<p>Relevante..</p>	<p>evaluaciones de los criterios.</p> <p>Se presiona el botón Aceptar.</p> <p>-El sistema arroja una evaluación ya sea Insatisfactorio, Destacado o Relevante.</p> <p>-Se gestionan las acciones formativas en caso de necesitarlo el usuario.</p> <p>-Se le da una observación al la usuario evaluado.</p> <p>-Se toma la decisión de mantener al usuario en su puesto de trabajo o cambiarlo y finaliza presionando el botón Terminar.</p>
--	--	--------------------	--

Tabla 4.1: Prueba Caja Negra. Evaluación según el Rol del Proyecto

Requisito a probar: Evaluación del profesor

Nombre del requisito	Descripción General	Escenarios de	Flujos del Escenario

IMPLEMENTACIÓN Y PRUEBA

		Prueba	
Evaluación del Profesor.	Se selecciona el profesor y el trimestre en el que se desea evaluar. Luego se presiona el botón Aceptar , inmediatamente se mostrara una interfaz con los datos del usuario, los criterios de los profesores y las tareas (que pueden ser de superior o de adecuado) a evaluar en ese trimestre. El evaluador podrá hacer una observación dependiendo del resultado obtenido y decidir su retribución salarial.	<p>Evaluar al profesor arrojando un resultado de su rendimiento ya sea Insatisfactorio, Destacado o Relevante.</p> <p>Además propone su retribución salarial teniendo en cuenta el cumplimiento de las tareas que le fueron asignadas.</p>	<p>-Se selecciona el usuario y el trimestre.</p> <p>Se presiona el botón Aceptar.</p> <p>-Se realizan las evaluaciones de los criterios.</p> <p>Se determina si las tareas fueron cumplidas o no.</p> <p>Se presiona el botón Aceptar.</p> <p>-El sistema arroja una evaluación ya sea Insatisfactorio, Destacado o Relevante, además propone su retribución salarial ya sea Básico, Adecuado o Superior.</p> <p>-Se le da una observación al la profesor evaluado.</p> <p>-Se decide cual será la</p>

			retribución salarial y finaliza presionando el botón Terminar .
--	--	--	--

Tabla 4.2: Requisito a probar: Evaluación del profesor

Se le realizaron varias iteraciones de pruebas al sistema. Los resultados más notables se obtuvieron durante las pruebas de Aceptación. Durante estas se comprobó la estabilidad y confiabilidad del sistema, además la aplicación implementada cumplió las expectativas del cliente obteniendo resultados satisfactorios.

4.5 Validación del Modelo de diseño propuesto

El desarrollo de software es algo muy complejo y la medida de su calidad real no es automatizable. Por esta razón la aplicación de métricas de calidad para evaluar el diseño orientado a objeto posee gran importancia. A continuación presentaremos un estudio que brinda un esquema sencillo de implementar y que a la vez cubre los principales atributos de calidad de software. Siendo esto la principal razón de la concepción de las métricas inspiradas en lo propuesto por Pressman (PRESSMAN, 1998).

Atributos de calidad que se abarcan:

1. Responsabilidad. Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
2. Complejidad del diseño. Consiste en la complejidad que posee una estructura de diseño de clases.
3. Complejidad de implementación. Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
4. Reutilización. Consiste en el grado de reutilización de presente en una clase o estructura de clase, dentro de un diseño de software.
5. Acoplamiento. Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, esta muy ligada a la característica de Reutilización.

6. Complejidad del mantenimiento. Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.

7. Cantidad de pruebas. Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, modulo, clase, conjunto de clases, etc.) diseñado.

8. Nivel de Cohesión. Consiste en el grado de especialización de las clases concebidas para modelar un dominio o concepto específico.

Tamaño operacional de clase (TOC): Está dado por el número de métodos asignados a una clase.

Atributo que afecta	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de Implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución en el grado de reutilización de la clase.

Tabla 4.3: Tamaño operacional de clase (TOC)

Relaciones entre clases (RC): Está dado por el número de relaciones de uso de una clase con otras

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad del mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 4.4: Relaciones entre clases (RC)

Resultados del instrumento de evaluación de la métrica Tamaño operacional de clase (TOC)

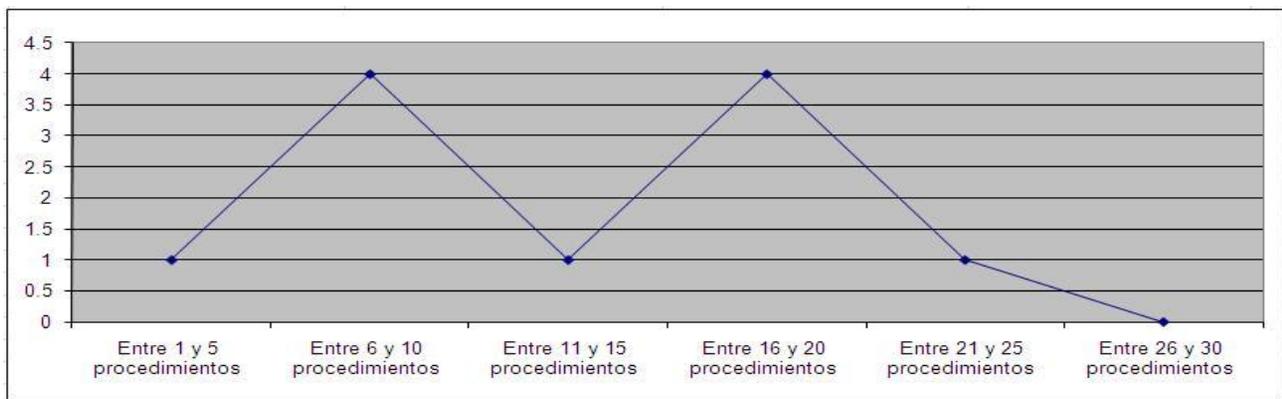


Figura 29: Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.



Figura 30: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.



Figura 31: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.

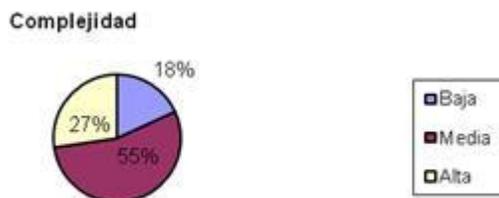


Figura 32: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación.

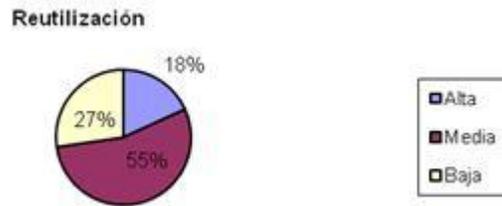


Figura 33: Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica TOC, se puede concluir que el diseño del Sistema de Gestión del Rendimiento del Personal tienen una calidad buena pudiéndose observar las clases poseer menos cantidad de operaciones que la media registrada en las mediciones. Además las clases poseen evaluaciones positivas en los atributos (Responsabilidad, Complejidad de Implementación y Reutilización).

Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC)

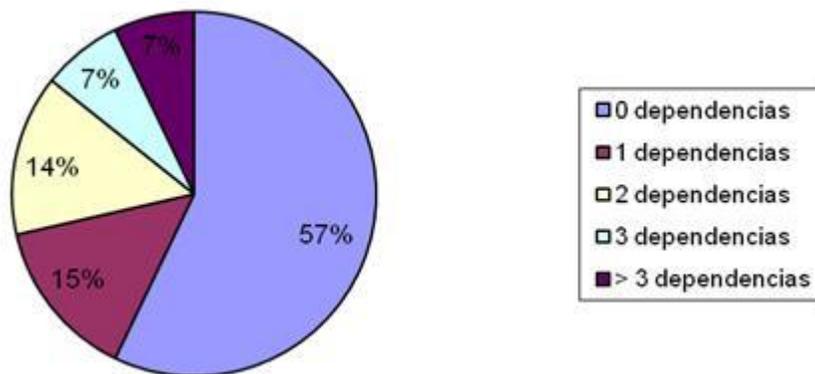


Figura 34: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

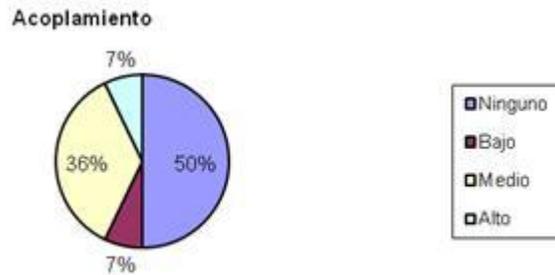


Figura 35: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

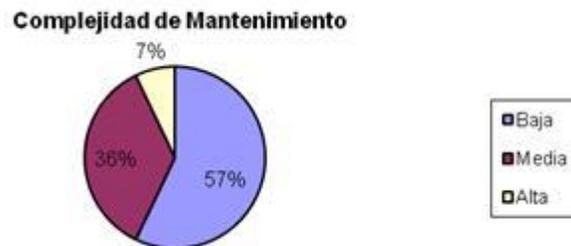


Figura 36: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento.

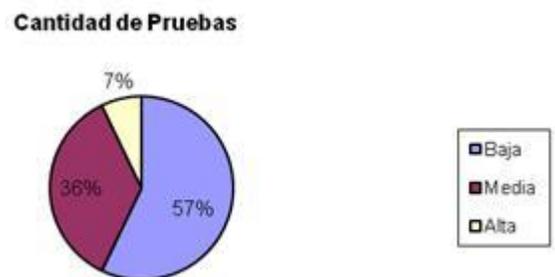


Figura 37: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.

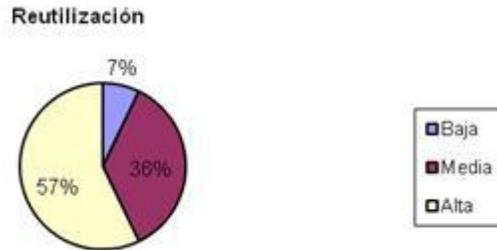


Figura 38: Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica RC, se puede concluir que el diseño de el componente Evaluación según Rol tienen una calidad buena pudiéndose observar que el 86% de las clases posee menos de 3 dependencias de otras clases. Además el 50% de las clases no poseen acoplamiento con otras y el 40% posee índices aceptables en cuanto a Acoplamiento. Así mismo los atributos de calidad Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización se comportan satisfactoriamente en un 57% de las clases.

A manera de resumen se han tabulado los resultados obtenidos en la siguiente tabla:

≤ 0	M
$0 < a \leq 0.5$	R
$0.5 < a \leq 1$	B

Tabla 4.5: Umbrales

	TOC	PH	RC	ND	NOR	Total
Complejidad Implementación	1	-	-	-	-	1
Reutilización	1	-	1	-	-	1
Acoplamiento	-	-	1	-	-	1
Complejidad Mantenimiento	-	-	0.5	-	-	0.5
Cantidad de Pruebas	-	-	0.5	-	-	0.5
Responsabilidad	1	-	-	-	-	1
Total	-	-	-	-	-	1

Tabla 4.6: Resumen de los resultados

4.6 Conclusiones

Finalizada la etapa de pruebas, se obtiene una aplicación funcional que cumple los requisitos del cliente. Estos resultados fueron posibles gracias a la realización de las pruebas descritas en el capítulo. Las no conformidades encontradas en las pruebas se fueron resolviendo a medida que se realizaban las mismas. El módulo Evaluación según el rol del proyecto obtuvo buenos resultados en las pruebas aplicadas y ya más específicamente en la capa de presentación las no conformidades encontradas se relacionaban con errores ortográficos. El módulo tuvo una buena integración con el sistema y las etapas de prueba validaron la solución que se adaptó más a los requisitos y a las necesidades del cliente final que tomó parte en dichas pruebas.

CONCLUSIONES GENERALES

Como resultado de este trabajo se tiene un software funcional, con toda la documentación generada durante su diseño e implementación y que se adapta a los procesos del proyecto ERP-Cuba, a partir del cual se llega a las siguientes conclusiones:

Se le dio solución a los requisitos funcionales y no funcionales del sistema.

La utilización del framework Symfony y ExtJs facilitó el trabajo de desarrollo, a pesar de la complejidad del sistema.

Después de implementada la solución, el sistema se sometió a varias iteraciones de pruebas en las cuales se detectaron un número pequeño de no conformidades, evidenciando el cumplimiento de los requisitos.

Por todo lo anteriormente expuesto, se puede decir que se han cumplido los objetivos trazados al inicio de la investigación.

RECOMENDACIONES

A partir del presente trabajo se proponen las siguientes recomendaciones:

Continuar con las pruebas, para certificar la eficiencia del sistema desarrollado, con el fin de elevar la calidad del producto.

Transmitir experiencias a otros proyectos que utilicen metodología, herramientas y tecnologías similares.

Realizar un estudio para determinar si pueden ser adicionadas nuevas funcionalidades de interés para beneficio de la administración del proyecto ERP-Cuba como puede ser editar la fórmula de evaluación del rendimiento del personal

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

- [1]. **Buschamann, F. 1996.** *Pattern-Oriented Software Architecture*. s.l. : John Wiley & Sons, 1996. ISBN 0471958697.
- [2]. **Eclipse. 2000.** Eclipse.org. [En línea] 2000. [Citado el: 2008 de Abril de 9.] <http://www.eclipse.org/>.
- [3]. **Arias, Yuniel Eliades Proenza. 2009.** *Diseño Avanzado de Aplicaciones Web. 8*, s.l. : Opentelematics, 2009.
- [4]. **Eguíluz Pérez, Javier. 2007.** *Introducción a CSS*. Madrid : s.n., 2007.
- [5]. —. **2007.** *Introducción a JavaScript*. Madrid : s.n., 2007.
- [6]. **Molina, Francisco Javier Pavón. 2006.** *Principles of Programming*. Honolulu : Atlantic International University, 2006.
- [7]. **PRESSMAN, R. S. 1998.** *Ingeniería de software. Un enfoque practico*. 1998.
- [8]. **MySQL AB. 2008.** *MySQL 5.0 Reference Manual*. 2008.
- [9]. **Network-Press.Org. 2003.** Network-Press.Org. [En línea] 2003. [Citado el: 9 de Abril de 2008.] http://www.network-press.org/?navegador_concepto.
- [10]. **Perl. 2008.** Perl.org. [En línea] 22 de Febrero de 2008. [Citado el: 8 de Abril de 2008.] <http://perldoc.perl.org/perlintro.html>.
- [11]. **Reenskaug, Trygve. 2007.** The original MVC reports. Oslo : s.n.,
- [12]. **Eudaldo Morales Gutiérrez. 2002.** OTROS CONCEPTOS Y HERRAMIENTAS DE RR.HH. <http://www.gestiopolis.com/canales/derrhh/articulos/30/grh.htm>

BIBLIOGRAFÍA

Achour, Mehdi, y otros. 2007. *PHP Manual*. Los Ángeles : PHP Documentation Group, 2007.

Álvarez, R. Qué es PHP. <http://www.desarrolloweb.com/articulos/392.php>.

Apache, Software Foundation. 2007. The Apache Software Foundation. [En línea] 2007. [Citado el: 8 de Abril de 2008.] <http://www.apache.org/>.

ASSETS NS. <http://assets.co.cu/assets.asp>

Daniel Pecos. PostgreSQL vs MySQL .[http:// www.netpecos.org/docs/mysql_postgres/x108.html#AEN115](http://www.netpecos.org/docs/mysql_postgres/x108.html#AEN115)

Desarrolloweb.com. Objetivos y usos del XML. <http://www.desarrolloweb.com/articulos/460.php>.

Docente Albert Anibal Osiris SOFIA.2002. Diagrama de Entidad-Relación.

Guervos, J. J. M. Introducción al lenguaje XML. [Online] 2004. <http://geneura.ugr.es/~jimerelo/xml/>.

INSAFORP(El Salvador):

http://www.cinterfor.org.uy/public/spanish/region/ampro/cinterfor/temas/complab/observ/i_d.htm

INTERCAP(Guatemala):

http://www.cinterfor.org.uy/public/spanish/region/ampro/cinterfor/temas/complab/observ/i_c.htm

León, Rolando Alfredo Hernández y González, Sayda Coello. 2002. *El paradigma cuantitativo de la investigación científica*. Ciudad de la Habana : EDUNIV, 2002. ISBN: 959-16-0343-6.

Pérez Subirats, Jorge Luis. 2003. *Diseño informacional de los sitios web*. Ciudad de la Habana, Ciudad de la Habana, Cuba : s.n., 21 de Septiembre de 2003.

Projects Subversion. 2001. Tigris.org. [En línea] 2001. [Citado el: 9 de Abril de 2008.] <http://subversion.tigris.org/>.

RODAS XXI. <http://www.rodasxxi.cu/>

Sanchez, M.A.M. Metodología de desarrollo de software.
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

SENAI(Brasil).

http://www.cinterfor.org.uy/public/spanish/region/ampro/cinterfor/temas/complab/observ/vargas/iii/i_a.htm

Sitio Web oficial Visual-Paradigm. *Sitio Web oficial Visual-Paradigm.* [Online] [Cited: enero 25, 2009.]

[http://www.visual-paradigm.com/product/vpuml/..](http://www.visual-paradigm.com/product/vpuml/)

Stallman, Richard. 2004. *Software Libre para una sociedad libre.* Madrid : Traficantes de Sueños, 2004.

Versat-Sarasola: <http://elecodecontador.blogspot.com/2008/10/el-versat-sarasola-sistema-cubano-de.html>

<http://www.vcl.desoft.cu/smf/index.php?topic=36.0>

ANEXOS

Anexo 1: Diagrama de clase del diseño

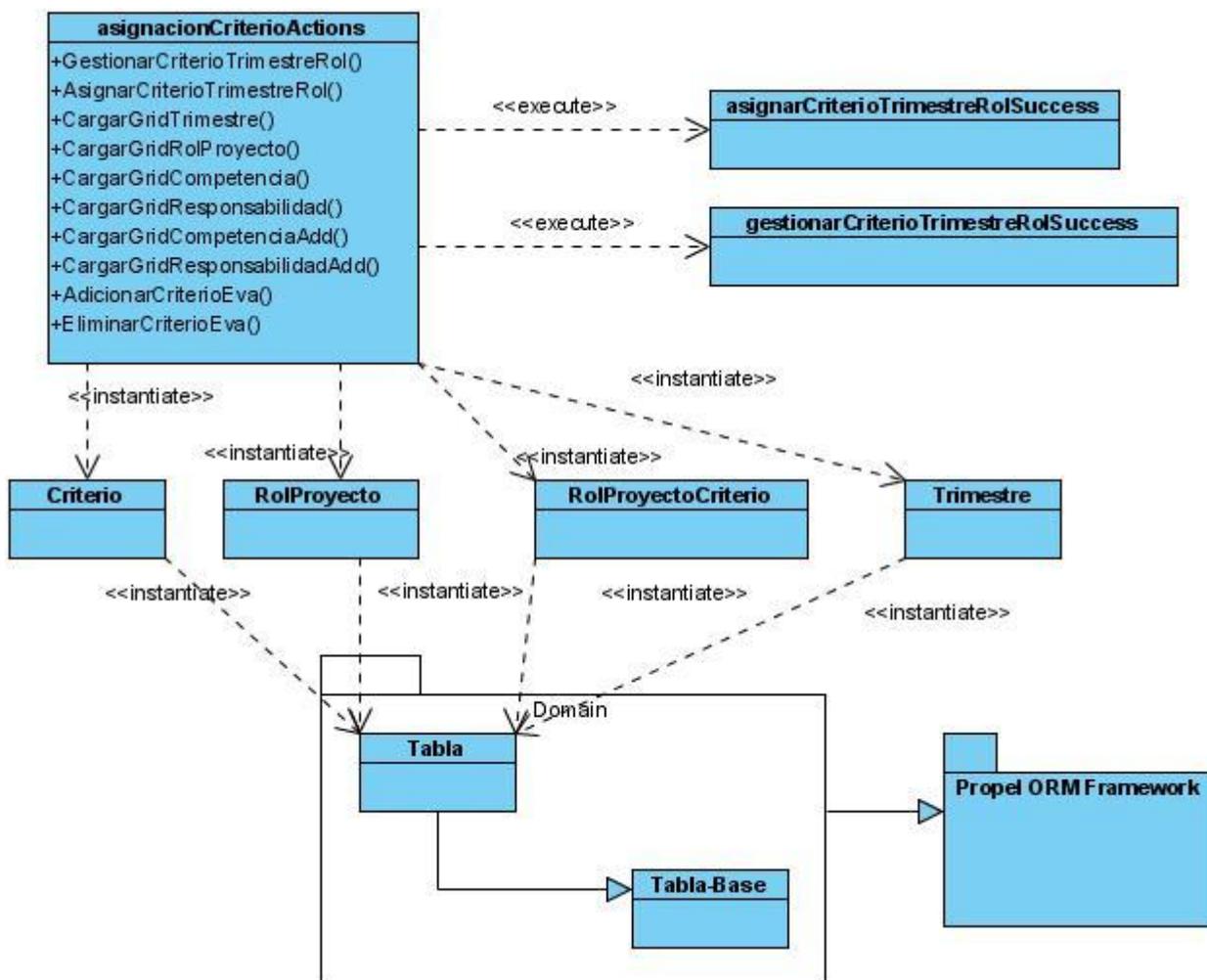


Figura 39: Diagrama de clase del diseño. Asignar Criterio

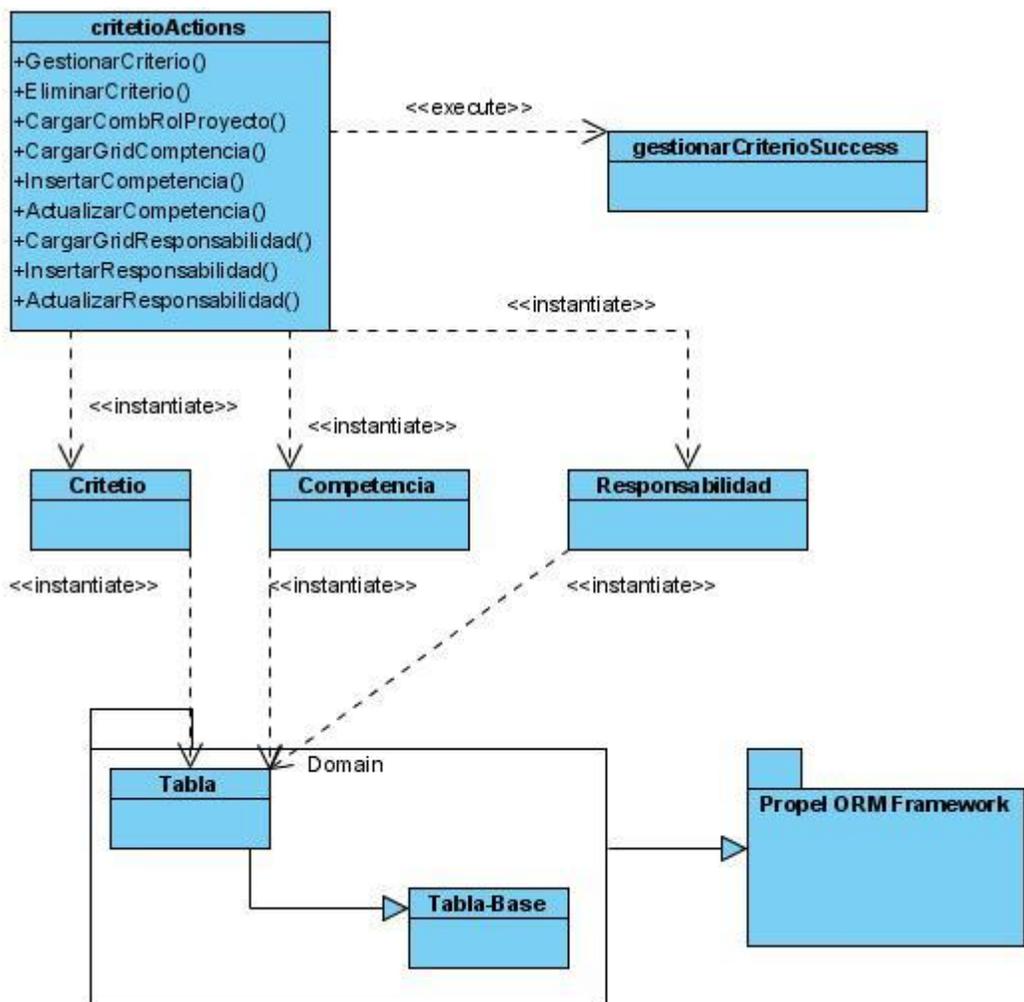


Figura 40: Diagrama de clase del diseño. Gestionar Criterio

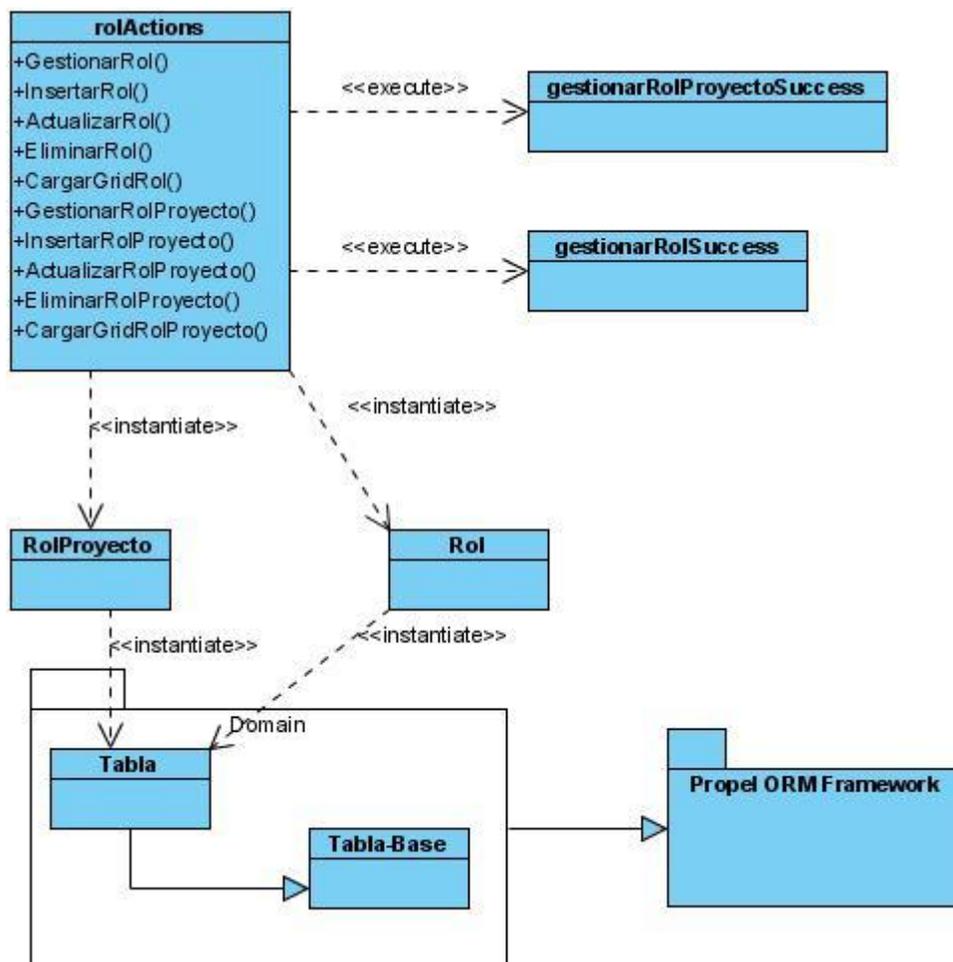


Figura 41: Diagrama de clase del diseño. Gestionar Rol

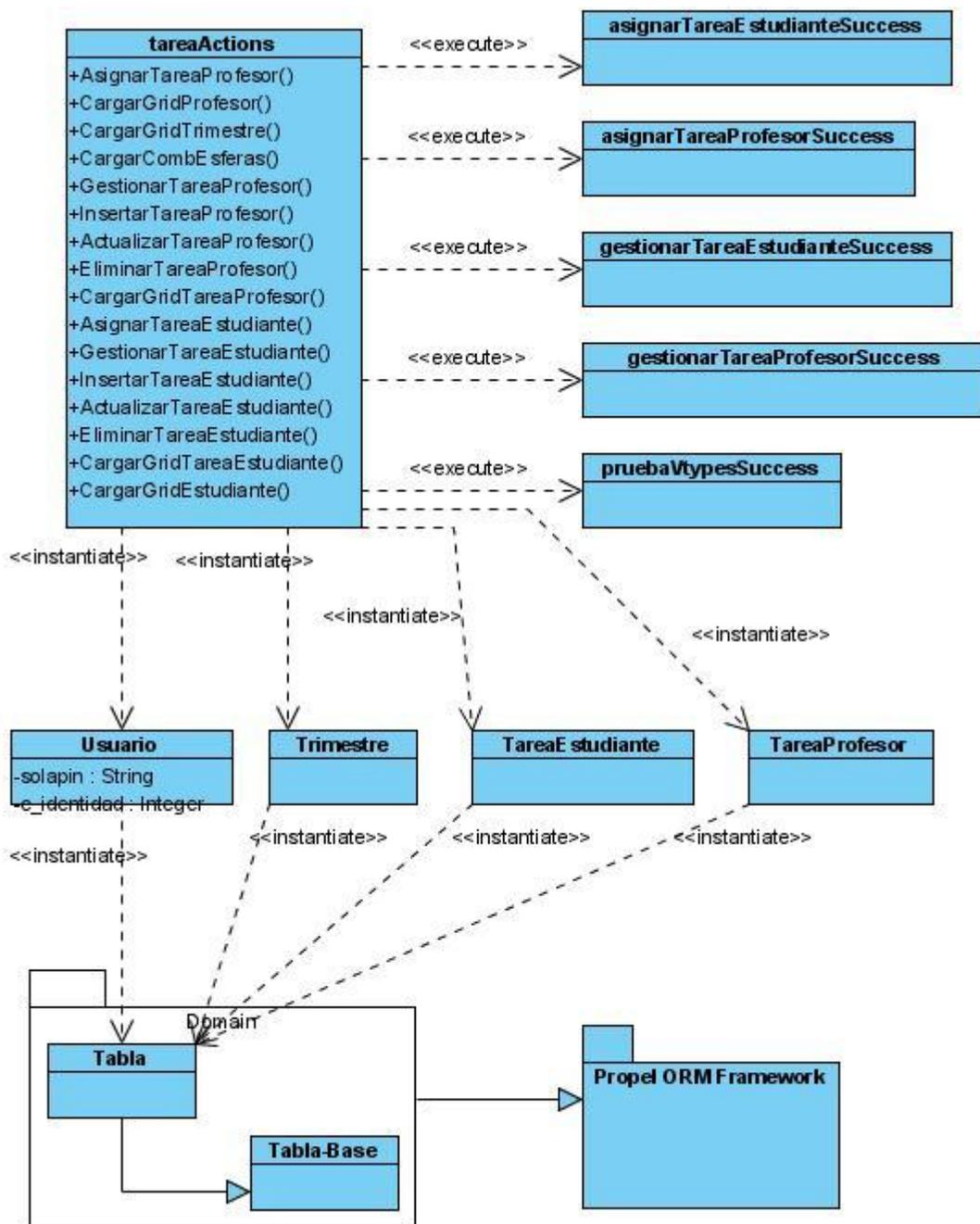


Figura 42: Diagrama de clase del diseño. Gestionar Tarea

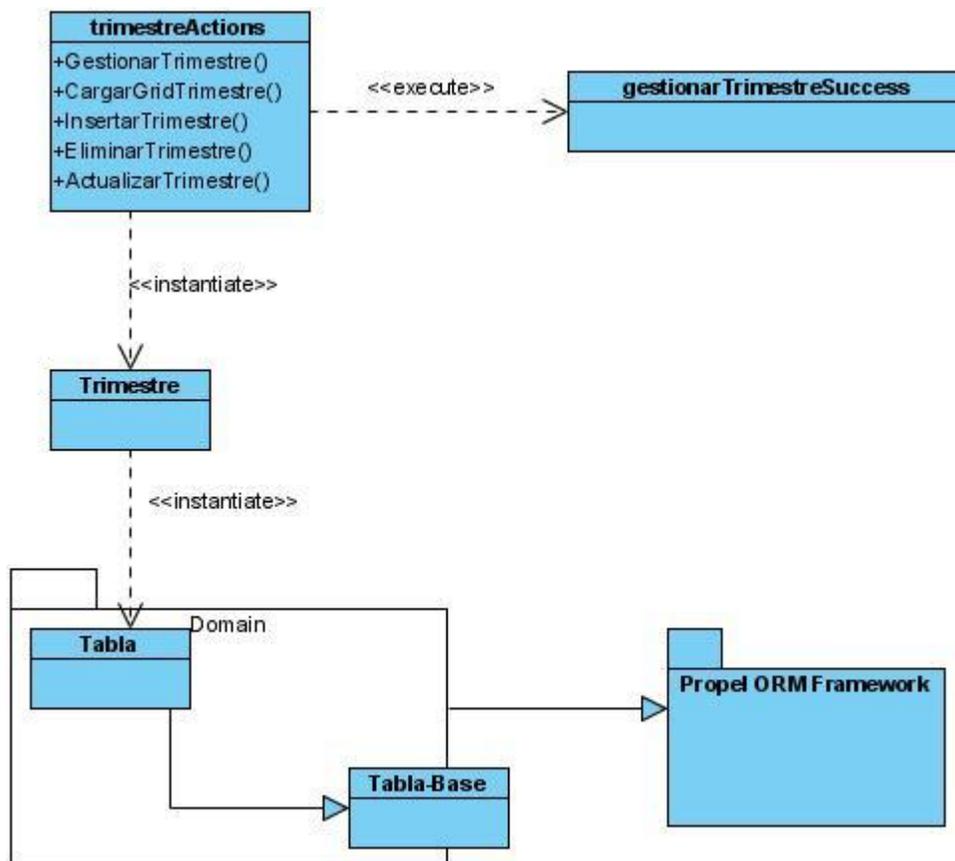


Figura 43: Diagrama de clase del diseño. Gestionar Trimestre

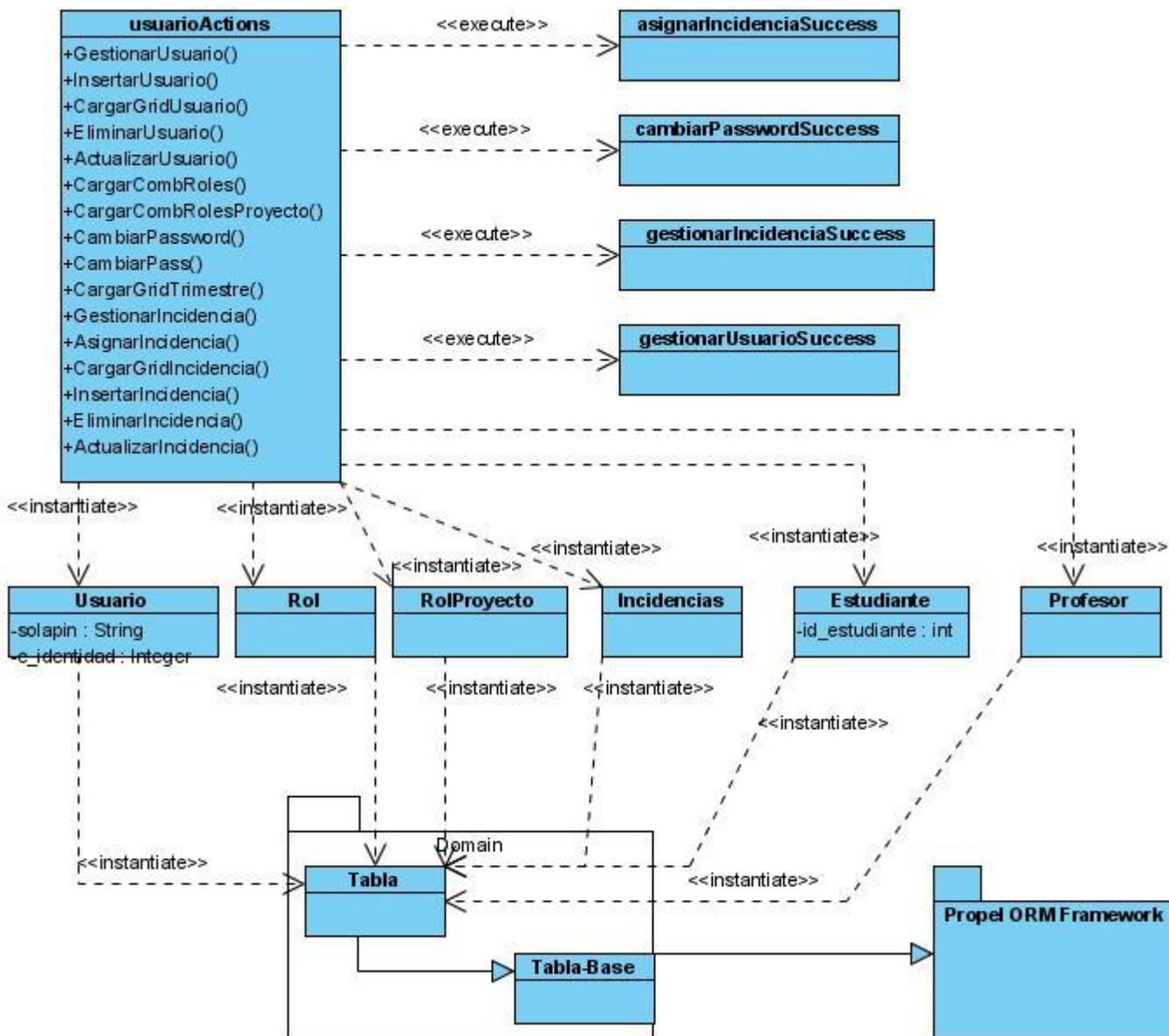


Figura 44: Diagrama de clase del diseño. Gestionar Usuario

Anexo3: Prototipo de interfaz del sistema

El prototipo de interfaz de usuario para 'Nuevo Usuario' presenta un formulario con los siguientes campos:

- Nombre y Apellidos:
- Usuario:
- Password:
- Confirmar Password:
- Solapin:
- Carnet de Identidad:
- Area:
- Tipo de Usuario:
- Rol:
- Rol Proyecto:

En la parte inferior del formulario, se encuentran dos botones: 'Aceptar' (con un icono de checkmark verde) y 'Cancelar' (con un icono de X roja).

Figura 46: Prototipo de interfaz. Adicionar Usuario

Este prototipo de interfaz, titulado "Modificar Usuario", presenta un formulario con los siguientes campos:

- Nombre y Apellidos: Administrador
- Usuario: admin
- Solapin: (campo vacío)
- Carnet de Identidad: (campo vacío)
- Area: (campo vacío)
- Tipo de Usuario: (menú desplegable)
- Rol: Administrador
- Rol Proyecto: Jefe de Proyecto

En la parte inferior del formulario se encuentran dos botones: "Aceptar" (con un icono de checkmark verde) y "Cancelar" (con un icono de prohibido rojo).

Figura 47: Prototipo de interfaz. Modificar Usuario

Este prototipo de interfaz, titulado "Adicionar Rol", presenta un formulario con el siguiente campo:

- Rol: (campo de texto vacío)

En la parte inferior del formulario se encuentran dos botones: "Aceptar" (con un icono de checkmark verde) y "Cancelar" (con un icono de prohibido rojo).

Figura 48: Prototipo de interfaz. Adicionar Rol

Modificar Rol

Modificar

rol:

Detailed description: This is a software interface window titled 'Modificar Rol'. It features a sub-section titled 'Modificar' containing a text input field labeled 'rol:'. Below the input field are two buttons: 'Aceptar' with a green checkmark icon and 'Cancelar' with a red 'X' icon.

Figura 49: Prototipo de interfaz. Modificar Rol

Adicionar Competencias

Adicionar

Competencia:

Rol del Proyecto: Seleccione...

Detailed description: This is a software interface window titled 'Adicionar Competencias'. It features a sub-section titled 'Adicionar' containing two input fields: a text field labeled 'Competencia:' and a dropdown menu labeled 'Rol del Proyecto:' with the text 'Seleccione...' and a downward arrow. Below the inputs are two buttons: 'Aceptar' with a green checkmark icon and 'Cancelar' with a red 'X' icon.

Figura 50: Prototipo de interfaz. Adicionar Competencia

Modificar Competencias

Modificar

Competencia:

Rol del Proyecto: Jefe de Proyecto

Detailed description: This is a software interface window titled 'Modificar Competencias'. It features a sub-section titled 'Modificar' containing two input fields: a text field labeled 'Competencia:' and a dropdown menu labeled 'Rol del Proyecto:' with the text 'Jefe de Proyecto' and a downward arrow. Below the inputs are two buttons: 'Aceptar' with a green checkmark icon and 'Cancelar' with a red 'X' icon.

Figura 51: Prototipo de interfaz. Modificar Competencia

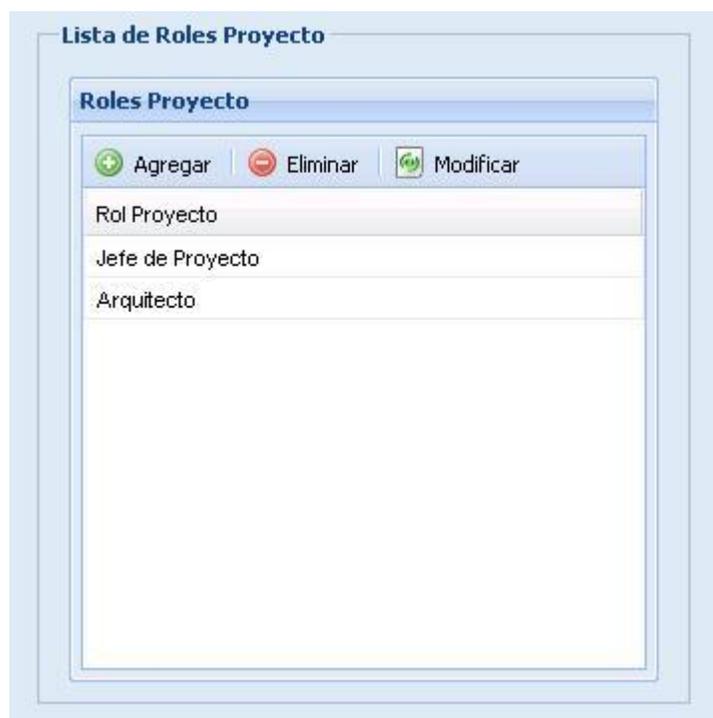


Figura 52: Prototipo de interfaz. Gestionar Rol del Proyecto

GLOSARIO DE TÉRMINOS

CSS: Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

XML: siglas en inglés de Extensible Markup Language («lenguaje de marcas »), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML.

DHTML: El HTML Dinámico o DHTML (del inglés Dynamic HTML) designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM. Una página de HTML Dinámico es cualquier página web en la que los scripts en el lado del cliente cambian el HTML del documento, después de que éste haya cargado completamente, lo cual afecta a la apariencia y las funciones de los objetos de la página. La característica dinámica del DHTML, por tanto, es la forma en que la página interactúa con el usuario cuando la está viendo, siendo la página la misma para todos los usuarios.

AJAX: acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas

operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

MySQL: Es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

PostgreSQL: Es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Oracle: Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

Linux: Es el núcleo o kernel del sistema operativo libre denominado GNU/Linux (coloquial pero erróneamente llamado Linux). Lanzado bajo la licencia pública general (GPL - General Public License) de GNU y desarrollado gracias a contribuciones provenientes de todo el mundo, Linux es uno de los mejores ejemplos de software libre cuyos desarrolladores originales siguieron la filosofía de ese movimiento. Linux fue creado por Linus Torvalds en 1991.

UNIX: (registrado oficialmente como UNIX®) es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.

Windows: Es una familia de sistemas operativos desarrollados y comercializados por Microsoft. Existen versiones para hogares, empresas, servidores y dispositivos móviles, como computadores de bolsillo y teléfonos inteligentes. Hay variantes para procesadores de 16, 32 y 64 bits. Incorpora diversas aplicaciones como Internet Explorer, el Reproductor de Windows Media, Windows Movie Maker, Windows Mail, Windows Messenger, Windows Defender, entre otros. Desde hace muchos años es el sistema operativo más difundido y usado del mundo; de hecho la mayoría de los programas (tanto comerciales como gratuitos y libres) se desarrolla originalmente para este sistema.

Apache: El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.