

Universidad de las Ciencias Informáticas

"Facultad 2"



**Título: "Sistema de apoyo a los procesos del
área de formación del Centro ISEC. (SUBAS)"**

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

Autor: Ana Karla Orret Moreno

Tutor: Ariagnis Yero Guevara

La Habana, 2012

"Año 54 de la Revolución"

Pensamiento

«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.»

Kent Beck.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Centro de Informatización de la Seguridad Ciudadana de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Ana Karla Orret Moreno

Ariagnis Yero Guevara

AGRADECIMIENTOS

A la Revolución, sin ella no habiéramos tenido la posibilidad de estudiar en esta universidad.

A mi familia, por darme el apoyo y la confianza necesaria para transitar segura por cada etapa de mi vida, en especial a mi mamá y a mi papá, a mis hermanos y primos, a mis abuelos, mis tíos, etc...

A mis amigos de la UCI, los cuales considero mi segunda familia. A mis colegas del movimiento de artistas aficionados, que tantas anécdotas y vivencias hemos compartido.

A Yannick y Yannick, por su apoyo incondicional, por ser un ejemplo perfecto para mí y por apoyarme y acompañarme a donde quiera que voy.

A mi novio José Miguel por apoyarme en cada momento, por soportarme cada berrinche, por darme todo el amor que necesite en cada una de las situaciones por las que pasé.

A Eidy Estupiñán que se ha portado como un hermano y que siempre ha estado dispuesto a sacarme de apuros.

A Osay González Fuentes, quien apenas sin conocerme me brindó su apoyo incondicional e hizo posible que todo se terminara en tiempo.

A los profesores que conformaron los talleres de tesis, a mi tribunal de tesis y a mi oponente por hacerme creer en mí, por demostrarme que siempre se puede un poquito más. A mi tutora que hizo maravillas para que se pudiera llevar a cabo esta tesis que hoy me permite incluirme en esta lista de ingenieros graduados en la Universidad de las Ciencias Informáticas.

DEDICATORIA

Dedico este trabajo de diploma a mi mamita linda y a mi papá, que tanto han confiado en mí y me han guiado rumbo al éxito con cada palabra, con cada momento, con cada experiencia.

A mis hermanos: Roxana, César, Carlitos que espero les sirva de ejemplo para que estudien mucho y sean seguidores del legado profesional de nuestra familia. A Mariana, que aun siendo la esposa de mi padre y la madre del pequeño Carlitos ha sido como una hermana para mí desde el instante en que nos conocimos.

A mis abuelitos que han sido junto a mis padres, mis principales educadores y a los cuales quiero agradecerles todo lo que soy, gracias a la confianza que depositaron en mí en todo momento.

A mis tíos y primos, dondequiera que estén, que han compartido junto a mí los buenos y malos momentos de esta vida que nos ha tocado vivir.

A mis amigos de toda una vida Nidia, Winnie, Wendy, Agustín. A los nuevos amigos que en los últimos cinco años han estado a mi lado y a los que no quisiera dejar a un lado mientras viva.

A los profesores responsables de mi formación profesional y a otros que siempre me dieron una mano en los momentos difíciles.

RESUMEN

Con el objetivo de lograr una formación interdisciplinaria en sus educandos, La Universidad de Ciencias Informáticas (UCI) implementa un modelo de formación que integra la docencia, la investigación y la producción. La puesta en práctica de dicho modelo involucra de forma activa a los Centros de Desarrollo de Software de la institución, pues es allí donde los estudiantes realizan su formación profesional. El Subdirector de Formación es el miembro del consejo de dirección del Centro responsable de organizar y controlar las actividades de formación de los estudiantes y profesores que se vinculan al mismo.

Con el objetivo de apoyar y mejorar la gestión de los procesos de formación del Centro de Informatización de la Seguridad Ciudadana (ISEC) se propone la solución informática "**Sistema de apoyo a los procesos del área de formación del Centro ISEC. (SUBAS)**", la cual permitirá almacenar y consultar información relacionada con los estudiantes, profesores y proyectos del centro ISEC y apoyará los procesos de gestión de tesis, cursos optativos y control de evaluaciones de los estudiantes.

En el presente documento se exponen los resultados del estudio realizado a varios sistemas de gestión de formación, así como aspectos y conceptos fundamentales e imprescindibles para una mejor comprensión del tema que se aborda en el desarrollo del trabajo. Se presenta además el análisis, diseño e implementación del sistema propuesto.

PALABRAS CLAVE

[formación, gestión, estudiantes]

TABLA DE CONTENIDOS

INTRODUCCIÓN	- 6 -
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	- 10 -
1.1. Introducción	- 10 -
1.2. Conceptos fundamentales	- 10 -
1.2.1. Formación docente	- 10 -
1.2.2. Procesos de formación docente	- 10 -
1.2.3. Gestión de procesos de formación docente	- 11 -
1.2.4. Herramientas de apoyo a la gestión de procesos de formación docente -	12 -
1.3. Estudio de las herramientas y tecnologías a utilizar en el desarrollo de la solución	- 14 -
1.3.1. Metodología o estándar	- 14 -
1.3.2. Herramienta CASE	- 16 -
1.3.3. Definición de la integración para el modelado de las funciones (IDEF0) -	16 -
1.3.4. Dia	- 17 -
1.3.5. Aplicaciones informáticas	- 18 -
1.3.6. Lenguajes de programación	- 19 -
1.4. Servidores Web	- 22 -
1.5. Arquitectura de Software	- 23 -
1.6. Sistemas Gestores de base de datos (SGBD)	- 24 -
1.7. Entorno de desarrollo integrado (IDE)	- 25 -
1.8. Marco de trabajo o Framework	- 26 -
1.9. JQuery 1.3.4	- 20 -
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	28
2.1. Introducción	28
2.2. Objetivos de la Organización	28
2.3. Objeto de Informatización	28
2.4. Propuesta de la aplicación	29
2.5. Modelado del negocio	29
2.5.1. Descripción del proceso base del negocio	30
2.5.2. Descripción de los sub-procesos del negocio.	31
2.5.3. Actor del negocio	32
2.5.4. Trabajador del negocio	33
2.5.5. Reglas del negocio	34
2.5.6. Especificación de requisitos de software	35
2.5.7. Descripción del sistema propuesto	40
2.5.8. Modelo de casos de uso del sistema	41
2.5.9. Diagrama de Caso de Uso del Sistema “Gestionar RH”	43

2.5.10.	Descripción de Caso de Uso del Sistema “Gestionar Estudiante”	44
2.5.11.	Descripción de Caso de Uso del Sistema “Buscar Estudiante”	47
2.5.12.	Conclusiones	49
CAPÍTULO 3: DISEÑO E IMPLEMENTACION DEL SISTEMA		50
3.1.	Introducción	50
3.2.	Selección del estilo arquitectónico	50
3.3.	Patrones de diseño	51
3.3.1.	Patrones GoF (Gang of four)	51
3.3.2.	Patrones (GRASP)	52
3.4.	Modelo de diseño	53
3.5.1.	Principios de diseño	55
3.6.	Diseño de la Base de datos	55
3.7.	Modelo de implementación	57
3.7.1.	Modelo de componentes	58
3.7.2.	Modelo de despliegue	59
3.8.	Pruebas	60
3.8.1.	Prueba de Caja Negra	60
3.8.2.	Pruebas de Validación	61
CONCLUSIONES		65
RECOMENDACIONES		66
REFERENCIAS BIBLIOGRÁFICAS		67
BIBLIOGRAFÍA		69
ANEXOS		¡ERROR! MARCADOR NO DEFINIDO.
Anexo1. Diagramas de Casos de Uso del Sistema		¡Error! Marcador no definido.
Anexo 2. Descripciones de Casos de Uso.		¡Error! Marcador no definido.
Anexo 3. Pruebas de Validación.		¡Error! Marcador no definido.
Anexo 4. Diagramas de clases del diseño.		¡Error! Marcador no definido.

INTRODUCCIÓN

La evolución de la ciencia y la técnica vinculadas a la informática imponen el reto de alcanzar nuevos caminos para continuar el proceso de desarrollo en los campos de la informatización y producción de software nacionales.

La Universidad de las Ciencias Informáticas (UCI), desde su creación hasta la fecha, se ha convertido en uno de los principales productores de software del país. A mediados del año 2010, se decide implementar en este centro universitario un nuevo modelo formativo, vinculando la formación, la producción y la investigación. Este nuevo modelo posibilitó la asociación de los estudiantes, desde el segundo semestre de tercer año y hasta quinto, a los procesos de desarrollo de software llevados a cabo en la universidad mediante su integración a los Centros de Desarrollo.

Los Centros de Desarrollo son estructuras organizativas que guían la producción de software dentro de las facultades. La Facultad 2 cuenta con dos Centros de Desarrollo: el **Centro de Telemática (TLM)** y el **Centro de Informatización de la Seguridad Ciudadana (ISEC)**. El centro ISEC está conformado por un director de centro, un Subdirector de investigación y postgrado, un Subdirector de formación y varios asesores, además de los estudiantes y profesores asociados a los proyectos del mismo.

El Subdirector de formación es el responsable de gestionar la actividad de formación docente que desarrollan los profesores del centro y la formación desde la producción que reciben los estudiantes.

Dentro de las actividades que realiza se encuentran la organización de las asignaturas optativas que reciben los estudiantes vinculados a los proyectos del centro, la orientación y control de las evaluaciones mensuales y finales de las asignaturas de Práctica Profesional, la organización de los procesos de certificación de roles, certificación de investigación y definición y defensa de los trabajos de diploma.

Para dar cumplimiento a estas actividades están definidas para el área de formación una serie de tareas o procesos que debe realizar el Subdirector del centro.

Del modo de realización actual de estas actividades o procesos se derivan inconvenientes que ralentizan la labor del Subdirector de Formación del centro ISEC. Entre los ejemplos que se pudieran citar se encuentra la generación de información en diversas fuentes y formatos como consecuencia del uso de las

herramientas ofimáticas. Esta diversidad imposibilita la centralización de la información, que a su vez implica la duplicación de la misma dentro de la organización de carpetas por actividades y dificulta la elaboración de reportes para la entrega en tiempo de información a la dirección de la facultad y la universidad. Otro de los inconvenientes identificados se basa en el hecho de almacenar la información procesada solamente en el ordenador del Subdirector de Formación, pues se corre el riesgo de sufrir una pérdida parcial o total de dicha información en caso de rotura del equipo. Lo antes expuesto provoca que la labor del Subdirector de Formación del Centro ISEC requiera más tiempo del que pudiera estar disponible.

Por la situación descrita anteriormente y la necesidad de realizar un trabajo más organizado y ágil se identifica el siguiente **problema a resolver**: ¿Cómo apoyar la gestión de los procesos de formación docente del Centro ISEC? Definiendo como **objeto de estudio**: la gestión de los procesos de formación docente y como **campo de acción**: las herramientas para la gestión de los procesos de formación docente. El **objetivo general** de esta investigación es: Desarrollar una aplicación que permita gestionar los procesos del área de formación de un centro de desarrollo.

De este objetivo general se derivan los siguientes **objetivos específicos**:

- Elaborar la fundamentación teórica de la investigación.
- Realizar el análisis y diseño de la herramienta propuesta.
- Implementar la herramienta propuesta.
- Desarrollar pruebas a la herramienta.

Con vistas a dar cumplimiento a estos objetivos específicos se definieron las siguientes **tareas de la investigación**:

- Elaboración del estado del arte del tema en cuestión.
- Realización del cronograma de la investigación.
- Selección de metodología, herramientas y lenguaje de desarrollo.

- Elaboración de los artefactos de análisis y diseño.
- Realización del diseño lógico y físico de la base de datos.
- Implementación de la aplicación.
- Selección de las pruebas que se aplicarán a la aplicación.
- Realización y documentación de las pruebas seleccionadas.
- Solución a los defectos encontrados en las pruebas realizadas.

Se propone para el presente trabajo la siguiente **idea a defender**: la realización y puesta en práctica del “**Sistema de apoyo a los procesos del área de formación del Centro ISEC. (SUBAS)**” permite al Subdirector de Formación contar con una herramienta destinada a la gestión de los procesos de formación docente del área de formación del Centro ISEC.

Los **métodos investigativos** utilizados para la realización de este trabajo de diploma son los siguientes:

Métodos Teóricos:

- **Analítico - Sintético**: Permite la descomposición de los elementos que conforman la situación problemática relacionando estos elementos entre sí, además de ayudar a analizar y comprender los conceptos fundamentales relacionados con los procesos de formación y la gestión de los mismos. Se utiliza en el análisis de las herramientas, lenguajes y métodos seleccionados para elaborar la solución propuesta, así como para mantener organizada y sintetizada toda la documentación utilizada para la bibliografía a lo largo de la investigación.
- **Histórico – Lógico**: se evidencia en el análisis de la evolución e importancia de los sistemas informáticos que apoyan la gestión de los procesos de formación, así como en el estudio de sus características y objetivos.
- **Inductivo - Deductivo**: se utiliza para el planteamiento del objetivo, la idea a defender y la extracción de las ideas fundamentales.

- **Modelación:** se utiliza en la descripción de los procesos del negocio mediante el uso de la técnica de modelado IDEF0 y en la representación de los caso de uso del sistema y sus descripciones.

Al finalizar esta investigación se espera obtener una herramienta mediante la cual se logre llevar a cabo los trámites o diligencias necesarios para el desarrollo de las actividades de formación definidas para el Centro ISEC, entre las cuales se incluyen la organización de las actividades de los profesores que prestan servicio en la docencia, la formación de los estudiantes dentro de los proyectos y la defensa de los trabajos de diploma de los estudiantes vinculados al centro.

El presente documento consta de cuatro partes fundamentales: resumen, introducción, desarrollo y conclusiones. El desarrollo está estructurado en 4 capítulos los cuales están organizados de la siguiente manera:

- **Capítulo 1:** Fundamentación Teórica. En este capítulo se realiza un estudio detallado de las tecnologías a utilizar para el desarrollo de la aplicación así como de los conceptos fundamentales a los que se hará referencia a lo largo de toda la investigación.
- **Capítulo 2:** Características del sistema. En este capítulo se ofrece una visión práctica del sistema, la cual incluye los requisitos funcionales y no funcionales, así como los diagramas de caso de uso y sus descripciones.
- **Capítulo 3:** Diseño e implementación del sistema. En este capítulo se presenta una vista interna del sistema, diagrama de clases del diseño, patrones, diagrama entidad relación, diagramas de componentes y diagrama de despliegue.
- **Capítulo 4:** Pruebas del sistema. Se especifican las pruebas realizadas al sistema, incluyendo los resultados obtenidos al ejecutar las mismas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

En este capítulo se abordan conceptos relacionados con los procesos de formación docente, dentro y fuera de los centros, así como la gestión de los mismos, con el objetivo de asegurar una perspectiva real sobre el contenido del presente trabajo. Se realiza además un estudio de las herramientas de apoyo a la gestión de procesos de formación docente y las tecnologías a utilizar para el desarrollo de la solución que se propone.

1.2. Conceptos fundamentales

1.2.1. Formación docente

El término “formación” suele estar asociado a la idea de formación docente, en la cual están comprendidos los estudios y aprendizajes referentes a la inserción, reinserción y actualización laboral. La formación docente se plantea como objetivo el aumento del conocimiento y de las habilidades del estudiante en su camino hacia la labor profesional, y se divide en tres tipos: la **específica o inicial** (destinada a los estudiantes que deciden iniciarse en el mundo laboral), la **ocupacional** (para los desempleados que desean reinsertarse en el mundo del trabajo) y la **continua** (para los trabajadores en activo que quieren adquirir mayores competencias y que intentan una actualización permanente de sus capacidades, lo que supone un aumento de su empleabilidad). [1]

1.2.2. Procesos de formación docente

El término “proceso” hace referencia al avance o secuencia evolutiva de un fenómeno natural o artificial, por lo que se puede entender como procesos de formación docente, toda aquella variación en el sistema de formación educacional. El objetivo fundamental de los procesos de formación docente apunta al seguimiento e intervención durante la implementación de experiencia de mejora institucional, favoreciendo de esta manera el desarrollo profesional de los profesores y estudiantes. [2] Estos procesos aliados a la introducción de la informática y sus tecnologías suponen un avance innegable para el desarrollo de la educación en el país.

“La utilización de las tecnologías informáticas en la educación exige el interés individual en su apropiación, y la formación de docentes con elementos computacionales que le permitan aprovechar los beneficios y sumergir en ellos a sus alumnos.” [3]

María Lina Jiménez Pardo (1), Leonel Pineda Folgoso (2), Eva de la Nuez Ramos (3).

Los actores principales de los procesos de formación docente son los profesores y estudiantes, y es a partir de los contenidos asimilados que los estudiantes deben desarrollar las competencias necesarias para su posterior desempeño profesional. Los procesos de formación docente constituyen un impulso fundamental para el desarrollo de la investigación, el pensamiento, la comunicación y la creatividad de todo aquel que se vea inmerso en el mismo.

1.2.3. Gestión de procesos de formación docente

El término “gestión” se refiere a la acción de realizar diligencias con el objetivo de alcanzar una meta previamente establecida. Entre las clasificaciones del término están: **la gestión social** (que se encarga de crear espacios que permitan la interacción entre los integrantes de la población), **la gestión ambiental** (destinada a desarrollar actividades dedicadas a establecer el desarrollo sostenible de nuestra sociedad), **la gestión del conocimiento** (que hace referencia al intercambio de conocimientos y experiencias) y **la gestión de proyectos** (que básicamente se encarga de la organización y administración de los recursos en el desarrollo de un proyecto).

La gestión de procesos de formación docente está incluida en la clasificación de gestión de proyectos, donde el proyecto a gestionar sería la formación en sí misma. En los procesos de formación participan estudiantes, profesores y trabajadores no docentes que forman parte de las instituciones educativas. Dentro de las acciones que se desarrollan como parte de la gestión de dichos procesos se encuentran la definición de los planes de formación docente, la planificación del horario docente, la elaboración de los programas docentes de las disciplinas y asignaturas, la vinculación de los profesores a las actividades de formación, el control y evaluación de los estudiantes y de la actividad desarrollada por los profesores, entre otras.

1.2.4. Herramientas de apoyo a la gestión de procesos de formación docente

Las herramientas de apoyo a la gestión de los procesos de formación docente brindan la posibilidad de llevar el control sobre la prestación de servicios en el ámbito educacional. La gestión tradicional y sin apoyo tecnológico se hace cada vez menos eficaz, cualquiera sea el ámbito laboral en que se desarrolle, pues si se compara el tiempo empleado en realizar la planificación, organización y administración de un proyecto sin el empleo de las tecnologías y el tiempo empleado una vez introducidas las mismas, sería simple percatarse del ahorro de recursos, de tiempo y de cerebro evidente.

A continuación se muestra el resultado del estudio realizado sobre herramientas de apoyo a los procesos de formación docente.

1.2.5. Análisis de soluciones existentes:

Sistema de Gestión Docente (SGD Web)

URL: <http://www.sgdweb.com>

SGD Web es la página web del Sistema de Gestión Docente colombiano, que presenta entre sus funcionalidades:

- Chequeo de horarios.
- Chequeo de asistencia y evaluaciones.
- Comunicación entre padres, profesores y alumnos desde casa.

La utilización de esta herramienta permitiría al Subdirector de Formación tener el control sobre los horarios, la asistencia y las evaluaciones de los estudiantes, pero las mismas constituyen solo una pequeña porción de las actividades definidas para el área de formación de los centros de desarrollo.

Centro de Formación, Investigación y Desarrollo de Soluciones de e-Learning.

URL: <http://elearning.sceu.frba.utn.edu.ar>

Este sitio funciona como herramienta para la gestión de la formación docente en Argentina, presentando como objetivo fundamental el acercamiento e integración de los ámbitos académicos y privados. Sus funcionalidades están encaminadas a:

- Ayudar a las organizaciones de manera eficaz y eficiente con sus proyectos de e-Learning.
- Formar profesionales capaces de desenvolverse en el ámbito del e-Learning con sólidos conocimientos, calidad en su trabajo y ética profesional.
- Investigar activamente acerca del desarrollo de los recursos humanos, el capital intelectual y la tecnología relacionada con e-Learning.
- Brindar a la comunidad información relevante para la toma de decisiones en proyectos educativos.

El principal inconveniente por el cual no es posible establecer esta herramienta como alternativa de solución al problema que origina la investigación es que no satisface en su totalidad las necesidades del área de formación de los centros de desarrollo, pues solo se muestran funcionalidades referentes a la gestión de estudiantes y cursos, quedando desatendidas funcionalidades tan importantes como la gestión de profesores, proyectos y tesis.

Sistema de Gestión Universitaria

URL: <https://gestionuniversitaria.uci.cu>

El Sistema de Gestión Universitaria es una herramienta para la gestión de los procesos de formación docente de la Universidad de las Ciencias Informáticas, las funcionalidades que incluye permiten realizar las siguientes tareas:

- Gestión de estudiantes y estructuras.
- Control y desarrollo del proceso docente, lo que incluye asistencia y evaluaciones.
- Registro de las asignaturas por cada disciplina dentro de una carrera.

- Planificación de profesores y grupos por cada asignatura.

Esta herramienta permitiría al Subdirector de Formación del Centro ISEC llevar el control de las asignaturas optativas del centro, incluyendo los estudiantes que la cursaron y sus respectivas evaluaciones, sin embargo no incluye funcionalidades que permitan asignar los estudiantes a los proyectos y especificar el rol que desempeñan, relacionarlos con sus Supervisores-Evaluadores-Tutores, registrar las evaluaciones mensuales de proyecto o de los procesos de certificación de roles, certificación de investigación o competencias genéricas, asignar los trabajos de diploma a los estudiantes con sus tutores y realizar la planificación de los talleres, predefensas y defensas de tesis

Partiendo del estudio previamente realizado se arribó a la conclusión de que las herramientas mencionadas no constituyen una alternativa de solución a la problemática que origina esta investigación, pues no ofrecen las funcionalidades necesarias para satisfacer las actividades o procesos definidos para el área de formación de los centros de desarrollo. Por este motivo se propone la creación de una aplicación informática que permita la gestión de los procesos de formación docente en los centros de desarrollo.

1.3. Estudio de las herramientas y tecnologías a utilizar en el desarrollo de la solución

1.3.1. Metodología o estándar

Los estándares o metodologías definen un conjunto de procedimientos, herramientas, técnicas y soporte documental que guían la forma en que se aplica la ingeniería del software. Si a lo largo del camino no se sigue metodología alguna no se sabrá con certeza si lo que se está desarrollando conduce a donde realmente se desea llegar. [13]

Proceso Unificado Ágil (AUP)

AUP (Agile Unified Process), es una versión simplificada del Proceso Unificado de Rational (RUP), donde se describe de manera fácil la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos válidos en RUP. Entre las técnicas que utiliza AUP se encuentran el Desarrollo Dirigido por Pruebas (TDD), Modelado Ágil, Gestión de Cambios Ágil y Refactorización de Base de Datos para mejorar la productividad.

El Modelo establecido por AUP es más simple que el que aparece en RUP, pues reúne en una las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de las disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con RUP. El ciclo de vida de AUP al igual que de RUP consta de cuatro fases fundamentales:

- **Concepción:** se establece la comunicación entre el cliente y el equipo de desarrollo para que ambos lleguen a comprender el alcance del nuevo sistema.
- **Elaboración:** el equipo de desarrollo profundiza en la comprensión de los requisitos del sistema y en validar la arquitectura.
- **Construcción:** se desarrolla el sistema y se prueba por completo en el ambiente de desarrollo.
- **Transición:** se lleva el sistema a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción. [\[14\]](#)

AUP es la adopción de muchas de las técnicas ágiles de XP y otros procesos ágiles de RUP, incorporando UML como Lenguaje Unificado de Modelado e incluyendo artefactos como de la definición de los requisitos y la estimación de recursos en la fase de Concepción. Otros de los artefactos generados en la fase de Elaboración, mediante el uso de esta metodología, son el Modelo del negocio o de procesos, la descripción de la arquitectura y el modelo funcional de alto nivel. Estos artefactos se generan luego de haber realizado el análisis del dominio y de los riesgos del problema, además de la definición de la arquitectura básica y la planificación del proyecto. De esta forma quedan documentados los elementos más necesarios para una posterior mejora del sistema por un equipo de trabajo diferente al inicial.

Debido a la necesidad de utilizar una metodología no tan resumida como XP, pero tampoco tan extensa como RUP, se decidió utilizar la metodología AUP, dado el hecho de que XP no muestra de forma explícita cómo crear algunas de las herramientas que la administración necesita y RUP es el gestor que más cantidad de artefactos genera. AUP permitirá al desarrollador de la aplicación obtener un sistema funcional en un corto o mediano plazo y a su vez generar la documentación mínima necesaria para continuar con el segundo ciclo de desarrollo.

1.3.2. Herramienta CASE

Las herramientas de Ingeniería de Software Asistida por Computadoras (CASE) son aplicaciones informáticas cuyo objetivo principal va encaminado al aumento de la productividad en el desarrollo de software logrando una notable reducción de los costos en términos de tiempo y dinero. [15]

Visual Paradigm for UML Enterprise Edition 5.0

Es una herramienta CASE capaz de dar soporte al ciclo de vida completo del software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue y utiliza como lenguaje de modelado UML, que es un lenguaje estándar para especificar, visualizar, construir y documentar los artefactos del sistema.

Se seleccionó Visual Paradigm for Enterprise Edition 5.0 como herramienta de modelado por su facilidad de uso, de instalación y de actualización, y la presencia de interfaces amigables. Se puede agregar el hecho de que la herramienta soporta el ciclo de vida completo de desarrollo de software y que es una de las herramientas utilizadas en la universidad por lo cual se cuenta con la licencia para su uso. Esta herramienta permitirá la generación de artefactos tales como: diagrama de casos de uso del sistema, diagrama de clases del diseño, diagrama de despliegue, y diagrama entidad relación. Se podrán utilizar los estereotipos web que incluye para los diagramas de clases del diseño y generar el script de la base de datos a partir del diagrama entidad relación.

1.3.3. Definición de la integración para el modelado de las funciones (IDEF0)

La Definición de la integración para el modelado de las funciones (IDEF0) es una técnica de modelación utilizada para representar de manera estructurada y jerarquizada las actividades, los objetos y datos que soportará la herramienta propuesta.

Un modelo IDEF0 se compone de una serie jerárquica de diagramas que permiten mediante niveles de detalle, describir las funciones especificadas en el nivel superior. En las vistas superiores del modelo la interacción entre las actividades representadas permite visualizar los procesos fundamentales que sustentan la organización.

Los elementos gráficos utilizados para la construcción de los diagramas IDEF0 son cuadros y flechas. El nombre de la caja será un verbo o una frase verbal que es descriptiva de la función que la caja representa. Las etiquetas de las flechas tendrán un significado u otro según a qué lado de la caja se encuentren, pueden representar entradas, salidas, control y sujeto.



La utilización de esta técnica facilitará la comunicación entre los desarrolladores y el cliente, porque permite modelar gráficamente los procesos de negocio con un alto nivel de detalle, de forma coherente y simple, sin dejar de ser rigurosa y exacta. Permite incorporar en la descripción de los procesos los datos que entran y salen en cada actividad. En el desarrollo de la solución propuesta permite además lograr una descripción detallada de los procesos del negocio en un corto período de tiempo pues no es necesario generar un gran número de artefactos, a diferencia de un modelo de negocio descrito a través de casos de uso y sus especificaciones detalladas.

1.3.4. Dia 0.97.1

Dia es una aplicación informática multiplataforma surgida para la creación de diagramas técnicos. Su interfaz y sus funciones se basan en Windows Visio ya que es un sustituto para esta herramienta. Entre sus características se incluye la exportación a muchos formatos (EPS, SVG, JPEG y PNG). Dia es útil para realizar el modelado de los procesos de un negocio utilizando técnicas de modelado como es el caso de IDEF 0. [\[20\]](#)

1.3.5. Aplicaciones informáticas

Las aplicaciones informáticas son los programas que permiten la interacción entre usuarios y computadoras, brindándoles la posibilidad a los usuarios de elegir opciones y ejecutar acciones que el programa ofrece. Existen dos clasificaciones fundamentales referentes a las aplicaciones informáticas: aplicaciones de escritorio y aplicaciones web.

Aplicaciones de escritorio

Las aplicaciones de escritorio o aplicaciones desktop, son las aplicaciones instaladas en el ordenador del usuario para ser ejecutadas sobre un sistema operativo de interfaz visual como Microsoft Windows, Mac OS X, Solaris o Linux. Su rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video, etc.

Desventajas de las aplicaciones de escritorio

Entre las principales desventajas de las aplicaciones de escritorio se encuentra su acceso limitado al servidor donde están instaladas, su dependencia del sistema operativo utilizado por el ordenador, la dispersión de la información y lógica en cada computadora que requiera de su uso, así como su necesidad de instalación y actualización personalizada.

Aplicaciones Web

Las aplicaciones Web son aplicaciones que los usuarios pueden utilizar a través de Internet o de una Intranet mediante un navegador. Para garantizar el funcionamiento y la eficiencia de una aplicación web se necesita un navegador web que cumpla con los estándares propuestos por el World Wide Web Consortium² (W3C) y que esté conectado al servidor web donde se almacena la aplicación.

Ventajas de las aplicaciones web

Las principales ventajas de las aplicaciones web explican el nivel de popularidad alcanzado entre los usuarios. Una de estas ventajas explica por qué en muchos casos se requiere del desarrollo de una aplicación web y no de una desktop, pues el desarrollo de una aplicación web brinda la posibilidad a los

usuarios de acceder al sistema cualquiera sea su ubicación. Otra de las características ventajosas de las aplicaciones web radica en el hecho de ser multiplataforma, pues con tan solo un programa y un navegador las aplicaciones pueden ser utilizadas desde múltiples plataformas. La actualización instantánea permite que el usuario siempre utilice la última versión del sistema debido a que todos los usuarios de la aplicación acceden a un mismo programa que radica en el servidor. [5]

Con la finalidad de desarrollar una herramienta que brinde el soporte necesario para llevar a cabo los procesos de formación docente que se realizan en los centros de desarrollo de la universidad se propone la creación de una aplicación web, como aplicación informática, que permita al Subdirector de Formación planificar, organizar y controlar las actividades de formación de los estudiantes y profesores vinculados al centro.

Para el desarrollo de la solución que se propone es necesario tener en cuenta algunas herramientas y tecnologías tales como los lenguajes de programación que se utilizarán, a partir de los cuales se realizará la selección del servidor web, el sistema gestor de base de datos, el entorno de desarrollo integrado en que se editará y depurará el código y el framework o marco de trabajo. Aunque no son dependientes de la selección de los lenguajes de programación se estudiaron herramientas para el modelado de los diagramas y procesos referentes a la fase de Análisis y Diseño de la solución propuesta. Estos elementos serán explicados a continuación.

1.3.6. Lenguajes de programación

En la actualidad se cuenta con un gran número de lenguajes de programación que responden a las necesidades de las plataformas existentes. Estos lenguajes surgen en el transcurso del tiempo con el avance de las tecnologías y la aparición de nuevos problemas. La selección de los lenguajes de programación es de vital importancia, siendo estos los principales elementos para la creación de los programas mediante un conjunto de instrucciones, operadores y reglas. A continuación se presentan los lenguajes de programación para la web.

Lenguaje de programación del lado del cliente

Los lenguajes de programación del lado del cliente son totalmente independientes del servidor y se usan para la programación de scripts y programas que se ejecutan en navegadores o clientes web. Su principal ventaja es que su ejecución depende del cliente con lo que se evita que el servidor esté cargado de trabajo innecesariamente. [6]

1.1. JQuery 1.3.4

JQuery es un framework JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Es software libre y de código abierto y ofrece funcionalidades basadas en JavaScript que de otra manera necesitarían mucho más código. Entre sus acciones fundamentales se encuentran la manipulación de hojas de estilos CSS, así como la creación de efectos y animaciones personalizadas. [21]

JavaScript 1.3.4

Es un lenguaje interpretado del lado del cliente, basado en objetos, guiado por eventos y diseñado específicamente para el desarrollo de aplicaciones cliente-servidor. Se integra dentro del código HTML de las páginas Web y responde a eventos en tiempo real.

El lenguaje JavaScript es interpretado por la mayoría de los navegadores más usados como son Netscape, Internet Explorer, Mozilla Firefox, Opera, entre otros. [6]

Entre las ventajas de este lenguaje de programación se destacan su sencilla curva de aprendizaje, su integración con HTML, la reutilización de códigos de programación implementados y su independencia de la plataforma hardware o sistema operativo.

HTML 5

HTML es el acrónimo de HyperText Markup Language o Lenguaje de Mercado de Hipertextos. Es considerado el sistema idóneo para compartir información a través de Internet por su condición de no ser privativo y por su usabilidad. Desde su creación se pretendió que las etiquetas fueran capaces de marcar la información de acuerdo a su significado, pero luego fueron añadiendo más etiquetas dirigidas a

controlar la representación de la información contenida en el documento. Los documentos creados en HTML incluyen imágenes, sonidos, videos y textos. [6]

CSS 1.3.4

Hojas de Estilo en Cascada (CSS), es un mecanismo que se encarga de describir la forma en que se va a mostrar un documento en la web. Este mecanismo le brinda a los desarrolladores la posibilidad de ejercer el control sobre el estilo y formato de los documentos, tanto HTML como XML, separando el contenido de la presentación. Son los estilos los que brindan la posibilidad de controlar el formato de múltiples páginas Web al mismo tiempo, de manera que si se modifica un elemento dentro de una CSS todas las páginas vinculadas a esta percibirán el cambio. [7]

Se decide utilizar como lenguaje de programación del lado del cliente HTML, CSS y JavaScript por la gran integración que poseen entre ellos. JavaScript será fundamental en la generación de formularios, CSS apoyará el diseño de interfaces permitiendo que los cambios realizados en la hoja de estilos modifiquen de inmediato las páginas en las que esté aplicada y HTML para la creación de las plantillas Twig a utilizar en el diseño del sistema.

Lenguaje de programación del lado del servidor

Los lenguajes del lado del servidor se encargan de las conexiones a la base de datos y al servidor web, así como de la generación de páginas que serán enviadas al cliente.

PHP 5.3.8

Se decidió utilizar PHP en su versión 5.3.8 por ser un lenguaje multiplataforma, de distribución gratuita bajo licencia abierta, totalmente orientado a la creación de aplicaciones web dinámicas con acceso a información almacenada en una base de datos y con una curva de aprendizaje muy corta. Este lenguaje permite la reutilización de código y puede ser utilizado en cualquiera de los principales sistemas operativos del mercado. Su código fuente es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable. [7]

PHP proporciona soporte para la mayoría de los protocolos de comunicación entre los cuales se destaca el protocolo de transferencia de hipertexto (HTTP), posibilita la conexión con PostgreSQL como manejador de base de datos y soporta Apache como servidor de aplicaciones. Este lenguaje es capaz de prescindir de la previa definición del tipo de variable a utilizar, así como de leer y manipular datos desde diversas fuentes, incluyendo los que son ingresados por los usuarios a partir de los formularios HTML.

Partiendo de la selección de PHP como lenguaje de programación del lado del servidor se fundamentará la selección de la tecnología a utilizar.

1.2. Servidores Web

Los servidores web son programas que se ejecutan de forma continua implementando el protocolo HyperText Transfer Protocol o Protocolo de Transferencia de Hipertexto (HTTP), manteniéndose a la espera de las peticiones de ejecución realizadas por un cliente o navegador. Su función principal es dar respuesta a estas peticiones, entregando como resultado una página web u otro tipo de información en dependencia de los comandos solicitados. Entre los servidores existentes se encuentran Microsoft, Sun Java™ System Web Server, Google, NCSA y Apache. [8]

Apache 2.2.21

Los servidores de aplicaciones Apache representan el complemento perfecto para el desarrollo de páginas web dinámicas por su robustez, configurabilidad y estabilidad, lo que los convierte en servidores web de excelencia, gratuitos y de código abierto, que son además multiplataforma y permiten la configuración de las respuestas ante los posibles errores que se puedan generar.

Estos servidores brindan una ventaja fundamental para el desarrollo web, la fiabilidad, pues los servidores con más alta disponibilidad funcionan bajo un servidor Apache. Otras de las ventajas que poseen estos servidores web radican en el hecho de que permiten mostrar una página web sin necesidad de albergarla en un web hosting o alojamiento web, además de probar el avance de la misma vía local y acceder desde equipos remotos. Es a razón de estas ventajas y de las características de este que se decide hacer uso del mismo para el desarrollo de esta aplicación.

1.3. Arquitectura de Software

La Arquitectura de Software se define como el diseño de más alto nivel de la estructura de un sistema conformado por patrones y abstracciones coherentes que conforman el marco de referencia necesario para guiar la construcción del software para un sistema de información en el cual, se definen de manera abstracta los componentes con tareas asignadas, sus interfaces y la comunicación entre ellos. [9]

Arquitectura Cliente – Servidor

Es una arquitectura que permite la comunicación entre cliente y servidor, donde el servidor es una aplicación que ofrece servicios a usuarios de Internet. A partir de una solicitud de servicio realizada al servidor por parte de un cliente, el servidor devuelve los resultados en forma de respuesta a dicha solicitud. Bajo esta forma de comunicación la información se puede mantener centralizada, evitando la pérdida de la misma por problemas de hardware u otros motivos relacionados con situaciones técnicas que puedan presentarse con los clientes.

Con la implantación de este modelo cada cliente tiene la posibilidad de trabajar desde cualquier lugar solamente haciendo una solicitud al servidor y manipulando la información que este le devuelva en forma de respuesta, en dependencia de las necesidades del cliente. Entre las ventajas más relevantes que brinda esta arquitectura para el desarrollo de esta aplicación se encuentran:

- **Centralización del control:** controla los accesos, recursos y la integridad de los datos de forma que un programa cliente defectuoso o no autorizado, no pueda dañar el sistema.
- **Fácil mantenimiento de los servidores:** como generalmente el servidor está en un ordenador independiente es posible reparar, reemplazar o actualizar sin que el cambio afecte a sus clientes. [10]

Patrón arquitectónico Modelo Vista Controlador (MVC)

EL término Patrón Arquitectónico se refiere al nivel en el cual la arquitectura de software define la estructura básica de un sistema, en la misma forma en que representa una plantilla de construcción que

proporciona un conjunto de subsistemas aportando las normas para su organización. El Patrón Arquitectónico MVC divide las aplicaciones en tres niveles de abstracción:

- **Modelo:** representa la lógica del negocio, se encarga de acceder directamente a los datos actuando como intermediario con la base de datos.
- **Vista:** es la página HTML encargada de mostrar la información al usuario de forma gráfica.
- **Controlador:** actúa de intermediario entre la vista y el modelo. Es el código que obtiene los datos dinámicamente mediante el cual controla las interacciones del usuario solicitando los datos al modelo y le entrega a la vista el contenido HTML generado. [\[11\]](#)

El empleo de este patrón resulta sumamente ventajoso para el desarrollo de la aplicación que se propone pues las vistas proveen mayor flexibilidad y agilidad, así como pueden ser anidadas y sincronizadas. Se pueden crear múltiples vistas de un modelo y cambiar el modo en que una vista responde al usuario sin cambiar su representación visual. El empleo de este patrón arquitectónico facilita el mantenimiento del sistema.

1.4. Sistemas Gestores de base de datos (SGBD)

Una base de datos es el lugar donde se almacenan los datos en reposo, al cual podrán acceder las aplicaciones de una organización. Un Sistema Gestor de Bases de Datos (SGBD), es un conjunto de programas que permiten al usuario acceder y modificar estos archivos mediante varios niveles de abstracción (nivel físico, nivel lógico y nivel de vistas), garantizando de esta manera la seguridad e integridad de los mismos.

Los SGBD poseen numerosas ventajas entre las que se encuentran:

- Protección de información, modificaciones, inclusiones y consultas.
- Independencia del tratamiento de información.
- Facilidad de manejo de grandes volúmenes de información.

- Seguridad de la información (acceso a usuarios autorizados).
- Gran velocidad de ejecución de las consultas.

Dentro de los gestores de base de datos más conocidos mundialmente se encuentran MySQL y PostgreSQL.

PostgreSQL 8.3

PostgreSQL es un SGBD relacional, publicado bajo la licencia BSD (Berkeley Software Distribution), con amplia variedad de tipos nativos, así como gran escalabilidad, alta concurrencia, funciones de ventanas, expresiones de tablas comunes y consultas recursivas.

PostgreSQL tiene la capacidad de comprobar la integridad referencial, así como la de almacenar procedimientos en la propia base de datos. Tiene además soporte completo para claves foráneas, uniones y procedimientos almacenados. Incluye la mayoría de los tipos de datos (INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL y TIMESTAMP). También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o video.

Se seleccionó PostgreSQL como SGBD para el tratamiento y almacenamiento de la información por ser un motor de base de datos que consta de instalaciones ilimitadas y de un diseño para ambientes de amplio volumen, que posee como ventaja su gran escalabilidad. Esto lo hace capaz de ajustarse al número de computadoras y a la cantidad de memoria que posea el sistema de forma óptima, soportando así una mayor cantidad de peticiones simultáneas de manera satisfactoria. PostgreSQL implementa además el uso de sub-consultas y transacciones mediante las cuales se ofrecen soluciones en campos en los que MySQL no podría, por lo que se considera la mejor elección a utilizar en la aplicación a desarrollar.

1.5. Entorno de desarrollo integrado (IDE)

Los IDE's (Integrated Development Environment) son los editores de código, compiladores, depuradores y constructores de interfaz gráfica que pueden funcionar con varios lenguajes de programación. Son considerados también como ambientes estructurados para la implementación de aplicaciones, los cuales facilitan en ocasiones el desarrollo de Interfaces de usuarios (GUI), apoyan el acceso a la información

referente a funciones (API), variables y procedimientos, y corrigen además los posibles errores léxicos, de sintaxis y semánticos.

Los IDE deben ser capaces de dar cumplimiento a los siguientes requisitos:

- Reconocimiento de sintaxis.
- Extensiones y Componentes para el IDE.
- Importar y exportar proyectos.
- Múltiples idiomas.
- Manual de usuario y Ayuda.

NetBeans 7.1

NetBeans es un entorno de desarrollo gratuito para los desarrolladores. Presenta potentes herramientas para el desarrollo de aplicaciones de escritorio profesionales, aplicaciones Web y aplicaciones móviles utilizando los lenguajes Java, C /C++, así como los lenguajes dinámicos PHP, JavaScript, Groovy y Ruby. Funciona en sistemas operativos compatible con la máquina virtual Java (Windows XP, Vista, Windows 7, Ubuntu 9.10, Solaris, Mac OS X 10.5 o superior).

NetBeans 7.1 es la última versión estable del entorno de desarrollo integrado que ofrece características para ayudar en la construcción y mantenimiento de aplicaciones web y móviles, siendo esta la razón principal para su selección como IDE para el desarrollo de esta aplicación. [12] Para este proceso de selección se tuvo en cuenta además el hecho de que permite el desarrollo de aplicaciones multiplataforma además de que soporta el lenguaje de programación seleccionado, se valoró la inclusión de herramientas para el desarrollo de aplicaciones profesionales y la presencia de una interfaz amigable e intuitiva.

1.6. Marco de trabajo o Framework

Un framework de desarrollo es una estructura de trabajo y soporte definida y utilizada por otro software con la finalidad de llevar a cabo un trabajo más organizado. Dentro de un framework de desarrollo se

pueden encontrar soportes de programas, bibliotecas y un lenguaje interpretado entre otros software para facilitar el desarrollo y unir las diferentes estructuras y componentes de un proyecto.

Symfony 2.0

Symfony es un framework para aplicaciones Web que acelera la creación y mantenimiento de las aplicaciones Web y permite la reutilización de código, garantizando y agilizando el desarrollo de las mismas. Entre sus características fundamentales radica el hecho de que separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web y proporciona clases y herramientas con vistas a minimizar el tiempo de desarrollo de una aplicación compleja. Otra característica fundamental es que automatiza las tareas comunes, permitiendo al desarrollador centrarse en los aspectos específicos de cada aplicación.

Symfony2 es un framework fácil de instalar y configurar en la mayoría de las plataformas existentes, independiente del SGBD, que se rige por diversos patrones de diseño. Presenta un código legible y es fácil de extender, lo que permite la integración con librerías desarrolladas por terceros.

Entre las características que influyeron en la selección de Symfony2 como framework para el desarrollo de la aplicación que se propone se encuentra el uso de plantillas en la capa de presentación, la validación automatizada, las técnicas de escape para evitar inyección de código y la paginación automatizada. La autenticación y la gestión de credenciales simplifican la creación de secciones restringidas y la gestión de la seguridad de usuario.

1.7. Conclusiones

Las herramientas informáticas de apoyo a los procesos de formación docente estudiadas en el presente capítulo no constituyen una alternativa de solución a la problemática planteada pues no permiten la gestión de todos los procesos definidos para el área de formación del Centro ISEC. Se definió entonces la necesidad de desarrollar una herramienta para apoyar las tareas del Subdirector de formación, para lo cual se identificaron las metodologías, herramientas y técnicas necesarias para su elaboración.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

Los procesos de desarrollo de software tienen como objetivo fundamental el desarrollo de un producto de software que reúna los requisitos manifestados por el cliente. Es por ello, que para lograr un mejor entendimiento de las características del sistema propuesto, en este capítulo se muestra parte del resultado del proceso donde se exponen los objetivos estratégicos de la organización, los temas relacionados con la fase de exploración y planificación de la metodología AUP y los artefactos generados. Para una mejor comprensión de las características específicas de esta aplicación se realiza la descripción de los requisitos funcionales y no funcionales a tener en cuenta para el diseño e implementación de la solución propuesta.

2.2. Objetivos de la Organización

ISEC es un centro dedicado al desarrollo de sistemas y servicios informáticos integrales de alta calidad y competitividad en el mejoramiento de la atención a emergencias ciudadanas, así como el tratamiento a la prevención del delito y la gestión de la información procedente de los órganos de seguridad ciudadana, capaz de integrar los procesos docentes, productivos e investigativos, garantizando la formación y superación de los estudiantes y especialistas de las diferentes áreas, los clientes y la organización.[16]

2.3. Objeto de Informatización

El uso de las herramientas ofimáticas en el desarrollo de las actividades o procesos definidos para el área de formación del centro ISEC ralentiza el buen desempeño del Subdirector de Formación. Teniendo en cuenta que la subdirección de formación de un Centro de Desarrollo es una de las áreas que mayor cantidad de actividades incluye, y el hecho de que la información a consultar por el Subdirector de Formación se encuentre en diversas fuentes y formatos hace el trabajo más engorroso y toma más tiempo del que pudiera estar disponible, por lo que se propone la implementación de una aplicación que apoye el desarrollo de estos procesos o actividades.

Las actividades que realiza el Subdirector de Formación son las siguientes:

- Planificación en cada semestre de las asignaturas optativas que recibirán los estudiantes del centro así como los profesores que las impartirán.
- Orientación a los Supervisores-Evaluadores-Tutores y líderes de proyecto de la realización de los talleres y reuniones mensuales y semestrales de proyecto para otorgar y analizar las evaluaciones a los estudiantes en cada período de tiempo.
- Organización y guía de los procesos de certificación de roles, certificación de investigación y evaluación por competencias genéricas.
- Orientación y explicación a los profesores de las asignaturas de Práctica Profesional sobre el proceso de obtención de las notas finales de estas asignaturas, la cual incluye la nota final de proyecto que recibe el estudiante, así como los resultados obtenidos en la evaluación de competencias genéricas, certificación de roles o certificación de investigación, según el año del estudiante.
- Asignación a cada estudiante de 5to año del centro de un trabajo de diploma y un tutor para el desarrollo del mismo.
- Planificación de los talleres, predefensas y defensas de tesis.

2.4. Propuesta de la aplicación

El sistema de apoyo a los procesos del área de formación docente del Centro ISEC es un sistema de soluciones informáticas que se encarga de todo lo referente a la gestión de los procesos de formación docente de un centro de desarrollo. Para llevar a cabo el desarrollo de este sistema se propone una aplicación web a la que podrá acceder el administrador (Subdirector de formación), para realizar la gestión de estudiantes, profesores, cursos, proyectos, tesis y tribunales mediante la web, así como la planificación de predefensas y defensas de tesis, la asignación de evaluaciones a los estudiantes y la realización de la matrícula de los mismos a los cursos optativos obligatorios. Además del Subdirector de Formación.

2.5. Modelado del negocio

Durante el análisis de la solución propuesta se identificaron los principales procesos del negocio que debían ser tratados y mediante el uso de la técnica de modelado IDEF0 se obtuvieron los diagramas referentes a dichos procesos del negocio.

2.5.1. Descripción del proceso base del negocio

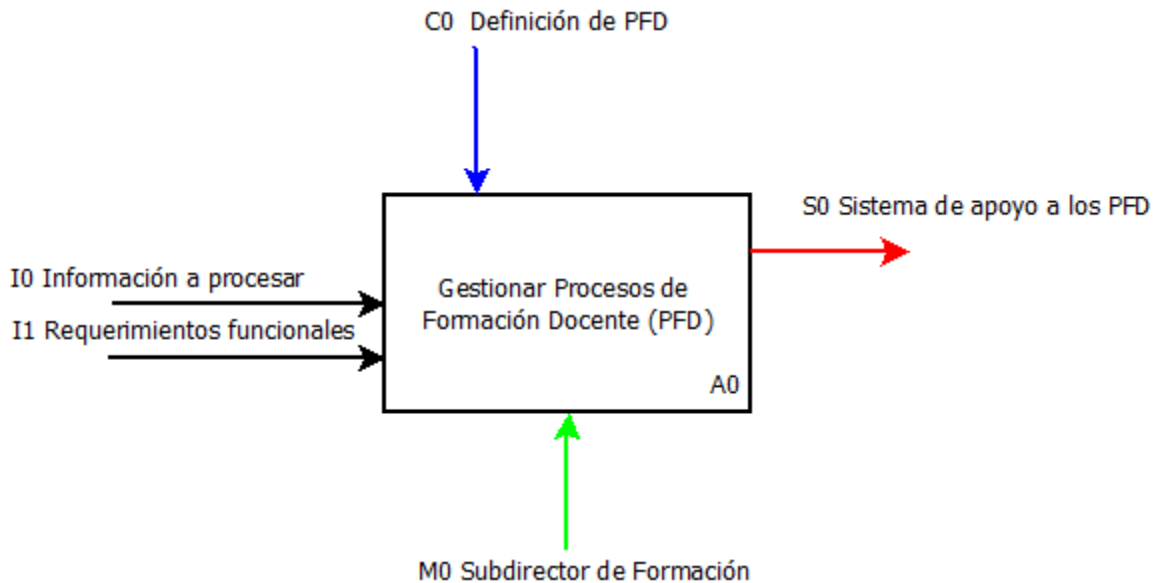


Fig. 1 Proceso base del negocio.

Para la realización del proceso base **Gestionar PFD** se necesita tener un dominio total de las informaciones referentes a las entidades que forman parte del proceso de gestión en sí, así como de los requisitos funcionales que regirán la aplicación. Este proceso se limita a los procesos o actividades del área de formación docente del Centro ISEC, además se necesita de la completa interacción del Subdirector de Formación del Centro ISEC para obtener como resultado el Sistema de apoyo a los Procesos de Formación Docente.

2.5.2. Descripción de los sub-procesos del negocio.

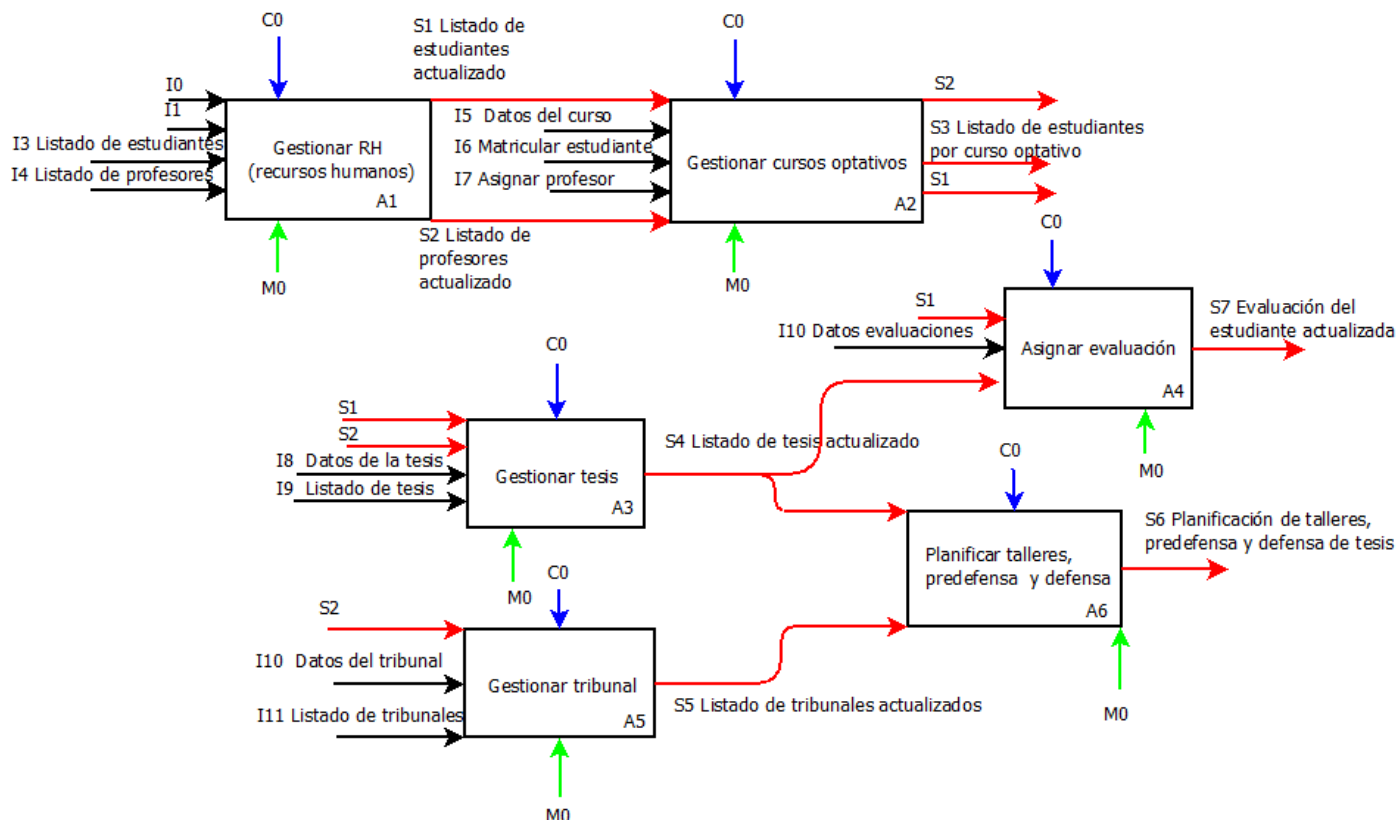


Fig. 2 Sub- procesos del negocio.

- **Gestionar RH:** luego de tener el dominio necesario sobre las informaciones a procesar y los requisitos funcionales, además de los listados de estudiantes y profesores, el sub-proceso “Gestionar RH”, condicionado por los Procesos de Formación y con la completa interacción del Subdirector de Formación del Centro ISEC, obtiene como resultado final la actualización de los listados, tanto de estudiantes como de profesores y administradores del sistema.
- **Gestionar cursos optativos:** a partir de la última actualización realizada a los listados de estudiantes y profesores, además de los datos específicos de los cursos y de la matrícula de estudiantes, este sub-proceso, condicionado por los Procesos de Formación Docente y con la

interacción del Subdirector de Formación obtiene como resultado los listados actualizados de estudiantes y profesores, así como el listado de estudiantes por cursos optativos.

- **Gestionar tesis:** a partir de la última actualización realizada a los listados de estudiantes y profesores, además de los datos específicos de las tesis y el listado de las tesis existentes, este sub-proceso, condicionado por los Procesos de Formación Docente, obtiene como resultado el listado actualizado de tesis contando con la completa integración del Subdirector de Formación.
- **Asignar evaluación:** a partir del dominio de los listados actualizados de estudiantes y de tesis, así como de los datos específicos de las evaluaciones existentes, este sub-proceso, que está condicionado por los Procesos de Formación Docente y que cuenta con la completa integración del Subdirector de Formación, obtiene como resultado la actualización de las distintas evaluaciones asignadas a los estudiantes.
- **Gestionar tribunal:** a partir de la última actualización realizada a los listados de profesores, y teniendo en cuenta el listado de tribunales y sus datos específicos, este sub-proceso, condicionado por los Procesos de Formación Docente, obtiene como resultado el listado actualizado de tribunales contando con la completa integración del Subdirector de Formación.
- **Planificar talleres, pre-defensa y defensa de tesis:** partiendo del listado actualizado de tesis y de tribunales, este sub-proceso, que está condicionado por los Procesos de Formación Docente y con la completa interacción del Subdirector de Formación, obtiene como resultado la planificación de los talleres, pre-defensas y defensas de tesis.

2.5.3. Actor del negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externa con lo que el negocio interactúa. Lo que se modela como actor es el papel que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. [17]

Actor	Descripción
Subdirector de formación	Se encarga del manejo de la información referente a los procesos asociados al área de formación del Centro ISEC.

Tabla. 1 Definición de actores del negocio

2.5.4. Trabajador del negocio

Un trabajador del negocio es una abstracción de una persona, máquina o sistema automatizado que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando las entidades del mismo. [17]

Trabajador	Descripción
Subdirector de formación	Se encarga del manejo de la información referente al negocio. Es la persona que organiza las actividades de Formación Docente del Centro ISEC.

Tabla. 2 Definición de trabajadores del negocio

Entidades: Las entidades son las clases que van a guardar la información perdurable en el tiempo

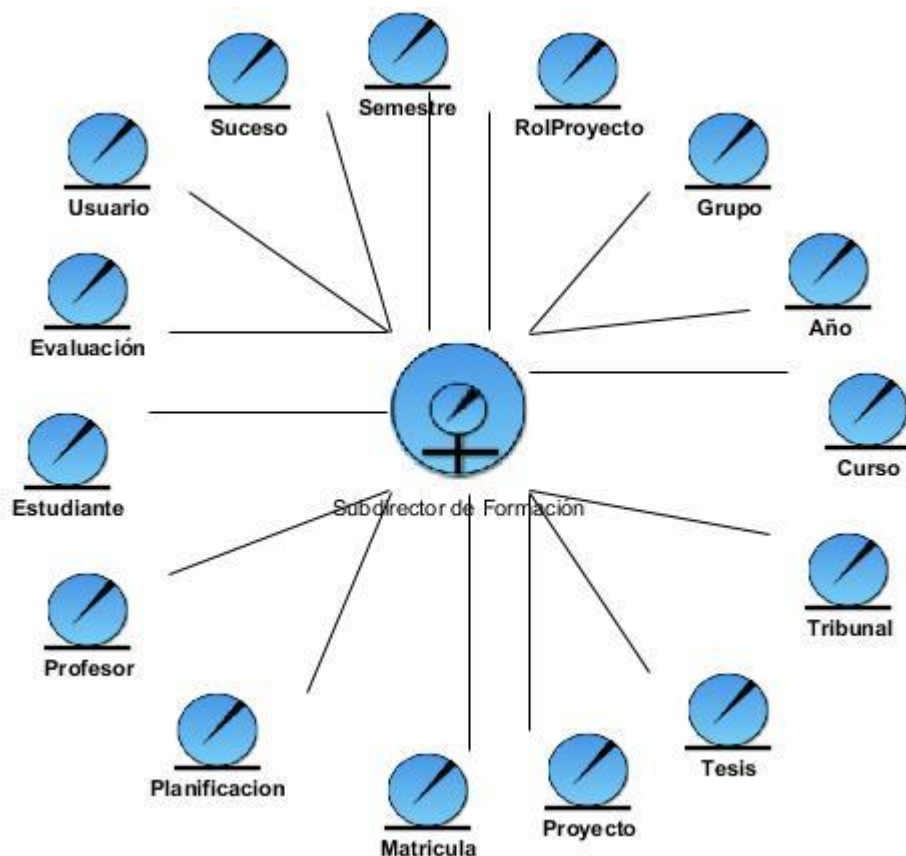


Fig. 3 Modelo de Objetos.

En la Fig. 3 se muestran las entidades o clases que intervienen en los procesos de gestión de formación docente del Centro ISEC y la relación que se establece con el trabajador del negocio, en este caso el Subdirector de Formación.

2.5.5. Reglas del negocio

Las reglas del negocio permiten definir los límites lógicos de las aplicaciones. En ellas las aplicaciones reflejan las restricciones que existen, evitando la realización de acciones no válidas. Estas reglas concretan y controlan la estructura, la actividad y la estrategia de una organización. Pueden estar definidas en manuales de procedimiento, contratos, o pueden existir como conocimiento o experiencia que tienen los empleados.

Las reglas del negocio identificadas para la solución que se propone son las siguientes:

- El estudiante no puede estar matriculado en más de dos cursos optativos en un mismo semestre.
- El estudiante no puede ser matriculado dos veces en un mismo curso optativo a lo largo de toda la carrera.
- El estudiante no puede tener más de un trabajo de diploma asignado.
- El estudiante no puede recibir:
 - Más de una evaluación mensual en el mismo mes y año.
 - Más de una evaluación final para cada asignatura de Práctica Profesional en cada curso.

2.5.6. Especificación de requisitos de software

Un proyecto no puede ser exitoso sin una descripción detallada, correcta y exhaustiva de los requisitos que definen lo que debe hacer un sistema y la forma en que debe hacerlo. La especificación de los requisitos ayuda a establecer y mantener el acuerdo con los clientes o interesados en la aplicación y proporciona a los desarrolladores del sistema una mejor comprensión de los requisitos, define las fronteras del software. Establece además una base para planificar el contenido técnico de las iteraciones, costo y tiempo de desarrollo del sistema. Mediante la especificación de requisitos se define una interfaz para el usuario enfocado en las necesidades y metas del mismo.

Requisitos funcionales

Los requisitos funcionales son aquellos que permiten plantear las principales funcionalidades que puede llegar a tener el sistema propuesto. [17]

Para el desarrollo de este sistema se han definido los siguientes requisitos funcionales:

NO.	REQUISITO	DESCRIPCIÓN
1.	Autenticar usuario	Somete al usuario a un proceso de validación con

		vistas a que solo los administradores del sistema tengan acceso.
2.	Registrar administrador	Adiciona un nuevo usuario y le asigna el rol de administrador.
3.	Buscar administrador	Busca los usuarios a partir de un criterio de búsqueda. Criterios de búsqueda: usuario, solapín.
4.	Registrar estudiantes	Adiciona nuevos estudiantes.
5.	Modificar estudiantes	Modifica la información de los estudiantes registrados.
6.	Eliminar estudiantes	Elimina a los estudiantes previamente registrados.
7.	Buscar estudiantes	Busca a los estudiantes a partir de un criterio de búsqueda. Criterios de búsqueda: nombre, solapín, proyecto, rol.
8.	Registrar profesor	Adiciona un nuevo profesor.
9.	Modificar profesor	Modifica la información de los profesores registrados.
10.	Eliminar profesor	Elimina la información de un profesor previamente registrado.
11.	Buscar profesor	Busca a los profesores a partir de un criterio de búsqueda. Criterios de búsqueda: nombre, solapín, departamento.
12.	Gestionar evaluaciones del estudiante	Gestiona evaluaciones mensuales, semestrales, de proyecto, de taller, de tesis, de práctica profesional, de competencias genéricas, así como de certificación de roles y de investigación.
13.	Registrar curso	Adiciona los Cursos que se impartirán.

14.	Modificar curso	Modifica los datos referentes a un Curso.
15.	Eliminar curso	Elimina un Curso.
16.	Buscar curso	Busca los Cursos a partir de un criterio de búsqueda. Criterios de búsqueda: nombre, semestre, año, descripción.
17.	Realizar matrícula	El subdirector matricula a los estudiantes en los cursos.
18.	Buscar matrícula	Busca las matrículas realizadas, a partir de un criterio de búsqueda. Criterios de búsqueda: curso, estudiante.
19.	Registrar tesis	Adiciona una nueva tesis.
20.	Modificar tesis	Modifica la información de las tesis registradas.
21.	Eliminar tesis	Elimina una Tesis de la base de datos.
22.	Buscar tesis	Busca las tesis a partir de un criterio de búsqueda. Criterios de búsqueda: título, autor, co-autor, tutor, co-tutor.
23.	Registrar tribunales	Adiciona un nuevo tribunal.
24.	Modificar tribunal	Modifica la información de los tribunales registrados.
25.	Eliminar tribunal	Elimina un tribunal de la base de datos.
26.	Buscar tribunal	Busca los tribunales a partir de un criterio de búsqueda. Criterios de búsqueda: No. Tribunal, presidente, secretario, vocal
27.	Registrar planificación	Adiciona una nueva planificación.

28.	Modificar planificación	Modifica la información de las planificaciones registradas.
29.	Eliminar planificación	Elimina una planificación de la base de datos.
30.	Buscar planificación	Busca las planificaciones a partir de un criterio de búsqueda. Criterios de búsqueda: evento, tesis, tribunal, lugar, fecha, hora, descripción.
31.	Registrar proyecto	Adiciona un nuevo proyecto.
32.	Modificar proyecto	Modifica la información de los proyectos registradas.
33.	Eliminar proyecto	Elimina un proyecto de la base de datos.
34.	Buscar proyecto	Busca los proyecto a partir de un criterio de búsqueda. Criterios de búsqueda: nombre, ubicación, teléfono.

Tabla 3. Requisitos funcionales

Requisitos no funcionales

Los requisitos no funcionales expresan las cualidades que deben tener los sistemas, tales como las características de la aplicación que señalan restricciones en la misma y hacen el producto atractivo, usable, rápido o confiable. Con vistas a desarrollar una aplicación de éxito se definieron los siguientes requisitos no funcionales:

Apariencia o interfaz externa

La aplicación presentará una interfaz agradable a la vista del usuario sin perder la profesionalidad y seriedad de su diseño, el cual se realizará acorde a la organización del contenido y en base a los requisitos funcionales definidos. Se realizará la selección de los colores azul, blanco y gris para el

diseño, se incorporarán sólo las imágenes necesarias para la comprensión de los procesos de formación y se utilizará tipo de letra **Arial**.

Usabilidad.

Se debe garantizar un acceso fácil y rápido a los usuarios autorizados. Al sistema solo accederán los usuarios autorizados por la administración del mismo. El subdirector de formación al autenticarse con su usuario y contraseña podrá acceder a todas las opciones del sistema, incluyendo las de administración del mismo.

Seguridad

Confidencialidad: El sistema será utilizado sólo por el Subdirector de Formación, este usuario tendrá permisos de administración sobre la aplicación.

Integridad: Hará uso de la aplicación el subdirector de formación, el cual tendrá permisos de administración, por lo que la información que se modifique en el sistema será responsabilidad de este usuario.

Disponibilidad: Se debe garantizar el acceso pleno a los usuarios autorizados para el uso de la aplicación las 24 horas del día. En este caso se podrán programar las acciones de soporte a la aplicación los días sábados no laborables o domingos para no afectar el uso del mismo en el resto de los días de la semana.

Político – Culturales

En la aplicación se deberá utilizar idioma español. Contará con logotipos e imágenes que se encuentren en correspondencia con el centro ISEC y los procesos de formación. Los cambios que se requieran realizar deben ser previamente consultados con la subdirección del centro ISEC.

Software

Los requisitos de software son los siguientes:

- La máquina servidor deberá disponer del servidor web Apache versión 2.2.21 o superior y contar con el SGBD PostgreSQL 8.3.
- El lenguaje de programación debe ser PHP versión 5.3.8.
- Las máquinas clientes podrán contar con sistema operativo Windows XP o superior o cualquier distribución de Linux.
- El acceso al sistema se deberá realizar a través de navegadores web como Firefox, Google Chrome, Internet Explorer 6.0 o superior.

Hardware

Los requisitos de hardware que se deberán tener en cuenta para el desarrollo de esta aplicación son los siguientes:

- Procesador Intel Dual Core 1.8 GHz, AMD Dual Core 1.8 GHz (o superior en cualquiera de ellos)
- Memoria RAM de 1 GB (800 MHz)
- Disco Duro con al menos 2 GB disponibles.
- CD –ROM: 12x o superior que soporte auto inicialización.

2.5.7. Descripción del sistema propuesto

Actor del sistema

Un actor del sistema es cualquier persona externa, proceso u objeto que interactúa con un sistema, subsistema o clase. Un actor del sistema es un trabajador del negocio que va a realizar actividades que serán automatizadas.

ACTOR	DESCRIPCION
Subdirector de Formación	Es el encargado de realizar las actividades que forman parte de los

	procesos de gestión de formación del centro.
--	--

Tabla 4. Definición de actores del sistema

2.5.8. Modelo de casos de uso del sistema

El modelo de casos de uso describe lo que hace el sistema para cada uno de los usuarios que interactúan con la aplicación.

A continuación se muestra el diagrama de paquetes de Casos de Uso, en el cual se agruparon los casos de uso teniendo en cuenta las entidades que se gestionan en cada caso.

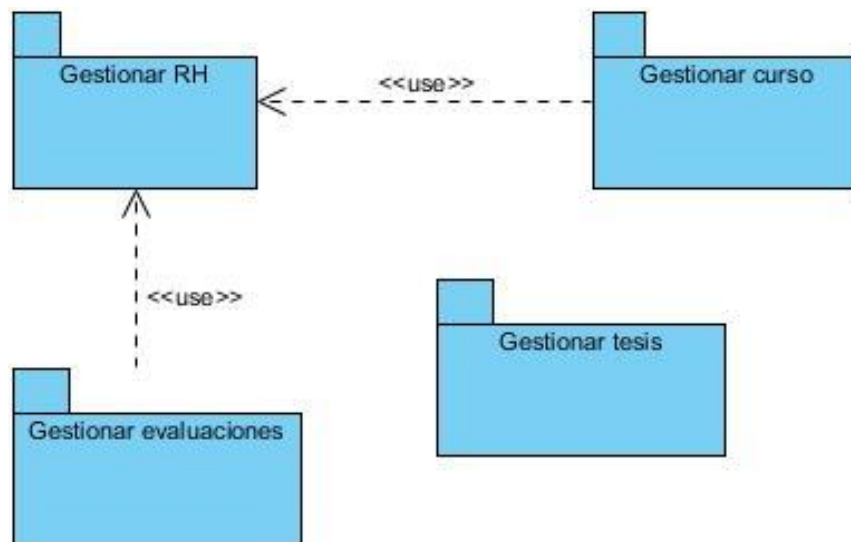


Fig 4. Diagrama de Paquetes de Casos de Uso del Sistema

En la siguiente tabla se establecen las relaciones entre los actores, casos de uso y requisitos funcionales del sistema propuesto.

ACTOR DEL SISTEMA	DEL	CASOS DE USO	REQUISITOS FUNCIONALES
-------------------	-----	--------------	------------------------

Usuario Genérico (Estudiante o Subdirector de Formación)	Autenticar	RF [1]
Subdirector de Formación	Gestionar RH	RF [2 - 3]
Subdirector de Formación	Gestionar Estudiante	RF [4 - 7]
Subdirector de Formación	Gestionar profesor	RF [8 - 11]
Subdirector de Formación	<p>Gestionar Evaluación Mensual de proyecto</p> <p>Gestionar Evaluación Semestral de proyecto</p> <p>Gestionar Evaluación Final de proyecto</p> <p>Gestionar Evaluación de Taller de tesis</p> <p>Gestionar Evaluación Final de Tesis</p> <p>Gestionar Evaluación de Práctica Profesional.</p> <p>Gestionar Evaluación por Competencia Genérica.</p> <p>Gestionar Evaluación de Certificación de Roles.</p> <p>Gestionar Evaluación de Certificación de Investigación.</p>	RF [12]
Subdirector de Formación	Gestionar Curso	RF [13 - 16]

Subdirector de Formación	Realizar matrícula	RF [17]
Subdirector de Formación	Buscar matrícula	RF [18]
Subdirector de Formación	Gestionar Tesis	RF [19- 22]
Subdirector de Formación	Gestionar Tribunal	RF [23 - 26]
Subdirector de Formación	Planificar Taller, Pre-Defensa y Defensa de Tesis	RF [27 - 30]
Subdirector de Formación	Gestionar Proyecto	RF [31 - 34]

Tabla 5. Relación Actor-Caso de Uso-Requisito Funcional

Un diagrama de casos de uso del sistema describe parte del modelo de casos de uso y muestra un conjunto de casos de uso con una asociación entre cada par actor/caso de uso que interactúan.

2.5.9. Diagrama de Caso de Uso del Sistema “Gestionar RH”



Fig 5. Diagrama de CU del Sistema “Gestionar RH”

Las descripciones de los Casos de Uso, que se mostrarán en la tabla 5, así como los Diagramas de Casos de Uso del Sistema se podrán encontrar en la sección de los **Anexos**, pues a partir de este epígrafe se hará uso del caso de uso “Gestionar Estudiante”, tomándolo como ejemplo ilustrativo mediante el cual se mostrarán los artefactos generados durante el diseño e implementación de esta aplicación.

2.5.10. Descripción de Caso de Uso del Sistema “Gestionar Estudiante”

CASO DE USO	GESTIONAR ESTUDIANTE	
Actores:	Subdirector de Formación (SDF)	
Propósito:	El propósito de este caso de uso es gestionar los estudiantes pertenecientes al Centro ISEC.	
Resumen:	En este caso de uso se brinda la funcionalidad de registrar un nuevo estudiante mediante su nombre, solapín, año, grupo, proyecto, rol de proyecto y SET.	
Referencia	RF. [4 – 6]	
Precondición	El Subdirector debe estar autenticado en el sistema.	
Postcondición	Se actualizó la tabla estudiantes de la base de datos.	
CU Incluido	Buscar estudiantes	
CU Extendido	No aplica.	
Requisitos no funcionales	No aplica.	
FLUJO NORMAL DE EVENTOS		
Acción del actor	Respuesta del Sistema	

<p>1. El caso de uso de inicia cuando el actor selecciona una de estas opciones:</p> <p>1.Opción 1, “Registrar” y del menú contextual escoge la entidad “Estudiante”.</p> <p>2.Opción 2, “Buscar y Mostrar” y del menú contextual escoge la entidad “Estudiante”.</p>	<p>2. Si el actor escoge:</p> <p>1.Opción 1, ver sección Registrar estudiante.</p> <p>2.Opción 2, ver CU Buscar estudiante.</p>
<p>Sección No.: 1. “Registrar estudiante”</p>	
<p>Acción del actor</p>	<p>Respuesta del Sistema</p>
	<p>2. Muestra la interfaz Registrar Estudiante con el formulario correspondiente a los datos de un estudiante.</p>
<p>3. Introduce los datos requeridos y pulsa el botón “Registrar”.</p>	<p>4. Verifica que no halla campos vacíos.</p> <p>5. Verifica que se entre el tipo de dato especificado para cada campo.</p> <p>6. Verifica que no exista un estudiante con el mismo nombre o número de solapín.</p> <p>7. Muestra los datos del estudiante en una nueva interfaz.</p>
<p>FLUJOS ALTERNOS</p>	
<p>Alternativa 4a: “Hay campos vacíos.”</p>	
<p>Acción del actor</p>	<p>Respuesta del sistema</p>
	<p>4.1. -Muestra el mensaje de información “Dato obligatorio”</p>

	4.2. Regresa al paso 3 del Flujo Normal de Eventos.
Alternativa 5a: "Tipo de datos incorrecto."	
Acción del actor	Respuesta del sistema
	5.1. Muestra el mensaje de error " <i>Valor incorrecto.</i> " Regresa al paso 3 del Flujo Normal de Eventos.
Alternativa 6a: "Ya existe."	
Acción del actor	Respuesta del sistema
	6.1. Muestra un mensaje de error " <i>Ya existe esta entidad en la base de datos</i> "
Sección No.: 2. "Modificar estudiante "	
Acción del actor	Respuesta del Sistema
5. Introduce los datos nuevos y oprime el botón " Modificar ".	6. Verifica que no halla campos vacíos. 7. Verifica que se entre el tipo de dato especificado para cada campo. 8. Verifica que no existan datos repetidos. 9. El sistema actualiza el listado de estudiantes del centro con los datos introducidos en los campos del formulario.
FLUJOS ALTERNOS	
Alternativa 6a: "Hay campos vacíos."	
Acción del actor	Respuesta del sistema
	6.1. Muestra el mensaje de información " <i>Dato obligatorio.</i> " 6.2. Regresa al paso 5 del Flujo Normal de Eventos.

Alternativa 7a: "Tipo de datos incorrecto."	
Acción del actor	Respuesta del sistema
	7.1. Muestra el mensaje de error "Valor incorrecto." 7.2. Regresa al paso 5 del Flujo Normal de Eventos.
Alternativa 8a: "Ya existe."	
Acción del actor	Respuesta del sistema
	8.1. Muestra un mensaje de error "Ya existe esta entidad en la base de datos"
Sección No.: 3. "Eliminar estudiante "	
Acción del actor	Respuesta del Sistema
5. El actor oprime el botón "Eliminar".	6. Actualiza el listado de estudiantes del Centro ISEC.

Tabla 6. Descripción de Caso de Uso "Gestionar estudiante"

2.5.11. Descripción de Caso de Uso del Sistema "Buscar Estudiante"

CASO DE USO	BUSCAR ESTUDIANTE
Actores:	Subdirector de Formación (SDF)
Propósito:	El propósito de este caso de uso es buscar los datos de un estudiante a partir de un criterio de búsqueda seleccionado.
Resumen:	El caso de uso inicia cuando el actor desea realizar la búsqueda de un estudiante.
Referencia:	RF. 7

Precondición:	El Subdirector debe estar autenticado en el sistema.
Postcondición:	No aplica.
CU Incluido	No aplica.
CU Extendido	No aplica.
Requisitos no funcionales	No aplica.
FLUJO NORMAL DE EVENTOS	
Acción del actor	Respuesta del Sistema
	1. Muestra la interfaz Buscar estudiante con un campo para introducir el dato del estudiante por el que se iniciará la búsqueda según el criterio de búsqueda seleccionado.
2. Selecciona el criterio de búsqueda, introduce los datos y pulsa el botón “Buscar” .	3. Verifica la existencia del estudiante. 4. Muestra una tabla con los resultados de la búsqueda y la opción “Editar” .
5. Pulsa sobre la opción “Editar” .	6. Muestra la interfaz “Editar” con los datos del curso seleccionado y las siguientes opciones. 3. Opción “Modificar” , ver sección Modificar estudiante del CU Gestionar estudiante . 4. Opción “Eliminar” , ver sección Eliminar estudiante del CU Gestionar estudiante .
FLUJOS ALTERNOS	
Alternativa 3a: “No existe el estudiante”	

Acción del actor	Respuesta del sistema
	2.1. El sistema muestra la interfaz Buscar estudiante con la tabla vacía.

Tabla 7. Descripción de Caso de Uso “Buscar estudiante”

2.5.12. Conclusiones

En este capítulo se identificaron y representaron los procesos del negocio mediante la técnica de modelado del negocio IDEF0. Se obtuvo un modelo de casos en el cual se representaron y describieron en términos de Casos de Uso del Sistema los 34 requisitos funcionales identificados, quedando especificada la relación que se establece entre los actores del sistema y los casos de uso.

CAPÍTULO 3: DISEÑO E IMPLEMENTACION DEL SISTEMA

3.1. Introducción

En el presente capítulo se traducen los requisitos funcionales definidos en el capítulo anterior, a una especificación que describe cómo se debe implementar el sistema. Se describen además, los patrones de diseño utilizados y los principales elementos de la arquitectura a través de la modelación de artefactos que se obtienen durante el diseño e implementación del software.

3.2. Selección del estilo arquitectónico.

El estilo arquitectónico a utilizar es el estilo Modelo – Vista – Controlador (MVC), también conocido como estilo de llamadas y retorno. Este estilo permite tener una organización lógica de los componentes del sistema al dividirlos en tres niveles diferentes: el modelo, la vista y la lógica del negocio, constituyendo esta su particularidad más ventajosa, pues elimina el complejo trabajo de tener que cambiar la lógica del negocio cada vez que cambie la interfaz de usuario.

Entre sus principales características se encuentran su versatilidad y adaptación al cambio, propiciando que se ajuste a las necesidades del sistema a desarrollar siendo su uso vital cuando se desea que el producto cuente con la menor cantidad de código posible.

En el Capítulo 1 se hace referencia al término patrón arquitectónico MVC, que aunque en muchas bibliografías lo relacionan con el de estilos arquitectónicos no resultan ser lo mismo. Su principal diferencia radica en el nivel de abstracción que alcanza cada uno, pues el estilo arquitectónico se encarga de la descripción del esqueleto estructural y general para la aplicación, es dependiente de otros estilos y expresa componentes y relaciones. Por otro lado los patrones arquitectónicos definen la estructura básica de una aplicación, pueden contener o estar contenidos en otros patrones y proveen un subconjunto de subsistemas predefinidos, incluyendo reglas y pautas para su organización. Para finalizar esta comparación es importante mencionar que los patrones de diseño son los esquemas finales para refinar subsistemas o componentes.

En analogía con el estilo arquitectónico seleccionado se decide utilizar el patrón arquitectónico MVC, siendo también uno de los patrones empleados por Symfony2 para el desarrollo de la aplicación.

3.3. Patrones de diseño

Un patrón es un conjunto de información que aporta la solución a un problema que se presenta en un contexto determinado. Para elaborarlo se aíslan sus aspectos esenciales y se añaden cuantos comentarios y ejemplos sean necesarios. [19] En el desarrollo de la aplicación propuesta se utilizaron los siguientes patrones de diseño:

3.3.1. Patrones GoF (Gang of four)

Decorator

Symfony 2 contienen un decorador que permite agregar funcionalidades dinámicamente a las aplicaciones desarrolladas bajo sus principios. Cada una de las vistas generadas heredan su diseño de la plantilla *layout.html.twig*, que a su vez hereda de *base.html.twig*, siendo esta la plantilla contenedora de la estructura y el diseño básico de las vistas.

Singleton

La clase *routing.yml* es utilizada por los controladores frontales de cada entidad y se encarga de enrutar todas las peticiones que se hagan a la aplicación.

Command

Este patrón se observa en el controlador frontal de cada entidad, en el método *dispatch()*. Esta clase está por defecto y es la encargada de establecer el módulo y la acción que se va a usar según la petición del usuario. En este método es parseada la URL con el objetivo de precisar los parámetros de la misma y de esta forma saber el Actions que debe responder a la petición.

Patrón Registry

Este patrón se aplica en la clase *config.yml*, que es la encargada de acumular todas las variables de uso global en el sistema. Mediante el uso de este patrón se hace posible compartir datos y objetos en la aplicación sin tener que conservar parámetros o utilizar variables globales. Cada clase de esta capa tiene su responsabilidad y es única, hay controladores que se encargan de la seguridad del sistema trabajando con ficheros YML, y otros que se encargan de identificar mediante algunos datos las clases que deben realizar determinadas tareas (Patrón GoF Command, clase *routing.yml*) y las clases relacionadas con la configuración del sistema (*config.yml*).

3.3.2. Patrones (GRASP)

Bajo acoplamiento

La clase Actions hereda únicamente de ***action.html.twig*** para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Alta cohesión

La clase Actions está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.

Creador

En la clase Actions se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase Actions es “creador” de dichas entidades.

Patrón Experto

Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión de la librería Doctrine para mapear la Base de Datos. Symfony utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos (Peer del Modelo) poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.

Patrón controlador

Todas las peticiones Web son manipuladas mediante un controlador frontal por cada entidad, siendo estos los únicos puntos de entrada de la aplicación en un entorno determinado.

3.4. Modelo de diseño

El modelo de diseño describe la realización física de los casos de uso, centrándose en los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, y el impacto que tienen en el sistema, constituyendo este la entrada principal para la implementación.

3.5. Diagrama de clases de diseño “Gestionar estudiante”

El diagrama que a continuación se muestra, describe el comportamiento por capas de la aplicación en cuanto a la gestión de las entidades involucradas. Se representa la relación existente entre las clases definidas para las capas modelo, vista y controlador y los componentes de Symfony:

- En la capa del Controlador se encuentra la clase: **CC_Gest_est.php**.

Paquete controlador: Una parte importante de su trabajo es contener las acciones fundamentales, que se realizarán sobre cada entidad, centralizadas en una clase controladora común para todas las vistas relacionadas con una entidad. Entre las tareas comunes se encuentran el manejo de las peticiones del usuario, la seguridad, cargar la configuración de la aplicación y otras tareas similares.

- En la capa del Modelo se encuentran las clases entidades (**CE_Estudiantes, CE_Curso_Estudiantes, CE_Profesor, CE_Curso**) y la clase de acceso a dato (**CAD_Estudiantes**).

Paquete del modelo: Solo contiene las clases encargadas del acceso a los datos almacenados en el gestor de base de datos.

- En la capa de la Vista se definieron **formularios y páginas clientes** para cada una de las acciones que se realizan sobre la entidad Estudiante y la página servidora **EstudianteController** que hereda del **layout.php** (plantilla).

Paquete vista: La vista se separa en un layout o plantilla, que será global para la aplicación a menos que existan casos de vistas a los que se les aplique otro layout. El layout sólo se encarga de visualizar las variables definidas en el paquete del controlador.

Subsistema Componente Symfony: Representa todas las clases del framework Symfony que serán utilizadas durante el funcionamiento del sistema. Dígase validadores de formularios, helpers de objetos y formularios, plantillas, componentes de seguridad, entre otros.

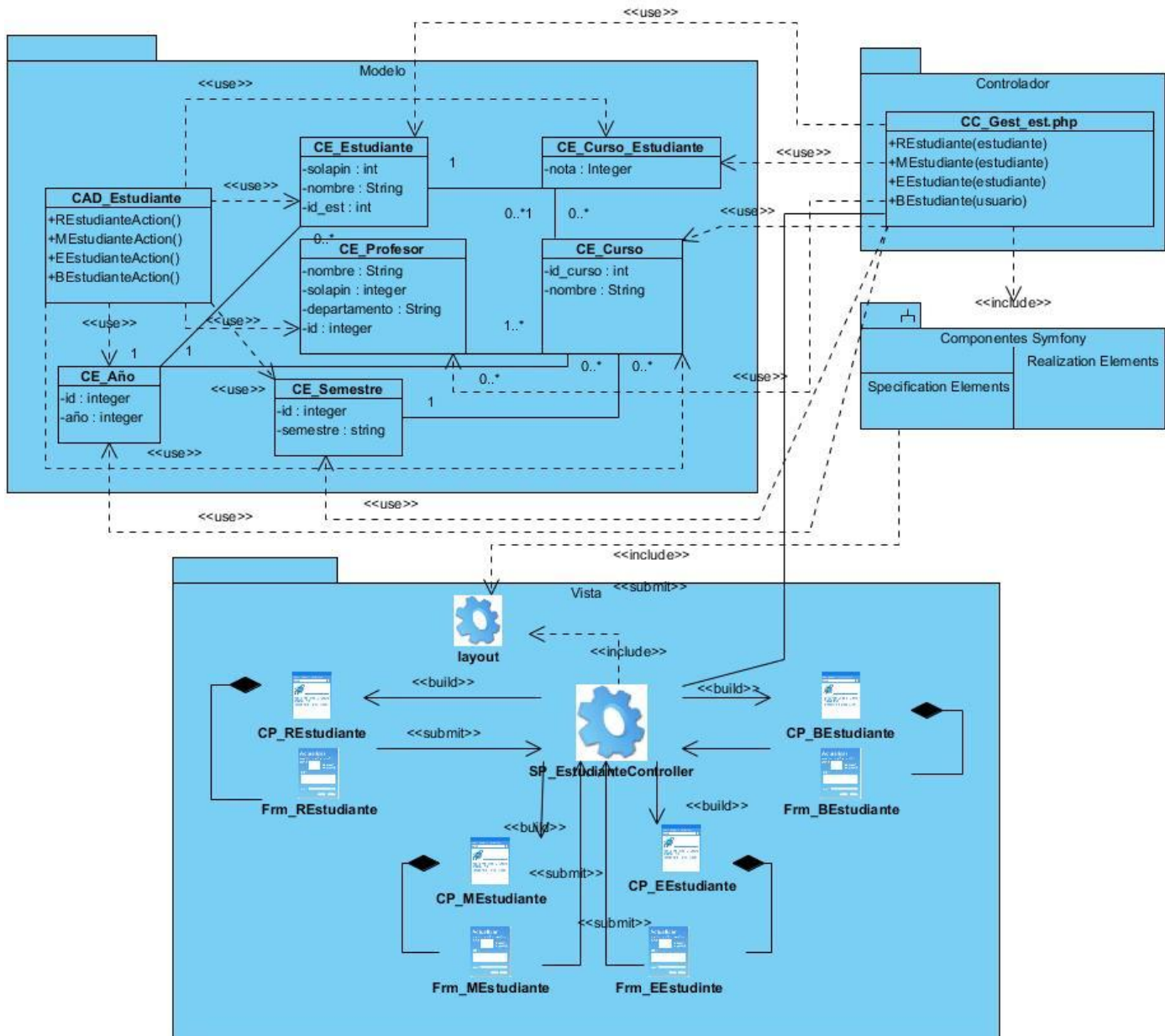


Fig 6. Diagrama de clases del diseño "Gestionar Estudiante"

Los diagramas de clase del diseño restantes se podrán visualizar en los anexos.

3.5.1. Principios de diseño

Para lograr un buen diseño en una aplicación es necesario definir principios que posibiliten que la misma sea atractiva para el usuario.

Para satisfacer un adecuado diseño en la interfaz de la aplicación, el sistema contará con una ventana principal la cual contendrá los vínculos a las diferentes funcionalidades que ofrece el sistema. Se garantizará la legibilidad de la información mediante la correcta selección de los colores de los textos y el fondo. Las notificaciones por acciones realizadas se harán mediante la visualización de interfaces actualizadas o de la aparición de mensajes en el cuerpo de la vista.

3.6. Diseño de la Base de datos

Diagrama entidad – relación de la base de datos

El diagrama entidad-relación (DER) constituye una guía para el modelado de bases de datos. Es mediante este diagrama que se visualizan los objetos que pertenecen a la base de datos como entidades, las cuales tienen atributos y se vinculan mediante relaciones.

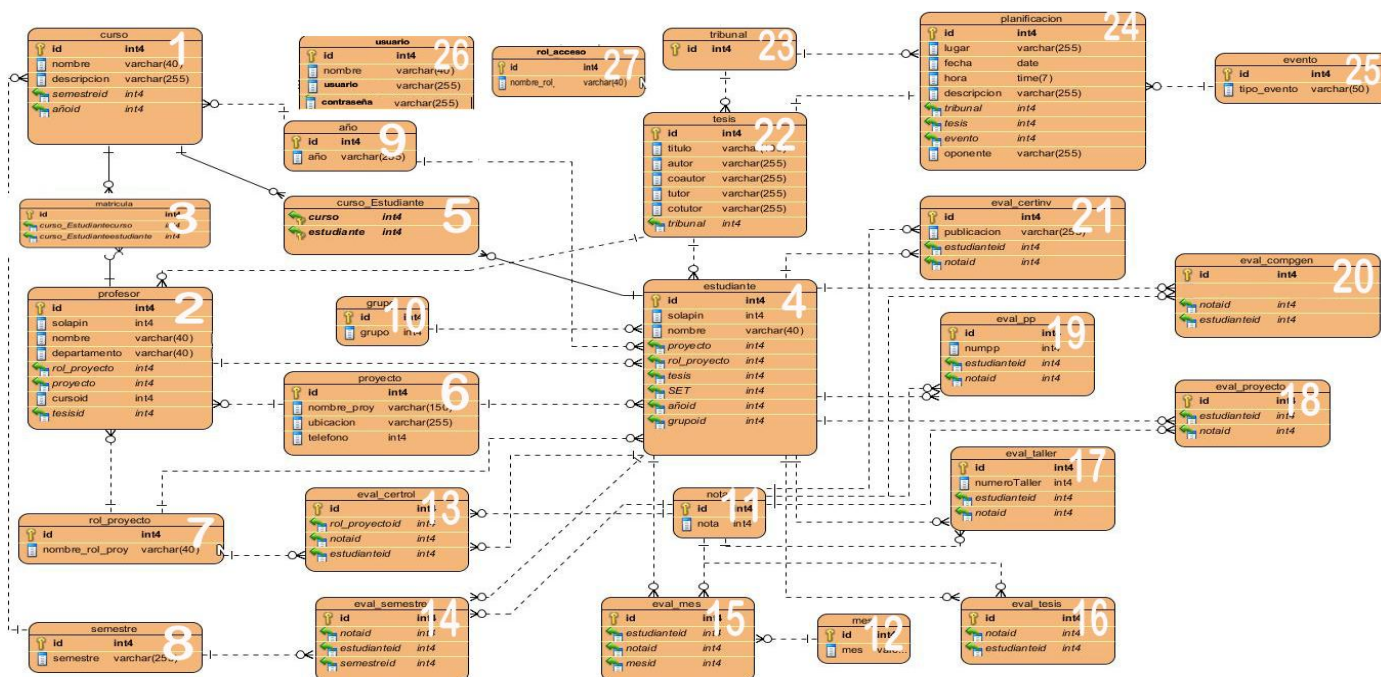


Fig 7. Diagrama entidad- relación.

Las relaciones presenten en el diagrama son mayoritariamente One to Many o Uno a Muchos. A continuación se listan las entidades que forman parte de dicho diagrama.

No.	ENTIDAD	ATRIBUTOS	RELACIONES
1	Curso	Id, nombre, descripción	Semestre, Año, Estudiante
2	Profesor	Id, nombre, solapín, departamento	Proyecto, Rol_proyecto, tesis
3	Matrícula	Id_curso, id_profesor	Curso, Profesor
4	Estudiante	Id, nombre, solapín	Proyecto, Rol_proyecto, SET, Año, Grupo, Tesis
6	Proyecto	Id, nombre_proy, ubicación, teléfono	Profesor, Estudiante
7	Rol_proyecto	Id, nombre_rol_proy	Profesor, Estudiante, Eval_Certrol
8	Semestre	Id, semestre	Curso, Eval_semestre
9	Año	Id, año	Curso, Estudiante
10	Grupo	Id, grupo	Estudiante
11	Nota	Id, nota	Eval_Certrol, Eval_semestre, Eval_mes, Eval_Certinv, Eval_Compgen, Eval_pp, Eval_tesis, Eval_proyecto, Eval_taller
12	Mes	Id, mes	Eval_mes
13	Eval_Certrol	Id	Nota, Rol_Proyecto, Estudiante
14	Eval_semestre	Id	Nota, Semestre, Estudiante

15	Eval_mes	Id	Nota, Mes, Estudiante
16	Eval_tesis	Id	Nota, Estudiante
17	Eval_taller	Id, númeroTaller	Nota, Estudiante
18	Eval_proyecto	Id	Nota, Estudiante
19	Eval_pp	Id, numpp	Nota, Estudiante
20	Eval_compgen	Id	Nota, Estudiante
21	Eval_certinv	Id, publicación	Nota, Estudiante
22	Tesis	Id, título, autor, co-autor, tutor, co-tutor	Tribunal, Planificación, Estudiante, Profesor
23	Tribunal	Id	Tesis, Planificación
24	Planificación	Id, fecha, hora lugar, descripción, oponente	Tribunal, Tesis, Evento
25	Evento	Id, tipo_evento	Planificación
26	Usuario	Id, nombre, usuario, contraseña	Rol_Acceso
27	Rol_Acceso	Id, nombre_rol	Usuario

Tabla 8. Entidades del Diagrama Entidad Relación

3.7. Modelo de implementación

Un modelo de implementación está constituido por una colección de componentes, subsistemas de implementación y paquetes empleados para agrupar los elementos de dicho modelo. Los componentes constituyen la parte modular del sistema que encapsula la implementación. Un componente contiene clases, puede ser un fichero de código fuente, scripts, ejecutables entre otros. Los diagramas de

componentes son utilizados para modelar la vista estática del sistema, mostrando la organización y las dependencias lógicas entre los componentes.

3.7.1. Modelo de componentes

El diagrama de componentes muestra las dependencias lógicas y las organizaciones de componentes de software. Estos pueden utilizar paquetes para agrupar elementos del modelo. Han sido agrupados, al igual que los diagramas de clases del diseño, en paquetes para facilitar su comprensión y poder detallarlos mejor.

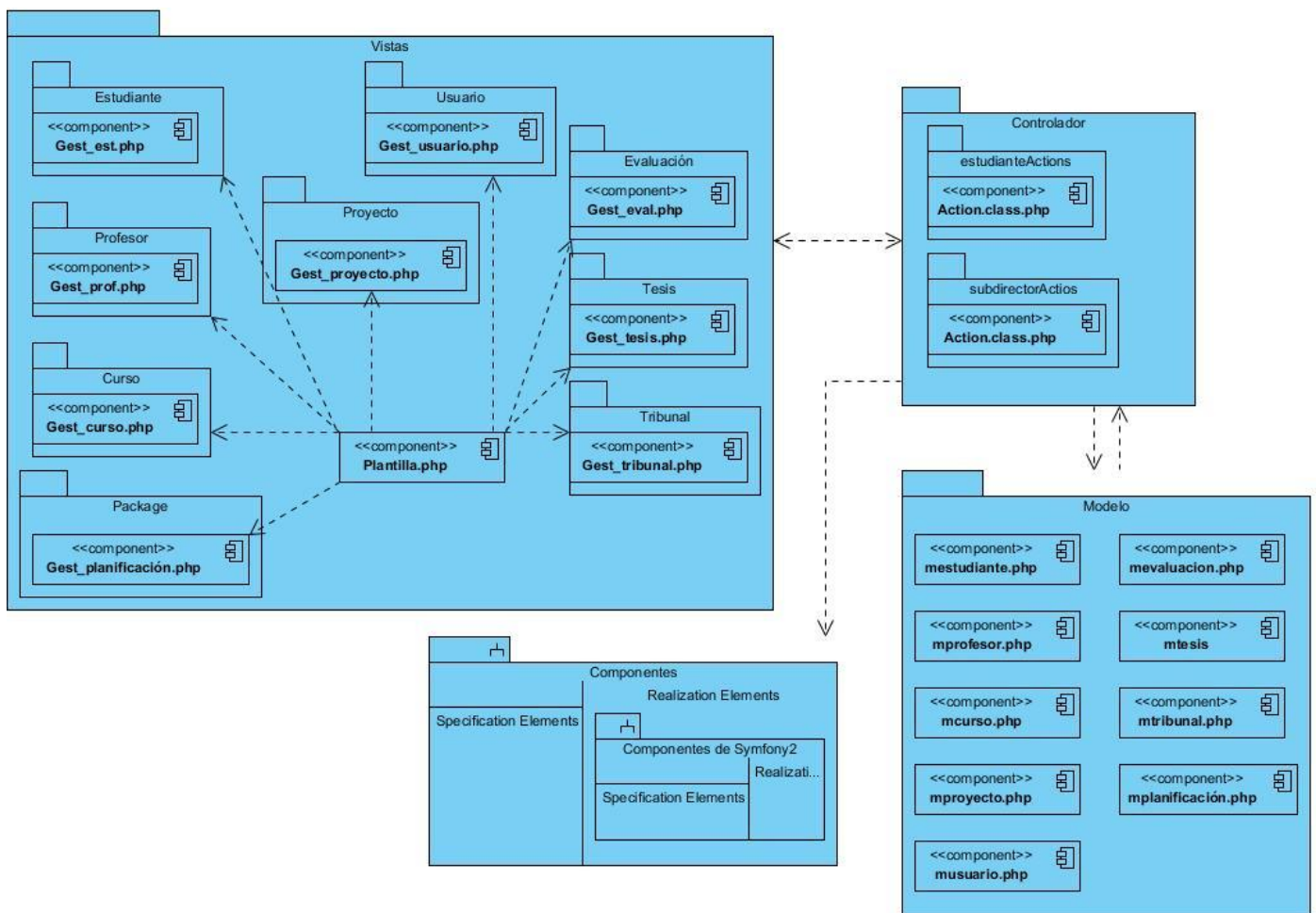


Fig. 8 Modelo de componentes

Se identificaron los componentes por cada una de las capas de la aplicación, donde se seleccionaron los ficheros .php como elementos a representar. Los componentes identificados por cada una de las capas son los siguientes.

Modelo: mestudiante, mprofesor, mevaluacion, mtesis, mcurso, mtribunal, mproyecto.

Vista: Gest_est.php, Gest_prof.php, Gest_curso.php, Gest_usuario.php, Gest_eval.php, Gest_tesis.php, Gest_tribunal.php, Gest_proyecto.php, Plantilla.php.

Controlador: Las Action.class de estudianteActions, subdirectorActions, InicioActions.

3.7.2. Modelo de despliegue

El Modelo de Despliegue define como va a ser la arquitectura física del sistema. Provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Este es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos.

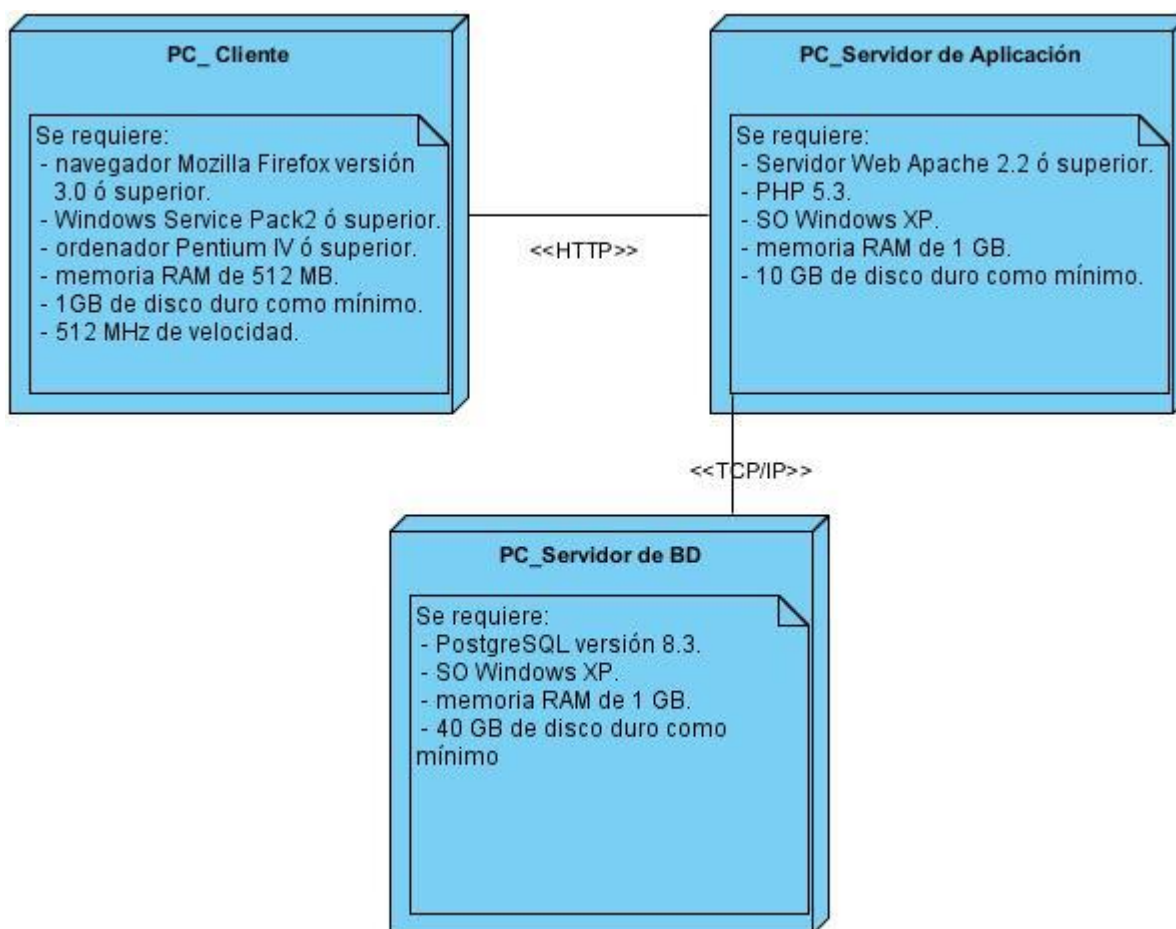


Fig. 9 Modelo de despliegue

Descripción de los nodos

Nodo: PC-cliente: Ordenadores que serán utilizados por los usuarios del sistema para acceder a la aplicación y operar la misma.

Nodo: Servidor de Aplicaciones: Es la PC donde se instalará el servidor web que será el encargado de atender todas las peticiones de los usuarios.

Nodo: Servidor de BD: Es la PC donde se instalará el servidor de base datos.

En la imagen mostrada, donde se ilustra la forma en que se realizará el despliegue de la aplicación se establece la comunicación entre las PC_Clientes y la PC_Servidor_de_la_Aplicación mediante el protocolo HTTP por el puerto 8080 y la comunicación de esta con la PC_Servidor_de_BD a través del protocolo TCP por el puerto 5432.

3.8. Pruebas

La disciplina de prueba es imprescindible para garantizar la calidad y el adecuado funcionamiento de un software. Esta disciplina permite comprobar la eficacia de un sistema de software, mediante la verificación del cumplimiento de los objetivos trazados en la etapa de implementación. De esta manera se logra reducir el número de errores no detectados durante la implementación, así como el tiempo de introducción de este error en el sistema y su detección. Para asegurar el éxito de las pruebas son imprescindibles los casos de pruebas, los cuales se presentan como un conjunto de condiciones bajo las cuales se determina si la respuesta de la aplicación es parcial o completamente satisfactoria.

Existen, para la realización de las pruebas al software, dos técnicas fundamentales: las pruebas de caja negra y las pruebas de caja blanca.

Las pruebas de caja blanca están dirigidas a las funciones internas del software. Se basan en un minucioso examen del código, por lo que se hace necesario conocer la lógica del programa.

Por otra parte, las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior, es decir, se realizan sobre la interfaz del programa.

3.8.1. Prueba de Caja Negra

Las pruebas de caja negra permiten demostrar si el sistema responde adecuadamente ante determinadas entradas de datos y si los resultados son los esperados. Permiten, por ejemplo, encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores de estructuras de datos o en accesos a las Bases de Datos externas
- Errores de rendimiento.
- Errores de inicialización y terminación.

3.8.2. Pruebas de Validación

Consecuentemente con los capítulos anteriores se realizan las pruebas al Caso de Uso “Gestionar Estudiante”, mostrando en la sección de los Anexos las pruebas realizadas a los Casos de Uso restantes.

Caso de prueba “Registrar Estudiante”

PRUEBA DE ACEPTACIÓN	
Caso de Uso: “Gestionar Estudiante”	Código: CU2_P1
Nombre: “Registrar estudiante”	
Descripción: Prueba para comprobarla funcionalidad <i>Registrar estudiante</i> y con ella la gestión de estudiantes.	
Condiciones de ejecución: El sistema debe estar levantado con el administrador autenticado.	
Entrada:	Nombre: Ana Karla Orret Moreno Solapín: 011738 Año: 5 Grupo: 2507 Proyecto: Distal Rol: Diseñador de Interfaz Gráfica. SET: Hamler Rodríguez
Salida: Todos los datos de los atributos son válidos y el estudiante no se encuentra en la base de datos, el sistema registra al nuevo estudiante y muestra una vista con sus datos a	

modo de notificación.	
Evaluación de la prueba:	<i>Prueba satisfactoria.</i>

Tabla 9. Caso de prueba # 1 “Registrar Estudiante”

PRUEBA DE ACEPTACIÓN		
Caso de Uso: “Gestionar Estudiante”	Código: CU2_P2	
Nombre: “Modificar estudiante”		
Descripción: Prueba para comprobarla funcionalidad <i>Modificar estudiante</i> y con ella la gestión de estudiantes.		
Condiciones de ejecución: El sistema debe estar levantado con el administrador autenticado y al menos un estudiante registrado.		
Entrada:	Antes	Después
	Nombre: Ana Karla Orret Moreno Solapín: 011738 Año: 5 Grupo: 2507 Proyecto: Distal Rol: Diseñador de Interfaz Gráfica. SET: Hamler Rodríguez	Nombre: Ana Karla Orret Moreno Solapín: 011738 Año: 5 Grupo: 2503 Proyecto: LABSI Rol: Diseñador de Interfaz Gráfica. SET: Ariagnis Yero Guevara
Salida: Los datos introducidos son válidos, el sistema actualiza la información del estudiante y muestra una vista con sus datos actualizados a modo de notificación.		
Evaluación de la prueba:	<i>Prueba satisfactoria.</i>	

Tabla 10. Caso de prueba # 2 “Modificar Estudiante”

PRUEBA DE ACEPTACIÓN	
Caso de Uso: “Gestionar Estudiante”	Código: CU2_P3
Nombre: “Eliminar estudiante”	
Descripción: Prueba para comprobarla funcionalidad <i>Eliminar estudiante</i> y con ella la gestión	

de estudiantes.	
Condiciones de ejecución: El sistema debe estar levantado con el administrador autenticado y al menos un estudiante registrado.	
Entrada:	Se elimina el estudiante seleccionado.
Salida:	Se muestra el mensaje de confirmación “El estudiante ha sido eliminado”.
Evaluación de la prueba:	<i>Prueba satisfactoria.</i>

Tabla 11. Caso de prueba # 3 “Eliminar Estudiante”

PRUEBA DE ACEPTACIÓN	
Caso de Uso: “Gestionar Estudiante”	Código: CU2_P4
Nombre: “Buscar estudiante”	
Descripción: Prueba para comprobarla funcionalidad <i>Buscar estudiante</i> y con ella la gestión de estudiantes.	
Condiciones de ejecución: El sistema debe estar levantado con el administrador autenticado.	
Entrada / Criterios de Búsqueda:	Nombre: Ana Karla Orret Moreno Solapín: 011738 Año: 5 Grupo: 2507 Proyecto: Distal Rol: Diseñador de Interfaz Gráfica. SET: Hamler Rodríguez
Salida: Los datos entrados en el campo de búsqueda sin válidos y se muestran los resultados en una tabla de la misma interfaz.	
Evaluación de la prueba:	<i>Prueba satisfactoria.</i>

Tabla 12. Caso de prueba # 4 “Buscar Estudiante”

3.9. Conclusiones

En este capítulo se obtuvo el diseño y la construcción del sistema en términos de clases del diseño. Se describió la interacción de los usuarios con el sistema basándose en el estilo arquitectónico MVC, así como la interacción entre las clases del sistema, con el apoyo de los diagramas de clase del diseño. Se obtuvieron los diagramas de componentes y de despliegue, mostrándose cómo se encontrará desplegado físicamente el sistema. A partir de la aplicación de las pruebas al sistema se comprobó la correspondencia del mismo con los requerimientos funcionales definidos.

CONCLUSIONES

Al finalizar este trabajo, ha sido desarrollada la versión 1.0 del Sistema de apoyo a los procesos del área de formación del Centro ISEC. Se han diseñado e implementado satisfactoriamente los casos de uso que pertenecen al primer ciclo de desarrollo del producto, por lo que la primera versión de la aplicación contará solo con las funcionalidades elementales y necesarias para ser una solución factible, teniendo un segundo ciclo de desarrollo que le proporcionará las funcionalidades restantes relacionadas con los reportes.

Se ha desarrollado el Sistema de apoyo a los procesos del área de formación del Centro ISEC, dando cumplimiento al objetivo general. La solución que se propone permite al Subdirector de Formación:

- Almacenar y consultar información relacionada con los estudiantes, profesores y proyectos del centro
- Registrar los temas de tesis de pregrado y relacionarlos con sus autores y tutores.
- Registrar los tribunales de predefensa y defensa de tesis.
- Vincular los estudiantes a los proyectos productivos.
- Registrar las evaluaciones de los estudiantes a los largo del ciclo profesional.
- Planificar las sesiones de los tribunales de predefensa y defensa de tesis.

Se verificó la correspondencia del sistema con los requisitos funcionales establecidos a través de las pruebas realizadas al mismo.

El desarrollo del trabajo permitió, además, adquirir conocimientos de herramientas nuevas como el marco de trabajo Symphony 2 y de técnicas de modelado de procesos como el IDEF0. La puesta en funcionamiento de este sistema va a contribuir con el buen desempeño del Subdirector de Formación, pues le permitirá almacenar toda la información en un formato común para luego poder ser consultada y mejorar el proceso de inscripción de los cursos optativos.

RECOMENDACIONES

Debido a la importancia que se le confiere al Sistema de apoyo a los procesos del área de formación del Centro ISEC, realizamos las siguientes recomendaciones:

- Incluir nuevas funcionalidades que permitan al estudiante realizar la matrícula de los cursos optativos a través del sistema.
- Incluir funcionalidades que permitan la generación de reportes a partir de la información gestionada.
- Desplegar la solución propuesta.
- Proceder al diseño e implementación del segundo ciclo de desarrollo del sistema.

REFERENCIAS BIBLIOGRÁFICAS

1. Definición de formación. [En línea] 2008 - 2011. [Citado el: 15 de febrero de 2012.] <http://definicion.de/formacion>.
2. Formación Docente. Escuelas del Bicentenario. IIIPE – UNESCO Buenos Aires. [En línea] [Citado el: 15 de febrero de 2012.] http://www.ebicentenario.org.ar/formacion_docente.php.
3. María Lina Jiménez Pardo (1), Leonel Pineda Folgoso (2), Eva de la Nuez Ramos (3). La tecnología educativa y la educación. Cuba BVS. Biblioteca Virtual Salud. [En línea] [Citado el: 15 de febrero de 2012.] http://bvs.sld.cu/revistas/mciego/vol9_02_03/educamed/em3_v9_0203.htm.
4. Gestión de procesos. Administración Electrónica. Aiteco Consultores. [En línea] 2006. [Citado el: 15 de febrero de 2012.] <http://www.aiteco.com/contacto.php>
5. Introducción a las Aplicaciones Web. Escuela técnica superior de ingeniería informática. 2004. [En línea] [Citado el: 16 de febrero de 2012.] <http://www.lsi.us.es/docencia/get.php?id=352>
6. Lenguaje de Programación del lado del servidor o cliente.
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html
7. PHP. [En línea] 2001 - 2012. [Citado el: 9 de febrero de 2012.] <http://php.net>.
8. The Apache Software Foundation. [En línea] [Citado el: 9 de febrero de 2012.] <http://www.apache.org>
9. ORTEGA, Á. L. [En línea] [Citado el: 9 de mayo de 2012.] http://ditec.um.es/ssdd/trabajos/0506/Hibernate-Angel_Luis_Calvo_Ortega.pdf.
10. Brito Acuña, Kareny. Selección de metodologías de desarrollo para aplicaciones web en la facultad de informática de la Universidad de Cienfuegos. BIBLIOTECA VIRTUAL de Derecho, Economía y Ciencias Sociales. [En línea] [Citado el: 16 de febrero de 2012.] http://www.eumed.net/libros/2009c/584/Descripcion_de_las_metodologias_existentes_para_el_desarrollo_de_software.htm

11. Modelo Vista Controlador [En línea] [Citado el: 9 de mayo de 2012.]
<http://www.neleste.com/modelo-vista-controlador/>
12. IDE NetBeans 7.1. DesarrolloWeb.com [En línea] [Citado el: 3 de junio de 2012.]
<http://www.desarrolloweb.com/actualidad/netbeans-ide-7-1-disponible-6378.html>
13. Jacobson, Ivar, Booch, Grady y Runbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Madrid: s.n., 2000
14. Flores, Ervin. Metodologías ágiles. Proceso Unificado Ágil (AUP). Universidad Unión Bolivariana. Carrera de Ingeniería de Sistemas.
15. Scribd. Herramientas CASE. Capitulo I. [En línea] [Citado el: 8 de febrero de 2012.]<http://es.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.
16. Portal ISEC. <http://portal.isec.prod.uci.cu/>
17. Somerville, Ian. Ingeniería del Software. 7ma Edición. 2005. Pág. 687. ISBN: 8478290745
18. Alarcos. Escuela Superior de Informatica de Ciudad Real. [En línea] [Citado el: 9 de febrero de 2012.]<http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
19. Ramírez, Alejandro. Introducción a los Patrones de Diseño. 09 agosto 2004
20. Herramienta Dia. [En línea] [Citado el: 10 de junio de 2012.]
<http://cajondesastreoracle.wordpress.com/2012/01/20/dia-herramienta-gratuita-para-crear-diagramas-entidadrelacion/>
21. JQuery. [En línea] [Citado el: 10 de junio de 2012.]
<http://www.desarrolloweb.com/articulos/introduccion-jquery.html>

BIBLIOGRAFÍA

1. Billy Reynoso, Carlos. Introducción a la Arquitectura de Software. Universidad de Buenos Aires. [En línea] Version 1.0 – Marzo de 2004. [Citado el: 9 de febrero de 2012.] <http://www.willydev.net/descargas/prev/IntroArq.pdf>
2. ASP.NET. ADRFORMACAION. [En línea] 2010. [Citado el: 9 de febrero de 2012.] <http://www.adrformacion.com/cursos/aspnet2010/aspnet2010.html>.
3. Fahnle, Pablo. ASP.NET. Programacion en castellano. [En línea] [Citado el: 9 de febrero de 2012.] <http://www.programacion.com/asp/articulos>.
4. Alvarez, Miguel Angel ,2003. Que es Python? desarrolloweb.com. [En línea] 19 de noviembre de 2033. [Citado el: 9 de febrero de 2012.] <http://www.desarrolloweb.com/articulos/1325.php>.
5. Base de datos. CEPEU. Centro de Especializacion profesional y Exension Universitaria. Paraguay. [En línea] 2004 -2012. [Citado el: 8 de febrero de 2012.] http://www.cepeu.edu.py/Base_de_Datos.pdf.
6. Sanchez, Jorge. 2004. MySQL. Guía rápida. scribd. [En línea] 2004. [Citado el: 8 de febrero de 2012.] <http://es.scribd.com/doc/2892777/MySQL-guia-rapida>.
7. PostgreSQL. PostgreSQL Global Development Group. [En línea] 1996 - 2012. [Citado el: 9 de febrero de 2012.] <http://www.postgresql.org>.
8. Joomla [En línea] [Citado el: 8 de febrero de 2012.] <http://www.joomla.org/about-joomla.html>
9. Drupal [En línea] [Citado el: 8 de febrero de 2012.] <https://www.drupal.org/>.
10. IDE de programacion. ECURED. [En línea] [Citado el: 9 de febrero de 2012.] http://www.ecured.cu/index.php/IDE_de_Programacion.
11. Gonzalez, Gustavo Herramientas Libres para Desarrollo de Aplicaciones en la Industria de Software. Universidad Icesi, Colombia. [En línea] [Citado el: 9 de febrero de 2012.] <http://www.icesi.edu.co/softwarelibre/memorias/Herramientas%20Libres%20en%20Industria%20de%20Software.pdf>.

12. Roberth G. Figueroa, Camilo J. Solís. 2011. Modelos De Procesos De Software. buenastareas. [En línea] Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, agosto 3, 2011. [Citado el: 9 de febrero de 2012.] <http://www.buenastareas.com/ensayos/Modelos-De-Procesos-De-Software/2600270.html>.
13. José H. Canós, Patricio Letelier y M^a Carmen Penadés. 2000 - 2012. Metodologías Ágiles en el Desarrollo de Software. Ingeniería de Software y Sistema de información. [En línea] Universidad Politécnica de Valencia, 200 - 2012. [Citado el: 15 de febrero de 2012.] <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>.
14. Sara Gutierrez, Hernan Zapata, Juan Pablo Arias, Cristian Zambrano. FDD: Feature Driven Development. Desarrollo Basado en Funcionalidades. Google. [En línea] [Citado el: 15 de febrero de 2012.] <https://docs.google.com/pisis.unalmed.edu.co/cursos/material/3004582/1/PresentacionFDD.ppt>.
15. LIC. ERVIN FLORES. 2012. METODOLOGIAS AGILES. PROCESO UNIFICADO AGIL (AUP). [En línea] UNIVERSIDAD UNION BOLIVARIANA, La Paz, Bolivia, 2012. [Citado el: 15 de febrero de 2012.] <http://www.adolfo.mex.tl/images/18149/METODOLOGIAS%20AGILES.pdf>.
16. Lenguajes de Programación Lado Cliente, Lado Servidor. Ambiente de Desarrollo de Aplicaciones Web. 6/10/2011
<http://www.buenastareas.com/ensayos/Lenguaje-De-Programacion-De-Lado-Cliente/2859649.html>
17. Sánchez, Jorge. MySQL guía rápida. 2004.
18. IDEFØ Función método de modelado. Visitado [En línea] [Citado el: 19 de junio de 2012.] <http://www.idef.com/IDEF0.html>