

Universidad de las Ciencias Informáticas.

Facultad 3.



Análisis y Diseño del módulo Desarrollo del personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3.

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

Autores: Armando Alexander Rosales Leyva
Miguel Angel Márquez Figueroa

Tutora: Ing. Janet Rodríguez Febles

Ciudad de la Habana, 18 de Junio de 2009.

DECLARACIÓN DE AUTORÍA

Nosotros, Armando Alexander Rosales Leyva y Miguel Ángel Márquez Figueroa declaramos que somos los únicos autores de este trabajo y otorgamos a La Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los _18_días del mes de _junio_ del año_2009_.

Armando Alexander Rosales Leyva

Miguel Ángel Márquez Figueroa

Tutor: Ing. Janet Rodríguez Febles

A todo el que contribuyó de forma directa o indirecta con la realización de este trabajo.

A todo el que tuvo disposición de ayudar y estuvo ahí para que se contara con él.

Al que criticó constructivamente.

Al que, aunque sin decirlo, siempre deseó que todo saliera bien.

Ustedes, siéntase mencionados aquí y reciban mis más sinceros agradecimientos que no acabarán con estas banas líneas.

Miguel A.

A mis abuelos y mis padres por confiar siempre en mí y motivarme para lograr los mejores resultados.

A Fidel y la Revolución por esta oportunidad tan hermosa.

A mis tios por su confianza y apoyo incondicional.

A mis compañeros que a lo largo de estos años de alguna forma me brindaron su ayuda y conocimientos.

A mis amigos y mi novia, por ser los mejores y estar cuando mas lo necesité.

A Yanet y su familia que fueron un pilar fundamental en este logro.

A mi tutora por sus regaños y por andar juntos este duro camino.

A los integrantes del tribunal por sus críticas constructivas y su ayuda.

A todos, reciban los mayores agradecimientos del mundo.

Armando A.

*Al que este trabajo pueda
servir de alguna forma.*

Miguel A.

A mis abuelos, que son mi vida.

A mis padres por saber guiarme.

*A mi abuela Noemí que siempre
la llevo junto a mi.*

A mi familia por ser tan especial.

Armando A.

RESUMEN

El desarrollo del personal por competencias es un nuevo enfoque que se integra a la gestión de recursos humanos, cuyo propósito radica en la formar las competencias necesarias para el desempeño exitoso en un puesto. La Universidad de las Ciencias Informáticas (UCI), presenta como reto la integración de los procesos de formación, investigación y producción que busquen una verdadera correspondencia entre lo adquirido en el aula y las necesidades productivas. El nuevo modelo de formación centrado en el aprendizaje, provee un Ciclo Profesional donde los estudiantes certifiquen roles y competencias, evidenciando la necesidad de contar con un sistema de ayuda, que posibilite el registro constante de los resultados obtenidos en el Expediente Productivo.

El presente trabajo provee el análisis y el diseño del módulo Desarrollo del Personal del “Sistema de Gestión de Recursos Humanos por Competencias en los Proyectos Productivos de la facultad 3”, que permita el mantenimiento del historial de certificaciones. Los estudios realizados permitieron seleccionar los lenguajes, tecnologías y herramientas a emplear, modelando los procesos del negocio con IDEF0, obteniendo el análisis y diseño, así como validando los resultados.

ÍNDICE DE CONTENIDO	
ÍNDICE DE CONTENIDO	VI
INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	4
Introducción.....	4
1.1 Gestión de recursos humanos por competencias	4
1.2.1 Definición de competencias.....	4
1.2.2 Antecedente y actualidad de las competencias.....	5
1.2.3 Clasificación de las competencias	6
1.2.4 Perfiles por competencias.....	7
1.2.5 Integración del enfoque de competencias a la Gestión de Recursos Humanos (GRH) 	8
1.3 Certificación de competencias	10
1.3.1 Principales organismos certificadores de competencias	10
1.3.2 Principales características de la certificación de competencias.....	11
1.3.3 Sistemas para la certificación de competencias laborales	12
1.4 Proceso de desarrollo del personal	14
1.4.1 Definición del proceso desarrollo del personal	14
1.4.3 Integración del enfoque por competencias al proceso de desarrollo del personal .	16
1.4.4 Sistemas Informáticos para el desarrollo del personal	17
1.5 Metodologías de Desarrollo de Software.....	19
1.6 Lenguajes de modelado.....	21
1.7 Herramientas para el modelado	24
1.8 Ingeniería de requisitos.	28
1.9 Patrones empleados	31
1.9.1 Patrones de casos de uso	31
1.9.2 Patrones de arquitectura	32
1.9.3 Patrones de diseño	34
1.10 Lenguajes de programación	40
1.11 Servidores web	45
1.12 Framework de desarrollo	47
1.13 Servidores de bases de datos	51
Conclusiones.....	54
CAPITULO II: CARACTERÍSTICAS DEL SISTEMA	55

Introducción.....	55
2.1 Situación actual. Su análisis crítico	55
2.2 Objeto de automatización	56
2.3 Descripción del proceso del negocio	57
2.4 Modelo del sistema	60
2.4.1 Requisitos funcionales	60
2.4.2 Requisitos no funcionales.....	64
2.4.3 Descripción de los actores.....	66
2.4.4 Diagrama de casos de uso del sistema	69
2.4.5 Descripción textual de los casos de uso	70
2.5 Seguridad que ofrece el framework para el sistema:.....	81
Conclusiones.....	84
CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA	85
Introducción.....	85
3.1 Modelo de análisis.....	85
3.1.2 Clases del análisis	86
3.2 Modelo de Diseño.....	89
3.2.1 Clases del diseño.....	90
3.2.2. Arquitectura del sistema	91
3.2.3 Diagramas de clases del diseño	92
3.2.4 Diagramas de iteración: Secuencia para el diseño.	97
3.3 Diagrama de clases persistentes	100
3.4 Modelo de datos	100
Conclusiones.....	101
CAPÍTULO IV: VALIDACIÓN DE LOS RESULTADOS.....	102
Introducción.....	102
4.1 Proceso de validación.....	102
4.2 Métricas para requisitos	103
4.3 Métricas para casos de uso.....	106
4.4 Métrica del índice de madurez de software	109
4.5 Métrica de la profundidad del árbol de herencia.....	109
4.6 Prototipado	110
Conclusiones.....	112
CONCLUSIONES GENERALES	113

RECOMENDACIONES 114
BIBLIOGRAFIA CONSULTADA 115

INTRODUCCIÓN

La Universidad de las Ciencias Informáticas (UCI) como centro de altos estudios, posee gran importancia social por la cantidad de ingenieros que aporta cada año y los compromisos productivos que enfrenta, los cuáles constituyen ingresos significativos en la economía del país. Sin embargo, en el contexto universidad-industria se requiere una integración óptima de los procesos de formación, investigación y producción, en aras de incorporar los conocimientos y habilidades adquiridos en el aula a la práctica del ejercicio. Esta es una de las razones por las que se implementa un experimento pedagógico, que constituye un cambio en el modelo de formación existente hacia otro centrado en el aprendizaje, el cual debe comenzar su aplicación en septiembre del 2009 e incluye un ciclo profesional, que requiere la rotación por varios roles con la finalidad de certificarlos.

En el marco de los proyectos productivos que se realizan en la facultad 3 se llevan los conocimientos a la práctica, logrando la construcción de productos informáticos con elevados índices de calidad. Sin embargo, no existe un portafolio donde se almacenen los resultados obtenidos de forma personalizada y las evaluaciones de desempeño que se efectúan no muestran las potencialidades individuales adquiridas. Además, no se cuenta con un procedimiento definido que apoye las certificaciones de competencias necesarias en el nuevo modelo de formación, las cuales brindan una guía fructífera para el desarrollo de personal, y su reflejo en el expediente productivo definido por el departamento docente central de práctica profesional. Un elemento importante lo constituye la creación de los perfiles por competencia de los roles, algunos de ellos han sido objeto de análisis en varias tesis de grado durante el curso 2008-2009, que se pueden utilizar como guía de las competencias genéricas a desarrollar. Con la situación problemática actual identificada, se desea construir una herramienta que estandarice y viabilice el proceso de desarrollo del personal por competencias en los proyectos productivos de esta facultad 3.

El problema a resolver es: ¿Cómo obtener los artefactos correctos que permitan a los desarrolladores implementar el módulo Desarrollo del Personal del “Sistema de Gestión de Recursos Humanos por Competencias en los Proyectos Productivos de la Facultad 3”?

El objeto de estudio es la gestión de recursos humanos en el desarrollo de software y el campo de acción es el análisis y diseño del módulo Desarrollo del Personal en los proyectos productivos de la Facultad tres.

El objetivo general consiste en: Realizar el análisis y el diseño del módulo Desarrollo del Personal del "Sistema de Gestión de Recursos Humanos por Competencias en los Proyectos Productivos de la Facultad 3", que permita a los desarrolladores implementarlo correctamente.

Los objetivos específicos son:

- ✓ Realizar el marco teórico de la investigación.
- ✓ Elaborar los artefactos correspondientes al modelo de procesos de negocio, sistema, análisis, diseño y datos del módulo desarrollo del personal.
- ✓ Validar el modelo de procesos de negocio, sistema, análisis, diseño y datos del módulo desarrollo del personal.

La hipótesis a demostrar es: Si se realiza correctamente los artefactos de entrada del módulo Desarrollo del Personal del "Sistema de Gestión de Recursos Humanos por Competencias en los Proyectos Productivos de la Facultad 3", se podrán utilizar como guía para implementarlo.

Los métodos utilizados en la investigación fueron los siguientes:

Métodos Teóricos:

- ✓ Histórico-Lógico: para el análisis de la trayectoria completa del fenómeno, su condicionamiento a los diferentes períodos de la historia.

Método Empírico:

- ✓ Entrevistas: para recopilar información con los involucrados con el objetivo de diseñar el proceso de Desarrollo del Personal por Competencias en los proyectos productivos de la facultad 3.

Aportes esperados:

La modelación del análisis y diseño, permitirá la implementación de uno de los módulos del "Sistema para la Gestión de Recursos Humanos por Competencias de la Facultad 3", que servirá de apoyo al proceso de desarrollo por competencias propuesto. El sistema automatizará el seguimiento de las certificaciones de competencias y roles reflejándolo en el expediente productivo, elemento de gran valor práctico en las proyecciones estratégicas de la UCI.

El presente trabajo se estructura de la siguiente forma:

Capítulo I: Fundamentación teórica.

En este capítulo se realiza un estudio de los antecedentes y tendencias modernas sobre el desarrollo del personal y la certificación de competencias, profundizando en el análisis de algunos productos de software existentes en el mercado y que dan solución a problemas similares. Además, se valora la situación de las tecnologías a utilizar en el desarrollo de la aplicación, comparándolas y seleccionando las mejores propuestas.

Capítulo II: Características del sistema

En este capítulo se presenta el proceso de negocio propuesto vinculado al Desarrollo del Personal por competencias en los proyectos productivos de la facultad 3, se especifican los requisitos funcionales, no funcionales, así como los casos de uso y sus respectivas descripciones textuales.

Capítulo III: Análisis y Diseño

En este capítulo se presenta el análisis y el diseño del sistema propuesto, generando los artefactos correspondientes a cada flujo, como son los diagramas de clases de análisis y clases de diseño, diagramas de interacción del diseño y la propuesta de la base de datos.

Capítulo IV: Validación de los Resultados

En él se muestran los resultados de las validaciones obtenidos a través de las diferentes métricas para los requisitos funcionales, los casos de uso y el prototipo no funcional del sistema propuesto.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se ofrece una panorámica general relacionada a la gestión de los recursos humanos por competencias, centrándose en el proceso vinculado al desarrollo del personal. Se expone un análisis de sistemas informáticos existentes orientados a la formación, evidenciando en algunos el tratamiento de la certificación. Se analizan las metodologías de desarrollo de software, lenguajes y herramientas de modelado, la ingeniería de requisitos, patrones, lenguajes de programación, servidores Web, framework de desarrollo y servidores de base de datos.

1.1 Gestión de recursos humanos por competencias

Para lograr un desempeño laboral competente, las personas recurren a diferentes combinaciones de tipos de competencias, aplicadas a problemas y situaciones concretas de su trabajo. Las competencias son valoradas como una herramienta capaz de proveer una conceptualización, un modo de hacer y un lenguaje común para el desarrollo de los recursos humanos, constituyen una visión y organización sistemática, se han expresado como un sistema de certificación legalmente establecido en varias regiones del mundo, incluida América Latina y se trata de un vínculo entre trabajo, educación y capacitación. (Chiavenato, 2002).

1.1.1 Definición de competencias

En el mundo existen diferentes definiciones y criterios sobre las competencias laborales y aunque no se ha logrado un consenso, sí se tiene una idea específica de que son y cómo interactúan en el ámbito laboral, productivo e investigativo, estas surgen de la necesidad de conocer las habilidades que tiene una persona para comprobar su capacidad al aplicarla en un entorno determinado para lograr un objetivo específico.

Las competencias se definen como:

“Un saber hacer frente a una tarea específica, la cual se hace evidente cuando el sujeto entra en contacto con ella. Esta competencia supone conocimiento, saberes y habilidades que emergen en la iteración que se establece entre el individuo y la tarea y que no siempre están de antemano.” (Bogota, 1999).

“Son habilidades, destrezas, aptitudes, actitudes, rasgos de personalidad, son estilos de dirección, formas de relaciones interpersonales, de experiencias que se adquieren durante toda la vida, tanto personal como laboral.” (Cuestas, 1999).

“Capacidad productiva de un individuo, que se define y se mide en términos de desempeño en un determinado contexto laboral, y no solamente como conocimientos, habilidades, destrezas y actitudes; éstas son necesarias pero no suficientes por sí mismas para un desempeño efectivo.” (Competencias, 1997).

Entonces se pueden definir por competencias laborales, las habilidades, capacidades, actitudes, condiciones que debe poseer una persona para desempeñarse en un puesto laboral y ejecutar eficientemente una tarea determinada, proporcionando beneficios a su institución.

1.1.2 Antecedente y actualidad de las competencias

Las competencias laborales se han convertido en una herramienta eficiente para lograr una mejor armonía entre la productividad y los conocimientos, donde se puede obtener un lenguaje común de que es lo que se debe hacer y cómo, así como para administrar el talento humano. En varios países se han establecido la certificación por competencias como un proceso legal y que permite a las organizaciones establecer un desarrollo constante de su personal.

El concepto que mueve actualmente a las organizaciones en la comunidad internacional, es el de lograr que su personal esté altamente capacitado y para esto se requiere dejar atrás mecanismos y metodologías ya caducas para lograr que las nuevas alternativas fomenten el aprendizaje y la superación, logrando una armonía entre los conocimientos que tiene el personal y la labor que desempeña. Esto sin dudas va a repercutir de forma positiva en conseguir una mayor productividad y eficiencia económica.

Hacia 1970 se comienza a ver con mayor auge el tema de las competencias laborales. Las grandes empresas y organizaciones hacían una selección del personal mediante pruebas de inteligencia y exámenes donde se medía los conocimientos que tenía para desempeñar una tarea determinada, aquellas personas que obtuvieran mayor puntuación eran consideradas las más apropiadas para conseguir el puesto y por lo tanto empleadas.

En la década de los 80 Inglaterra crea el Consejo Nacional para las Calificaciones Profesionales, cuyo objetivo es definir y certificar el sistema de calificaciones profesionales de ese país. A partir de entonces en varios países de Europa, Estados Unidos, Canadá y Australia se comienzan a crear sistemas muy parecidos. En México se crea en la década de los 90 un sistema normalizado de competencia laboral, en otros países de América Latina se están instrumentando y refinando sistemas similares.

En la actualidad, con el nuevo orden económico mundial y sus transformaciones en el campo laboral, cuyo dinamismo configura una economía internacional más competitiva y una virtual revolución en la organización, tipo y contenido del trabajo, la flexibilidad tecnológica y organizacional aportan a las empresas un mayor potencial de adaptación y respuesta oportuna a los cambios de los mercados, por lo que se hace inminente trabajar en base a elevar las competencias del capital humano que conforman dichas empresas, para optimizar los resultados y que la calidad de los productos sea la mejor.

En la comunidad internacional, se vienen desarrollando diversas teorías sobre cómo mejorar las competencias laborales internamente en el ámbito empresarial y son ya muchos los países que han optado por hacer un cambio radical en su sistema interno para lograr una mejor productividad, competitividad y calidad de sus productos, destacando de manera particular la función activa e integradora que aporta.

El reconocimiento de las competencias desarrolladas durante la vida laboral puede influir positivamente en la obtención de nuevas competencias por el capital humano y que los mecanismos y teorías sobre el desarrollo de éste puedan mejorarse para lograr una mejor formación del mismo.

1.1.3 Clasificación de las competencias

En las diferentes bibliografías que se han consultado se clasifican las competencias laborales de diferentes maneras pero se utiliza la que se maneja en la universidad y que han sido definidas por el Dr. Armando Cuestas. Las competencias laborales entonces se clasifican en: (Cuestas, 1999).

- ✓ Genéricas: Estas pueden responder a cualquier puesto de trabajo o a la mayoría según su misión, objetivos y metas.

- ✓ Específicas: Estas se adecuan a puestos de trabajo, con objetivos de trabajo específicos y pueden cambiar de acuerdo a los escenarios y las estrategias que se traza la organización.
- ✓ Compuestas: Son aquellas en la que intervienen varias competencias, son más difíciles de dimensionar y dificultan el análisis de desempeño.
- ✓ Simples: Son las que se refieren a una sola competencia para evaluar, siendo fáciles de dimensionar y por lo tanto facilitan su evaluación.
- ✓ Importantes: Todas no tienen la misma importancia porque depende de las características del puesto de trabajo y de la misión, objetivos y estrategias del puesto y la organización.
- ✓ Esenciales: Dentro de las anteriores hay otras que le llaman esenciales, que debe poseer una persona independientemente del puesto que desempeña.

Luego de realizar el estudio de las clasificaciones de competencias y teniendo en cuenta las características de la UCI, se toman como base las competencias genéricas, ya que abarcan los comportamientos asociados con desempeños comunes a diversas ocupaciones y ramas de actividad productiva. Esto permitirá la normalización de los perfiles por competencias de los roles, necesarios para la evaluación de competencias y roles.

1.1.4 Perfiles por competencias

Los actuales perfiles por competencias se centran más bien en las responsabilidades y actuaciones que deberá tener el individuo para desempeñar exitosamente su trabajo, bajo una situación contextualizada. (Cuestas, 1999).

El término de perfil por competencia, le imprime una alta flexibilidad al diseño de roles y constituye un patrón de selección, formación y evaluación muy válida para la gestión del talento humano. (Cuestas, 1999).

El perfil por competencias es un documento que contiene una descripción de las competencias, así como un control de las habilidades que posee una persona para que puedan ser gestionadas con el fin de lograr una mayor eficiencia. Se elaboran para los cargos o profesiones específicas de la entidad, en función de las particularidades de los procesos y tecnología existente en la misma.

El desarrollo de personal tiene que centrarse sobre la base de problemas o necesidades reales, su punto de partida son las particularidades de la actividad, reflejadas a partir del análisis y descripción

de los roles a desempeñar, lo que permite definir las principales características, exigencias y requerimientos de los mismos; posibilita evaluar las cualidades necesarias en los trabajadores de forma integrada en la valoración de las competencias como elemento esencial que caracteriza la relación hombre-trabajo, y también determinar los métodos, técnicas e instrumentos a emplear en el proceso de desarrollo de personal.

Una de las mejores formas de averiguar qué se necesita para lograr un desempeño exitoso en un puesto de trabajo determinado, es estudiando a los individuos que tengan un desempeño exitoso en el mismo y analizar qué hacen para lograrlo. Esto quiere decir: realizar una evaluación no del puesto de trabajo, sino de la persona que realiza el trabajo. La gestión por competencias busca a partir de la definición de un perfil por competencias y de posiciones dentro del perfil, que los momentos de verdad entre una organización y su personal sean conscientes y apunten en definitiva, a aumentar la contribución de cada empleado a la generación de valor de dicha organización.

La tendencia actual es hacia puestos de trabajo polivalentes o de múltiples habilidades, habrá que garantizar que el profesiograma se mantenga en consecuencia con esa tendencia, de manera que no signifiquen un encasillamiento, sino un rango flexible con un perfil amplio, propiciando el enriquecimiento del desempeño, tanto en su vertiente horizontal como en la vertical. Contar con las personas que posean las características adecuadas, se ha convertido en la directriz de la gestión de recursos humanos.

En la realización del presente trabajo se hace de vital importancia establecer una plantilla que contemple las competencias genéricas necesarias para ocupar un rol en la producción, la misma no entra en contradicción con las especificidades que contemplan los perfiles por competencias de los cargos. Esta podrá retomarse posteriormente, como un elemento fundamental contemplado en el diseño del perfil por competencias que se establezca en la facultad 3 y en la UCI. (Ver Anexo1)

1.1.5 Integración del enfoque de competencias a la Gestión de Recursos Humanos (GRH)

La visión contemporánea que existe sobre La Gestión de Recursos Humanos es que el hombre está considerado un recurso, el cual hay que desarrollar y en el que es necesario invertir para lograr un resultado deseable, sus resultados tienden a ser difícilmente medibles, intangibles y sus ciclos de retroalimentación largos.

En la actualidad se han venido insertando en el marco empresarial una serie de conceptos que abarcan la determinación por competencias, normalización de competencias, formación por competencias, certificación por competencias y perfiles por competencias entre otros. El Dr. Carlos Álvarez de Zayas en el 2002 comienza a hablar sobre la gestión por competencias como el factor que abarca todas las áreas del proceso Gestión de Recursos Humanos, el estudio y aplicación de éstas se enfoca desde diferentes ángulos desde a nivel de la organización como un sistema o desde el punto de vista particular de cada individuo.

“La gestión por competencias permite evaluar las competencias específicas para cada puesto y favorecer el desarrollo personal de los empleados. Se entiende como Gestión por Competencias la capacidad de una empresa de atraer, desarrollar y mantener el talento mediante la alineación consistente de los sistemas y procesos de Recursos Humanos, sobre la base de las capacidades y resultados requeridos para un desempeño competente.” (Clara, 2001).

Se entiende entonces que la gestión por competencias es la herramienta estratégica indispensable para enfrentar los nuevos desafíos que impone el medio, es impulsar a nivel de excelencia las competencias individuales, de acuerdo a las necesidades operativas. Garantiza el desarrollo y administración del potencial de las personas de lo que saben hacer o podrían hacer. El aprendizaje organizacional debe ser permanente y continuo.

Si se logra realizar una eficiente Gestión de Recursos Humano basados en las competencias, los conocimientos de las capacidades de los trabajadores serían mayores, generando el desarrollo individual y organizacional dentro de la empresa, enfocándose en las necesidades de capacitación y apoyando a los sistemas de desarrollo del personal.

Se ha planteado en muchas organizaciones la necesidad de crear estrategias para la evaluación del desempeño así como para el desarrollo del personal, cabría tener en cuenta las ventajas que brinda la gestión por competencias: (Cabezas, 2001).

- ✓ La evaluación del desempeño.
- ✓ El desarrollo de equipos que posean las competencias necesarias para su área específica de trabajo.

- ✓ Identificación de los puntos débiles, permitiendo intervenciones de mejora que garantizan los resultados.
- ✓ La eliminación de la costosa e improductiva práctica del adiestramiento tradicional. Cuando se sitúa la gestión por competencias, se evita que los directivos y sus colaboradores pierdan el tiempo en programas de entrenamiento y desarrollo que no tienen que ver con las necesidades de la empresa o las necesidades de los puestos de trabajo.
- ✓ La posibilidad de definir perfiles profesionales que favorecerán a la productividad.
- ✓ El aumento de la productividad y la optimización de los resultados.
- ✓ La sustitución urgente de las descripciones del cargo como eje de la gestión de los recursos humanos.
- ✓ La compensación justa con base en el aporte al valor agregado.

Como se puede apreciar al integrar el enfoque de competencias a La Gestión de los Recursos Humanos se obtienen solo beneficios que ayudarán a desarrollar a las personas en un marco productivo más eficiente y permitirá a las empresas a consolidar sus resultados.

1.2 Certificación de competencias

La certificación de competencias se conoce como un reconocimiento formal de las competencias demostradas por un individuo, a partir de un proceso de evaluación realizado sobre la base de normas de competencias, establecidas por una empresa para un rol laboral determinado, o consensuadas por los actores de un sector de actividad. La certificación se otorga como un reconocimiento a la culminación de un proceso de formación, basada fundamentalmente en el tiempo de capacitación, la práctica laboral y del dominio de los contenidos examinados. El certificado de una competencia laboral permite hacer referencia al desempeño donde un trabajador demuestra ser competente por medio de la evaluación de competencias.

1.2.1 Principales organismos certificadores de competencias

En la actualidad existen varios organismos certificadores, son entidades destinadas para realizar la certificación de las competencias de conformidad con una norma técnico de competencia laboral, garantizando que el individuo es competente en una o varias funciones productivas.

Con el avance de las nuevas tecnologías no bastan los métodos tradicionales de enseñanza para que una persona pueda desempeñar eficazmente una tarea dentro de una organización debido a la

informatización de muchos puestos laborales, por lo que a diario son mayores los países que crean las organizaciones necesarias para controlar y acreditar competencias laborales. Algunas de esas organizaciones más importantes son:

En Brasil, SENAI, adelanta un proyecto estratégico nacional que realiza aplicaciones piloto en varios sectores ocupacionales para certificar competencias adquiridas en la experiencia.

En Chile, SENCE, trabajando con el sector privado, maneja un componente que articula las acciones de formación con la certificación de competencias forjadas en el trabajo para nueve sectores ocupacionales inicialmente.

En Argentina, desde el Ministerio del Trabajo, desarrolla un programa en cuatro sectores de la economía que diseña y prueba mecanismos para la certificación y la formación de trabajadores. El reconocimiento de las competencias desarrolladas durante la vida laboral se puede convertir en un efectivo motor que impulse la adquisición de nuevas competencias; así como la generación de nuevos mecanismos y programas de formación. La certificación de competencias para el nuevo milenio ha adquirido un valor relacionado profundamente con la oferta de empleos para los trabajadores, en la medida en que los certificados se refieren a competencias de base amplia, que facilitan su transferibilidad entre diferentes contextos ocupacionales.

1.2.2 Principales características de la certificación de competencias

Las competencias laborales presentan una gran variedad de características las cuales se describen a continuación: (Cuestas, 1999).

- ✓ Se basan en un aprendizaje permanente e incremental, ya que posibilita que el personal pueda completar otras competencias para mejorar su desempeño en una tarea determinada.
- ✓ La formación de competencias debe ofrecerle al personal la posibilidad de aumentar sus conocimientos, para que lo estimule a superarse en aras de acreditarse más habilidades.
- ✓ El proceso de certificación por competencias, debe realizarlo un organismo o institución creada con tales fines, de manera que los certificados sean creíbles y realmente sean otorgados a candidatos que tienen las habilidades para desempeñarse eficazmente en un puesto laboral.
- ✓ Los certificados que se acreditan a las competencias deben tener un valor permanente, tanto para instituciones, como para el sistema laboral en general, de forma que le permita a su

acrededor poder desempeñar cualquier actividad que requiera de las competencias vencidas en cualquier organismo. El certificado tendría la validez de un título universitario.

- ✓ Al certificarse una competencia laboral el certificado debe brindar la información tanto a la persona como a quien va a emplear del alcance de la competencia.
- ✓ Debe resultar un proceso legal y fácil de conseguir sin necesidad de desperdiciar tiempo y recursos en trámites innecesarios y brindar el contenido de la ocupación que se está certificando.

1.2.3 Sistemas para la certificación de competencias

Los sistemas que han sido creados para la certificación de competencias, tienen como objetivo fortalecer y mejorar las ofertas de empleo de las personas y por ende la situación laboral. Mediante diferentes normas y reglas obtenidas a través de encuestas y planificación con actores de todo el mundo del trabajo, se ayuda a reconocer las calificaciones del personal con diferentes procesos de comprobación y certificación.

Dentro de los sistemas más importantes para la certificación por competencias en el mundo se encuentran: (Espinoza, 2005).

BrainBench (<http://www.brainbench.com>)

Es un sistema de fácil uso para la certificación que ofrece un gran número de cursos gratis y un examen para certificar las competencias, ha sido creado siguiendo el certificado ISO 9001. Los exámenes son evaluados por profesionales certificados anteriormente.

Sigue una teoría sobre que los activos más importantes son las personas sobre su equipo y que éstas, si están capacitadas para ejercer las tareas que les asignan en su organización, pueden hacer la diferencia entre eficiencia y la mediocridad.

Los Test de BrainBench consisten en 40 preguntas que toman entre 45 minutos y 1 hora para completar. Cada pregunta dispone generalmente de 180 segundos mostrando una advertencia cuando restan 30 segundos. Los resultados incluyen lo siguiente:

- ✓ Una puntuación entre 1.0 y 5.0 puntos donde 5 es el máximo, acompañado del texto con la descripción del significado de la puntuación.

- ✓ Una puntuación porcentual mostrando la comparación con todos los que se hayan evaluado antes.
- ✓ Un resumen de tus fortalezas y debilidades comparadas con tu puntuación en conjunto.
- ✓ Aproximadamente el 40% de los que completan un Test de certificación son certificados (alcanzan una puntuación de 2.75). De estos solo el 12% alcanzan el nivel máster (4.0 puntos).

NCSAcademy (<http://www.ncsacademy.com>)

El examen consta de 35 a 50 preguntas en dependencia del tema a evaluar, tomándose hasta cerca de una hora para contestarlas todas. Todas las preguntas son de selección múltiple. La puntuación es en base a 4.0 puntos. Una puntuación superior a los 2.0 puntos indica que se ha superado el examen y es elegible para la certificación. Una puntuación inferior no te permite ser certificado. Se deben esperar 7 días para repetir un examen.

Se tienen 90 segundos para contestar cada pregunta. El sistema proporciona un contador que muestra el tiempo restante para responder la pregunta. El tiempo también es controlado por el servidor y cuando termina no hay posibilidades de cambiar la respuesta dada. Ofrece una evaluación por temáticas permitiendo conocer donde se encuentran los problemas.

Exams Online (<http://www.examsonline.com>)

Los exámenes gratis constan de 15 preguntas y los pagados de 105. Brinda 3 opciones para la realización del examen:

- ✓ Simulación de examen aleatorio: Las preguntas se muestran en un orden aleatorio con un límite de tiempo de 20 minutos. Permite simular un examen real, las respuestas no son mostradas durante la realización del examen.
- ✓ Examen completo: Todas las preguntas son mostradas, no hay límite de tiempo. Puedes ver las respuestas durante la realización del examen.
- ✓ Examen personalizado: No hay límite de tiempo. Se puede modificar la pregunta inicial y la final. Este modo se utiliza para ver las preguntas en un rango determinado. Las preguntas no son aleatorias. Se podrán ver las mismas preguntas cada vez que se seleccione el mismo rango de preguntas. Puedes ver las respuestas durante la realización del examen.

Todas las opciones permiten una revisión al finalizar el examen y se permite bajar exámenes.

CERTIMAX

El Centro de Validación y Certificación de Competencias Profesionales UCI, propone ofrecer cursos y contenidos a los estudiantes para evaluarlos, primeramente a través de documento que reflejen las principales irregularidades que se han detectado y crear los mecanismos que permitan vencer dichos contenidos.

El sistema permite tanto a los directivos de la universidad como a jefes de proyectos realizar un análisis exhaustivo de los principales problemas encontrados, así como mejorar o proponer nuevos cursos de capacitación para realizar una selección del personal para integrar los proyectos, así como una visualización de los expedientes productivos de los estudiantes.

1.3 Proceso de desarrollo del personal

1.3.1 Definición del proceso desarrollo del personal

¿Por qué el estudio de los procesos? Porque las organizaciones son tan eficaces y eficientes como lo son los procesos que producen los bienes y servicios. La mayoría de las organizaciones que han tomado conciencia de esto han reaccionado ante la ineficiencia y se plantean como mejorar los procesos y evitar algunos males habituales como: el bajo rendimiento, las barreras departamentales, subprocesos inútiles debido a la falta de visión global del proceso, entre otras.

“Los procesos de desarrollo de personas se relacionan estrechamente con la educación, educar significa extraer, traer, arrancar. En otras palabras, representa la necesidad de traer del interior del ser humano las potencialidades interiores”. (Chiavenato, 2002).

El desarrollo de las personas “es el conjunto de experiencias organizadas de aprendizaje (intencionales y con propósito), proporcionadas por la organización dentro de un periodo específico, para ofrecer la oportunidad de mejorar el desempeño del personal.” (Chiavenato, 2002).

Por tanto no es más que mejorar las competencias existentes y formar nuevas con el fin de lograr una mayor eficiencia y rendimiento en la actividad que se realiza, además de crear un ambiente agradable en el marco de la organización que permita la superación constante.

El proceso de Desarrollo del Personal por competencias está basada en un modelo de 4 sub-procesos los cuales se describen en el Project Managment Institute como: (PMBOOK, 2004).

Planificación de los Recursos Humanos: identificar y documentar los roles del proyecto, las responsabilidades y las relaciones de informe, así como crear el plan de gestión de personal.

Adquirir el Equipo del Proyecto: obtener los recursos humanos necesarios para concluir el proyecto.

Desarrollar el Equipo del Proyecto: mejorar las competencias y la interacción de los miembros del equipo para lograr un mejor rendimiento del proyecto.

Gestionar el Equipo del Proyecto: hacer un seguimiento del rendimiento de los miembros del equipo, proporcionar retroalimentación, resolver polémicas y coordinar cambios a fin de mejorar el rendimiento del proyecto.

Estos procesos interactúan entre sí y también con los procesos de las demás áreas de conocimientos. Cada proceso puede implicar el esfuerzo de una o más personas o grupos de personas, dependiendo de las necesidades del proyecto. Cada proceso tiene lugar por lo menos una vez en cada proyecto y se realiza en una o más fases del proyecto, si el proyecto se encuentra dividido en fases. (PMBOOK, 2004).

Cuando se desarrollan personas no es solo darles más información para que puedan superarse en un rol determinado sino formarlos de forma tal que puedan adquirir nuevas actitudes que modifiquen sus hábitos y comportamientos, logrando ser más eficientes en la tarea que desarrollan. El PMBOOK brinda una teoría del desarrollo de las personas pero centrado específicamente en la vida de un proyecto. Se basa en cuatro procesos fundamentales, para la realización de este trabajo se profundiza en el proceso de desarrollo del equipo de proyecto el cual es definido como:

“Mejorar las competencias y la interacción de los miembros del equipo para lograr un mejor rendimiento del proyecto.” (PMBOOK, 2004).

Los procesos de negocio describen un conjunto de conceptos que se detallan a continuación:

Procesos o actividades: Se representa por una caja en la cual se encierran todas las actividades que forman parte del proceso.

Entradas: Representa el material o la información que es consumida o transformada por el proceso con el objetivo de producir las salidas. Es posible que algunos procesos no tengan entrada.

Salidas: material o información producida por el proceso. Cada proceso, para ser considerado como tal, debe tener al menos una salida.

Controles: reglamentan, limitan o establecen la forma en que los procesos desarrollan sus actividades para producir las salidas a partir de las entradas. Cada proceso debe tener por lo menos un control. Los más comunes son leyes, decretos, normativas, directrices, procedimientos.

Mecanismos: aquellos recursos que el proceso necesita y que generalmente no son consumidos durante el mismo. Ejemplo de mecanismo: personal cuantitativa y cualitativamente adecuado, máquinas, equipamiento de informática, copiadoras, etc.

Para realizar el presente trabajo se propone un proceso de Desarrollo de Personal con nuevas actividades pero siguiendo el enfoque de Chiavenato y del PMBOOK aplicado a los proyectos productivos de la facultad 3.

1.3.2 Integración del enfoque por competencias al proceso de desarrollo del personal

Se torna cada vez más evidente como diferentes tipos de organizaciones están adoptando un modelo de competencia con el fin de integrar más efectivamente sus estrategias competitivas y de desarrollo con las tendencias globales y sus requerimientos de capital humano.

Los modelos de competencias están ayudando a las organizaciones a tomar un enfoque más integrado y coordinado cuando se diseñan mejoras en los sistemas de gestión de recursos humanos. De hecho estos enfoques incluyen, entre algunas de sus aplicaciones, la articulación entre el diseño de reformas a nivel organizacional, el re-diseño de funciones, el reclutamiento y la selección del personal a todos niveles, la organización del aprendizaje y actualización gerencial y técnica, la gestión del desarrollo de carreras, la gestión de los procesos de evaluación de desempeño y la promoción del personal y los sistemas de compensación.

La necesidad de que se perfeccione el desarrollo de las persona a diferentes niveles y que sean capaces de superarse constantemente con el objetivo de lograr mayor cantidad de habilidades para desempeñarse en su puesto de trabajo, propicia que se utilice la gestión por competencias como una de las herramientas más poderosas en este sentido, así se forman personas que ganan en competitividad y éxito empresarial.

Las organizaciones competentes y competitivas requieren de trabajadores competentes, capaces. La capacidad de respuesta alta, la aceptación, manejo y producción de los cambios, no significa sólo un cambio de comportamiento del individuo. La alteración de sus prácticas habituales y la asimilación y modificación de los nuevos hábitos y procedimientos para adaptarse, requiere de la interiorización y aceptación por parte del individuo de las nuevas situaciones, requiere de aprendizaje continuo. Estos son imperativos tecnológicos y organizativos que imponen nuevas demandas al trabajador.

1.3.3 Sistemas Informáticos para el desarrollo del personal

Con el desarrollo de las nuevas tecnologías y las comunicaciones a nivel mundial y con los nuevos enfoques que se le están dando al desarrollo de las personas, como medio para garantizar productos de mejor calidad y mayor eficiencia en el mercado, las empresas han desarrollado numerosos sistemas informáticos para lograr tales fines. Ya son muchos los países que tienen este tipo de sistemas que brindan capacitación en diferentes esferas del marco productivo, principalmente en los sistemas y herramientas de la informática.

Se expondrán a continuación algunos ejemplos de los sistemas más importantes en el mundo y que han obtenido los mejores resultados:

Esimagen (www.esimagen.com):

Es un sistema del Perú dedicado a desarrollar productos y servicios de capacitación en el manejo de las herramientas de software, cuenta con instructores especializados, de amplia experiencia, capacitando al personal de las empresas más importantes del país, los cuales además de dominar las materias que tienen a su cargo, cuentan con amplia experiencia aplicando estos conocimientos a tareas reales de trabajo.

Tienen publicados online gran números de cursos que van desde productividad, donde se capacitan las principales herramientas para el desarrollo informático, el diseño, el desarrollo con diferentes lenguajes de programación, hasta negocios web y empresas.

Presenta varios tipos de productos de capacitación entre los que se destacan:

Talleres: Dirigidos a nivel intermedio y avanzado. Son de aplicación concreta en temas puntuales y de efecto inmediato en sus actividades de trabajo. No está dirigido a usuarios nuevos en el manejo de las herramientas de software.

Cursos: Generalmente van del nivel básico al intermedio. Están dirigidos a aquellas personas que no entienden los procesos generales de uso de los productos de software.

Las modalidades de capacitación son:

- ✓ Convencional: De acuerdo a programación regular, la cual se distribuye a través de la web.
- ✓ Corporativo: Ajustan el contenido a los requerimientos específicos de trabajo de cada persona e implementan su ejecución en el horario y días que mejor se acomode a las actividades de trabajo de dichas personas.
- ✓ Personalizado: Dirigido a ejecutivos con escaso tiempo, donde el horario se adecua a su disponibilidad de tiempo, en sus instalaciones de oficina o domicilio. Manteniendo el temario, su duración es menor al convencional.

Cognos (www.cognos.com):

Este es un sistema que ha alineado sus soluciones de aprendizaje para apoyar a los individuos a alcanzar sus metas profesionales y las certificaciones más beneficiosas, a elegir los niveles apropiados de habilidades que necesitan desarrollar, y encontrar las oportunidades profesionales disponibles para ellos con las disciplinas elegidas.

Cognos alcanzó la insignia dorada, máxima credencial que lo convierte en la primera institución educativa boliviana, que cumple con todos los requisitos que Microsoft exige para proveer a este nivel de competencia recursos y herramientas flexibles para mantener un margen competitivo en un mercado que cambia continuamente.

Ofrecen 500 o más cursos detallados en un catálogo con la oferta más confiable de la región. Variedad y calidad donde se encuentra, desde ofimática y productividad, carreras de certificación y diseño gráfico, cursos de tecnología para la nueva gerencia, para los docentes en escuelas, cursos de Informática educativa, talleres computacionales para niños y adolescentes y mucha más diversidad para todas las personas que quieren estar a la vanguardia con la tecnología.

Tienen sistemas de intercambio con otras empresas donde publican las necesidades de las habilidades que se necesitan obtener para desempeñar un cargo eficientemente, además de contar con el apoyo de empresas como la Microsoft Certified para certificar conocimientos adquiridos con este sistema.

1.4 Metodologías de Desarrollo de Software

El desarrollo de software es un proceso bastante difícil y gran parte del éxito que se tenga para obtener un buen producto es la metodología que se utilice, ya que incide de forma diferente en su desarrollo. De nada sirven buenas notaciones y herramientas si no se proveen directivas para su aplicación.

Dentro de las metodologías más importantes para el desarrollo de software se encuentran:

Proceso Unificado de Rational (RUP)

Es una metodología utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, se considera la más usada y se caracteriza por ser iterativa e incremental, centrada en la arquitectura y guiada por casos de usos. Es un producto de Rational del cual IBM es propietario.

RUP es caracterizado por:

- ✓ Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- ✓ Pretende implementar las mejores prácticas en Ingeniería de Software.
- ✓ Desarrollo iterativo.
- ✓ Administración de requisitos.
- ✓ Uso de arquitectura basada en componentes.
- ✓ Control de cambios.
- ✓ Modelado visual del software.
- ✓ Verificación de la calidad del software.

Extreme Programming (XP)

Es una metodología ágil que centra sus prioridades en las personas y no en los procesos, se diseñan e implementan las pruebas antes de programar la funcionalidad. Se basa en cuatro

principios fundamentales que son: simplicidad, comunicación, retroalimentación y valor. Es sencillo, común y adaptable a las características cambiantes de los clientes y empresas.

Extreme Programming es caracterizado por:

- ✓ Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- ✓ Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- ✓ Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- ✓ Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- ✓ Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- ✓ Refactorización del código, es decir: reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento.
- ✓ Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.

SCRUM

Es una metodología ágil que se caracteriza por ser iterativa e incremental, puede utilizarse en equipos de mantenimiento de software pero está enfocada principalmente a la gestión de desarrollo de software.

SCRUM se caracteriza por:

- ✓ Define un conjunto de prácticas y roles.
- ✓ En periodos de tiempo establecidos (Sprint), el equipo crea un incremento de software potencialmente entregable (utilizable).
- ✓ Se realizan reuniones al comienzo de cada Sprint donde se especifican que parte del proyecto se va a completar.
- ✓ Durante cada Sprint los requisitos a completarse no pueden ser cambiados.

Una vez analizadas las diferentes metodologías de desarrollo se decide utilizar RUP por sus grandes ventajas que permiten que el sistema a implementar proporcione los resultados esperados.

Se emplea RUP además por:

Utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema de software. Permite la producción de software determinando un tiempo y un costo y logra cumplir con los requisitos de los usuarios, dándoles solución a sus necesidades.

Optimiza la productividad de los miembros del equipo de desarrollo ya que pone al alcance de estos, la experiencia derivada de miles de proyectos anteriores, así como plantillas y guías que permiten lograr un resultado en un tiempo aceptable. Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar software.

RUP se integra estrechamente con herramientas Rational, permitiendo a los equipos de desarrollo aprovechar todas las ventajas de las características de los productos Rational, el Lenguaje de Modelado Unificado (UML) y otras prácticas óptimas de la industria de software.

1.5 Lenguajes de modelado

Un proceso de negocio es un conjunto de actividades relacionadas lógicamente que se ejecutan para lograr un resultado de negocio definido. Cada uno tiene sus entradas (información que se necesita), roles (quién hace qué), funciones (como se hace) y salidas (objetivo final de la actividad). Cuando una función es aplicada a las entradas de un método, se tienen ciertas salidas resultantes. El modelado de procesos es la representación gráfica de los procesos que una persona o empresa efectúa para alcanzar una meta.

Un modelo de procesos debe permitir visualizar el comportamiento de un sistema, donde cada proceso pueda ser analizado, automatizado y mejorado. Existen varios lenguajes que permiten representar los procesos de una empresa u organización, tales como: Método para la Modelación Funcional de Procesos (IDEF0), Notación de Modelado de Procesos de Negocio (BPMN) y Lenguaje Unificado de Modelado (UML).

Método para la modelación funcional de procesos (IDEF).

Es un estándar para el desarrollo estructurado de gráficos representativos de procesos. Su propósito es proveer técnicas de modelado simples y formales que permitan describir, analizar y evaluar distintos puntos de vista de una organización.

IDEF0 permite la modelación funcional de procesos de manera estructurada y jerarquizada, teniendo en cuenta las decisiones, acciones y objetos que soportan la interacción de esas actividades de la organización. Representa lo que se hace en la empresa de un modo no temporal. Su principal preocupación es: qué actividades se llevan a cabo. (Knowledge, 2006).

La representación de un proceso consta de la combinación de cinco unidades básicas que interactúan: Entrada (s), Salida (s), Control (es), Mecanismo (s) y la actividad. Las cuatro primeras representadas por flechas y la última mediante una forma rectangular. (Loyola, 2006).

Uno de los aspectos de IDEF0 más importantes es que como concepto de modelización va introduciendo gradualmente más y más niveles de detalle a través de la estructura del modelo. De esta manera, la comunicación se produce dando al lector un tema bien definido con una cantidad de información detallada disponible para profundizar en el modelo.

Las entradas expresan los datos u objetos que serán transformados por la función en salidas. Los controles expresan condiciones requeridas para producir las salidas correctas. Los datos u objetos tratados como controles pueden ser transformados por la función creando una salida. Las salidas expresan los datos u objetos producidos por la función. Los mecanismos expresan los medios utilizados para ejecutar la función.

De manera general IDEF0 tiene las siguientes características: (Porcesos, 1993).

- ✓ Es una técnica genérica que permite modelar gráficamente procesos de sistemas de diferentes propósitos a cualquier nivel de detalle.
- ✓ Es comprensivo y expresivo, capaz de representar gráficamente reglas y una amplia variedad de negocios.
- ✓ Es un lenguaje simple y coherente.
- ✓ Realza la comunicación entre los analistas de sistemas, los desarrolladores y los usuarios por la facilidad de aprender sus distintas representaciones y por su énfasis en la exposición jerárquica del detalle.

- ✓ Está bien probado y comprobado, con muchos años de uso por la Fuerza Aérea de Estados Unidos (EEUU), otros proyectos gubernamentales e industrias privadas.
- ✓ Puede ser generado por una variedad de herramientas de modelado gráfico de computadora.
- ✓ Facilita el análisis para identificar puntos de mejora.

Notación de Modelado de Procesos de Negocio (BPMN)

BPMN es un nuevo estándar de modelado de procesos de negocio. Esta notación está diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

Su principal objetivo es proveer una notación estándar que sea fácilmente entendible por parte de todos los involucrados e interesados del negocio. Entre estos interesados están los analistas de negocio, los desarrolladores técnicos y los gerentes y administradores del negocio. En síntesis: BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación.

Los objetos de flujo tienen tres elementos centrales: evento, actividad y decisión. Los mismos se conectan en un diagrama para crear el esqueleto básico de la estructura de un proceso de negocio. La versión actual de BPMN predefine sólo tres tipos de artefactos: objeto de datos, grupo y comentario.

Lenguaje Unificado de Modelado (UML)

UML es definido por sus creadores Booch, Jacobson y Rambaugh como: lenguaje para la especificación, visualización y construcción de artefactos de sistemas de software. Es un estándar para construir modelos, no guía al desarrollador en la forma de realizar el análisis de sistema, ni le indica cual proceso de desarrollo a adoptar. [Larman, 1999].

Es un lenguaje estándar de modelado orientado a objetos que se usa para especificar y no para describir métodos o procesos. Puede integrarse y utilizarse con cualquier metodología de desarrollo de software, incluso con un híbrido entre dos o más metodologías. Existen varios software como por ejemplo: Visual Paradigm, Pacestar UML Diagrammer, Rational Rose entre otros, para apoyar el modelado UML, brindando importantes funcionalidades. Permite el desarrollo de distintos tipos de

diagramas, cada uno de los cuales representa el sistema a especificar, analizar o diseñar desde distintas perspectivas.

Se propone para este trabajo modelar el proceso del negocio con IDEF0 y emplear UML para especificar el mismo a través de la generación de los diferentes diagramas de iteración y no se emplea inicialmente porque no está dirigido para describir procesos.

Se emplea IDEF0 además por:

Por ser un lenguaje muy simple y comprensivo. Estas características ayudarán al entendimiento entre el equipo de desarrollo y clientes. Otros negocios tratados por proyectos pertenecientes a la Universidad de las Ciencias Informáticas han tenido satisfactorios resultados con clientes similares haciendo uso del mismo.

Como se puede apreciar, es un lenguaje fácil de utilizar y los resultados son muy buenos atendiendo al nivel de detalle que se puede obtener al modelar los procesos y actividades relacionadas con los sistemas a desarrollar.

Se emplea UML además por:

Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos, la arquitectura del mismo. Permite describir un sistema en diferentes niveles de abstracción, simplificando la complejidad sin perder información, para que los usuarios y desarrolladores comprendan las características de la aplicación. Se ha convertido en un lenguaje estándar con el que se puede modelar todos los componentes del desarrollo de una aplicación, sin pretender definir un modelo de desarrollo sino únicamente un lenguaje de modelado.

1.6 Herramientas para el modelado

Dentro de las herramientas más utilizadas para el desarrollo de software hoy día, están las herramientas CASE (Computer Aided Software Engineering), que son aplicaciones que facilitan el desarrollo, reduciendo el esfuerzo, el costo y el tiempo.

Visual Paradigm

Es una herramienta profesional que ayuda a la construcción de aplicaciones con mayor calidad y a menor costo. Proporciona un gran número de tutoriales, demostraciones y soporta todo el ciclo de vida del desarrollo del software.

Sus características principales son:

- ✓ Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.
- ✓ Generación de bases de datos.
- ✓ Ingeniería inversa de bases de datos.
- ✓ Integración con Visio.
- ✓ Editor de Detalles de Casos de Uso.

Enterprise Architect (EA)

Provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue. Combinados con la ubicación de recursos y tareas incorporados. Los equipos de administradores de proyectos y calidad están dotados con la información que ellos necesitan para ayudarles a entregar proyectos en tiempo.

Es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. Tiene buena velocidad, estabilidad y buen rendimiento.

EA soporta transformaciones de Arquitectura avanzada dirigida por Modelos usando plantillas de transformaciones de desarrollo muy fáciles de usar. Con transformaciones incorporadas para DDL, C#, Java, EJB y XSD, se puede desarrollar rápidamente soluciones complejas desde los simples "modelos independientes de plataforma" (MIP) que son el objetivo en "modelos específicos de plataforma" (MEP).

Sus principales características son:

- ✓ Es rápido.
- ✓ Tiene incorporado gran cantidad de funciones que facilitan el trabajo.
- ✓ Presenta una buena interfaz gráfica.
- ✓ Muy buena ingeniería inversa

Algunas de sus desventajas son:

- ✓ El control de versiones es lento.
- ✓ Es complicado hacer combinaciones entre dos importaciones con elementos en común.

Pacestar UML Diagrammer:

Es una herramienta creada para explotar el mejor aspecto de UML y está especializada para el dibujo y diseño, además simplifica el proceso de modelado.

Sus principales características son:

- ✓ Crea de forma rápida y fácil diagramas de UML.
- ✓ Desarrolla y documenta diseños en una notación comúnmente extendida.
- ✓ Genera todas las variedades de diagramas cómodamente.
- ✓ Presenta una interfaz sencilla y cómoda.
- ✓ Tiene una gran galería de símbolos que permiten crear diagramas personalizados.

BPWin 4.0

Es una potente herramienta para modelar, la cual analiza, documenta y mejora los procesos de negocio en una organización. Permite documentar de manera clara los elementos más importantes, como que actividades son necesarias, cómo se realizan y qué recursos consumen, lo cual nos proporciona una visión exacta, no solo de qué es lo que hace una organización, sino si lo hace de forma eficiente.

Bpwin proporciona un marco de trabajo para poder representar y entender los procesos de negocio, determinando el impacto de los diferentes sucesos y definiendo cómo los procesos interactúan unos con otros mediante flujos de información permitiéndonos identificar actividades poco eficientes o redundantes. Actualmente el BPwin ofrece un conjunto de bondades que lo hacen más utilizable en el mundo de los procesos y la ingeniería.

Entre sus beneficios tenemos:

- ✓ Interface Intuitiva: Se puede navegar fácilmente en todo el entorno.
- ✓ Diseño automatizado de procesos: Se pueden corregir los resultados en el diseño de los procesos.

- ✓ Propiedades Definidas por el Usuario: Se puede usar BPWin para capturar información relevante a sus procesos y utilizarlas cuando se requiera.
- ✓ Técnicas de Integración: Provee las técnicas más comerciales para documentar los procesos empresariales y flujos de proceso a través del IDEF0.

Rational Rose

Es una herramienta de modelado muy utilizada y la más completa dentro de la familia de Rational Rose.

Sus principales características son:

- ✓ Facilita la creación de un mejor software en menor tiempo debido a su lenguaje común de modelado.
- ✓ Soporte para análisis de patrones ANSI C++, Rose J y Visual C++.
- ✓ Capacidad de análisis de calidad de código.
- ✓ Modelado UML para trabajar en diseños de base de datos.
- ✓ Integración con otras herramientas de desarrollo de Rational.

Luego del estudio de las herramientas de modelado, se decide hacer uso de BPWin para modelar el proceso del negocio y para obtener los diagramas de casos de usos, clases del análisis, clases del diseño y diagramas de iteración se utiliza Visual Paradigm.

Se emplea BPWin además por:

La solución de modelación de procesos empresariales de BPwin proporciona el marco para comprender estos procesos, determinando el impacto de eventos empresariales y definiendo la forma de interacción de estos con los datos que circulan por la empresa. Un modelo de proceso puede permitir claramente documentar factores importantes dentro de la empresa.

Se emplea Visual Paradigm además por:

Genera código ya sea a través de un modelo o un diagrama lo que permite minimizar el tiempo de trabajo y puede generar bases de datos transformando los diagramas de entidad relación en tablas de bases de datos.

Tiene un editor de figuras y de detalles de casos de uso que incluyen la especificación del modelo general y de las descripciones de casos de uso. Distribuye automáticamente los diagramas, reorganizando las figuras y conectores de los diagramas UML.

1.7 Ingeniería de requisitos.

La ingeniería de requisitos del software es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refina en detalle los requisitos del sistema y el papel asignado al software. En la ingeniería de requisitos tanto el desarrollador como el cliente tienen un papel activo. El cliente intenta replantear un sistema confuso, mientras que el desarrollador pretende solucionar el problema y negociar.

La ingeniería de requisitos proporciona el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. (Pressman, 2001).

Actividades de la Ingeniería de requisitos:

Pressman en su libro “Ingeniería del software. Un enfoque práctico”, en su quinta edición, define el proceso de ingeniería de requisitos en 5 pasos principales, dividiendo el último en dos actividades diferentes, a continuación se definen los pasos a seguir en el proceso:

Identificación de requisitos: De las seis actividades tratadas es la de mayor interacción con los clientes, en esta actividad se recurre a la realización de entrevistas, talleres entre otras técnicas, además de ser en la que el analista y los expertos en el tema trabajan conjuntamente con el cliente para obtener los requerimientos reales de una forma óptima. En esta etapa es donde se adquiere toda la información referente al trabajo del cliente y se busca comprender sus necesidades y detallando las restricciones del negocio, objeto de estudio y como resultado se obtiene los requerimientos de todas las parte involucradas en el negocio.

Análisis y Negociación de requisitos: Se realiza un estudio de los requerimientos obtenidos previamente para detectar la presencia de áreas no especificadas, requisitos contradictorios y peticiones vagas e irrelevantes. Como resultado a este análisis puede que se necesite volver a la primera actividad para eliminar todo tipo de inconsistencia.

Especificación de requisitos: La especificación del sistema es el producto final sobre los requisitos del sistema obtenidos por el analista. Esta especificación describe la información que entra y sale del sistema y delimita cada elemento del sistema. (Pressman, 2001).

Modelado del sistema: En esta actividad se modela el sistema con el objetivo de evaluar los componentes del mismo y sus relaciones entre sí, además de reflejar en dicho sistema todos los requerimientos capturados.

Validación de requisitos: La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto. (Pressman, 2001).

Gestión de requisitos: la gestión de requisitos es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir requisitos y cambios en cualquier momento. (Pressman, 2001).

De la ingeniería de requisitos se aplicará la identificación, especificación y validación de requerimientos; a continuación se explican las técnicas utilizadas en las etapas seleccionadas:

1- Identificación de requisitos:

Cuestionarios: Documento redactado con preguntas cuyas respuestas deben ser cortas y concretas, o incluso cerradas por unas cuantas opciones. Apoya la realización de entrevistas individuales o colectivas. (Pressman, 2001).

Entrevistas: Resulta una técnica muy aceptada dentro de la IR y su uso está ampliamente extendido; se realiza para comprender el problema y los objetivos de la solución buscada. La estructura de las entrevistas abarcan tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados (protocolo de la entrevista). (Toro, y otros, 2000).

Tormenta de ideas (Brainstorming): Técnica de reuniones en grupos cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consisten en una mera acumulación de ideas o información sin evaluar las mismas. El grupo de personas que participan en estas reuniones no debe ser muy numeroso (máximo 20 personas). Ofrecen una visión general de las necesidades del sistema, pero normalmente no sirven para obtener detalles concretos del mismo, por lo que generalmente se aplican en los primeros encuentros. (Toro, y otros, 2000).

Casos de Uso (CU): La ventaja esencial es que resultan muy fáciles de entender para el usuario o cliente. Sin embargo, carecen de la precisión necesaria si no se acompañan con una información textual o detallada con otra técnica como pueden ser los diagramas de actividades [UML, 2001]. Describe la secuencia de interacciones que se producen entre el sistema y los actores, para realizar una determinada función, permitiendo mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso). (UML, 2001).

2- Especificación de los requisitos

Lenguaje natural: Muchas veces resulta una técnica muy ambigua para la definición de los requisitos y con ella se definen en un lenguaje natural. A pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando, debido a la complejidad y la gran variedad de lenguajes formales que dificultan el entendimiento entre los clientes y analistas.

3- Validación de requisitos

Matriz de trazabilidad: Consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo, es necesario ir viendo qué objetivos cubre cada requisito, ya que de esta forma se podrán detectar inconsistencias u objetivos no cubiertos. Esto garantiza de manera más precisa, que todos los requisitos funcionales estén asociados con al menos un caso de uso. (Toro, y otros, 2000).

Prototipos: Se basa en obtener prototipos de los requisitos, sin tener la totalidad de la funcionalidad del sistema. Permiten al usuario hacerse una idea de la estructura de la interfaz del sistema, representación visual estándar sujeta a cambios para que el cliente y los desarrolladores obtuvieran una visión general de lo que se propone implementar. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final. (Olsina, 1999).

1.8 Patrones empleados

1.8.1 Patrones de casos de uso

Los patrones son soluciones simples y elegantes a problemas específicos y comunes. Son soluciones basadas en la experiencia, que se ha demostrado que funcionan y pueden emplearse en diferentes contextos. Los patrones para casos de uso proveen una gran ayuda a la hora de confeccionar modelos de casos de uso que sean re-usables y de alta calidad.

Extensión Concreta (Concrete Extensión or Inclusión: Extensión): Es un patrón de estructura. Consiste en dos casos de usos y una relación de extensión entre ellos. El caso de uso extendido es concreto, esto quiere decir que: este puede ser instanciado por sí solo, así como extender el caso de uso base. El caso de uso base puede ser concreto o abstracto. Es aplicable cuando un flujo de un caso de uso puede extender el flujo de otro caso de uso, así como ser ejecutado por sí solo.

Inclusión Concreta (Concrete Extensión or Inclusión: Inclusión): Es un patrón de estructura. Consiste en dos casos de usos y una relación de inclusión entre el caso de uso base y el caso de uso incluido. Este último puede ser instanciado por sí solo. El caso de uso base puede ser concreto o abstracto. Se utiliza este patrón cuando un flujo de datos puede ser incluido en el flujo de otro caso de uso y también puede ejecutarse por sí solo.

Completo (CRUD: Complete): CRUD cuyas siglas significan Creating, Reading, Updating and Deleting, es un patrón de estructura. Propone identificar en un caso de uso, todas las diferentes operaciones que se pueden realizar sobre una parte de información de cierto tipo (o sea en una misma entidad), tal como crearla, leerla, actualizarla y eliminarla. Debe ser usado cuando todos los flujos contribuyen al mismo valor de negocio, son cortos y sencillos.

Actores múltiples: Rol común (Multiple Actors: Common Role): Es un patrón de estructura que plantea que cuando dos actores juegan el mismo papel hacia un caso de uso se representa otro actor, del que heredan los actores que comparten este rol. Este patrón es aplicable cuando, desde el punto de vista de un caso de uso, hay solo una entidad externa interactuando con cada instancia del caso de uso.

Reglas de Negocio: Definición Estática (Business Rules: Static Definition): El patrón Reglas de Negocio: Definición Estática es de tipo descripción, por lo que no influye sobre la estructura del

modelo de casos de uso. Este patrón es aplicado a todos los casos de uso que modelan servicios que son afectados por reglas de negocio definidas en la organización. Las reglas son descritas en un documento por separado, referenciado por la descripción del caso de uso. Es apropiado cuando no se necesita cambios dinámicos en las reglas del negocio cuando el sistema está en uso.

Concordancia: Re-uso (Commonality: Reuse): Es un patrón de estructura que consiste en tres casos de uso. El primero llamado "Sub-secuencia Común", modela la secuencia de acciones que aparecen en múltiples casos de uso del modelo. Los otros dos casos de uso comparten de esta sub-secuencia común de acciones (dos es la menor cantidad que puede existir).

La sub-secuencia tiene que estar en un fragmento, es decir, todo lo que requiere estar incluido tiene que estar en un único fragmento completo. Además no se puede hacer referencia desde la sub-secuencia a donde esta es utilizada, porque el caso de uso incluido tiene que ser independiente del caso de uso base. (Övergaard, 2004).

1.8.2 Patrones de arquitectura

El patrón de arquitectura Modelo Vista Controlador(MVC) permite obtener una arquitectura robusta, que guía el proceso de desarrollo, y que define de manera abstracta, los componentes que llevan a cabo alguna tarea, sus interfaces y la comunicación ente ellos. (Potencier, y otros, 2008). Él separa la Vista (interfaz) del Modelo (base datos) y del Controlador (lógica negocio).

Modelo

El modelo lo divide en una capa de acceso a datos y otra de abstracción de datos. La abstracción indica que se quiere de la base de datos y la capa de acceso hace las consultas necesarias para obtener esa información. Así, si cambiamos de base de datos, tan solo cambiaremos la capa de acceso, y la capa de abstracción podrá seguir haciendo las mismas operaciones.

Vista

En la presentación de las páginas tendremos varios elementos comunes en la mayoría de ellas, como son la cabecera, la navegación, el pie de página y las layout, cambiando tan solo el interior o el contenido de la página. Así estos 3 elementos quedan separados.

Controlador

El trabajo del controlador se repite para muchas acciones. Symfony crea un controlador frontal, único en la aplicación, que está encargado de realizar estas labores comunes, como es el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Además es el único punto de entrada a la aplicación.

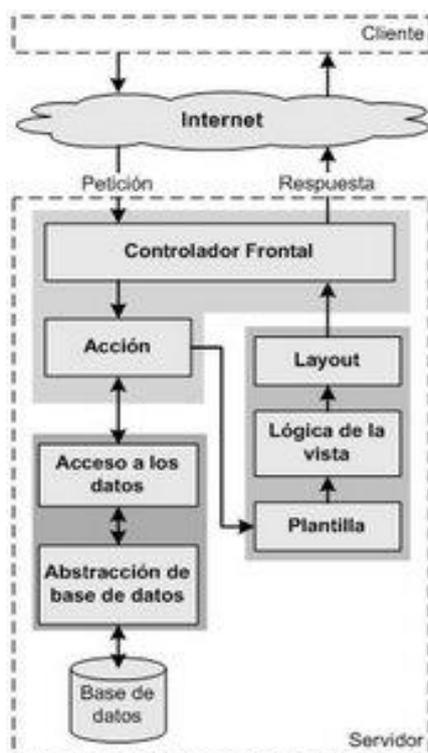


Fig.1.1 Patrón MVC para el framework de desarrollo Symfony.

El principio más importante de la arquitectura MVC es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. Symfony además utilizando otros patrones de diseño puede simplificar la programación y de esta forma, las capas del modelo, la vista y el controlador se pueden subdividir en más capas. (Potencier, y otros, 2008).

Unido a esto, este patrón incorpora ventajas importantes tales como:

- ✓ Al separar de manera clara la lógica de negocio (modelo) de la vista permite la reusabilidad del modelo, de modo que la misma implementación de la lógica de negocio que maneja una aplicación pueda ser usado en otras aplicaciones, sean éstas web o no.
- ✓ Permite una sencilla división de roles, dejando que sean diseñadores gráficos sin conocimientos en programación o en desarrollo de aplicaciones, los que se encarguen de la realización de la capa vista, sin necesidad de mezclar código php entre el código visual que desarrollen.

1.8.3 Patrones de diseño

Los patrones de diseño son un buen método para resolver pequeños problemas que pueden adaptarse a una variedad mucho más amplia de problemas específicos. En el contexto de los sistemas y aplicaciones basados en Web, los patrones de diseño pueden aplicarse en el nivel jerárquico, nivel de componentes y nivel de hipertexto (de navegación). (Pressman, 2001).

Los patrones de diseño identifican: Clases, instancias, roles, colaboraciones y la distribución de responsabilidades. Los patrones de diseño no son fáciles de entender, pero una vez que se logra conocer su funcionamiento, el diseño será mucho más flexible, modular y reutilizable. Han revolucionado el diseño orientado a objetos y todo buen arquitecto de software debería conocerlos.

El objetivo que se tiene al utilizarlos es:

- ✓ Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- ✓ Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- ✓ Formalizar un vocabulario común entre diseñadores.
- ✓ Estandarizar el modo en que se realiza el diseño.
- ✓ Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Los patrones han sido creados para ser aplicados en diferentes esferas de la creación de un software pero siempre acumulando la experiencia de los programadores, en dependencia de para qué son aplicados se pueden clasificar en: (Gamma, 2000).

Patrones de creación: Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos delegará responsabilidades otro objeto.

Patrones estructurales: Describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.

Patrones de comportamiento: Se utilizan para organizar, manejar y combinar comportamientos.

Teniendo en cuenta que se propone trabajar con el framework Symfony, hay que tener en cuenta que este ha sido creado para brindar mayores y mejores facilidades a los arquitectos de software, para ello se le han incorporado varios patrones de diseños.

A continuación se relacionan cuales son los principales patrones que tiene asociado este framework:

De los creacionales: (GOF, 1995).

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia desde cualquier parte del framework.

El patrón Singleton provee una única instancia global gracias a que:

- ✓ La propia clase es responsable de crear la única instancia.
- ✓ Permite el acceso global a dicha instancia mediante un método de clase.
- ✓ Declara el constructor de clase como privado para que no se cree una instancia directamente.

Se usa generalmente cuando deba haber exactamente una instancia de una clase y ésta deba ser accesible a los clientes desde un punto de acceso conocido, cuando la única instancia debería ser extensible mediante herencia y los clientes deberían ser capaces de utilizar una instancia extendida sin modificar su código.

Las ventajas de usar este patrón son:

- ✓ Acceso controlado a la única instancia. Puede tener un control estricto sobre cómo y cuándo acceden los clientes a la instancia.
- ✓ Espacio de nombres reducido. El patrón Singleton es una mejora sobre las variables globales.

- ✓ Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.
- ✓ Permite un número variable de instancias. El patrón hace que sea fácil cambiar de opinión y permitir más de una instancia de la clase Singleton.

Abstract Factory (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las estas no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando. Cuando el framework necesita por ejemplo crear un nuevo objeto para una petición, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

Este patrón es usado generalmente cuando un sistema debe ser independiente de los procesos de creación, composición y representación de sus productos o debe ser configurado con una familia múltiple de productos. Además si se quiere proporcionar una librería de productos y no revelar su implementación (simplemente revelando sus interfaces).

Las ventajas de usar este patrón son:

- ✓ Aísla las clases de implementación ya que ayuda a controlar los objetos que se creen y encapsula la responsabilidad de creación.
- ✓ Hace fácil el intercambio de familias de productos sin mezclarse, permitiendo configurar un sistema con una de entre varias familias de productos:
- ✓ Fomenta la consistencia entre productos.

De los estructurales: (Gamma, 2000).

Decorator (Envoltorio): Añade dinámicamente funcionalidad a una clase. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación de este patrón de diseño.

El patrón es usado generalmente cuando se desea adicionar responsabilidades a un objeto pero no a toda la clase. Las responsabilidades se pueden adicionar por medio de los mecanismos de herencia, pero este mecanismo no es flexible porque la responsabilidad es adicionada

estáticamente. La solución flexible es la de rodear el objeto con otro que es el que adiciona la nueva responsabilidad. Este nuevo objeto es el Decorator. (Potencier, y otros, 2008).

Algunas ventajas de este patrón son:

- ✓ Mayor flexibilidad que la herencia estática, incluso puede decorarse un objeto varias veces con el mismo decorador, lo que sería imposible con la herencia.
- ✓ Favorece la definición de interfaces y clases bases ligeras. Sólo se carga con las funcionalidades que se necesitan, no son necesarias megaclasses que sean capaces de hacerlo todo.

El Symfony tiene implementado a su vez otros patrones conocidos como patrones GRASP con sus siglas en ingles que significan “patrones generales de software para asignar responsabilidades”. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos. Estos describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Dentro de los patrones que entran en esta clasificación podemos encontrar:

- ✓ Creador.
- ✓ Experto.
- ✓ Alta Cohesión.
- ✓ Controlador.
- ✓ Bajo Acoplamiento.
- ✓ Polimorfismo.
- ✓ Fabricación Pura.
- ✓ Indirección.
- ✓ No Hables con Extraños (Ley de Demeter).

Aunque todos son importantes y se utilizan a menudo, se consideran los más importantes los 5 primeros.

De los patrones GRASP se utilizan: (Gamma, 2000).

Creador: Este patrón como su nombre lo indica es el que crea, guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un Objeto de la clase A solamente cuando:

- ✓ B contiene a A.
- ✓ B es una agregación (o composición) de A.
- ✓ B almacena a A.
- ✓ B tiene los datos de inicialización de A (datos que requiere su constructor).
- ✓ B usa a A.

En las clases Actions que se representan en las clases de diseño con estereotipo web, se encuentran las acciones definidas para el sistema y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es "creador" de dichas entidades.

Las ventajas que ofrece este patrón son:

- ✓ Es sencillo y de fácil uso.
- ✓ Con solo utilizar un poco de lógica podemos conseguir un diseño sencillo y claro con una cohesión y acoplamiento adecuados.

Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada.

Este es uno de los patrones más utilizados, puesto que Propel es la librería externa que utiliza Symfony para realizar su capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

Hay que tener en cuenta que esto es aplicable mientras estemos considerando los mismos aspectos del sistema:

- ✓ Lógica de negocio.
- ✓ Persistencia a la base de datos.
- ✓ Interfaz de usuario.

Alta Cohesión: La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento, cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-

identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las properties, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a varios cambios.

Las ventajas de utilizar este patrón son que en un modelo podemos tener clases que hacen varias cosas, los ejemplos de buen diseño se producen cuando se crean los denominados "paquetes de servicio" o clases agrupadas por funcionalidades que son fácilmente reutilizables (bien por uso directo o por herencia).

Controlador: Es un evento generado por actores externos. Normalmente un controlador delega en otros objetos el trabajo que se necesita hacer, coordina o controla la actividad, no realiza mucho trabajo por sí mismo.

En el caso de Symfony, todas las peticiones Web son manejadas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario, sfController es la clase del controlador. El controlador gestiona estas peticiones, y realiza algunas funciones como: comprobación de restricciones de seguridad, manejo de errores, mapear y delegación de las peticiones a otros componentes de la aplicación que se encargarán de generar la vista adecuada para el usuario.

Dentro de las ventajas que proporciona este patrón se encuentran:

- ✓ Centraliza en un único punto la gestión de las peticiones.
- ✓ Aumenta la reusabilidad de código.
- ✓ Mejora la gestión de la seguridad.

Bajo Acoplamiento: El acoplamiento es una medida de fuerza con que un elemento: está en, tiene conocimiento de, confía en, otros elementos. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

La idea es tener las clases lo menos ligadas entre sí que se pueda, para que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases. Potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Para Symfony, Las clases Actions heredan solamente de sfActions para lograr un bajo acoplamiento de clases.

Las principales ventajas de este patrón son:

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

De esta manera se resume el uso de los principales patrones que tiene implementado el framework de desarrollo que se propone para utilizar y que repercuten en el diseño del sistema.

1.9 Lenguajes de programación

En la actualidad existen lenguajes de programación y compiladores muy avanzados que contienen palabras reservadas para rutinas mucho más complejas y elaboradas. Lo que permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural.

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

PHP

Es un lenguaje de programación interpretado, diseñado especialmente para la creación de páginas web dinámicas. El hecho de que sea interpretado lo hace entrar dentro de las polémicas diferencias que se atribuyen entre estos y los lenguajes compilados. La real diferencia, se encuentra en si se usa un compilador o un intérprete y no en el lenguaje como tal. Por ser interpretado, PHP entra dentro de la clasificación de lenguaje de Script donde se encuentran otros como Python y Java Script.

Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Algunas ventajas de PHP son:

- ✓ Es un lenguaje multiplataforma.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- ✓ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos.
- ✓ Posee una amplia documentación, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite las técnicas de Programación Orientada a Objetos.
- ✓ Biblioteca nativa de funciones sumamente amplia e incluida.
- ✓ No requiere definición de tipos de variables.
- ✓ Tiene manejo de excepciones (desde PHP5).

C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de

objetos de la plataforma.NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes como Delphi.

Entre las principales características de C# podemos encontrar:

- ✓ Sencillez: Elimina muchos elementos que otros lenguajes incluyen y que son innecesarios.
- ✓ Modernidad: Incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones.
- ✓ Orientación a objetos: Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos.
- ✓ Orientación a componentes: La sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.
- ✓ Gestión automática de memoria: Todo lenguaje de .NET tiene a su disposición un recolector de basura.
- ✓ Seguridad de tipos: C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente.
- ✓ Instrucciones seguras: Para evitar errores muy comunes, en C# se han impuesto una serie de restricciones en el uso de las instrucciones de control más comunes.
- ✓ Sistema de tipos unificado: A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada System.Object.

Las desventajas principales de la aplicación o uso de C# son que consume en su instalación plena una cantidad de recursos de hardware en el orden medio-alto y que no es recomendable para programadores o desarrolladores novatos.

Python

Python es un lenguaje de programación interpretado creado por Guido van Rossum en el año 1991. Se compara habitualmente con TCL, Perl, Scheme, Java y Ruby. En la actualidad Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. La última versión estable del lenguaje es la 3.0.1.

Es un lenguaje muy liviano, de sintaxis muy elegante. La propia estructura del lenguaje obliga a escribirlo de forma legible y clara. Es muy recomendable para principiantes y desarrolladores de poca experiencia.

Dentro de sus características se encuentran:

- ✓ Simple: Python es en lenguaje simple.
- ✓ Sencillo de Aprender: Es recomendable para iniciarse en la programación ya que ofrece una sintaxis extraordinariamente simple.
- ✓ Libre y Fuente Abierta: Es un ejemplo de software de fuente abierta, gratuito y libre.
- ✓ Lenguaje de Alto Nivel: Cuando se escribe programas en Python nunca se deben atender detalles de bajo nivel, como manejar la memoria empleada por el programa, por ejemplo.
- ✓ Orientado a Objetos: Python permite programación orientada a objetos.
- ✓ Ampliable: Si se necesita que una pieza de código se ejecute muy rápido, se puede lograr escribiéndola en C y después combinándola con el programa de Python.
- ✓ Incrustable: Se puede insertar Python dentro de un programa en C/C++ para ofrecer las facilidades de "scripting" dentro del mismo.
- ✓ Librerías Extendidas: La librería estándar de Python es de hecho muy amplia. Sirve de ayuda para hacer varias cosas que involucran: expresiones regulares, generación de documentos, evaluación de unidades, pruebas, procesos, bases de datos, navegadores web, CGI, ftp, correo electrónico, XML, XML-RPC, HTML, archivos WAV, criptografía, GUI (interfaz grafica del usuario), entre otros.

La principal desventaja de Python es: que por ser un lenguaje interpretado hace que los programas sean más lentos que con un lenguaje compilado.

Java

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Es un lenguaje independiente de la plataforma por lo que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Es una tecnología orientada a la

programación de software con el cual se puede realizar cualquier tipo de programas. Hoy en día, la tecnología Java ha cobrado mucha importancia en el ámbito de Internet gracias a su plataforma J2EE.

Una de las principales características que favoreció el crecimiento y difusión de este lenguaje es su capacidad de que el código fuente funcione sobre cualquier plataforma de software y hardware, es decir, un mismo programa puede ejecutarse en varios sistemas sin tocar el código fuente.

C++:

Es un lenguaje de programación orientado a objetos, dentro de sus principales características están las facilidades que proporciona para la programación y para el uso de plantillas o programación genérica. Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel.

Sus principales características son:

- ✓ Posibilidad de redefinir los operadores (sobrecarga de operadores).
- ✓ Identificación de tipos en tiempo de ejecución (RTTI).
- ✓ Además el mismo está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel.

Por las características descritas anteriormente y teniendo en cuenta los requerimientos del sistema en cuestión, se decide hacer uso de PHP como lenguaje de programación del lado del servidor, es gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación, además ofrece una mejor adecuación al entorno de las aplicaciones web y cuenta con una amplia aplicación y validación en este tipo de sistemas.

Se emplea PHP además por:

PHP es software libre, lo que implica menos costos y servidores más baratos, por lo que podemos usarlo en proyectos diferentes sin necesidad de pagar altos precios por una licencia. Posee galerías estándar muy amplias y una de las comunidades más grandes que se pueden encontrar en Internet, por lo que resulta fácil encontrar tanto ayuda como documentación ya que es la gran tendencia actual en el mundo del desarrollo web.

En caso de estar montada la aplicación sobre un servidor Linux o Unix es más rápida que ASP ya que se ejecuta en un único espacio de memoria y es un lenguaje interpretado en HTML. Posee una gran compatibilidad con las bases de datos más comunes, como MySQL, MSSQLServer, SQLServer, Oracle, Postgre, entre otras.

PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unos 20 servidores de bases de datos y contiene unas 40 extensiones estables y se continúa extendiendo su uso. No cabe duda de las magníficas facilidades que nos brinda este lenguaje por tanto se considera el más propicio a emplear en el desarrollo de este sistema.

1.10 Servidores web

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

A los computadores encargados de ofrecer este servicio también se les conoce como servidores Web. De algunos de los servidores web más conocidos y utilizados se ofrece un estudio a continuación.

Apache

El servidor HTTP Apache es un software servidor 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. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo.

Algunas de las ventajas de Apache son:

- ✓ Modular: La arquitectura del servidor Apache es muy modular. El servidor consta de diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web.
- ✓ Código Abierto: Es Libre de costo para usarse y redistribuirse, además por estar disponible su código fuente puede ser reconfigurado o reprogramado para obtener funcionalidades muy específicas o personalizadas.

- ✓ Multi-plataforma: Puede ser utilizado en plataformas como: Unix y derivadas, Windows, Macintosh entre otras.
- ✓ Extensible: Se le pueden agregar fácilmente nuevas funcionalidades.
- ✓ Popular: El hecho de que sea popular ayuda mucho a la hora de conseguir documentación y resolver problemas puesto que cuenta con una comunidad amplia de usuarios muy colaborativos.

Internet Information Server (IIS)

Internet Information Server, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del paquete de opciones para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Aunque brinda muchas funcionalidades características de estos programas, su principal desventaja consiste en que al formar parte del paquete Windows y no ser multiplataforma, el costo de su uso es igual al del sistema operativo por lo que se necesita la licencia del sistema operativo para hacer uso de este software incluido.

Cherokee

Servidor HTTP Cherokee es un Servidor web libre, multiplataforma, abierto bajo la licencia GPL. Apunta a ser un servidor web bastante rápido que también soporta las funcionalidades más comunes de estos. Está escrito completamente en C, es escalable y puede usarse como un sistema integrado. Cherokee tiene características como: puede ejecutar CGI, PHP y además: PHPCGI o FastCGI. También soporta registro y autenticación de usuarios.

Cherokee puede también realizar redirecciones y soporta la configuración de Servidores Virtuales.

Tomando en cuenta las características de los servidores web antes descritas, se llega a la conclusión de que el más óptimo para ser aplicado al sistema es: Apache.

Se emplea Apache además por:

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, siendo el servidor empleado en el 70% de los sitios web en el mundo, cuando PHP es usado como un módulo de Apache, hereda los permisos del usuario de este. Es el servidor web que por su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto. Corre en una multitud de sistemas operativos, y tiene una alta configurabilidad en la creación y gestión de logs.

1.11 Framework de desarrollo

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Los Framework ayudan en el desarrollo de software. Tienen una estructura definida que se puede reutilizar varias veces la cual ayuda a crear aplicaciones con mayor rapidez. Permite realizar un proceso de mantenimiento efectivo y rápido teniendo en cuenta que haya sido organizado en el proceso de desarrollo. Teniendo en cuenta que para realizar la implementación del trabajo se selecciono PHP como lenguaje de programación, se verán algunos de los framework más importantes que son compatibles con este lenguaje y sus principales características.

Zend Framework

Es simple, no necesita instalación especial, requiere PHP 5 e incorpora el patrón MVC. Se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios web robustos y seguros. Además es Open Source. Fue desarrollado por la empresa Zend quien respalda comercialmente a PHP.

Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services como Amazon, Flickr, Yahoo. Es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.

Sus principales características están basadas en:

- ✓ Presenta una documentación bastante completa y test de una alta calidad.
- ✓ Tiene un buscador compatible con Lucene.
- ✓ Tiene clases muy fuertes para la autenticación y el filtrado de entradas.
- ✓ Clientes para servicios web, incluidos Google Data APIs y Strikelron.
- ✓ Muchas otras clases útiles para hacerlo tan productivo como sea posible.

CakePHP

Es un framework para php que nos permite programar de forma rápida evitándonos escribir código tedioso de tareas muy comunes. Es una estructura de librerías y clases para programar aplicaciones web. Brinda la posibilidad de interactuar con las base de datos, usando ActiveRecord. Incorpora el patrón MVC, compatible con PHP4 y PHP5, URLs amigables, Soporta AJAX e incluye validación.

Sus principales características son:

- ✓ Tiene una arquitectura MVC (modelo, vista y controlador) lo cual permite un trabajo más rápido.
- ✓ Tiene Helpers para AJAX, Javascript, formularios HTML y más.
- ✓ Tiene listas de control de acceso.
- ✓ Componentes para el manejo de la seguridad, sesiones y peticiones.
- ✓ Tiene una caché bastante sensible.
- ✓ Tiene una licencia flexible.
- ✓ Cada línea de código está escrita por el equipo de desarrollo de CakePHP
- ✓ Es un framework orientado a objetos.

Prado

Está basado en componentes con el objetivo de acelerar el desarrollo de aplicaciones web usando PHP 5. El concepto del desarrollo de aplicaciones en Prado es diferente, se utilizan componentes, eventos y propiedades en vez de procedimientos, URL y parámetros. Este Framework combina

especificaciones en un archivo XML, plantillas HTML y una clase PHP. Prado, cuenta con soporte para AJAX, validación, autenticación, plantillas, múltiples servidores de bases de datos.

Sus principales características son:

- ✓ Presenta gran reutilización para los códigos y componentes.
- ✓ Tiene un uso fácil a la hora de crear componentes y manejar los existentes.
- ✓ Presenta un técnica de Caché bastante sensible.
- ✓ Permite la separación del contenido y la presentación.

Zoop Framework

Es un Framework PHP Orientado a Objeto basado en el modelo MVC, sus desarrolladores lo caracterizan por ser rápido, eficiente y fácil, destinado para programadores. Requiere PHP 4.3.10 o superior. Cuenta con soporte e integración con AJAX, validación, sistemas de plantillas inteligentes, creación de PDF, plantillas para el envío de correo electrónico usando SMTP, cuenta con compatibilidad con múltiples servidores de bases de datos.

CodeIgniter

Es un buen framework, utilizado por una gran comunidad de usuarios. Construido para codificadores PHP que necesitan una herramienta de desarrollo fácil para crear aplicaciones web simples y elegantes. Entre sus características podemos encontrar su compatibilidad con PHP 4 y PHP 5, incorpora el modelo MVC, soporte para múltiples servidores de bases de datos, plantillas, validaciones, no requiere instalación, podemos encontrar una librería con un gran número de clases.

Es Open Source, muy pequeño y posee un acceso a sus librerías bien estructurado. Es liviano y fácil de instalar. Fácil de aprender y familiarizarse con el concepto que propone, permite concentrarse en el problema y no en cómo hacerlo evitando tener que volver a escribir una y otra vez lo mismo.

Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la

presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

Este framework automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Está desarrollado completamente con PHP 5 y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de los gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix, Linux, etc. como en plataformas Windows.

Una vez analizado los diferentes framework se acuerda proponer Symfony para la solución del sistema.

Se emplea Symfony además por:

Dentro de la gran cantidad de características y ventajas que podemos encontrar están:

- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- ✓ Preparado para aplicaciones empresariales y es lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Fácil de instalar y configurar en la mayoría de plataformas.
- ✓ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Este framework automatiza la mayoría de elementos comunes de los proyectos web, como por ejemplo:

La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos. La capa de presentación utiliza plantillas y capas que pueden ser creadas por diseñadores HTML sin ningún tipo de conocimiento del framework. Los Helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones. Los formularios incluyen validación automatizada y relleno automático de datos, lo que asegura la obtención de datos correctos y mejora la experiencia de usuario.

El soporte de e-mail incluido y la gestión de APIs permiten a las aplicaciones web interactuar más allá de los navegadores. Los listados son más fáciles de utilizar debido a la paginación automatizada, el filtrado y la ordenación de datos.

1.12 Servidores de bases de datos

Los Database Management System (DBMS), Gestores de bases de datos o servidores de bases de datos como también se le conoce, son software cuya función principal consiste en administrar o gestionar bases de datos donde se encuentran los datos persistentes de cualquier sistema informático. Los DBMS aportan una interfaz que sirve de abstracción entre los usuarios o aplicaciones que acceden a los datos y el almacenaje físico de los datos por lo que hace que a los usuarios o aplicaciones les sea transparente si la base de datos ocupa uno o cien archivos físicos en el disco.

Entre los principales objetivos que persiguen los DBMS se encuentran: Abstracción de la información, seguridad, consistencia, independencia, integridad, control de concurrencia, respaldo, manejo de transacciones y tiempo de respuesta. Las ventajas del uso de un sistema gestor de base de datos son claras aunque no siempre es recomendable usarlos puesto que son software complejos con alto costo de administración y funcionamiento. En caso de aplicaciones pequeñas y sencillas es recomendable usar otra forma de almacenar los datos, sin embargo para aplicaciones medianas o grandes y con alto grado de complejidad, con concurrencia de acceso de usuarios o programas, es muy beneficioso el uso de un gestor de base de datos para manejar la información persistente de estos sistemas.

PostgreSQL

PostgreSQL es un gestor de bases de datos multiplataforma, por su carácter “libre” no es manejado por una sola compañía sino por un grupo de desarrollo global, una comunidad muy amplia y muy colaborativa.

Dentro de sus principales características podemos encontrar:

- ✓ Alta concurrencia: Mediante una buena implementación de un sistema llamado “Acceso Concurrente Multiversión” PostgreSQL permite que un proceso realice una operación de

escritura sobre una tabla mientras otro realiza una consulta sin problemas de bloqueos de información.

- ✓ Amplia variedad de tipos nativos: PostgreSQL brinda soporte para tipos como por ejemplo figuras geométricas, direcciones ip, texto largo ilimitado, entre otros.

Oracle

Oracle es básicamente una potente herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que se vea en empresas muy grandes y multinacionales. En el desarrollo de páginas web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc.

Oracle es un sistema de gestión de base de datos relacional, desarrollado por Oracle Corporation. Se considera como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y su soporte multiplataforma.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.

Dentro de las desventajas que posee este servidor de bases de datos se encuentran que necesita una gran cantidad de recursos de hardware y que es software propietario con licencia muy costosa.

SQL Server

Microsoft SQL Server es un gestor de bases de datos desarrollado por Microsoft entre sus principales características podemos encontrar:

- ✓ Soporte de transacciones.
- ✓ Escalabilidad, estabilidad y seguridad.
- ✓ Soporta procedimientos de almacenado.
- ✓ Incluye también un potente entorno gráfico de administración.
- ✓ Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

- ✓ Además permite administrar información de otros servidores de datos.
- ✓ Incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que se distribuye en forma gratuita en las versiones 2005 y 2008.

Se pueden desarrollar proyectos completos integrando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se vincula la base de datos (Microsoft SQL Server), y el entorno de desarrollo (VBA Access), con la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET. La principal desventaja de este gestor de Bases de Datos es que sólo está disponible para el Sistema Operativo Windows.

MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Es desarrollado en su mayor parte en ANSI C por Mysql AB subsidiaria de Sun Microsystems, como software libre en un esquema de licenciamiento dual. Para software compatible con la licencia GPL este es libre pero si se quiere hacer uso de este en software privativo se debe obtener una licencia especial.

Entre sus principales características podemos encontrar:

- ✓ Soporte a multiplataforma.
- ✓ Procedimientos almacenados.
- ✓ Vistas actualizables.
- ✓ Soporte a VARCHAR.
- ✓ Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación (savepoints) con InnoDB.
- ✓ Soporte para SSL.
- ✓ Soporte completo para Unicode.
- ✓ Corre sobre una amplia gama de plataformas, aunque posee requisitos de hardware en el orden medio-alto.

Realizado un estudio y análisis de estos servidores se concluye que sería muy ventajoso el uso de PostgreSQL por las características antes mencionadas además su opción multiplataforma y el perfecto balance entre necesidades-funcionalidades, también por la facilidad de contar con una amplia comunidad.

Se emplea Postgre además por:

Su gran compatibilidad con PHP y Symfony, los cuales se proponen para emplear en el sistema que se desarrollará. Además soporta un subconjunto de SQL92 mayor que el que soporta MySQL y posee la característica de poder conformar vistas, procedimientos almacenados, herencia de tablas y funciones, soportando estas últimas gran variedad de lenguajes como son: PL/PgSQL (lenguaje propio), C, C++, Java, pI PHP, pI Python entre otros.

Conclusiones

En este capítulo se realiza un estudio profundo sobre la gestión de los recursos humanos y los nuevos enfoques de competencias, además se hace una comparación de las mejores tecnologías y las tendencias del mundo actual y así se llega a las siguientes conclusiones:

- ✓ La construcción de un sistema empleando software libre, contribuye al impulso y al crecimiento del prestigio de la comunidad y contribuye además a la socialización de nuevas soluciones.
- ✓ La metodología a utilizar en el desarrollo del proyecto será Rational Unified Process (RUP), debido a que es robusta y resulta más adaptable para los proyectos a largo plazo que necesitan un desarrollo iterativo capaz de mantener un buen control sobre los cambios. En base a esta decisión se define como lenguaje de modelado UML y para modelar el proceso del negocio se emplea IDEF0.
- ✓ Como lenguaje de programación se propone PHP por ser software libre y fácil de usar además de su robustez, para obtener otras funcionalidades compatibles con el lenguaje de programación, se propone el uso del framework de desarrollo Symfony y como servidor web el Apache.
- ✓ Como necesidad de desarrollar un producto en software libre, además de las características y ventajas que presentan estas herramientas se recomienda como Gestor de Base de Datos PostgreSQL, y como herramienta de modelado se utiliza Visual Paradigm en su versión 6.0

CAPITULO II: CARACTERÍSTICAS DEL SISTEMA

Introducción

El presente capítulo tiene como propósito la realización del modelo de procesos de negocio vinculado al desarrollo del personal por competencias, el cual se propone que sea aplicado a los proyectos productivos de la facultad 3. Se muestran los actores junto a los casos de uso correspondientes, se identifican los requisitos funcionales y no funcionales de la aplicación a desarrollar, permitiendo tener una concepción general del sistema y crear el diagrama de casos de usos del sistema con las relaciones entre los actores y las secuencias de acciones con que interactúan.

2.1 Situación actual. Su análisis crítico

Actualmente, el mayor problema evidente que impide un mejor desarrollo del personal en los proyectos, teniendo en cuenta un enfoque de competencias, es que no existen de manera institucionalizadas las entidades responsables de certificar dichas competencias, no se encuentran oficializados los artefactos necesarios para apoyar este proceso, como es el caso de los perfiles de competencias por roles y el expediente productivo, como agravante no se cuenta con una aplicación informática que controle el desarrollo del personal por competencias que seguramente mejoraría dicho proceso en los proyectos productivos de la Facultad 3.

Por otra parte a la hora de realizar el proceso de desarrollo del personal se hace mediante técnicas como pruebas de conocimiento, sin tener en consideración una verdadera retroalimentación de la trayectoria productiva e investigativa de los candidatos, además de que no hay una manera estandarizada de poder comparar las competencias necesarias para ocupar los roles en los proyectos, con las presentadas por los candidatos, volviéndose sumamente tediosas e imprecisas las actividades para realizar este proceso.

En la UCI, se encuentran diseñando un experimento pedagógico que comenzará su implantación en septiembre del 2009, donde se concibe un ciclo profesional, en el cual se roten las personas por varios roles que deben certificarse. Sin embargo, existen algunas limitantes para la ejecución de la propuesta, como es el constante mantenimiento que se debe lograr del expediente productivo, el que debe reflejar los resultados individuales obtenidos, así como los roles y las competencias certificadas.

Esta situación existente solo puede ser enmendada con la creación de un sistema automatizado que agilice y facilite las funcionalidades implícitas en el proceso de desarrollo del personal por competencias en la Facultad 3, en aras de lograr una mayor organización en las actividades a ejecutar.

2.2 Objeto de automatización

La realización de entrevistas dirigidas a los capacitadores, líderes de proyectos, jefes de formación, jefes de polos y al Vicedecano de producción, permitió diseñar el proceso de desarrollo del personal por competencias. Además, la realización de una tormenta de ideas en el marco del consejo de producción de la facultad 3 y talleres con especialistas que trabajan directamente en el tema a nivel de universidad, permitieron la identificación de los elementos de automatización. (Ver Documento de validación del proceso Desarrollo de Personal) (Rosales, y otros, 2009).

Los principales procesos de automatización lo constituyen las certificaciones vía proyecto y reserva, en las cuáles se presentan un conjunto de actividades que serán analizadas para su implementación.

Certificación vía proyecto:

- ✓ Consultar perfiles por competencias.
- ✓ Solicitud de certificación de roles desde la producción
- ✓ Solicitud de certificación de competencias desde la producción.
- ✓ Actualización del Expediente Productivo por concepto de certificaciones de roles desde la producción.
- ✓ Actualización del Expediente Productivo por concepto de certificaciones de competencias desde la producción.
- ✓ Comunicación de los resultados de las certificaciones de roles desde la producción.
- ✓ Comunicación de los resultados de las certificaciones de competencias desde la producción.
- ✓ Convocatorias del ejercicio de defensa para certificar un rol desde la producción.

Certificación vía reserva:

- ✓ Consultar perfiles por competencias.
- ✓ Solicitud de certificación de roles desde la reserva
- ✓ Solicitud de certificación de competencias desde la reserva

- ✓ Actualización del Expediente Productivo por concepto de certificaciones de roles desde la reserva.
- ✓ Actualización del Expediente Productivo por concepto de certificaciones de competencias desde la reserva.
- ✓ Comunicación de los resultados de las certificaciones de roles desde la reserva.
- ✓ Comunicación de los resultados de las certificaciones de competencias desde la reserva.
- ✓ Convocatorias del ejercicio de defensa para certificar un rol desde la reserva.

2.3 Descripción del proceso del negocio

Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí, que a partir de una o varias entradas obtienen determinadas salidas, con valor añadido para los usuarios. En otras palabras, es la manera en que se hacen las cosas en la organización y se caracterizan por ser observables, medibles, mejorables y repetitivos. A continuación se describen las entradas, actividades, salidas, controles y recursos humanos que intervienen en el proceso de Desarrollo del Personal por competencias que se propone aplicar en los proyectos productivos de la facultad 3.

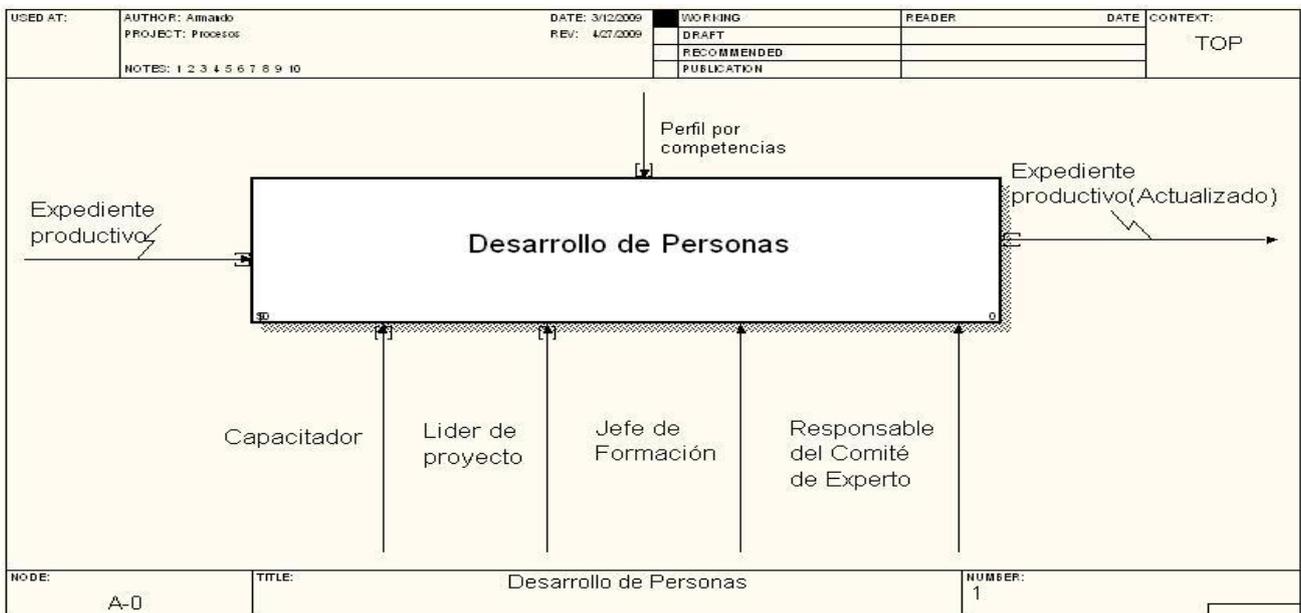


Fig. 2.1 Modelo de proceso del negocio “Desarrollo del Personas”.

Entradas:

- ✓ Expediente Productivo: mantiene actualizado la trayectoria productiva e investigativa del personal evidenciando las competencias y roles que han sido certificados, la participación en eventos, publicaciones y otros datos personales.

Actividades:

El proceso de Desarrollo del Personal, se divide en dos sub-procesos con objetivos concretos relacionados con la certificación de roles y competencias por dos vías alternativas: proyectos productivos y reserva para el personal que no se encuentra vinculado a la producción. (Ver Anexo 2)

Actividades del sub-proceso certificación vía proyecto (Ver Anexo 3):

- ✓ Solicitar certificación: Se realiza la solicitud para certificar una competencia o un rol de una persona determinada.
- ✓ Atender solicitud: Se revisan las evidencias que se presentan al pedir la certificación para determinar si son suficientes para completar el proceso.
- ✓ Analizar tipo de solicitud: Al recibir la solicitud de certificación, se analiza para saber si es de competencia o de rol y en dependencia de cuál sea se continúan con los pasos a seguir.
- ✓ Convocar ejercicio para certificación de rol: Se convoca a un ejercicio para comprobar los conocimientos de la persona en el rol a certificar.
- ✓ Mostrar resultados de la certificación del rol: Una vez que la persona se presenta al ejercicio se muestran los resultados comunicando si se certifica o no.
- ✓ Mostrar resultados de la certificación de competencia: Al analizar las evidencias presentadas, se muestran los resultados comunicándole a las personas interesadas en el proceso si la competencia se certifica o no.

Actividades del sub-proceso certificación vía reserva (Ver Anexo 4):

- ✓ Solicitar certificación: Se realiza la solicitud para certificar una competencia o un rol de una persona determinada.
- ✓ Analizar tipo de solicitud: Al recibir la solicitud de certificación, se analiza para saber si es de competencia o de rol y en dependencia de cuál sea se continúan con los pasos a seguir.
- ✓ Convocar ejercicio para certificación de rol: Se convoca a un ejercicio para comprobar los conocimientos de la persona en el rol a certificar.
- ✓ Mostrar resultados de la certificación del rol: Una vez que la persona se presenta al ejercicio se muestran los resultados comunicando si se certifica o no.

- ✓ Mostrar resultados de la certificación de competencia: Al analizar las evidencias presentadas, se muestran los resultados comunicándole a las personas interesadas en el proceso si la competencia se certifica o no.

Salidas:

- ✓ Expediente Productivo (actualizado): mantiene actualizado la trayectoria productiva e investigativa del personal evidenciando las competencias y roles que han sido certificados, la participación en eventos, publicaciones y otros datos personales.

Control:

- ✓ Perfil por competencias: Contiene las competencias genéricas que necesita una persona para certificarse en un rol.

Recursos humanos

- ✓ Capacitador del proyecto: Inicializa el proceso de Desarrollo de Personal vía proyecto, al hacer la solicitud al líder del mismo sobre las competencias que necesita certificar una persona.
- ✓ Líder de proyecto: Revisa la propuesta hecha por el capacitador y si considera que las evidencias son suficientes hace la solicitud al responsable del comité de expertos para realizar la certificación.
- ✓ Jefe de Formación: Inicializa el proceso de Desarrollo de Personal vía reserva, al hacer la solicitud al responsable del comité de expertos sobre las competencias que necesita certificar una persona.
- ✓ Responsable del Comité de Expertos: Una vez recibida la solicitud de certificación, convoca al comité de expertos para proceder a ejecutar la misma, una vez terminado el proceso informa de los resultados a las personas interesadas.

Para la validación del proceso del negocio expuesto se consultó con especialistas en el tema los cuales trabajan en la creación del nuevo modelo de formación en la universidad y que asesoran varias tesis sobre la gestión de recursos humanos, ellos son: Msc. Edistio Yoel Verdecia Martínez y Dr. José Lavandero García. Se participó en varias conferencias sobre el mejoramiento de los procesos que se lleva a cabo en la UCI, impartidas en el rectorado de la universidad por especialistas en el tema y donde en varias ocasiones se realizaron tormentas de ideas de donde se tomaron las mejores propuestas que conformaron el proceso del negocio actual. La propuesta se presentó en el consejo de producción de la facultad 3, donde se hicieron varias recomendaciones lo

cual ayudó a la mejora considerable del proceso y que propició se obtuviera un modelo refinado del mismo. Están contemplados como reuniones los cuatro cortes de tesis donde se le hicieron útiles aportes y críticas al trabajo en vistas de su mejoramiento. (Ver Anexo 1).

2.4 Modelo del sistema

2.4.1 Requisitos funcionales

Los Requisitos Funcionales son capacidades que el sistema cumplirá. Es una tarea simple enunciada con un solo verbo y se corresponde con futuras opciones, acciones ocultas y condiciones extremas a determinar por el software. (Villanueva, 2007).

Existen varias técnicas para realizar un buen levantamiento de requisitos, según la metodología utilizada se escogen las técnicas que se muestran a continuación:

- ✓ Entrevistas y cuestionario.
- ✓ Comparación con sistemas existentes.
- ✓ Brainstorming (tormenta de ideas).
- ✓ Prototipos.

El método más efectivo para la realización del levantamiento de los requisitos fue el de las entrevistas, donde los analistas de sistemas pudieron presentar las necesidades de los clientes de forma concreta y directa, así como verificar las respuestas que se recibieron por su parte de acuerdo a las preguntas realizadas, teniendo un grado de aceptación óptimo entre el cliente y el sistema que se desea diseñar. También fue de gran ayuda las tormentas de ideas que se realizaron con los especialistas en el tema y con el consejo de producción de la facultad 3.

A continuación se muestran los requisitos funcionales del sistema:

RF1 Gestionar Polo: Es realizado por el Vicedecano de Producción. Se refiere a la gestión de la información que maneja el sistema sobre los polos existentes en la facultad.

RF1.1 Crear Polo: Comprendido dentro de la gestión de polo también lo realiza el Vicedecano de Producción y es la acción de insertar la información referente a un polo que se desea dar de alta.

RF1.2 Modificar Polo: Comprendido dentro de la gestión de polo también lo realiza el Vicedecano de Producción y es la acción de modificar la información referente a un polo que existe registrado con anterioridad en el sistema.

RF1.3 Eliminar Polo: Comprendido dentro de la gestión de polo también lo realiza el Vicedecano de Producción y es la acción de eliminar la información referente a un polo que existe registrado con anterioridad en el sistema.

RF1.4 Consultar Polo: Comprendido dentro de la gestión de polo también lo realiza el Vicedecano de Producción y es la acción de solicitar un listado con la información de los polos registrados en el sistema y dentro de un rango de filtrado.

RF1.5 Imprimir Información: Comprendido dentro de la gestión de polo también lo realiza el Vicedecano de Producción y es la acción de imprimir la información referente a uno o varios polos que se encuentren registrados con anterioridad en el sistema.

RF2 Gestionar Proyecto: Se refiere a la gestión de toda la información que maneja el sistema sobre los proyectos pertenecientes a los polos de la facultad. Es realizado por el jefe de polo.

RF2.1 Crear Proyecto: Realizado por el jefe de polo y contenido dentro de la gestión de proyectos se refiere a la inserción de todos los datos necesarios para la creación de un nuevo proyecto en el sistema.

RF2.2 Modificar Proyecto: Realizado por el jefe de polo y contenido dentro de la gestión de proyectos se refiere a la modificación de todo o parte de los datos registrados con anterioridad de un proyecto en el sistema.

RF2.3 Eliminar Proyecto: Realizado por el jefe de polo y contenido dentro de la gestión de proyectos se refiere a la eliminación de todos los datos de un proyecto existente en el sistema.

RF2.4 Consultar Proyecto: Realizado por el jefe de polo y contenido dentro de la gestión de proyectos se refiere a la consulta de todos los datos de uno o varios proyectos registrados con anterioridad en el sistema.

RF2.5 Imprimir Información: Realizado por el jefe de polo y contenido dentro de la gestión de proyectos se refiere a la impresión de todos los datos de uno o varios proyectos registrados en el sistema.

RF3 Gestionar Permisos: Realizado por el administrador del sistema. Se refiere a la gestión de los permisos que definen roles en el sistema.

RF3.1 Crear Permiso: Es la asignación de un permiso a un usuario del sistema. Está comprendido dentro de la gestión de permisos y es realizado por el administrador del sistema.

RF3.2 Modificar Permiso: Es la modificación de un permiso a un usuario del sistema. Está comprendido dentro de la gestión de permisos y es realizado por el administrador del sistema.

RF3.3 Eliminar Permiso: Es la eliminación de un permiso a un usuario del sistema. Está comprendido dentro de la gestión de permisos y es realizado por el administrador del sistema.

RF3.4 Consultar Permiso: Es la consulta de uno o varios permisos asignados a usuarios del sistema. Está comprendido dentro de la gestión de permisos y es realizado por el administrador del sistema.

RF4 Gestionar noticias: Es la gestión de toda la información relacionada con la publicación o almacenamiento de noticias en el sistema. Es realizado por el editor.

RF4.1 Crear Noticia: Es la inserción de una noticia en el sistema. Esta comprendido dentro de la gestión de noticias y lo realizan usuarios con privilegios de editor.

RF4.2 Modificar Noticia: Es la modificación de una noticia en el sistema. Esta comprendido dentro de la gestión de noticias y lo realizan usuarios con privilegios de editor.

RF4.3 Eliminar Noticia: Es la eliminación de una noticia en el sistema. Esta comprendido dentro de la gestión de noticias y lo realizan usuarios con privilegios de editor.

RF4.4 Consultar Noticia: Es la consulta de una o varias noticias en el sistema. Esta comprendido dentro de la gestión de noticias y lo realizan usuarios con privilegios de editor.

RF5 Gestionar Expediente: Es lo referente a la gestión de toda la información contenida en el expediente de la persona, está conformado por diferentes actividades que se desglosan a continuación y son realizadas por diferentes roles dentro del sistema.

RF5.1 Modificar Expediente: Es la acción de modificar la información registrada en el expediente del estudiante se desglosa a su vez en diferentes actividades repartidas entre diferentes roles en el sistema.

RF5.1.1 Gestionar Publicaciones: Es la acción de gestionar (adicionar, eliminar, modificar, listar) las publicaciones en el expediente de uno o varios estudiantes.

RF5.1.2 Gestionar competencias: Es la acción de gestionar (adicionar, eliminar, modificar, listar) las competencias en el expediente de uno o varios estudiantes.

RF5.1.3 Gestionar roles: Es la acción de gestionar (adicionar, eliminar, modificar, listar) los roles en el expediente de uno o varios estudiantes.

RF5.1.4 Gestionar eventos: Es la acción de gestionar (adicionar, eliminar, modificar, listar) los eventos en el expediente de uno o varios estudiantes.

RF5.1.5 Gestionar anotaciones: Es la acción de gestionar (adicionar, eliminar, modificar, listar) las anotaciones en el expediente de uno o varios estudiantes.

RF5.2 Consultar expediente: Es la acción de consultar la información contenida en el expediente de un estudiante. Es realizado por cualquier usuario autenticado en el sistema.

RF5.3 Imprimir expediente: Es la acción de imprimir la información contenida en el expediente de un estudiante. Es realizado por cualquier usuario autenticado en el sistema.

RF6 Solicitar certificar competencia o rol: Es realizado por el líder de proyecto o por el jefe de formación de polo y es la acción de solicitar la adición de una competencia o rol al expediente productivo de un estudiante.

RF7 Atender solicitud de certificación de rol, competencia: Es realizado por el jefe de la comisión de expertos y es la acción de atender las solicitudes de certificación realizadas.

RF8 Cargar información de persona de LDAP: Es la acción de cargar la información de los estudiantes de la base de datos central de la uci mediante un actor externo que sería el webservice del servidor LDAP. Es iniciada por cualquier usuario autenticado en el sistema.

RF9 Realizar Propuesta de certificación: Es realizada por el capacitador de proyecto y es la acción de realizar una propuesta de solicitud que atenderá el líder de proyecto.

RF 10 Atender Propuesta de certificación: El líder de proyecto atiende las propuestas de solicitud realizadas por el capacitador del proyecto.

RF11 Convocar ejercicio para certificar rol: El jefe de la comisión de expertos convoca a uno o varios estudiantes a realizar un ejercicio para verificar si se le certifica un rol o no.

RF12 Consultar reporte de cambios: EL administrador del sistema consulta el reporte de los cambios realizados sobre tablas controladas de la base de datos.

2.4.2 Requisitos no funcionales

Los requisitos no funcionales son los que le permiten al software ejecutarse mejor, atendiendo a las condiciones mínimas necesarias del hardware y proporcionan mejor interfaz para el usuario. Se especifican a continuación las condiciones que deben cumplirse para ejecutar de forma óptima el sistema a implementar:

Requerimientos de apariencia o interfaz externa:

- ✓ El diseño debe ser sencillo y funcional, fácil de navegar y de rápida adaptación para los usuarios.
- ✓ Paginación de reportes de búsqueda, y listados.

Requerimientos de usabilidad:

- ✓ Para usar el sistema no se necesitará ser especialista en la informática, solo es necesario poseer conocimientos mínimos sobre manejo de aplicaciones Web y para lograr los resultados esperados: tener conocimientos sobre el procedimiento para el control del desarrollo del personal.

- ✓ La aplicación deberá estar optimizada para los principales navegadores existentes en la Universidad (Internet Explorer, Firefox, Opera, Maxthon).
- ✓ Las personas que interactúen con el sistema solo deberán tener conocimientos básicos sobre computación.
- ✓ El sistema debe permitir ser ejecutado sobre diferentes sistemas operativos, ya sea Windows, Linux o Unix.

Requerimientos de rendimiento:

- ✓ El sistema operará con volúmenes medios de información, por tanto, se hacen necesarios tiempos de respuestas cortos, para ello deberá implementar técnicas de procesamiento y respuesta de información empleando recursos propios de la Web 2.0.
- ✓ Se deben optimizar las consultas a realizar en la base de dato para lograr el mismo resultado en el menor tiempo posible.

Requerimientos de soporte:

- ✓ El sistema tendrá implementado un sistema de ayuda que le permitirá al usuario hacer cualquier consulta cuando lo necesite.
- ✓ Se proporcionará un documento adjunto con el software que le permitirá a todas las personas interesadas poder capacitarse en el manejo del sistema.
- ✓ Se requiere un servidor de bases de datos con Postgre que permita procesar vistas y procedimientos almacenados.
- ✓ Se requiere un servidor Web con Apache SSL a partir de la versión 2.0

Requerimientos de portabilidad:

- ✓ El sistema tendrá independencia del Sistema Operativo.
- ✓ Sera independiente del Sistema de Gestión de Bases de Datos.

Requerimientos de seguridad:

- ✓ Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.
- ✓ Solo permitir acceder a materiales de información a los usuarios anónimos.
- ✓ Existencia de roles que establezcan las acciones que pueden realizar los usuarios.
- ✓ Garantizar que la información sea vista únicamente por quien tiene derecho a verla.
- ✓ Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo.

- ✓ Garantizar que el servidor Web opere sobre protocolo seguro.

Confiabilidad:

- ✓ Debe realizarse un trabajo serio con el control de versiones.
- ✓ Funcionalidad:
- ✓ Altos niveles de velocidad en la respuesta del servidor a las peticiones del cliente.
- ✓ Lograr altos niveles de dinamismo y reutilización de páginas en el cliente.

Requerimientos de Software:

- ✓ Apache 2.0 ó posterior
- ✓ Navegador Internet Explorer 5.5. o superior, o compatible con Firefox, Opera, Maxthon.
- ✓ Se utilizará un servidor de Base de Datos con software libre (PostgreSQL).
- ✓ Intérprete de PHP 5
- ✓ Sistema operativo Linux.
- ✓ Servidor Web Apache 2.0.
- ✓ PHP 5.
- ✓ Framework Symfony.
- ✓ Servidor de Base de Datos PostgreSQL 8.3.

2.4.3 Descripción de los actores

Los actores son el conjunto de usuarios, personas, máquinas, software, hardware, etc. que interactúan con el sistema, y que “pueden intercambiar información o pueden ser un recipiente pasivo de información”.

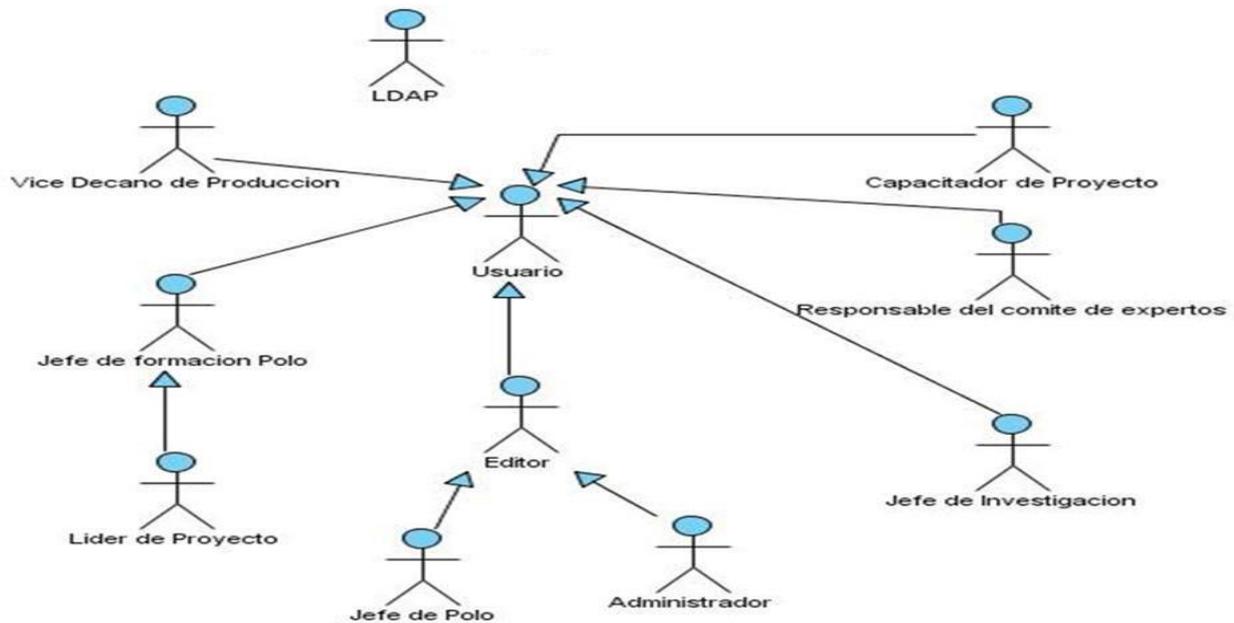


Fig. 2.2 Actores del sistema.

Actor	Justificación
Vicedecano de Producción	Es el encargado de Gestionar información de los polos (Insertar, Actualizar, Eliminar, Listar) en la Facultad.
Jefe de Polo	El jefe de polo se encarga de gestionar la información de los proyectos en su facultad. También puede Publicar noticias y/o información en el sistema.
Líder de Proyecto	Es el encargado de atender las propuestas de solicitud que le haga el capacitador y realizar en base a estas las solicitudes de certificación vía proyecto, al jefe del comité de expertos.
Administrador	El administrador es el encargado de gestionar todo lo referente a los permisos de acceso del sistema además puede editar noticias y/o información en el sistema y consultar el reporte de cambios en la base de datos del sistema.
Editor	Se encarga de editar noticias en el sistema, adicionar, cambiar, eliminar, consultar.

Usuario	El usuario genérico representa al usuario registrado con credenciales del dominio uci.cu y puede consultar el expediente productivo de cualquier estudiante
Capacitador de proyecto	Consulta el expediente de los estudiantes periódicamente y realiza las propuestas de solicitudes de certificación al líder de proyecto.
LDAP	Es el sistema de Servicios Web que ofrece el servidor de la base de datos de personas de la uci, donde almacena la información de todos los estudiantes que será utilizada como apoyo al sistema con el objetivo de optimizar el funcionamiento de la base de datos.
Responsable del Comité de Expertos	Es el encargado de atender las solicitudes de certificación de rol o competencia. Gestionar los cambios y/o actualizaciones en el expediente de las competencias o los roles. Y convocar a un ejercicio para la certificación de un rol.
Jefe de formación del Polo	Realiza las solicitudes de certificación vía reserva al jefe del comité de expertos.
Jefe de investigación	Es el encargado de gestionar la inclusión o cambio en el expediente de los eventos las publicaciones y las anotaciones especiales.

Tabla 2.1 Descripción de los actores del sistema.

En el sistema, el nivel de acceso de los usuarios estará controlado por roles que definen los privilegios de cada persona, cuando esta se conecta al sistema. Se prevé que el acceso al software sea por el dominio uci.cu por lo que el control de la seguridad se verá fortalecido por el que rige este dominio. El software debe ser capaz de reconocer el usuario que se ha autenticado y en base a esto visualizar solo las acciones que están definidas para este rol. Existen precisados una serie de roles que coinciden con los actores en el sistema y para cada uno se delimitan una serie de acciones o casos de uso a realizar.

2.4.4 Diagrama de casos de uso del sistema

Luego de realizar un estudio de los patrones de casos de usos más utilizados y teniendo en cuenta las características específicas del sistema que se desea implementar, se presenta a continuación el diagrama de casos de uso del sistema y seguidamente una breve explicación de cuales patrones fueron empleados para confeccionar el diagrama.

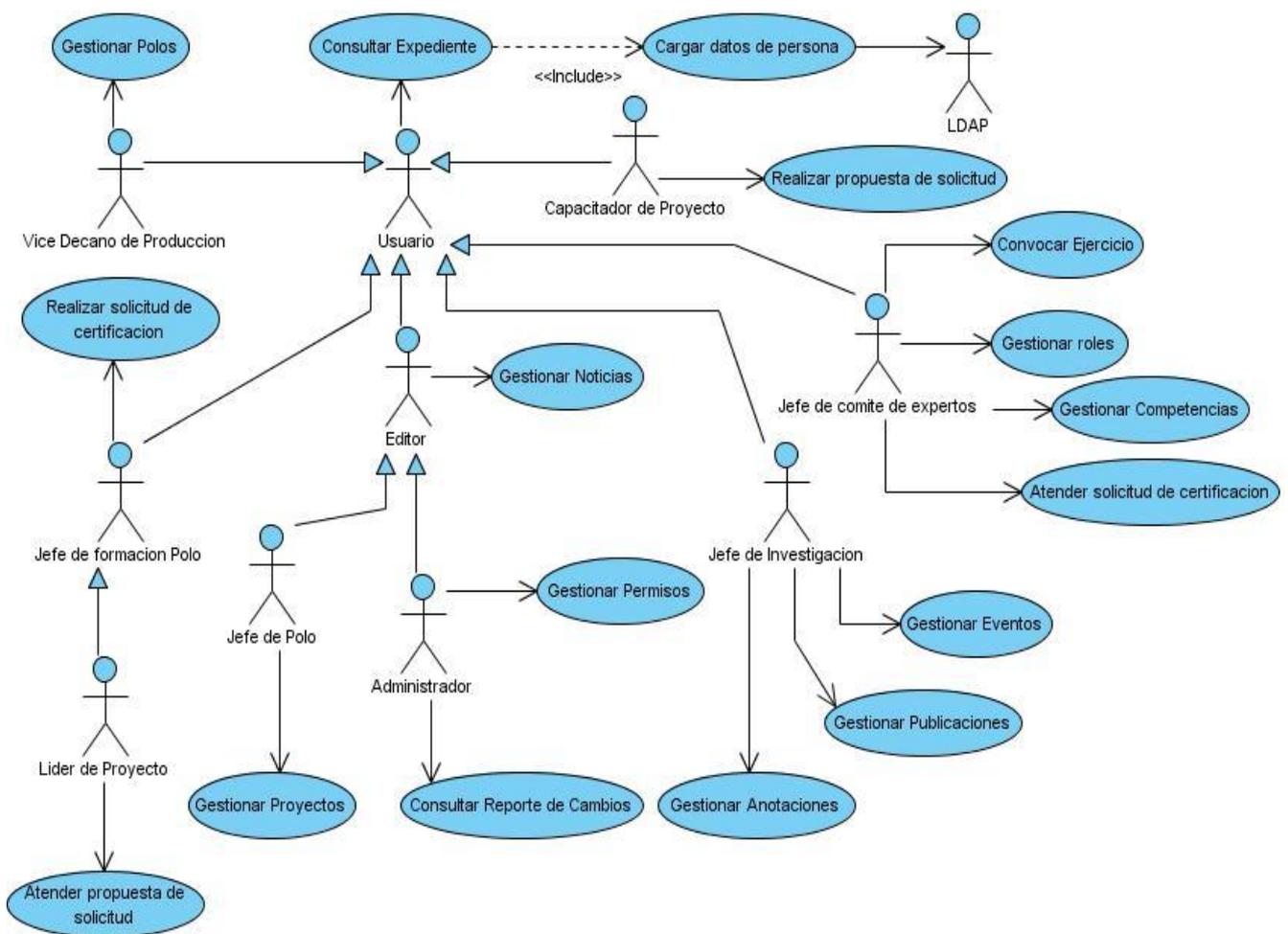


Fig. 2.3 Diagrama de casos de uso del sistema.

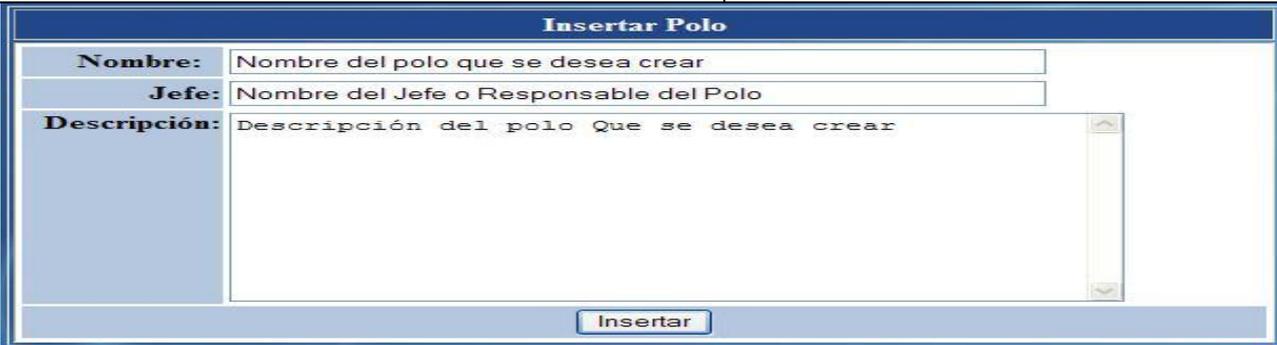
En el diagrama de casos de usos se hace uso de los patrones CRUD para todos los casos de uso que gestionan algún tipo de información, por ejemplo, en el caso de uso “Gestionar roles”. Se emplea el patrón inclusión concreta entre el caso de uso “Consultar expediente” y el caso de uso “Cargar datos de personas” y por último se aplica el patrón actores múltiples: rol común entre el actor usuario-editor, editor-jefe de polo, editor-administrador y jefe de polo-líder del proyecto.

2.4.5 Descripción textual de los casos de uso

Seguidamente se muestran las descripciones de los flujos normales de eventos de dos de los principales casos de uso del módulo Desarrollo del Personal, con sus descripciones básicas; para acceder a la descripción íntegra de todos los casos de usos, ver el documento Modelo de casos de uso del Sistema de Gestión del Talento Humano por competencias laborales en los proyectos productivos de la Facultad 3, módulo Desarrollo de Personal (Rosales, Márquez, et al., 2008).

CU1: Gestionar Polos:

Plantilla de descripción CU 1	
Nombre:	Gestionar Polos.
Actor:	Vice Decano de Producción.
Resumen:	El Vice decano de producción se encarga de realizar las operaciones correspondientes a los polos en el sistema (Buscar, Insertar, Eliminar, Actualizar sus datos así como Listar).
Referencias:	RF1
Pre-condiciones:	El Actor debe estar autenticado en el sistema.
Pos-condiciones:	Se buscan, listan, insertan, eliminan Polos además de actualizar sus datos.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice Decano de producción selecciona la opción de Insertar o Eliminar o Buscar o Listar o Actualizar los datos de un Polo.	2. El sistema ejecuta alguna de los siguientes escenarios: 2.1 Si decide Insertar un Polo ir al escenario "Insertar". 2.2 Si decide eliminar un Polo ir al escenario "Eliminar". 2.3 Si decide listar mediante una búsqueda un grupo de Polos ir al escenario "Listar". 2.4 Si decide buscar un Polo ir al escenario "Buscar". 2.5 Si decide actualizar los datos de un Polo ir a escenario "Actualizar".

Escenario 1 Insertar Polos	
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el actor desea acceder en el menú Polo a la interfaz Insertar.	2. El sistema visualiza la interfaz.
3. El actor entra los datos necesarios: Nombre, Responsable, Descripción, Contacto, etc. 3.1 Presiona el botón Insertar.	4. El sistema inserta el(los) usuario(s) en la Base de datos.
	5. EL sistema envía un mensaje de confirmación. Culminando así el caso de uso.
Curso Alternativo de Eventos	
Acciones del Actor	Respuesta del Sistema
3.2 El actor presiona el botón Cancelar.	4.1. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.
	
Escenario 2 Eliminar Polos	
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El actor selecciona Eliminar.	2. El sistema visualiza la interfaz.
3. El actor entra el(los) Polo(s) a eliminar. 3.1 Presiona el botón Eliminar.	4. El sistema elimina el(los) Polo(s) de la base de datos.
	5. Envía mensaje de confirmación, culminando así el caso de uso.
Curso Alternativo de Eventos	
Acciones del Actor	Respuesta del Sistema
1.1 El actor decide en la misma interfaz de Eliminar,	2.1 El sistema visualiza la interfaz de la

realizar una búsqueda del polo a eliminar.	búsqueda.
3.2 El actor entra los datos pertinentes a la búsqueda. 3.2.1 Presiona el botón Buscar.	4.1 El sistema muestra los resultados
5. El administrador selecciona el(los) polos(s) a eliminar. 5.1 Presiona el botón Eliminar.	6. El sistema los elimina de la base de datos y envía mensaje de confirmación, culminando así el caso de uso.
3.2.2 Presiona el botón cancelar	4.2 El sistema cierra la interfaz culminando así el caso de uso.

Eliminar Polo

Seleccione un Polo:

Información del Polo Seleccionado

Nombre:

Jefe:

Descripción:

Escenario 3 Listar Polos

Curso Normal de Eventos

Acciones del Actor	Respuesta del Sistema
1. El actor selecciona Listar.	2. El sistema visualiza en la interfaz la opción de búsqueda para los Polos que el actor desea listar.
3. El actor entra los datos de la búsqueda. 3.1 Presiona el botón listar.	4. El sistema lista los resultados, culminando así el caso de uso.

Curso Alterno de Eventos

Acciones del Actor	Respuesta del Sistema
3.2. El actor presiona el botón Cancelar.	4.1. El sistema cancela la operación y cierra la interfaz, culminando así el caso de uso.

Escenario 4 Actualizar Datos

Curso Normal de Eventos

Acciones del Actor	Respuesta del Sistema
1. El actor selecciona Actualizar Datos.	2. El sistema visualiza la interfaz.
3. El actor entra el Polo al cual desea actualizarle los datos.	4. El sistema visualiza una interfaz con los datos del usuario en cuestión.
5. El Administrador entra los datos a actualizar. 5.1 Presiona el botón Actualizar Datos.	6. El sistema actualiza los datos del Polo en la base de datos y envía mensaje de confirmación, culminando así el caso de uso.

Curso Alterno de Eventos

Acciones del Actor	Respuesta del Sistema
1.1 El actor decide en la misma interfaz de Actualizar Datos, realizar una búsqueda del Polo al que va a actualizar sus datos. 1.1.1 Presiona el botón Búsqueda.	2.1 El sistema visualiza la interfaz de búsqueda.

3.1 El actor entra los datos de la búsqueda 3.1.1 Presiona el botón buscar.	4.1 El sistema muestra los resultados.
5.2 El actor selecciona el polo al cual va a actualizarle los datos 5.2.1 Presiona el botón Aceptar.	6.1 El sistema muestra los datos pertenecientes al polo que se desea actualizar.
7. El actor entra los nuevos datos a actualizar. 7.1 Presiona el botón Actualizar.	8. El sistema actualiza los datos del polo en la base de datos y envía mensaje de confirmación, culminando así el caso de uso.

CU2: Realizar Solicitud de Certificación:

Plantilla de descripción CU 2	
Nombre:	Realizar Solicitud de Certificación
Actor:	Jefe de Formación de Polo, Líder de Proyecto
Resumen:	El jefe de formación del Polo realiza una solicitud al jefe del comité de expertos de acreditar competencias o roles a un estudiante de la reserva; al igual que el líder de proyecto para un estudiante de su proyecto.
Referencia:	RF6
Pre-condiciones:	El Jefe de Formación de Polo o Líder de Proyecto según sea el caso debe estar autenticado en el sistema.
Pos-condiciones:	Se registra una solicitud de acreditación para un estudiante tanto del polo como de un proyecto.

Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1-El actor selecciona la opción: "Realizar Solicitud de Certificación"	2-El sistema muestra la interfaz correspondiente.
3-El actor completa los campos de datos para la solicitud. 3.1- Presiona el botón: "Aceptar"	4- El sistema verifica que los datos estén correctos.
	5-El sistema inserta en la base de datos la nueva solicitud, muestra mensaje de confirmación y culmina el caso de uso.
Curso Alternativo de Eventos	
Acciones del Actor	Respuesta del Sistema
3.2- Presiona el botón: "Cancelar"	4.1- El sistema cierra la interfaz correspondiente y culmina el caso de uso.
	5.1- El sistema encuentra errores en los datos.
	6- Muestra un mensaje informando el error detectado y permite rectificarlo.
Prioridad:	Crítico

CU3: Atender Solicitud de Certificación:

Plantilla de descripción CU 3	
Nombre:	Atender Solicitud de Certificación

Actor:	Jefe del comité de expertos
Resumen:	El Jefe del comité de expertos atiende una solicitud de acreditación, se da por enterado de la solicitud y la define como aceptada, en proceso, pospuesta, o denegada.
Referencia:	RF7
Pre-condiciones:	El Jefe del comité de expertos debe estar autenticado en el sistema.
Pos-condiciones:	Se resuelve una solicitud de acreditación para un estudiante tanto del polo como de un proyecto.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1-El Jefe del comité de expertos selecciona la opción: "Atender Solicitud de Certificación"	2-El sistema muestra la interfaz correspondiente
	3-Carga una lista de todas las solicitudes que no han sido atendidas
4-El Jefe del comité de expertos selecciona la solicitud que desea atender.	5-El sistema carga todos los datos específicos de la solicitud seleccionada.
6-El Jefe del comité de expertos selecciona un nuevo estado para la solicitud mostrada. 6.1- Presiona el botón: "Aceptar"	7-El sistema registra el nuevo estado de la solicitud, muestra mensaje de confirmación y termina el caso de uso.
Curso Alterno de Eventos	
Acciones del Actor	Respuesta del Sistema
6.2- Presiona el botón: "Cancelar"	8- El sistema cierra la interfaz y muestra nuevamente la lista de solicitudes que no han sido atendidas.
Prioridad:	Crítico

Atender Solicitud de Acreditación				
Solicitudes sin atender en el sistema				
id	Nombre	Rol	Competencia	Ampliar
00	Miguel Angel Marquez	Analista	Saber Hacer Levantamiento de requisitos	Ver
01	Armando Alezander	Programador	Programar enPython	Ver
02	Julio Antonio	Diseñador	Utilizar Visual Paradigm	Ver
Datos de la solicitud seleccionada				
Solapin:	Solapin del estudiante que s ele hace la solicitud			
Rol:	<input type="text"/>	Competencia:	<input type="text"/>	
Descripción:	Descripción de la solicitud <input type="text"/>			
Atender:	<input type="radio"/> Aceptada <input type="radio"/> Denegada <input type="radio"/> Pospuesta			<input type="button" value="Aceptar"/>

CU4: Convocar ejercicio:

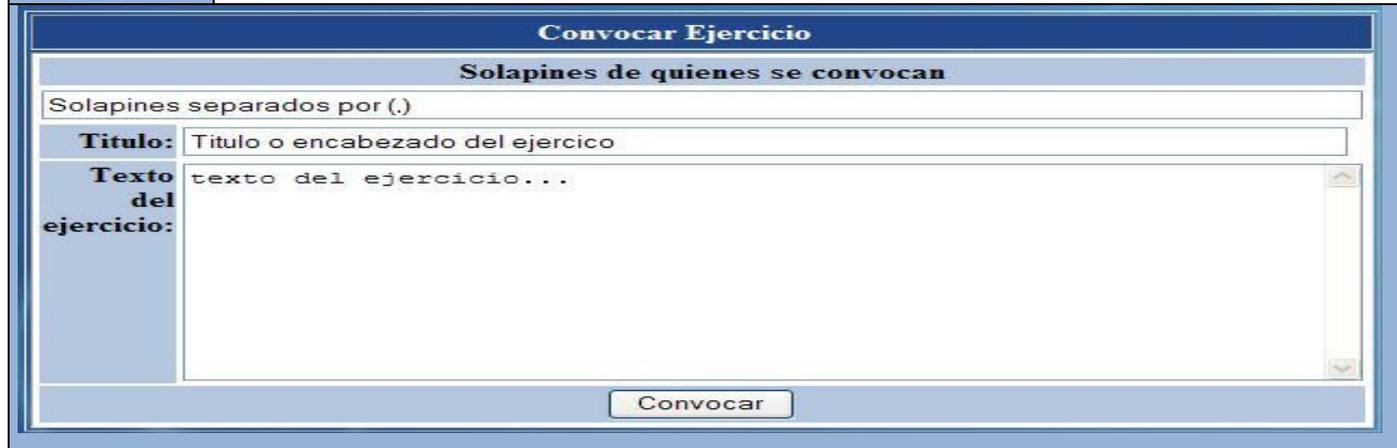
Plantilla de descripción CU 4	
Nombre:	Convocar Ejercicio
Actor:	Jefe de comité de expertos
Resumen:	El jefe del comité de expertos convoca a un ejercicio para certificar un roll solicitado a uno o varios estudiantes específicos
Referencia:	RF11
Pre-condiciones:	El Jefe de comité de expertos debe estar autenticado en el sistema.
Pos-condiciones:	Queda registrado un ejercicio a realizarse por uno o varios estudiantes.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1-El actor selecciona la opción: "Convocar ejercicio".	2- El sistema muestra la interfaz correspondiente.
3- El actor completa los campos de los estudiantes que se convocan al ejercicio. 3.1- Presiona el botón: "Aceptar"	4-El sistema verifica que los datos estén correctos.
	5- Carga la interfaz correspondiente a la inserción del ejercicio

6-El actor completa los campos para insertar un ejercicio. 6.1- Presiona el botón: "Aceptar"	7-El sistema verifica que los datos estén correctos.
	8- Inserta los datos en la base de datos, muestra mensaje de confirmación y culmina el caso de uso.

Curso Alternativo de Eventos

Acciones del Actor	Respuesta del Sistema
3.2 Presiona el botón cancelar	4.1- El sistema cierra la interfaz re direccionando hacia el nivel inferior
	4.2- El sistema detecta errores en los datos
	5.1- Muestra un mensaje informando del error.
	7.1- El sistema detecta errores en los datos
	8.1- Muestra un mensaje informando del error.

Prioridad: Crítico



CU5: Consultar Reporte de Cambios

Plantilla de descripción CU 5	
Nombre:	Consultar Reporte de Cambios
Actor:	Administrador
Resumen:	El Administrador consulta el reporte de los cambios realizados sobre tablas controladas de la base de datos.
Referencia:	RF12
Pre-condiciones:	El Administrador debe estar autenticado en el sistema.

Pos-condiciones:	Se muestra el reporte solicitado de los cambios efectuados en tablas controladas de la base de datos.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1- El administrador escoge la opción de consultar reporte de cambios	2- El sistema muestra una lista con los cambios recientes realizados en las tablas controladas de la base de datos.
3- El administrador selecciona el cambio que quiere ampliar.	4- El sistema carga la información ampliada del cambio seleccionado.
5- El administrador selecciona la opción aceptar	6- El sistema cierra la información ampliada y regresa a la lista de los cambios recientes.
7- El administrador selecciona la opción salir.	8- El sistema cierra la interfaz y termina el caso de uso.
Curso Alternativo de Eventos	
Acciones del Actor	Respuesta del Sistema
3.1- El administrador selecciona la opción de mostrar cambios más antiguos.	4.1 el sistema carga una nueva lista con los cambios más antiguos en la base de datos
Prioridad:	Crítico

Reporte de Cambios

Cambios realizados en el sistema

Tabla	Cambió(solapin)	Fecha	Ampliar
Polo	51277	25/1/2009	Ver
Proyecto	24151	14/2/2009	Ver
Expediente	12356	1/5/2009	Ver

Datos del cambio seleccionado

Id de Tupla:	<input type="text" value="identificador de la tupla cambiada"/>
Nombre:	<input type="text" value="Nombre de quien realiza el cambio"/>
Descripción del cambio realizado :	<div style="border: 1px solid gray; padding: 2px; min-height: 100px;"> Descripción y consulta SQL que se realizo sobre la base de datos y que esta relacionada al cambio en cuestion </div>

CU6: Consultar Expediente:

Plantilla de descripción CU 6	
Nombre:	Consultar Expediente
Actor:	Usuario
Resumen:	El usuario consulta el expediente productivo de otro estudiante cualquiera.
Referencia:	RF5.2
Pre-condiciones:	El usuario debe estar autenticado en el sistema bajo dominio uci.cu.
Pos-condiciones:	Se muestran los datos todos los datos del expediente solicitado.
Curso Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1-El usuario selecciona la opción de consultar expediente.	2- Muestra la interfaz correspondiente.
	3- Carga un filtro para estudiantes. Por nombre, por solapín, ci, etc.
4- Inserta los parámetros de filtrado	5- Filtra los estudiantes por la información proporcionada
	6- muestra una lista generada con las coincidencias totales y parciales con los parámetros de filtrado
7- selecciona el estudiante buscado, en la lista	8- muestra el expediente productivo del estudiante
9- selecciona la opción aceptar	10- cierra la interfaz y regresa al filtro para seguir buscando.
11- selecciona la opción salir	12- cierra la interfaz y termina el caso de uso.
Curso Alternativo de Eventos	
Acciones del Actor	Respuesta del Sistema
7.1 selecciona la opción salir	8.1- cierra la interfaz y termina el caso de uso.
	6.1- no encuentra coincidencias ni totales ni parciales
	7.2- muestra un mensaje informando que deben cambiarse los parámetros de filtrado
Prioridad:	Crítico

Consultar Expediente			
Buscar Estudiante			
miguel		Buscar	
Resultados de la Búsqueda			
Nombre	Solapin	Polo	Expediente
Miguel Angel Marquez Figueroa	51277	Gestion Juridica	Ver
Miguel Alfredo Moya	24151	Gestion Turistica	Ver
Alfredo Miguel Garcia Orozco	12356	Informatización	Ver

Para acceder a las restantes descripciones de los casos de usos ver el informe: Descripción textual de casos de usos del Módulo Desarrollo del Personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3. (Rosales & Márquez, 2009).

2.5 Seguridad que ofrece el framework para el sistema:

El mecanismo de seguridad puede ser entendido como un filtro, por el que debe pasar cada petición antes de ejecutar la acción. Según las comprobaciones realizadas en el filtro, se puede modificar el procesamiento de la petición, Symfony extiende esta idea a clases de filtros. Se puede especificar cualquier número de clases de filtros a ser ejecutadas antes de que se procese la respuesta y hacerlo de forma sistemática para todas las peticiones. Se pueden entender los filtros como una forma de empaquetar cierto código de forma similar a `preExecute()` y `postExecute()`, pero a un nivel superior, es decir, para toda una aplicación en lugar de para todo un módulo. [Potencier. Zaninoto, 2008]

Symfony trata la seguridad del sistema para diferentes situaciones que se explican a continuación:
 Seguridad de la acción: La posibilidad de ejecutar una acción puede ser restringida a usuarios con ciertos privilegios. Las herramientas proporcionadas por Symfony para este propósito permiten la creación de aplicaciones seguras, en las que los usuarios necesitan estar autenticados antes de acceder a alguna característica o a partes de la aplicación. Añadir esta seguridad a la aplicación requiere dos pasos: declarar los requerimientos de seguridad para cada acción y autenticar a los usuarios con privilegios para que puedan acceder estas acciones seguras.

Restricciones de acceso: Antes de ser ejecutada, cada acción pasa por un filtro especial que verifica si el usuario actual tiene privilegios de acceder a la acción requerida. En Symfony, los privilegios están compuestos por dos partes:

- ✓ Las acciones seguras requieren que los usuarios estén autenticados.
- ✓ Las credenciales son privilegios de seguridad agrupados bajo un nombre y que permiten organizar la seguridad en grupos.

Para restringir el acceso a una acción se crea y se edita un archivo de configuración YAML llamado `security.yml` en el directorio `config/` del módulo. En este archivo, se pueden especificar los requerimientos de seguridad que los usuarios deberán satisfacer para cada acción o para todas las acciones.

Las acciones no incluyen restricciones de seguridad por defecto, así que cuando no existe el archivo `security.yml` o no se indica ninguna acción en ese archivo, todas las acciones son accesibles por todos los usuarios. Si existe un archivo `security.yml`, Symfony busca por el nombre de la acción y si existe, verifica que se satisfagan los requerimientos de seguridad.

Lo que sucede cuando un usuario trata de acceder una acción restringida depende de sus credenciales:

- ✓ Si el usuario está autenticado y tiene las credenciales apropiadas, entonces la acción se ejecuta.
- ✓ Si el usuario no está autenticado, es redireccionado a la acción de login.
- ✓ Si el usuario está autenticado, pero no posee las credenciales apropiadas, será redirigido a la acción segura por defecto.

Las páginas login y secure son bastante simples, por lo que seguramente será necesario personalizarlas. Se puede configurar que acciones se ejecutan en caso de no disponer de suficientes privilegios en el archivo `settings.yml` de la aplicación cambiando el valor de las propiedades.

Otorgando acceso: Para obtener acceso a áreas restringidas, los usuarios necesitan estar autenticados o poseer ciertas credenciales. Se pueden extender los privilegios del usuario mediante llamadas a métodos del objeto `sfUser`. El estado autenticado se establece con el método `setAuthenticated()` y se puede comprobar con el método `isAuthenticated()`. Las credenciales son un poco más complejas de tratar, ya que se pueden verificar, agregar, quitar y borrar.

Si el usuario tiene la credencial "parámetro", entonces ese usuario podrá acceder a las acciones para las cuales el archivo security.yml requiere esa credencial. Las credenciales se pueden utilizar también para mostrar contenido autenticado en una plantilla. Para el estado de autenticación, las credenciales normalmente se dan a los usuarios durante el proceso de login. Este es el motivo por el que el objeto sfUser normalmente se extiende para añadir métodos de login y de logout, de forma que se pueda establecer el estado de seguridad del usuario de forma centralizada.

Métodos de validación y manejo de errores: Symfony incluye un sistema de validación, utilizando métodos de la clase acción. Cuando un usuario hace una petición a miAccion, Symfony siempre busca primero un método llamado validateMiAccion (). Si lo encuentra, Symfony ejecuta ese método. El valor de retorno de esta validación determina el siguiente método que se ejecuta: si devuelve true, entonces se ejecuta el método executeMiAccion (); en otro caso, se ejecuta handleErrorMiAccion (). En el caso de que handleErrorMiAccion () no exista, Symfony busca un método genérico llamado handleError (). Si tampoco existe, simplemente devuelve el valor sfView::ERROR para producir la plantilla miAccionError.php.

La clave para un correcto funcionamiento de la validación es respetar la convención de nombres para los métodos de la acción:

- ✓ validateNombreAccion () es el método de validación, que devuelve true o false. Se trata del primer método buscado cuando se solicita la acción NombreAccion. Si no existe, la acción se ejecuta directamente.
- ✓ handleErrorNombreAccion () es el método llamado cuando el método de validación falla. Si no existe, entonces se muestra la plantilla Error.
- ✓ executeNombreAccion es el método de la acción. Debe existir para todas las acciones.

Se puede incluir cualquier código en el método validate (). La única condición es que devuelva un valor true o false. Como es un método de la clase sfActions, tiene acceso a los objetos sfRequest y sfUser, que pueden ser realmente útiles para validación de los datos de la petición y del contexto.

Se pueden utilizar este mecanismo para implementar la validación de los formularios (esto es, controlar los valores introducidos por el usuario en un formulario antes de procesarlo), pero se trata de una tarea muy repetitiva para la que Symfony proporciona herramientas automatizadas.

Conclusiones

En el capítulo se expone el proceso del negocio para el módulo de Desarrollo de Personal que se propone aplicar a los proyectos productivos de la facultad 3, obteniéndose una lista de funcionalidades que debe tener el sistema. Estas funcionalidades fueron representadas mediante un diagrama de casos de uso junto a la descripción de las acciones que realizan los actores a través del análisis del proceso del negocio. Por último se refleja la manera en que se trata la seguridad empleando el framework de desarrollo propuesto.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

El presente capítulo refleja la propuesta del análisis y el diseño del módulo Desarrollo del Personal por competencias, utilizando como metodología de desarrollo de software RUP y la herramienta Visual Paradigm. Los artefactos que se obtienen son los diagramas de clases del análisis y diseño, así como los diagramas de secuencia para el diseño de todos los casos de uso propuestos en el sistema. Además, la realización del modelo de datos es un artefacto fundamental que se presenta en este capítulo.

3.1 Modelo de análisis

El modelo de análisis contiene los resultados del análisis de los casos de uso. Es donde se obtienen las clases pertenecientes a este flujo. En adición a esto, en el modelo de análisis se refinan los requisitos, no se toma en cuenta el lenguaje de programación a usar en la construcción, ni se define la plataforma en la que se ejecutará la aplicación ya que el objetivo principal del análisis es garantizar una total comprensión de los requisitos del software y no precisar cómo se implementará la solución propuesta.

Para la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso y las relaciones entre ellas y con esta información se desarrollan los Diagramas de Clases del Análisis para el sistema objeto de estudio.

Las clases van a representar abstracciones de conceptos, en las cuales deben incluirse atributos y operaciones a un nivel alto, por lo que no se incluye el paso de parámetros ni el tipo de datos.

La realización de los casos de uso del análisis son los diagramas que expresan el comportamiento de las clases del análisis, presentando una trazabilidad de uno a uno. En la realización de casos de uso se expresan los diagramas de interacción donde se comunican las clases, y representa la vista dinámica del sistema. Estos diagramas pueden ser de dos tipos: diagrama de secuencia y diagramas de colaboración.

RUP propone clasificar a las clases en:

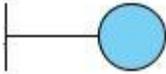
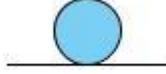
	<p>Clases Interfaz: Modelan la interacción entre el sistema y sus actores.</p>
	<p>Clases Controladoras: Coordinan la realización de uno o varios casos de uso, coordinando la actividades de los objetos que implementan la funcionalidad del caso de uso.</p>
	<p>Clases Entidad: Modelan información que posee larga vida y que es a menudo persistente.</p>

Tabla 3.1 Clases del análisis

3.1.2 Clases del análisis

Los diagramas de clases de análisis, representan los conceptos en un dominio del problema, con una vista estática de los mismos, a través de las diferentes relaciones que se establecen entre las clases. A continuación se muestra algunos de los diagramas de clases realizados, que sirvieron como punto de partida para la modelación del diseño.

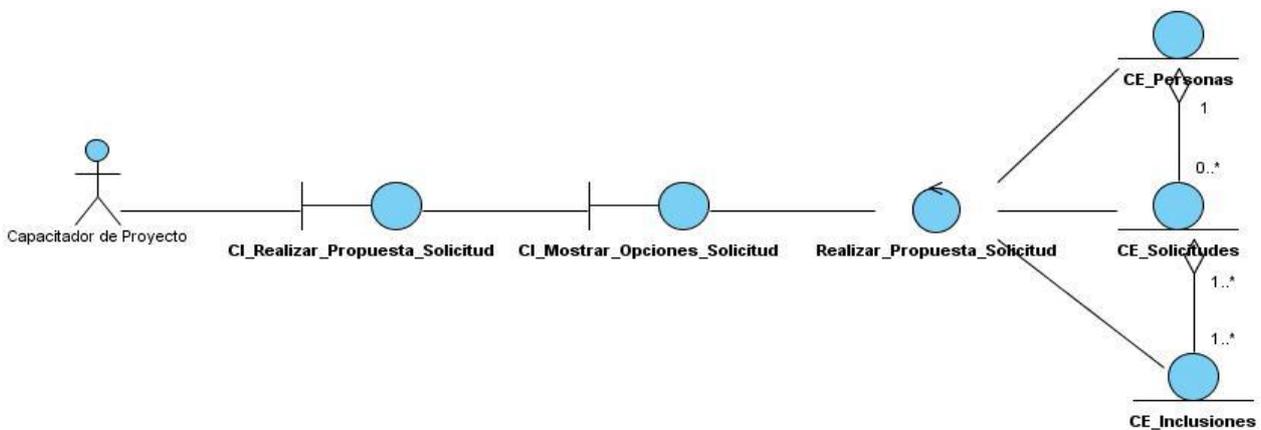


Fig. 3. 1 Diagrama de clases: Realizar Propuesta de solicitud de Certificación.

El diagrama muestra la relación que existe entre las clases, cuando el capacitador del proyecto hace una solicitud para certificar una competencia o un rol de uno de los integrantes de su proyecto.

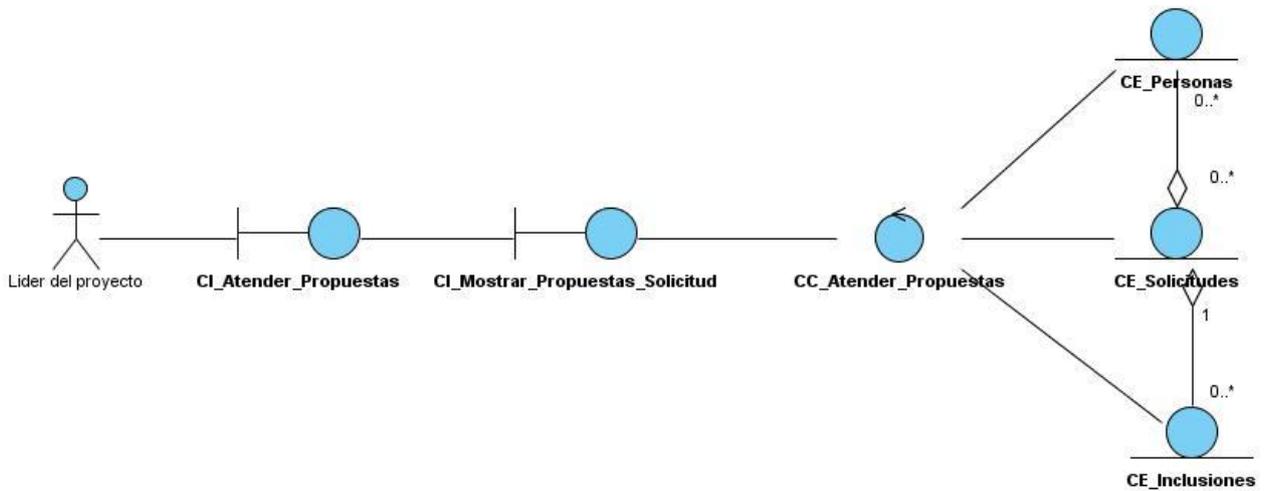


Fig. 3. 2 Diagrama de clases: Atender Propuesta de Solicitud.

El diagrama muestra la relación que existe entre las clases, cuando el líder del proyecto recibe una solicitud para certificar una competencia o un rol de uno de los integrantes de su proyecto.

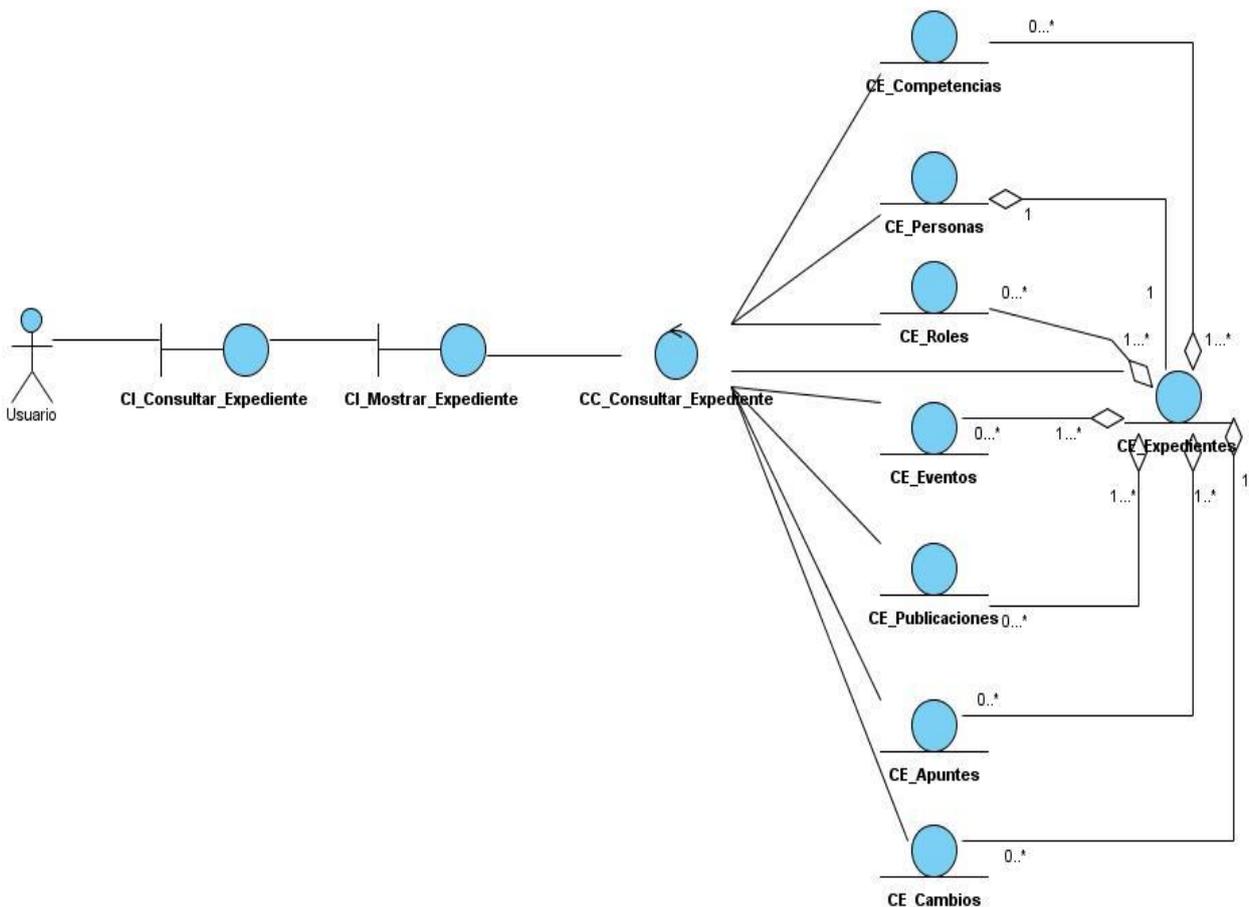


Fig. 3. 3 Diagrama de clases: Consultar Expediente Productivo.

El diagrama muestra la relación entre las clases, cuando los usuarios desean consultar el expediente productivo.

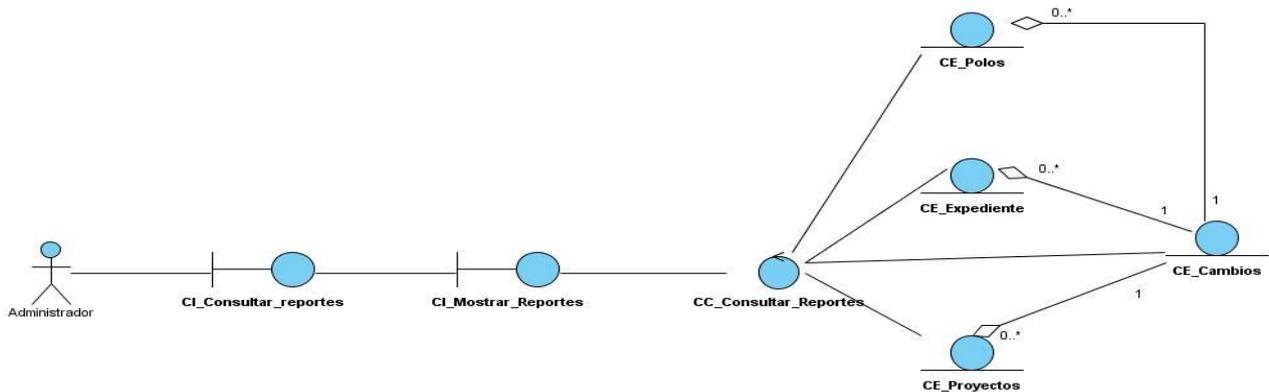


Fig. 3. 4 Diagrama de clases: Consultar Reportes de Cambio.

El diagrama muestra la relación existente entre las clases, cuando el administrador consulta el reporte de cambios que ocurren en el sistema.

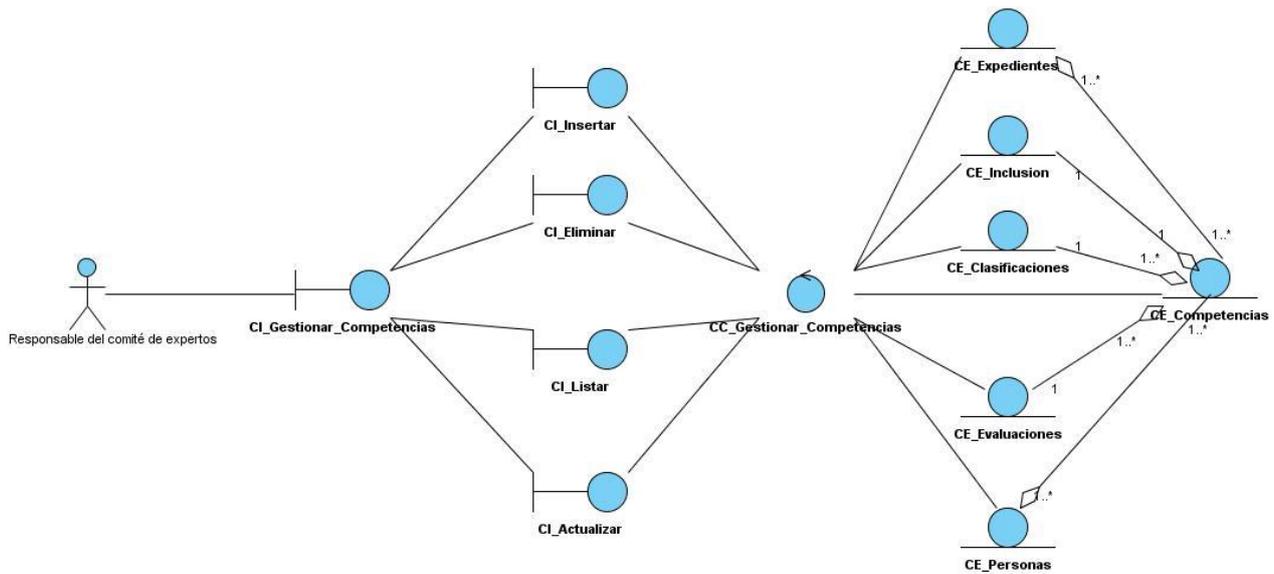


Fig. 3. 5 Diagrama de clases: Gestionar Competencias.

El diagrama muestra la relación entre las clases, cuando el responsable del comité de expertos realiza la gestión de las competencias, ya sea insertando, eliminando, listando o actualizando una o varias competencias.

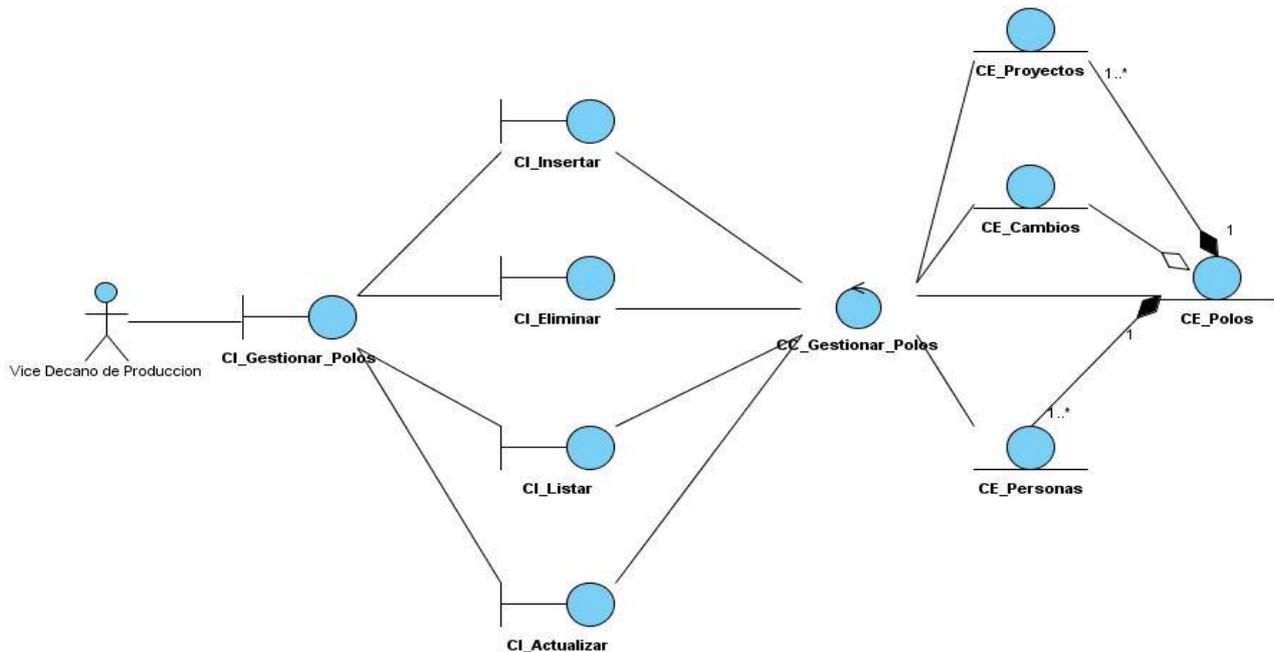


Fig. 3. 6 Diagrama de clases: Gestionar Roles.

El diagrama muestra la relación entre las clases, cuando el responsable del comité de expertos realiza la gestión de los roles, ya sea que las inserte, elimine, modifique o las liste.

Para acceder a los restantes diagramas de clases del análisis ver el informe: Diagramas de clases del análisis del Módulo Desarrollo del Personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3. (Rosales & Márquez, 2009).

3.2 Modelo de Diseño

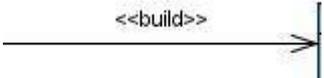
El modelo de diseño, es un modelo de objetos que aporta una comprensión en profundidad de los aspectos relacionados con los requisitos no funcionales y restricciones de los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución, concurrencias y tecnologías de interfaz de usuario. Además, ayuda a descomponer los trabajos de implementación en partes, que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia. [Jacobson, 2000].

En el diseño, al mismo tiempo que se modela el sistema se le da forma a la arquitectura, soportando así todos los requisitos, reglas y restricciones. Un artefacto importante que constituye un punto de partida para modelar el diseño, es el modelo de análisis. RUP, plantea que no es

obligatoria la confección del modelo de análisis, pero esta resulta de gran ayuda en la creación del diseño. [Booch, 2004].

3.2.1 Clases del diseño

Los diagramas de clases de diseño muestran, un conjunto de interfaces, colaboraciones y sus relaciones. Se utilizan para modelar principalmente la vista de diseño estática de un sistema. Los diagramas de clases, son importantes, no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa o inversa.

	<p>Client Page: Una instancia de Página Cliente es una página Web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script que son accesibles para páginas cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.</p>
	<p>Form: Colección de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields).</p>
	<p>Server Page: Representa la página Web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos también localizados en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página. Esta clase sólo puede tener relaciones con objetos en el servidor.</p>
	<p>Build: Representa una asociación especial que relaciona las páginas cliente con las páginas servidor, de forma general se expresa como que las páginas que se encuentran en el servidor, construyen las páginas en el cliente. Debe ser una relación direccional, donde una página servidor puede</p>

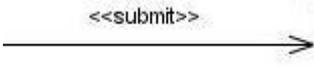
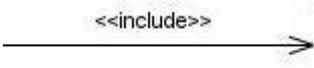
	construir una o más páginas cliente.
	Submit: Es la relación que se crea siempre entre una página servidor y un formulario, a través de esta relación el formulario manda los valores de sus campos al servidor, para ser procesados por la página servidor.
	Include: Relación que se establece entre dos componentes en el que el primer componente contiene al segundo. Vincula una página template con las páginas que referencia.

Tabla 3.2 Clases del diseño

3.2.2. Arquitectura del sistema

Symfony como framework de desarrollo que permite optimizar el trabajo de aplicaciones web, tiene implementados patrones para lograr obtener un mejor diseño de cualquier sistema que se esté desarrollando. En el caso de la arquitectura él define el patrón Modelo Vista Controlador (MVC) implementando todas sus ventajas.

Con el objetivo de eliminar importantes problemas de escalabilidad, disponibilidad, seguridad e integración y de aislar el negocio de la interfaz del usuario y del manejo de la persistencia se propone para la implementación de la aplicación el estilo arquitectura en 3 capas:

Capa de presentación:

Es la que interactúa directamente con el usuario, captura la información entrada por éste (realiza un filtrado previo para comprobar que no hay errores de formato) y hace las peticiones a la capa inferior mostrando al usuario la respuesta proveniente de ésta. Únicamente se comunica con la capa de negocio.

Capa de negocio:

Está conformada por los subsistemas, los cuales se ajustan a los requisitos y casos de uso arquitectónicamente significativos. Desde el punto de vista de diseño, esta capa es contenedora de las clases entidades y controladoras. Únicamente se comunica con la capa de Acceso a Datos.

Capa de Acceso a Datos:

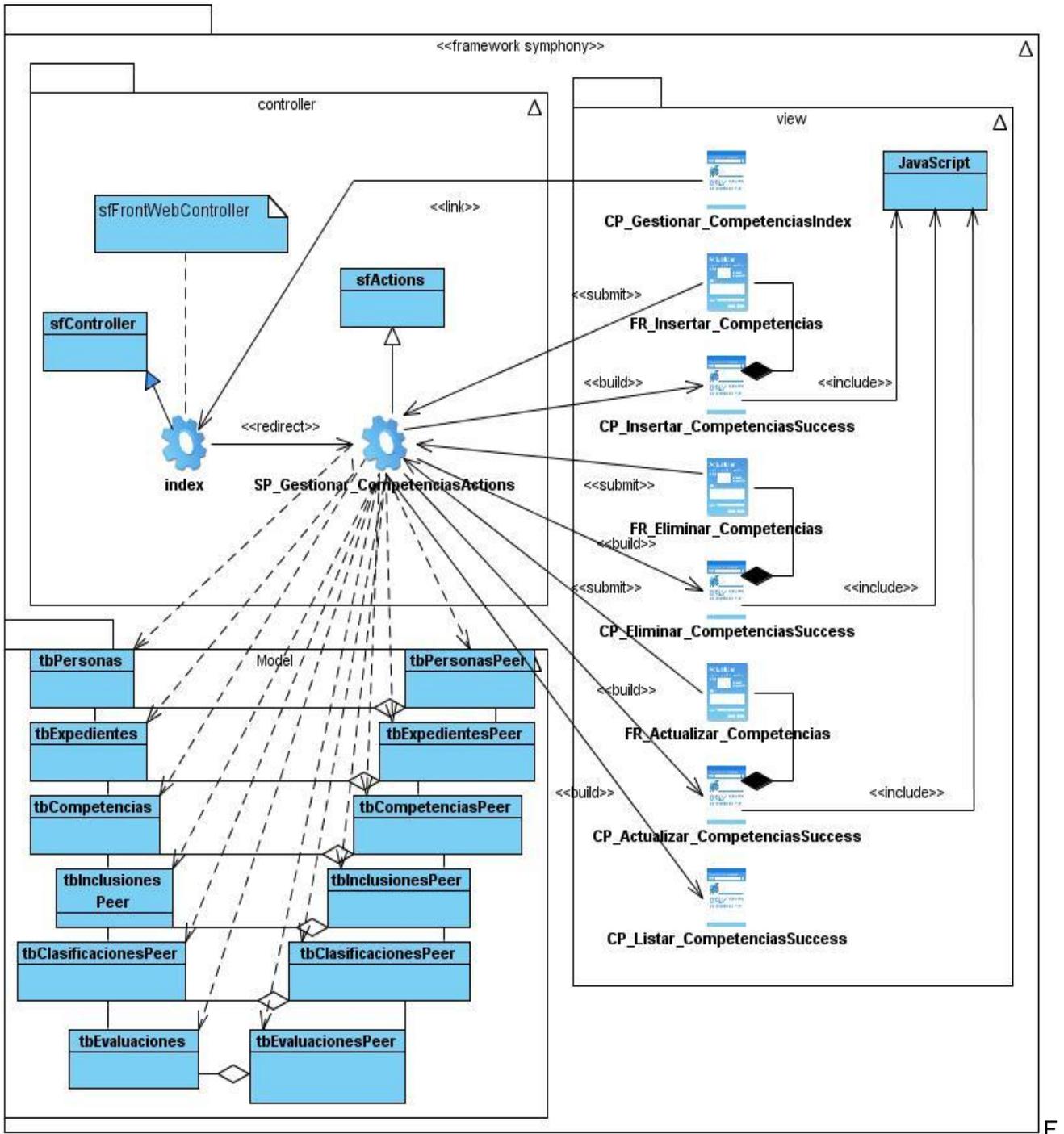
Contiene clases que interactúan con la base de datos y permiten, utilizando los procedimientos almacenados generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.

Una vez agrupada la funcionalidad en capas lógicas es fácil relacionar unas con otras. El usuario interactuará con la capa de presentación, solicitando datos o desencadenando acciones. Las solicitudes serán atendidas por la capa de negocios, que se encargará de su gestión o de la traducción necesaria para que la capa de servidor realice la tarea solicitada. Si la capa de servidor debe proporcionar datos estos se devolverán a la capa de negocios, la cual los gestionará o transmitirá a la capa de presentación.

3.2.3 Diagramas de clases del diseño

Los diagramas de clases describen, gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Contienen información como clases, asociaciones y atributos, interfaces con sus operaciones y constantes, métodos, navegabilidad, dependencias. Para construir los diagramas de clases del diseño, se aplican los patrones que incluye Symfony.

A continuación, se muestran algunos de los diagramas de clases más importantes del sistema realizados teniendo en cuenta el framework de desarrollo empleado:



ig. 3.7: Diagrama de clases del diseño: Gestionar Competencias.

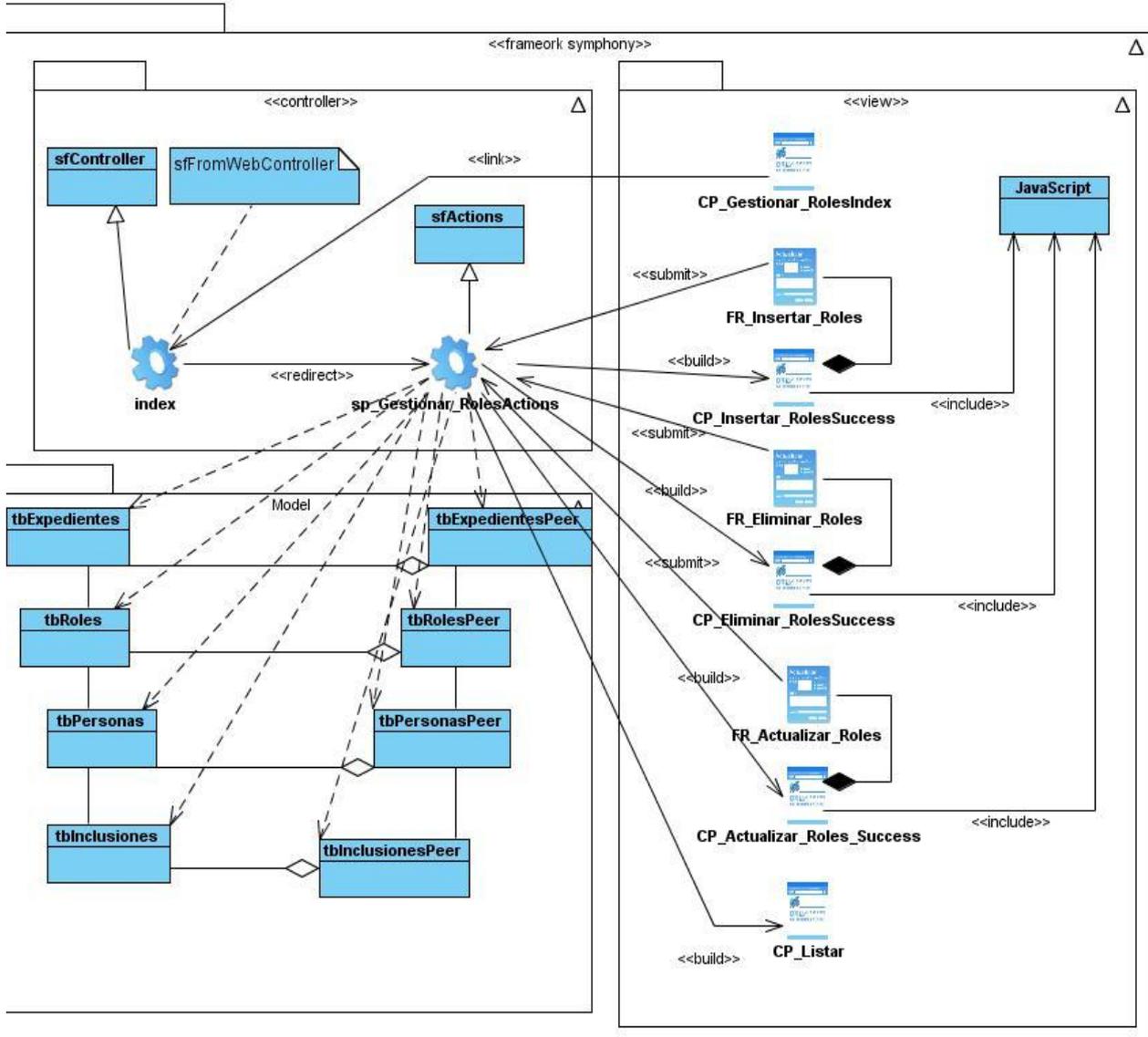


Fig. 3.8: Diagramas de clases de diseño: Gestionar Roles

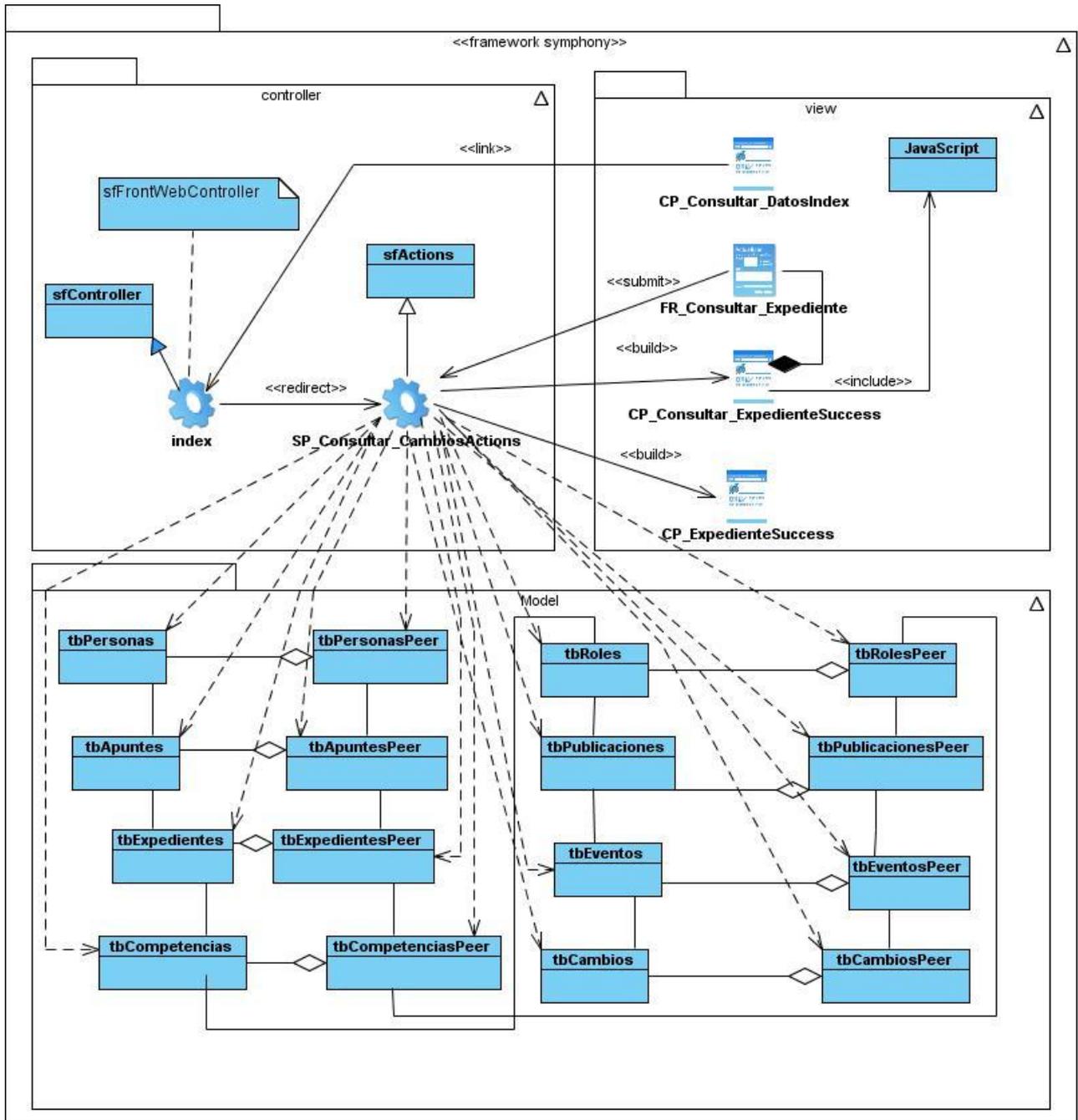


Fig. 3.9 Diagrama de clases del diseño: Consultar Expediente Productivo.

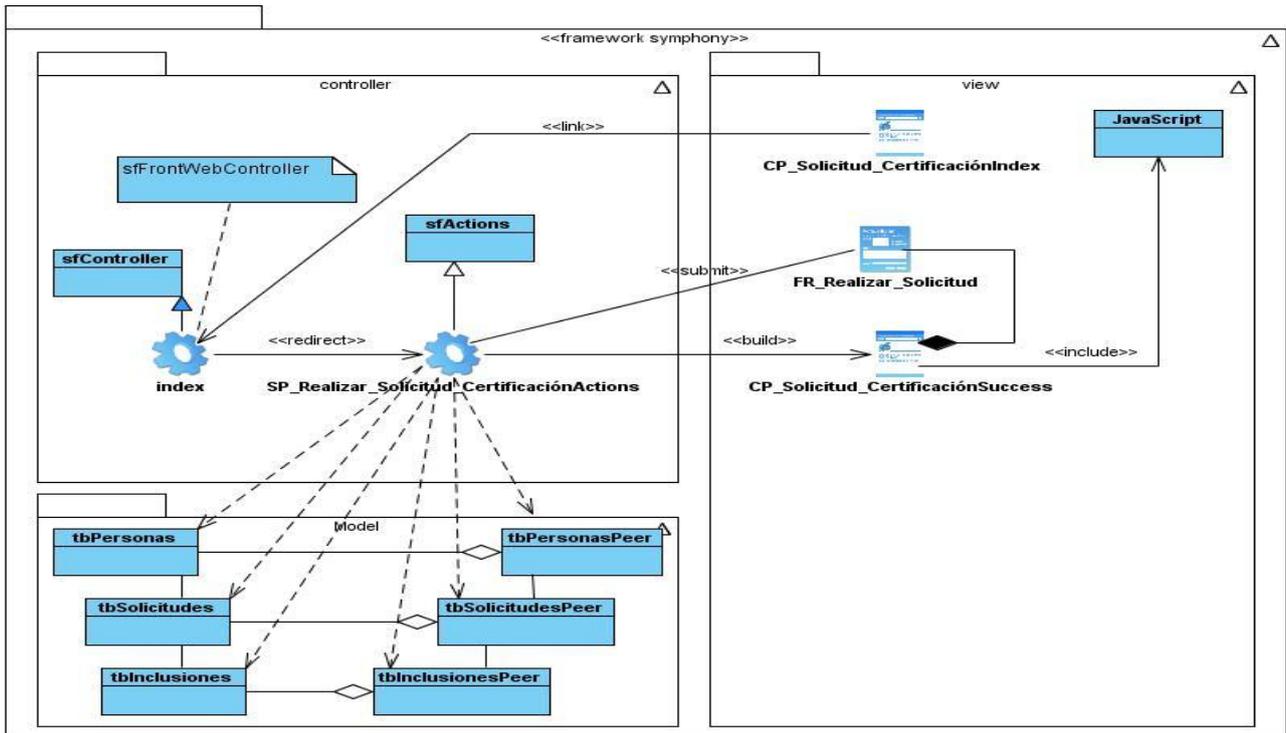


Fig. 3.10 Diagrama de clases del diseño: Realizar Solicitud de Certificación.

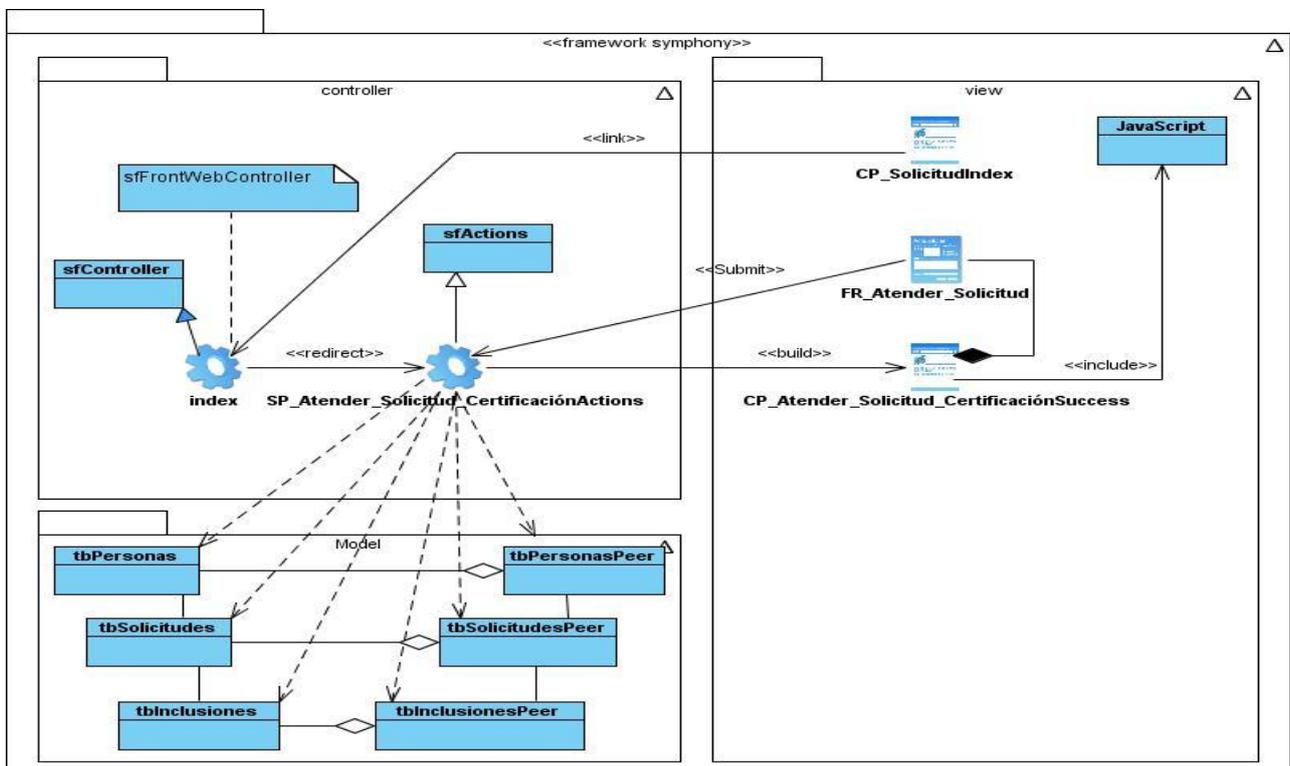


Fig. 3.11 Diagrama de clases del diseño: Atender Solicitud de Certificación.

Para acceder a los restantes diagramas de clases del diseño ver el informe: Diagramas de clases del diseño del Módulo Desarrollo del Personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3. (Rosales, y otros, 2009).

3.2.4 Diagramas de iteración: Secuencia para el diseño.

La realización de los casos de uso del análisis es otro de los artefactos que se generan en el flujo de análisis, y colaboran de forma tal que describen como se lleva a cabo y se ejecuta un caso de uso en términos de las clases del análisis y de sus objetos en interacción, por tanto es el que proporciona una trazabilidad directa hacia un caso de uso del modelo de casos de uso.

Por su parte los diagramas de interacción muestran: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos. Modelan el comportamiento dinámico del sistema, el flujo de control en una operación. Describe la interacción entre objetos, los objetos interactúan a través de mensajes para cumplir ciertas tareas. Las interacciones proveen un comportamiento y típicamente implementan un Caso de Uso.

Los diagramas de interacción, se pueden representar a través de diagramas de secuencias o colaboración. Los diagramas de secuencia, ordenan los mensajes de forma tal que colocan los objetos que participan en la interacción y los mensajes que envían y reciben estos objetos, en orden de sucesión y en el tiempo. Los diagramas de colaboración son los que construyen los objetos como un nodo de un grafo y a continuación se representan los mensajes que conectan a dichos objetos como si fueran arcos de este grafo, los cuales se envían y se reciben a través de los objetos.

A continuación, se muestran algunos de los diagramas de secuencias más importantes del sistema realizados teniendo en cuenta el framework de desarrollo empleado:

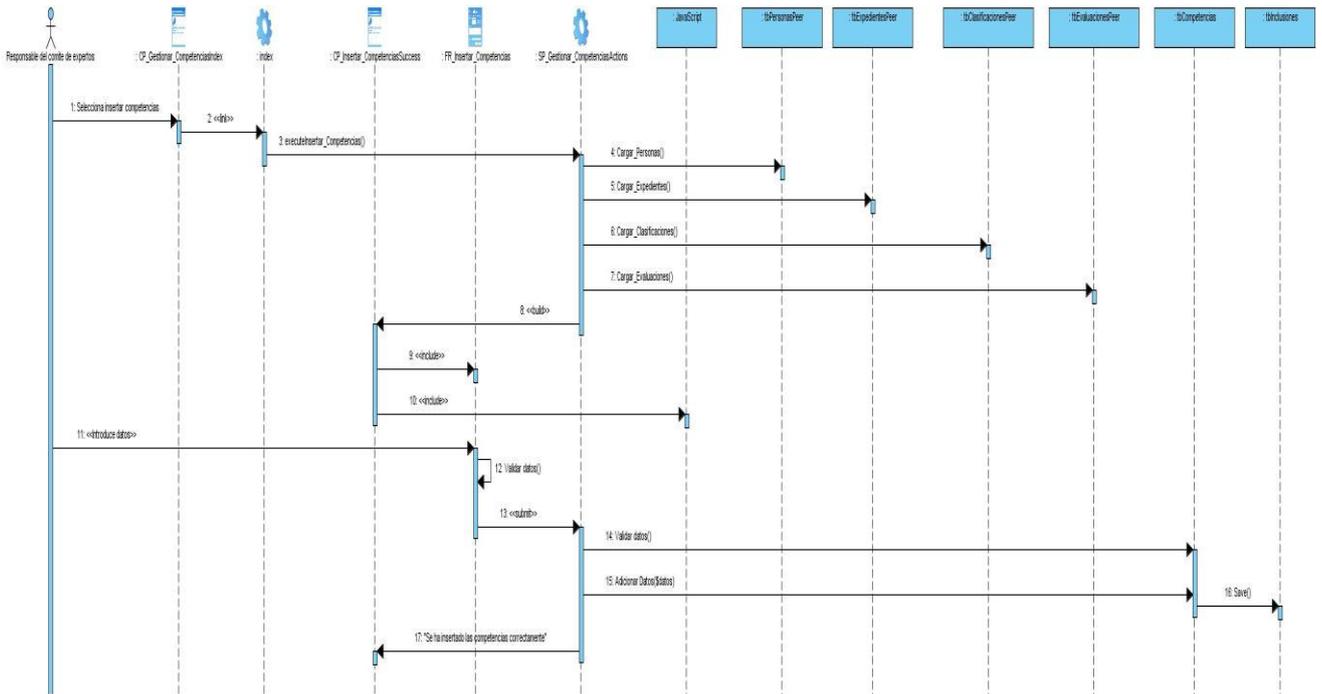


Fig. 3.12 Diagrama de secuencia: Insertar Competencia.

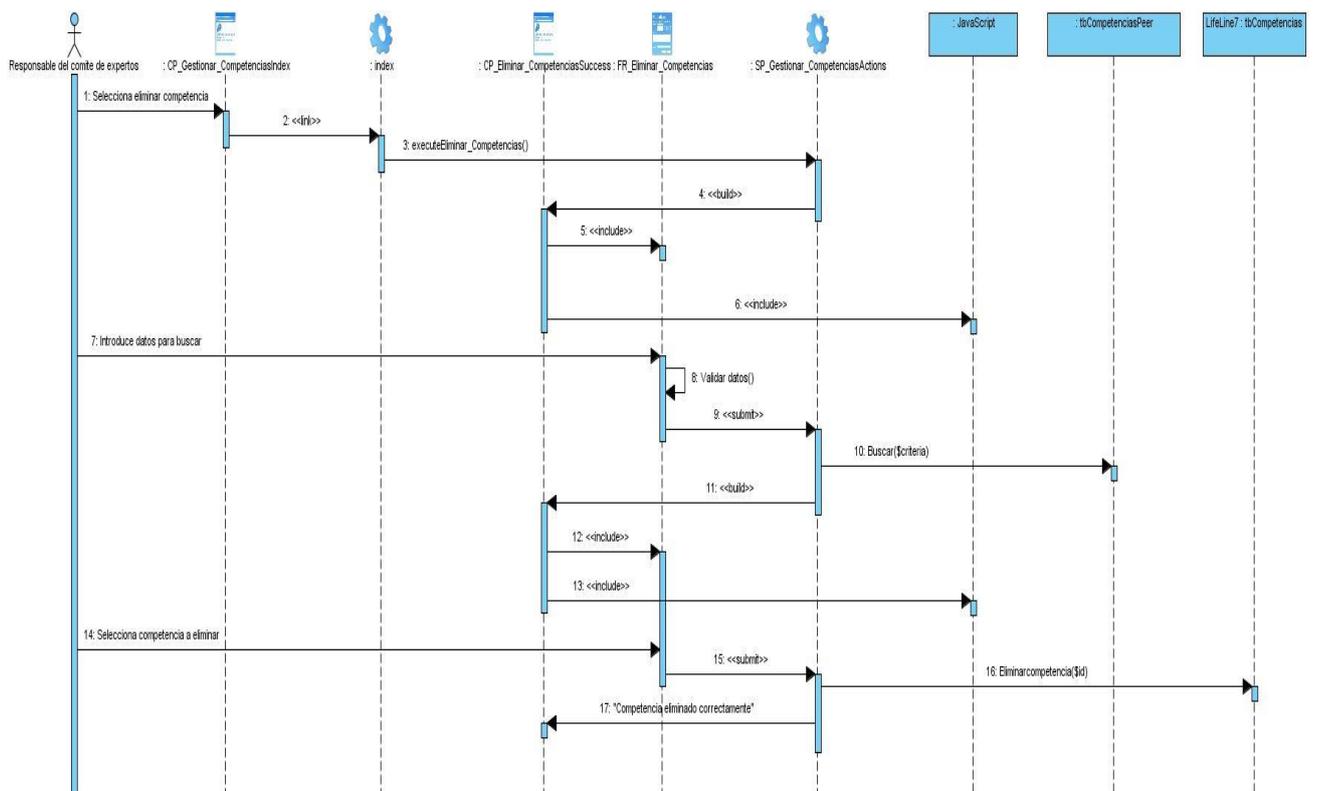


Fig. 3.13 Diagrama de secuencia: Eliminar Competencia.

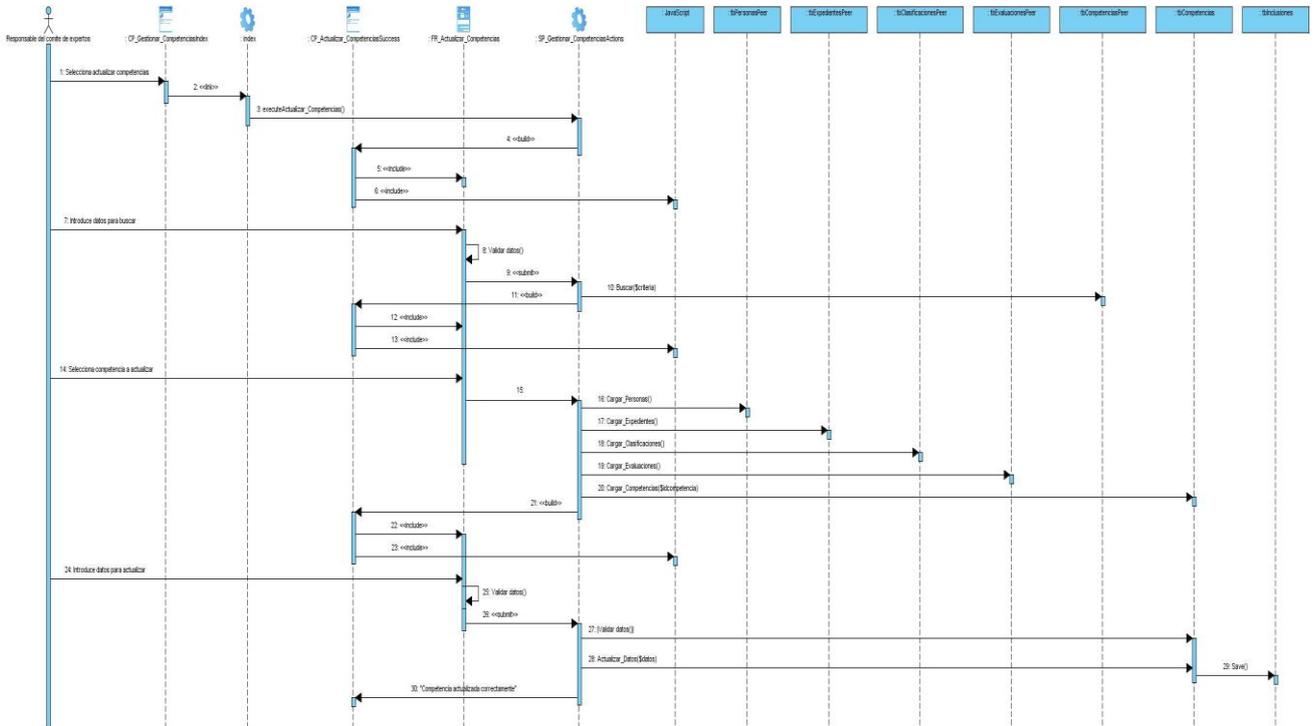


Fig. 3.14 Diagrama de secuencia: Actualizar Competencia.

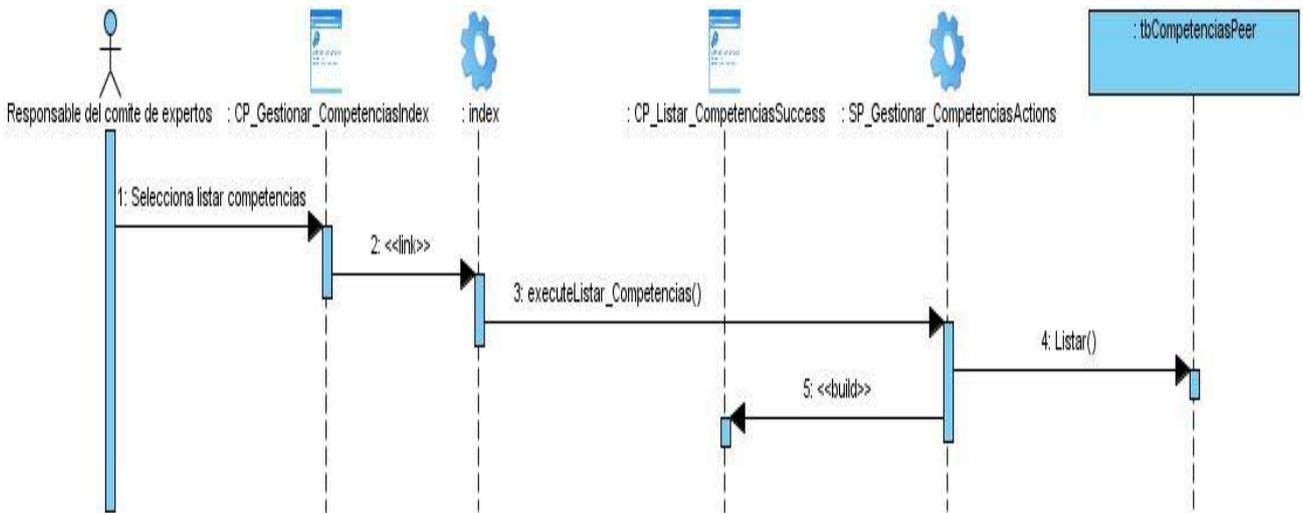


Fig. 3.15 Diagrama de secuencia: Listar Competencia.

Para acceder a los restantes diagramas de secuencias ver el informe: Diagramas de secuencias del Módulo Desarrollo del Personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3. (Rosales, y otros, 2009).

3.3 Diagrama de clases persistentes

Una clase persistente es una clase entidad que tiene la capacidad de mantener su valor en el espacio y en el tiempo, y el diagrama de clases persistentes no es más que dichas clases y las relaciones que se establece entre ellas (asociación, agregación/composición). (Ver Anexo 5)

3.4 Modelo de datos

Para construir la base de datos se utiliza el diagrama de clases persistentes mediante el cual se obtiene el modelo de datos que no es más que la representación lógica de la base de datos, luego a partir de este se puede generar un diagrama físico se obtiene un script de la base de datos que se pretende construir.

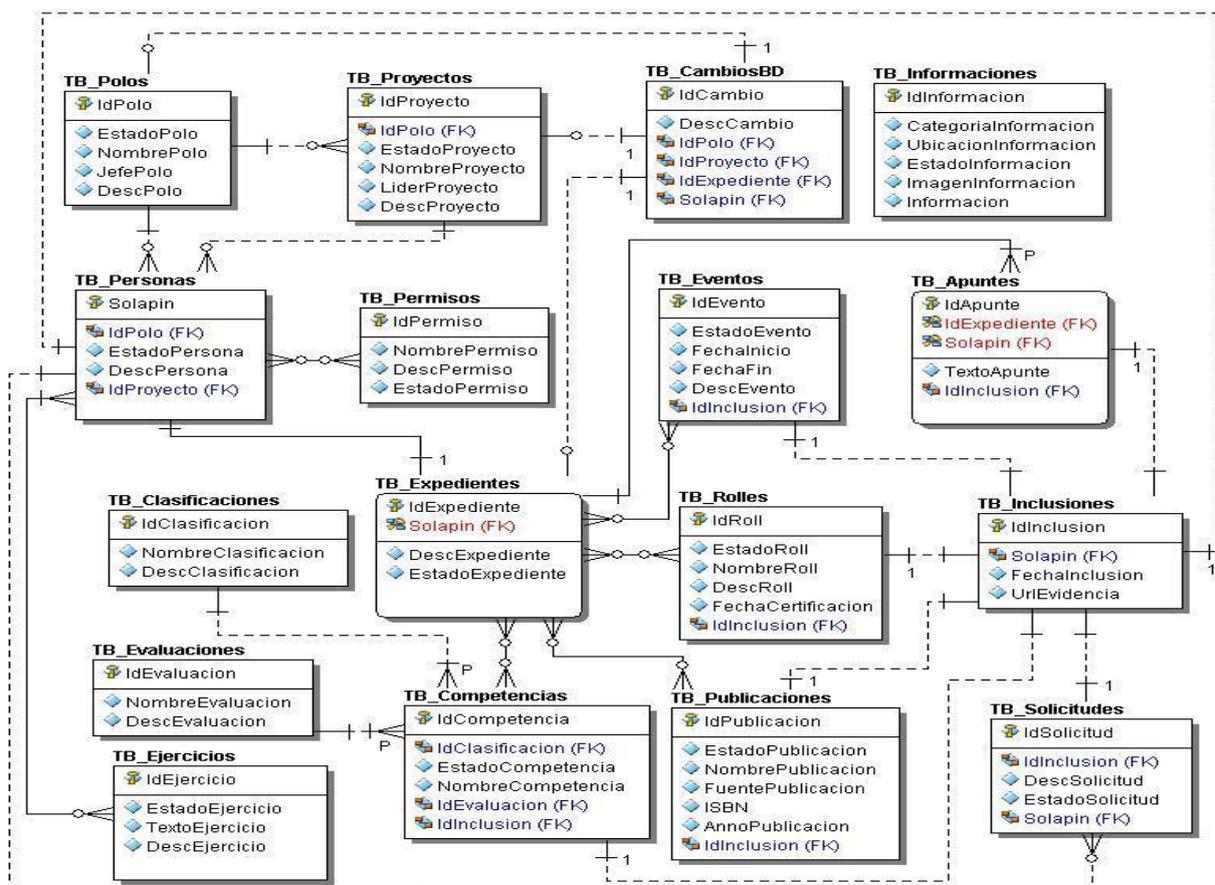


Fig. 3.16 Diagrama de entidad relación.

Conclusiones

En el presente capítulo se construyeron los diagramas de clases del análisis y del diseño, los diagramas de interacción y el modelo de datos, generándose los artefactos definidos en el flujo de trabajo que propone RUP. Se hace uso de patrones de diseño, los cuales mejoran considerablemente la calidad del software y agilizan en gran medida el trabajo de los programadores, con lo que finalmente quedaron sentadas las bases para seguir trabajando en la futura implementación del sistema.

CAPÍTULO IV: VALIDACIÓN DE LOS RESULTADOS

Introducción

En el presente capítulo se muestran los resultados obtenidos luego de haber realizado el análisis y diseño del Módulo Desarrollo de Personal del Sistema de Gestión de Recursos Humanos por Competencias en los Proyectos Productivos de la Facultad 3. Se verifica la calidad con que se realizó cada actividad a partir de los artefactos generados en cada una de ellas, empleándose métricas para comprobar que los resultados obtenidos son adecuados y que el trabajo realizado fue satisfactorio.

4.1 Proceso de validación.

Un sistema de software, como producto al fin, viene a solucionar un problema que parte de una necesidad dada, en un medio determinado. La medida de su calidad real se obtiene de hasta qué punto da solución a ese problema y con qué eficacia y eficiencia lo hace. No se tiene una medida exacta de la calidad de un sistema hasta que no está desplegado y en pleno funcionamiento, pero hay formas de obtener una idea de la calidad y la validez de un sistema durante sus etapas de desarrollo iniciales, en nuestro caso: análisis y diseño.

Para validar las etapas de análisis y diseño de un sistema se usan en la ingeniería de software varias técnicas de medición denominadas: “métricas”. Aunque los términos medida, medición y métrica se utilizan indistintamente hay diferencias “sutiles” entre ellos: (Pressman, 2001).

Medida: Proporciona una indicación cuantitativa de la extensión, Cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto.

Medición: Es el acto de determinar una medida.

Métrica: Una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. (IEEE, 1990).

La aplicación e métricas nos sirven para saber cuando nuestros proyectos se están saliendo de la línea base determinadas para ellos, nos sirven para plantearnos objetivos lógicos y aceptables tanto de coste como de calidad, alcance o esfuerzo.

En la definición y aplicación de las métricas un enfoque muy utilizado es el enfoque: objetivo pregunta métrica, Goal-Question-Metric (GQM). (Maryland, 1992).

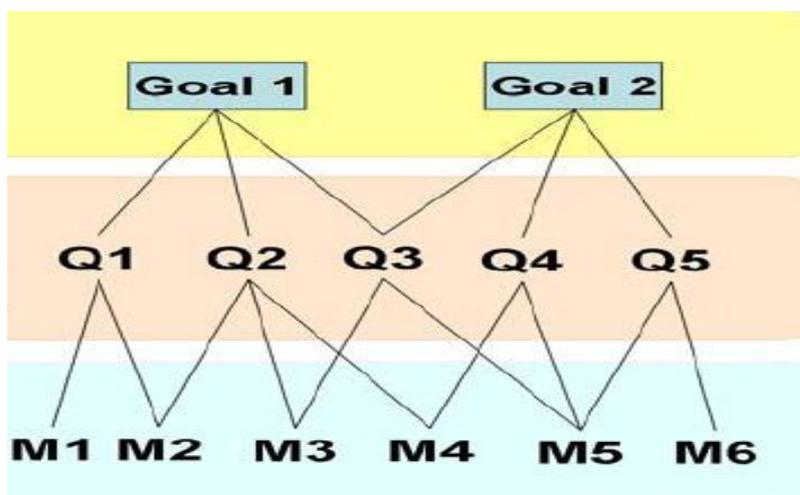


Fig. 4.1 Descripción del enfoque GQM.

GQM define un objetivo, refina este objetivo en preguntas y define métricas que intentan dar información para responder a estas preguntas. Se puede aplicar a todo el ciclo de vida del producto, procesos, y recursos y se puede alinear fácilmente con el ambiente organizacional.

Cabe añadir que una métrica cuenta además con una medida que es el valor tangible que arroja como resultado la medición de la métrica definida para un objetivo determinado mediante una pregunta específica.

4.2 Métricas para requisitos

1.- Para determinar requisitos que no están incluidos en ningún caso de uso:

Con esta métrica se pretende medir los requisitos incluidos en las funcionalidades del sistema. Propone la realización de una “Matriz de trazabilidad” que no es más que un esquema QFD (Quality Function Deployment) adaptado a los productos de software. Este esquema consta de filas, en este caso los “cómo”, de columnas los “qué” y de intercepciones entre ambas, lo que representa la inclusión de un requisito funcional en un caso de uso, determinando que este caso de uso es la realización (objetiva) del requisito funcional (subjetivo).

Casos de Uso		Requisitos Funcionales	
Núm.	Nombre del Caso de Uso	Núm.	Nombre del Requisito Funcional
1	Gestionar polos	1	Gestionar Polos
2	Realizar solicitud de certificación	2	Gestionar Proyectos
3	Atender solicitud de certificación	3	Gestionar Permisos

CAPÍTULO IV: VALIDACIÓN DE LOS RESULTADOS

4	Convocar ejercicio	4	Gestionar Noticias
5	Consultar reporte de cambios	5	Gestionar Publicaciones
6	Consultar expediente	6	Gestionar Competencias
7	Gestionar publicaciones	7	Gestionar Roles
8	Gestionar eventos	8	Gestionar Eventos
9	Gestionar roles	9	Gestionar Anotaciones
10	Gestionar proyectos	10	Consultar Expediente
11	Gestionar permisos	11	Solicitar certificar competencia o rol
12	Gestionar Noticias	12	Atender solicitud de certificación
13	Gestionar competencias	13	Cargar información de persona de LDAP
14	Gestionar anotaciones	14	Realizar Propuesta de certificación
15	Atender propuesta de solicitud	15	Atender Propuesta de certificación
16	Cargar datos de persona	16	Convocar ejercicio para certificar rol
17	Realizar propuesta de solicitud	17	Consultar reporte de cambios

Tabla 4.1 Numeración de los requisitos y casos de uso del sistema.

Requisitos Funcionales																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
Casos de Uso	1	X																	
	2											X							
	3												X						
	4																X		
	5																	X	
	6											X							
	7					X													
	8									X									
	9								X										
	10		X																
	11			X															
	12				X														
	13							X											
	14										X								

15																X		
16													X					
17														X				

Tabla 4. 2 Matriz de Trazabilidad o esquema QFD.

2.- Para la calidad de la especificación de los requisitos de software:

Para determinar la especificidad de los requisitos, Alan Davis sugiere una métrica basada en la consistencia de la interpretación de los revisores para cada requisito:

Q1 = nui / nr

Q1: consistencia de la interpretación por los revisores y cuanto más cerca se encuentre de 1, menos inconsistencias habrá y menos ambigüedad en la especificación de los requisitos.

nui: número de requisitos para los que todo el equipo de desarrollo tuvieron la misma interpretación.

nr: cantidad de requisitos en una especificación, y se calcula usando la formula: **nr = nf + nnf** donde,

nf: número de requisitos funcionales.

nnf: número de requisitos no funcionales.

Esta métrica fue aplicada en tres etapas durante el desarrollo del proceso, distribuidas en orden cronológico proporcional al avance del mismo. Las medidas arrojadas y su análisis se muestran en la tabla y gráficos siguientes:

Mediciones	Ambiguos	No Ambiguos	Total de CU	Valor de Q
Primera	4	13	17	0.76470588
Segunda	1	16	17	0.94117647
Tercera	0	17	17	1

Tabla 4.3 Valor de Q por mediciones.

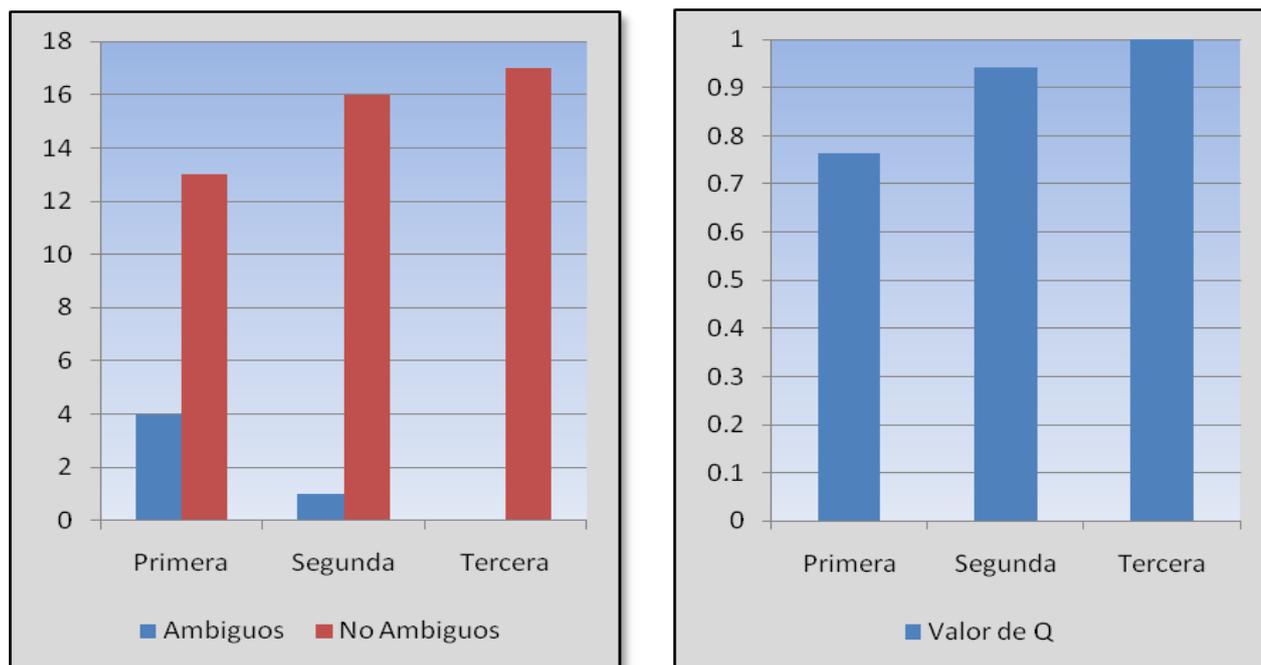


Fig. 4.2 Resultados de la métrica para detectar requisitos ambiguos.

Por tanto se concluye que no existe ambigüedad en la especificación de los requisitos, ya que el equipo de desarrollo y los revisores siempre interpretaron de igual forma el significado de la especificación de los requisitos.

4.3 Métricas para casos de uso

1.- Para la evaluación de la calidad del diagrama de casos de uso:

Este grupo de métricas tienen como objetivo evaluar la calidad de los artefactos que se generan para el sistema. Utiliza cuatro atributos u objetivos genéricos como propiedades evaluadoras de la calidad:

- ✓ Complejidad: Mide claridad y re-huso del artefacto
- ✓ Completitud: Determina en qué grado se han descrito en forma clara y concisa todos los elementos.
- ✓ Correctitud: Determina en qué grado se adecua el artefacto para satisfacer los requisitos capturados.
- ✓ Consistencia: Permite establecer el grado de ambigüedad en los elementos del artefacto.

El diagrama de casos de uso del sistema es uno de los artefactos más importantes en el desarrollo del software, muchas veces es lo que el cliente ve, trata de entender y corregir para que se adecue

CAPÍTULO IV: VALIDACIÓN DE LOS RESULTADOS

a los requisitos que se plantean. Por tanto se hace necesario aplicarle métricas de calidad siguiendo el enfoque **GQM** (Goal-Question-Metric) (Basili, 1992) que proporciona una manera útil para definir métricas.

Objetivo	Pregunta	Métrica	Medida
Complejidad	¿Hay partes reutilizables en el artefacto?	Porcentaje de reusabilidad del artefacto.	70 %
	¿Se han aplicado patrones en el diagrama de casos de uso?	Porcentaje del artefacto con patrones.	46%
	¿Se necesita reubicar algún elemento?	Porcentaje de elementos sin ser reubicados.	100%
	¿Ha sido adecuada la partición por paquetes?	Grado de adecuación del particionado por paquetes.	100%
Promedio Total:	79%		
Complejitud	¿Han sido modelados todos los aspectos necesarios?	Grado de inclusión de los aspectos necesarios	100%
	¿Se definieron todos los roles involucrados?	Grado de definición de los roles involucrados	100%
	¿Existen requisitos que no están contemplados en algún caso de uso?	Grado de contemplación de los requisitos en los Casos de Uso	100%
	¿Está todo las acciones del flujo de eventos especificadas en función de un responsable?	Grado de acciones redactadas en función de un responsable	100%
Promedio Total:	100%		
Correctitud	¿Se representan en los Casos de Uso requisitos comprensibles?	Grado en que cada caso de uso representa requisitos comprensibles	100%
	¿Se ajusta el diagrama de casos de uso a las necesidades reales?	Grado de ajuste a las necesidades reales	100%

	¿Las descripciones se ajustan a la funcionalidad que se desea?	Grado de ajuste de las funciones a las funcionalidades deseadas	100%
	¿Todos los casos de uso tienen un usuario responsable?	Porcentaje de casos de uso con usuario responsable	100%
Promedio Total:	100%		
Consistencia	¿Todos los casos de uso tienen un nombre correcto?	Porcentaje de casos de uso con un nombre correcto	100%
	¿Todas las descripciones nombres y aclaraciones están redactadas en un lenguaje de usuario?	Grado de descripciones nombres y aclaraciones redactadas en lenguaje de usuario	100%
	¿Todos los casos de uso representan interacciones observables por un actor?	Grado de casos de uso que representan interacciones observables	100%
	¿Existe una adecuada separación entre los flujos básico y alterno?	Grado de separación entre flujos básico y alterno	100%
Promedio Total:	100%		

Tabla 4.4: Resultado de las métricas para la calidad del diagrama de casos de uso.

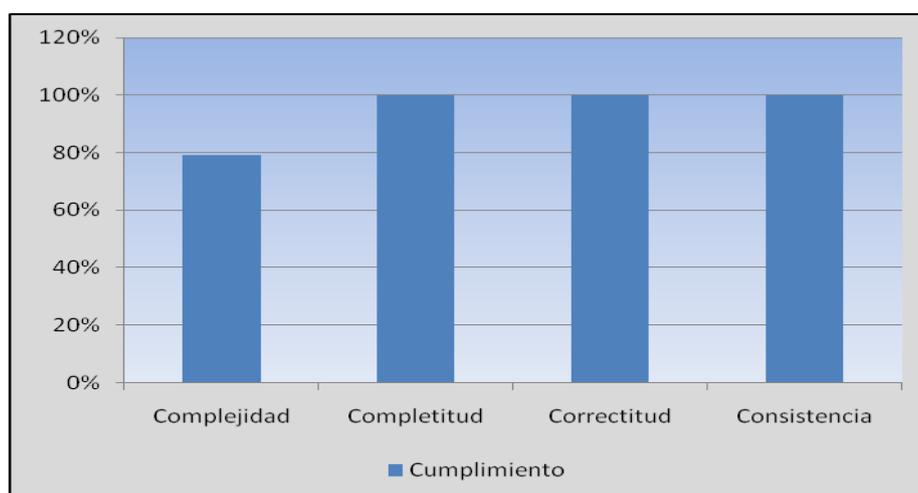


Fig. 4.3 Gráfico de cumplimiento de los objetivos de calidad para el diagrama de casos de uso.

4.4 Métrica del índice de madurez de software

Dentro de las métricas de mantenimiento en el estándar IEEE 982.1-1998 (Basili, 1992), se propone un Índice de Madurez del Software (IMS) el cual proporciona una indicación de la estabilidad del producto de software y se expresa como:

- ✓ $IMS = [Mt - (Fc + Fa + Fd)] / Mt$
- ✓ Mt = número de módulos en la versión actual
- ✓ Fc = número de módulos en la versión actual que han cambiado
- ✓ Fa = número de módulos en la versión actual que se han añadido
- ✓ Fd = número de módulos en la versión actual que han eliminado

En el caso del módulo, se ha decidido subdividirlo en otros mucho más pequeños para el cálculo del índice de madurez. Considerando así cada caso de uso como un módulo de igual nombre y con iguales funcionalidades. Por lo que la métrica se aplicaría de la siguiente forma:

$$IMS = [17 - (0 + 1 + 0)] / 17$$

$$IMS = 16 / 17$$

$$IMS = 0.94$$

El valor del índice de madurez es bastante cercano a 1 por lo que se considera que el sistema tiene un alto índice de madurez.

4.5 Métrica de la profundidad del árbol de herencia

El objetivo de esta métrica consiste en determinar el nivel de acoplamiento de las clases y la facilidad que brindan para dar mantenimiento al sistema. Las preguntas que se realizan serían: ¿Existe bajo acoplamiento entre las clases del sistema?, ¿Es fácil dar mantenimiento al sistema una vez implementado? La medida de esta métrica consiste en el promedio de la profundidad de un árbol de herencia (Pressman, 2001) para cada clase, donde la raíz sería la clase padre y las hojas las hijas.

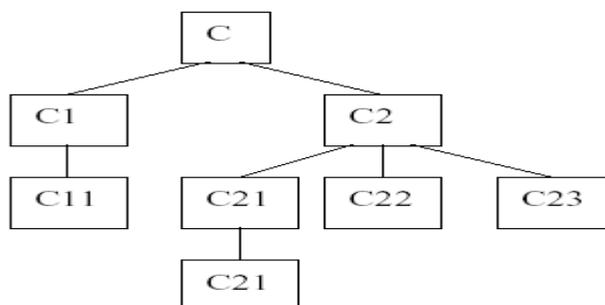


Fig. 4.4: Jerarquía de clases.

A medida que crece la profundidad, la herencia de operaciones y atributos por parte de las clases hijas es mayor, por tanto la complejidad y el acoplamiento se hacen mayor y la facilidad para el mantenimiento menor. La medida de esta métrica es una razón entre la cantidad de clases y la sumatoria de la profundidad de cada clase. Se define a continuación como:

$$PAH = \frac{\sum \text{Profundidad de cada Clase}}{\text{Número de Clases}}$$

En el sistema que se propone se aplica esta métrica en la capa de acceso a datos obteniendo un resultado para la profundidad del árbol de herencia (PHA) igual a 1. Es baja, por tanto: el sistema tiene poca complejidad, existe un bajo acoplamiento y es fácil el mantenimiento del sistema.

4.6 Prototipado

Una forma muy usada y eficaz de validar los requisitos de un sistema es la elaboración de prototipos. Se conoce como prototipo, la implementación concreta de un sistema que, se crea para explorar cuestiones sobre aspectos muy diversos durante el desarrollo de un sistema. Sirven de comunicación entre clientes y desarrolladores (Pressman, 2001), ya que generalmente el cliente sabe lo que quiere cuando lo ve. Los principales problemas de la captura de requisitos se centran en la comunicación cliente-desarrollador.

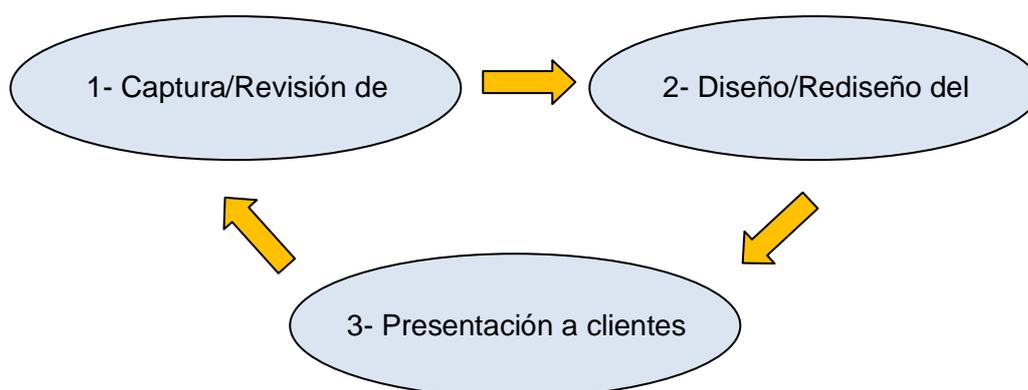


Fig. 4.5 Proceso de Captura/Revisión de requisitos mediante prototipado.

El sistema que se propone realiza la técnica del prototipado en la validación de los requisitos funcionales del software, elaborando modelos de prototipos no funcionales.

Tamaño de Clase (TC):

Las métricas basadas en clases se dividen en cuatro categorías (Lorenz): tamaño, herencia, valores internos y valores externos. Las métricas orientadas a tamaños para una clase se centran en cálculos de atributos y de operaciones para una clase individual, y promedian los valores para el sistema en su totalidad.

El tamaño general de una clase se puede determinar empleando las medidas siguientes:

- ✓ El número total de operaciones (tanto operaciones heredadas como privadas de la instancia) que están encapsuladas dentro de la clase.
- ✓ El número de atributos (tanto atributos heredados como atributos privados de la Instancia) que están encapsulados en la clase.

Si existen valores grandes de TC éstos mostrarán que una clase tiene demasiada responsabilidad, lo cual reduce la reutilización de la clase y complica la implementación y la comprobación, por otra parte cuanto menor sea el valor medio para el tamaño, más probable es que las clases existentes dentro del sistema se puedan reutilizar ampliamente.

Se aplica esta métrica para la capa de acceso a datos arrojando los resultados que se listan a continuación:

#	Nombre de la clase	Cant. de atributos	Cant. de Operaciones	Tamaño
1	CE_Polos	5	5	Pequeña
2	CE_Proyectos	5	5	Pequeña
3	CE_CambiosBD	2	5	Pequeña
4	CE_Informaciones	6	5	Pequeña
5	CE_Personas	3	5	Pequeña
6	CE_Permisos	4	5	Pequeña
7	CE_Expedientes	3	5	Pequeña
8	CE_Apuntos	2	5	Pequeña
9	CE_Inclusiones	3	5	Pequeña

10	CE_Evaluaciones	3	5	Pequeña
11	CE_Ejercicios	4	5	Pequeña
12	CE_Competicencias	3	5	Pequeña
13	CE_Roles	5	5	Pequeña
14	CE_Solicitudes	3	5	Pequeña
15	CE_Eventos	5	5	Pequeña
16	CE_Publicaciones	6	5	Medio
17	CE_Clasificaciones	3	5	Pequeña
Promedio Tamaño de Clase:		3.82	5	Pequeña

Tabla 4.6 Resultado de la métrica para el tamaño de clases.

De la medida obtenida anteriormente (Promedio: Atributos = 3.82, Operaciones = 5) se puede concluir: que las clases dentro del sistema tienen un tamaño óptimo por lo que son ampliamente reutilizables.

Conclusiones

Con la aplicación de las diferentes métricas se llegan a las siguientes conclusiones:

- ✓ La aplicación de las métricas para la calidad de la especificación de los requisitos de software demostró que el grado de ambigüedad en la especificación de los requisitos fue muy bajo.
- ✓ La aplicación de las métricas para la calidad de la funcionalidad del DCUS demostró que se construyó un diagrama con calidad.
- ✓ Aplicando la etapa de Validación de requisitos de la Ingeniería de Requisitos se lograron requisitos consistentes y sin ambigüedades.

CONCLUSIONES GENERALES

Durante el presente trabajo, se da cumplimiento a los objetivos trazados, obteniendo el análisis y diseño del módulo Desarrollo del Personal, lo que permitirá a los desarrolladores la implementación del mismo.

- ✓ La realización del marco teórico de la investigación, permitió profundizar en el proceso de Desarrollo del Personal por competencias, así como en la selección de las herramientas, lenguajes y metodologías a emplear para el desarrollo del sistema.
- ✓ El modelo de procesos de negocio propuesto, posee gran valor práctico. Sirve de guía metodológica para las certificaciones de competencias y roles, tan necesaria en el nuevo modelo de formación que se prevé aplicar en la UCI. Las entrevistas dirigidas a los involucrados y las tormentas de ideas con especialistas, permitieron su enriquecimiento y perfeccionamiento.
- ✓ Los requisitos funcionales, no funcionales, el modelo de casos de uso del sistema y sus descripciones textuales, sirvieron como elemento fundamental para la concepción del futuro sistema. Además, se realiza el análisis y el diseño en conjunto con el modelo de datos; artefactos que contribuyen de forma significativa a la continuidad de su implementación.
- ✓ La aplicación de métricas y técnicas, contribuyeron a la validación del proceso de negocio, sistema, análisis, diseño y datos del módulo Desarrollo del Personal.

RECOMENDACIONES

De forma general los objetivos propuestos al inicio de este trabajo fueron cumplidos en su totalidad, no obstante, se realizan una serie de recomendaciones que podrán tenerse en cuenta para el desarrollo futuro del sistema:

- ✓ Refinar el modelo del proceso de Desarrollo del Personal por competencias, en correspondencia con las políticas futuras que siga la universidad, con vista de ser aplicado en todas las facultades.
- ✓ Refinar los artefactos obtenidos en el análisis y el diseño acorde con las nuevas necesidades organizacionales, antes de comenzar con la implementación del sistema.
- ✓ Conectar el sistema con Akademos v2, y con ERP.
- ✓ Implementar el sistema propuesto.

BIBLIOGRAFIA CONSULTADA

- Basili, V. 1992.** *Software modeling and measurement.* 1992.
- Bogota. 1999.** *Evaluación de competencias.* Bogota : s.n., 1999.
- Cabezas, M. 2001.** *CABEZA M. A. Indicadores de Gestión en la Educación Superior como Herramienta de la Planificación Estratégica.* Caracas : s.n., 2001.
- Chiavenato, Idalberto. 2002.** *Gestión del talento humano. El nuevo papel de los recursos humanos en la organización.* Mexico : s.n., 2002.
- Clara, Elena. 2001.** *Tecnología de Gestión para la formación de los recursos humanos en las organizaciones.* Holguin : s.n., 2001.
- Competencias. 1997.** *Normalización y certificación de la competencia laboral.* 1997.
- Cuestas, Armando. 1999.** *Conformación de Perfiles de Competencias de los Perfiles de Competencia Directivas por el Método Delphi.* Ciudad de la Habana : s.n., 1999.
- Espinoza, Y. 2005.** *Introducción de los Test Adaptativos Informatizados en las Universidades Cubanas.* 2005.
- Gamma, Erich. 2000.** *Design Patterns: Elements of Reusable Object-Oriented Software.* 2000.
- GOF. 1995.** *Patrones GOF.* 1995.
- IEEE. 1990.** *Software Engineering Standards.* 1990.
- Knowledge. 2006.** *Knowledge Based Systems.* 2006.
- Lorenz.**
- Loyola, William. 2006.** *Maestría en sistemas de información gerencial.* 2006.
- Malparaiso. 2006.** *Modelado de procesos con IDEF0.* Malparaiso : s.n., 2006.
- Maryland, University. 1992.** 1992.
- Olsina. 1999.** 1999.
- Övergaard. 2004.** 2004.

PMBOOK. 2004. *Guía de los Fundamentos de la Dirección de Proyectos.* 2004.

Porcesos. 1993. *Procesos estandar de modelado.* 1993.

Potencier, Fabien y Zaninotto, François. 2008. *Symfony, la guía definitiva.* 2008.

Pressman, Roger. 2001. *Ingeniería del Software un enfoque práctico.* 2001.

Rosales, Armando Alexander y Márquez, Miguel Angel. 2009. *Descripción textual de los casos de usos del Módulo Desarrollo de Personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la Facultad 3.* Ciudad de la Habana : s.n., 2009.

—. **2009.** *Diagramas de clases del análisis del Módulo Desarrollo del Personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3.* Ciudad de la Habana : s.n., 2009.

—. **2009.** *Diagramas de clases del diseño del Módulo Desarrollo del Personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3.* Ciudad de la Habana : s.n., 2009.

—. **2009.** *Diagramas de secuencias del Módulo Desarrollo del Personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3.* Ciudad de la Habana : s.n., 2009.

—. **2009.** *Validación del proceso de Desarrollo del Personal.* Ciudad de la Habana : s.n., 2009.

Toro, Duran y Jiménez, Bernárdez. 2000. *Doctorado en Tecnologías e ingeniería de software.* España : s.n., 2000.

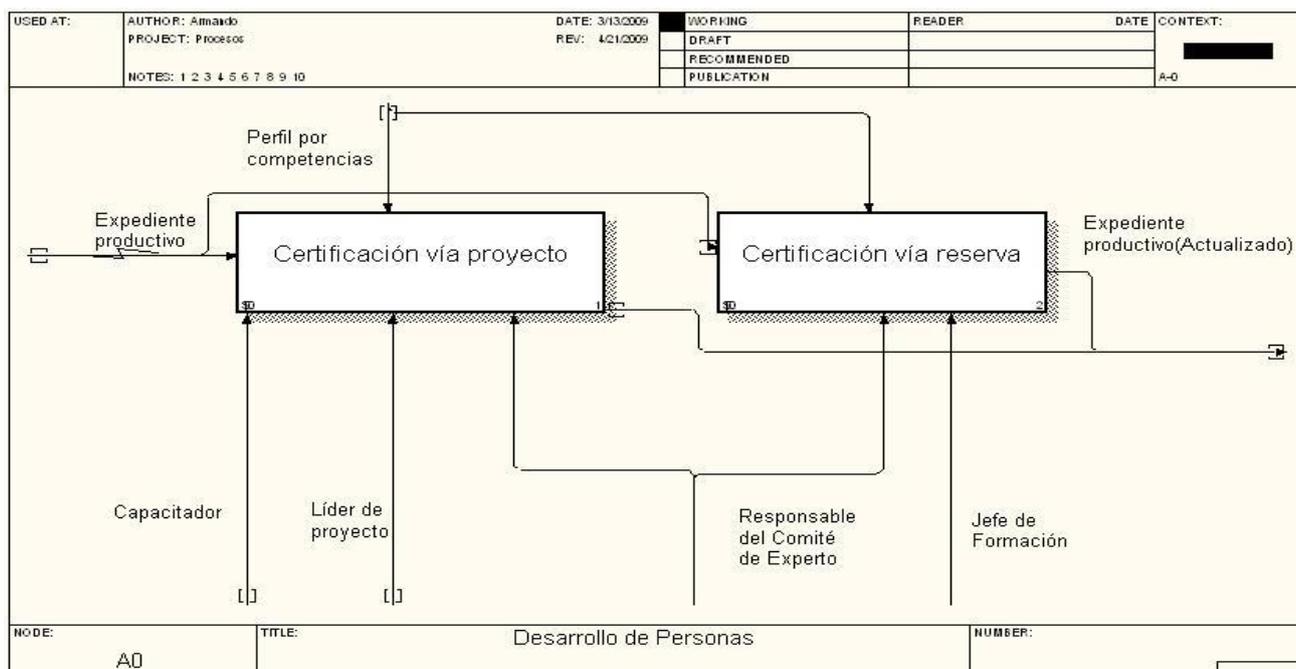
UML. 2001. *UML y la Modelación de datos.* 2001.

Villanueva, Martin. 2007.

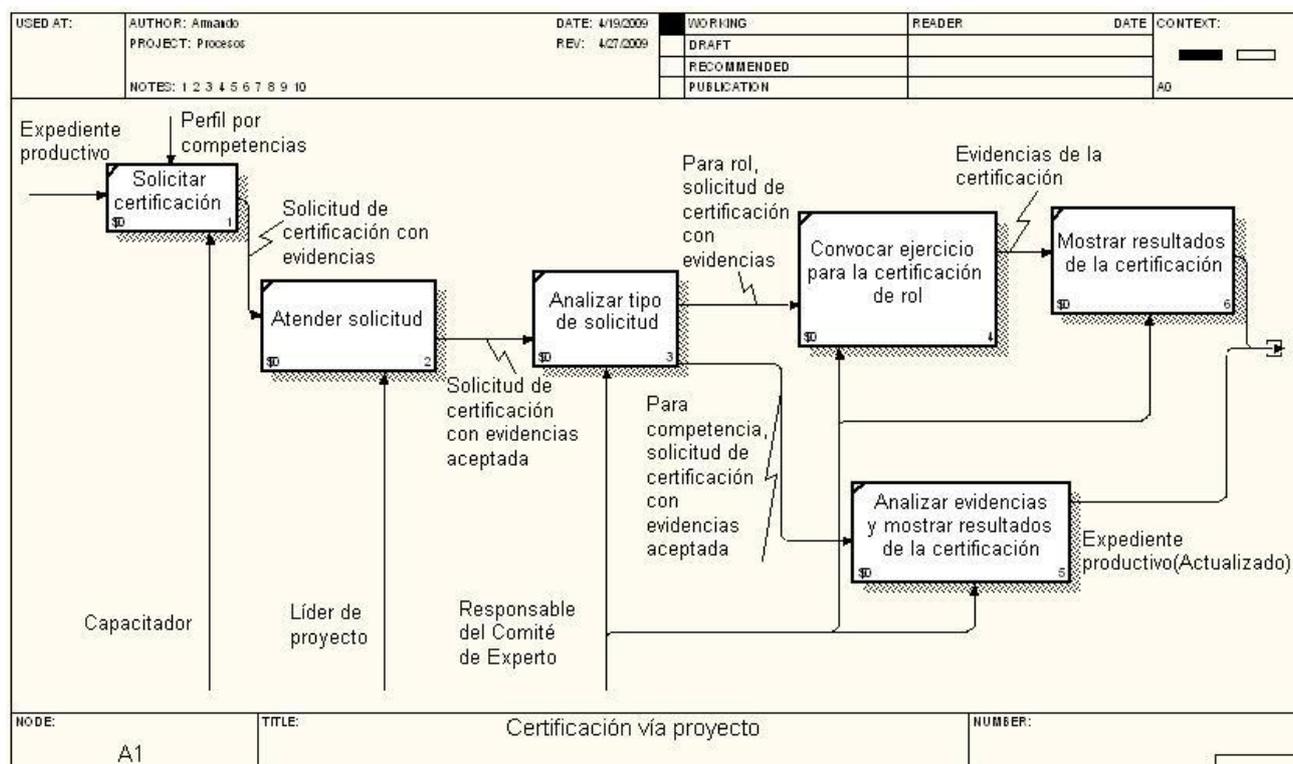
Anexo 1: Propuesta de plantilla del Perfil por Competencias.

PERFIL POR COMPETENCIAS	
Facultad#:	
Rol:	
Listado de Competencias Genéricas	

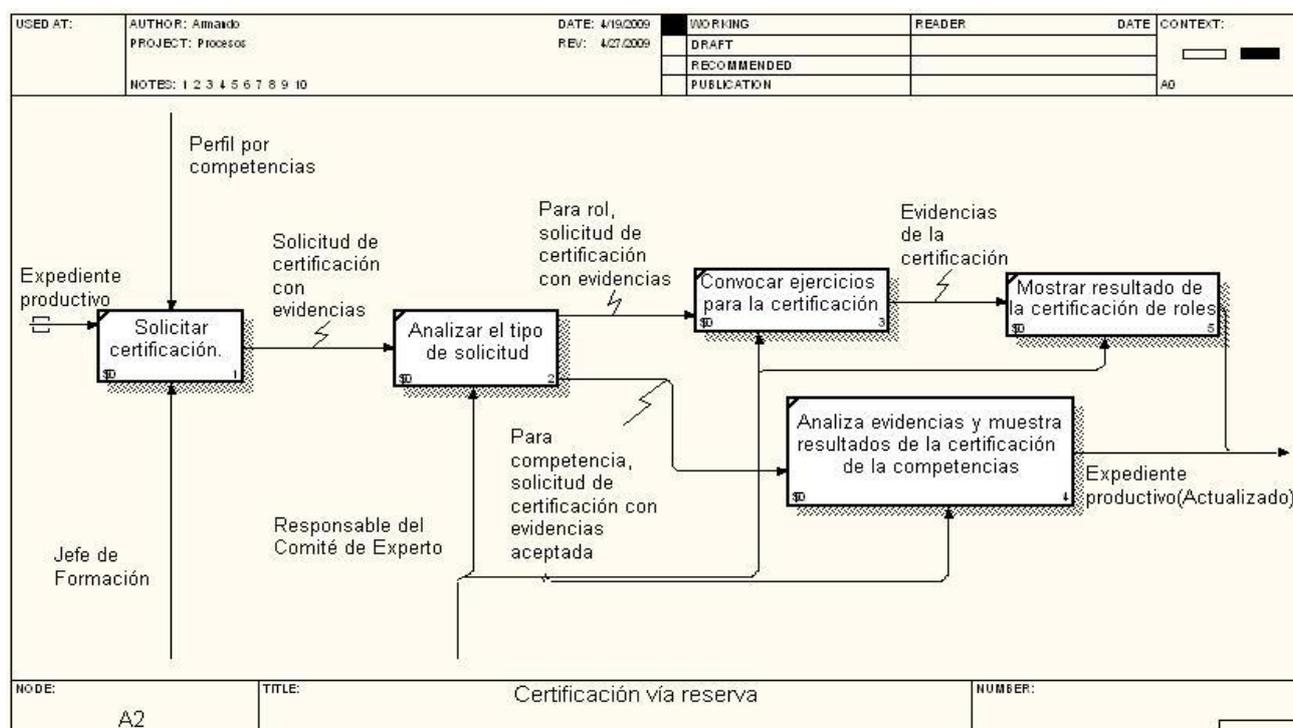
Anexo 2: Subprocesos del Proceso de Desarrollo del Personal.



Anexo 3: Actividades del sub-proceso de Desarrollo de Personal vía proyecto



Anexo 4: Actividades del sub-proceso de Desarrollo de Personal vía reserva.



Anexo 5: Diagrama de clases persistentes.

