

Universidad de las Ciencias Informáticas

Facultad 3



Título: Sistema de gestión académica para la facultad 3.
Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor: Beatriz Molina Legrá.

Tutores: MsC. Yalice Gámez Batista.

MsC. Johanny Rivera López

La Habana, julio 2012

“Año 54 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Beatriz Molina Legrá

Firma del Autor

MsC. Yalice Gámez Batista de la Nuez

MsC. Johanny Rivera López

Firma del Tutor

Firma del Tutor

Agradezco...

A Diosito por darme los padres que tengo y haberme permitido vencer todas las barreras.

A mi madre por el amor, cariño y dedicación sin límites, por estar presente en todos los momentos y vivir todas mis agonías. A ti que eres la mejor te debo mi vida.

A mi padre por su constante preocupación, por su magnífica forma de ser, por darme siempre su sonrisa...Te adoro

A mi familia que siempre ha estado presente en especial a mis abuelos, mi tío Edivito, mis hermanos Yordanis y Yeli, a Noel por haber sido otro padre para mi.

A mi novio Javier por ser una excelente pareja, por brindarme tanto amor, por comprenderme, por también pasar malos ratos de tesis, por apoyarme y ayudarme. Te Amo

A la familia de mi novio por hacerme sentir como uno más de ellos.

A los tutores por todas las recomendaciones y los momentos que me dedicaron para que el trabajo saliera adelante.

A la oponente y el tribunal que confiaron en mí, por su ayuda, por sus críticas constructivas para alcanzar una mejor calidad.

A la Universidad por tantos amigos y lindos momentos, por permitirme crecer como persona.

A mi gran grupo de amigos de la uni por tan buenos momentos, por las risas, por aportarme su granito de arena, son ellos: Yanet, Santiago, Mayelín, Yesmin, Raidel, Juan Javier, Surima, Charito, Marlon, Yoel, Yalína, Mailín, Laura, Hanny, Arlethy, Yaniel, Jorgito, Jordán, Adrian Gámez, Yaneisi, Roilan, Aylin, Víctor Manuel, Ailenis...

A todos los que me brindaron su mano y ayuda incondicional, que no puedo decir los nombres para no dejar a nadie, pero les estoy eternamente agradecida.

A mi querido grupo 4107, al 4207 y al 59104.... A todos muchas gracias.

Dedicatoria

Con todo el amor y cariño a mis padres por ser mis razones de ser y mis principales inspiraciones para alcanzar las metas.

RESUMEN

La Universidad de Ciencias Informáticas por sus disímiles aportes y características excepcionales aspira convertirse en una universidad de excelencia y líder en el desarrollo de software e informatización de la sociedad cubana. La institución cuenta con una estructura de centros de desarrollos que permiten dar soluciones a diferentes problemas existentes a partir de su informatización. En los centros educacionales la gestión académica se identifica como un proceso fundamental que define el funcionamiento del centro y por ende sus resultados. En el presente trabajo se propone realizar la propuesta de un sistema que gestione los procesos docentes de la facultad. Dentro de los principales procesos a desarrollar se encuentran el control de repitencias y arrastres, ajuste del plan de estudio, gestión de perfiles de tesis, y seguimiento y control de alumnos ayudantes. Para la implementación del sistema, el trabajo se rige por la plataforma tecnológica encontrada en el documento PIN-F03001_2011 propuesta de informatización definida por la facultad 3.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Introducción.....	4
1.2 Sistemas de Gestión	4
1.3 Gestión Académica	5
1.4 Sistemas de gestión académica	6
1.5 Metodología y plataforma tecnológica a utilizar	11
1.6 Conclusiones del capítulo.....	20
CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA.....	21
2.1 Introducción.....	21
2.2 Modelo de Historia de Usuario del Negocio.....	21
2.4 Lista de reserva del producto	30
2.5 Historias de usuario (HU)	33
2.6 Patrones de diseño	41
2.7 Estructuras de las clases del sistema.....	¡Error! Marcador no definido.
2.8 Diseño de la Base de Datos	45
2.9 Tareas de ingeniería	47
2.10 Conclusiones del capítulo.....	54
CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA	55
3.1 Introducción.....	55
3.2 Estándares de codificación.....	55
3.3 Diagrama de despliegue.....	56
3.4 Diagrama de componentes	57
3.5 Interfaz de usuario.....	58
3.6 Pruebas.....	60
3.7 Plantillas de casos de prueba de aceptación.....	61
3.8 Conclusiones del capítulo.....	66
CONCLUSIONES GENERALES.....	67
RECOMENDACIONES	68
REFERENCIAS BIBLIOGRÁFICAS	69
ANEXOS.....	72

INTRODUCCIÓN

El desarrollo tecnológico en el mundo durante las últimas décadas, ha traído consigo el surgimiento de las Tecnologías de la Información y las Comunicaciones (TICs), dando paso a la llamada sociedad de la información. Esto ha permitido la realización de cambios en los sistemas educativos de forma que estos se tornen más flexibles y accesibles.

En los centros educacionales la gestión académica se identifica como un elemento fundamental que define el funcionamiento del centro y por ende sus resultados, permite conocer y almacenar todas las informaciones que se generan en el proceso docente. [1]

Con el transcurso del tiempo se han desarrollado diversos sistemas encaminados a dar solución de forma automatizada a aspectos elementales de la gestión del proceso docente. El objetivo principal de estos sistemas ha sido proveer a los centros estudiantiles de mecanismos que permitan de una forma eficiente la planificación, organización y control de sus actividades académicas, elementos imprescindibles para lograr que el proceso de enseñanza aprendizaje sea efectivo. La Universidad de Ciencias Informáticas (UCI) no se ha quedado atrás y ha comenzado la informatización de varios procesos para una mejor gestión y manejo de documentos.

La UCI cuenta con varias aplicaciones encargadas de la gestión académica, las cuales han brindado resultados favorables y han facilitado gran parte de la información de los estudiantes, pero aún así no cubre con todas las necesidades. En la secretaría de la facultad 3 se hace engorroso el manejo y control de un grupo de informaciones referidas a los estudiantes debido a la amplia matrícula con que se cuenta, y que dicha información es almacenada en documentos de textos y hojas de cálculo que no son eficientes para estos fines. Además de que los sistemas de la UCI existentes no presentan todos los procesos de la gestión académica. Esto provoca que el proceso sea difícil, lento y propenso a errores. El envío de la información a la dirección de la facultad se realiza a través del correo electrónico. Esto dificulta que se pueda acceder a la información de forma automática o simultánea por varias personas. El presente trabajo propone desarrollar un sistema que permita dar solución a los inconvenientes planteados.

De la situación problemática anteriormente planteada se deriva el siguiente **problema a resolver**:

¿Cómo contribuir a mejorar la gestión de los procesos de planificación, seguimiento y control de la actividad académica en el área de formación de la facultad 3?

Luego de un análisis de la situación actual, la investigación enmarca su **objeto de estudio** en los Software de gestión académica.

Centrando su **campo de acción** en Software de gestión y planificación docente para la facultad 3.

Teniendo en cuenta lo planteado anteriormente, el **objetivo general** de este trabajo es: Desarrollar un sistema informático que permita mejorar la gestión de los procesos de planificación, seguimiento y control de la actividad académica en el área de formación de la facultad 3.

Para cumplir con el objetivo general expuesto, se trazan varios **objetivos específicos** que facilitan y organizan el trabajo:

- Estudio del estado del arte de sistemas existentes con similitud al propuesto a desarrollar.
- Modelación de los procesos de negocios para una mejor comprensión del mismo.
- Captura de los requerimientos funcionales o no funcionales.
- Análisis y diseño del sistema.
- Implementación del sistema.
- Pruebas de validación.

Los **Métodos teóricos** utilizados fueron:

- **Histórico – Lógico:** Para realizar el análisis del estudio del estado del arte de los sistemas con similitud al propuesto a desarrollar.
- **Analítico - Sintético:** Para arribar a conclusiones abordadas en los distintos temas tratados a lo largo de la investigación.
- **Modelación:** Para modelar el negocio y comprender los procesos implicados en el mismo.

Los **Métodos empíricos** utilizados fueron:

- **Entrevistas:** Para conocer como funcionaba el proceso docente y sus participantes.
- **Análisis de documentos:** Para comprender el proceso de plan de estudio, las características del la metodología y tecnologías de desarrollo.

El presente trabajo está estructurado en tres capítulos, a continuación se muestra una breve descripción de cada uno de ellos:

Capítulo 1: Fundamentación teórica

En este capítulo se realiza el estudio del estado del arte de sistemas existentes con alguna similitud con el que se desea desarrollar. Se realiza un análisis de la metodología, tecnologías y herramientas a utilizar así como la fundamentación de su uso.

Capítulo 2: Análisis y diseño de la solución propuesta

En este capítulo se muestra una detallada descripción del negocio actual, de los procesos, actores, trabajadores, relación entre actores y procesos que se relacionan. Se realiza la captura de requisitos y sus descripciones a través de las historias de usuario. Se generan los artefactos correspondientes de la metodología en el análisis y diseño.

Capítulo 3: Implementación y validación del sistema

En este capítulo se lleva a cabo el desarrollo del sistema, se realizan los diagramas de despliegue y componentes, se llevan a cabo las pruebas de aceptación para verificar que se correspondan con las historias de usuarios y con el resultado esperado.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se realiza el estudio de estado del arte de algunos sistemas de gestión académica, se lleva a cabo el análisis de la metodología y las herramientas a utilizar así como su fundamentación. Para una mejor comprensión del tema se aborda sobre algunos conceptos relacionados con el objeto de estudio.

1.2 Sistemas de Gestión

Un sistema de gestión es un conjunto de etapas unidas en un proceso continuo, que permite trabajar ordenadamente una idea hasta lograr mejoras y su continuidad. [2]

Este proceso establece cuatro etapas que hacen de un sistema un proceso circular virtuoso, pues en la medida que el ciclo se repita recurrente y recursivamente se logrará en cada ciclo obtener una mejora.

Las cuatro etapas del sistema de gestión son:

1. Etapa de Ideación
2. Etapa de Planeación
3. Etapa de Implementación
4. Etapa de Control

➤ Etapa de Ideación:

El objetivo de esta etapa es trabajar en la idea que guiará los primeros pasos del proceso de creación de los sistemas de gestión.

➤ Etapa de Planeación (Planificación):

En esta etapa se definen las estrategias que se utilizarán, la estructura organizacional que se requiere, el personal que se asigna, el tipo de tecnología que se necesita, el tipo de recursos que se utiliza y la clase de controles que se aplican en todo el proceso. Dentro del proceso, la planificación constituye una etapa fundamental y el punto de partida de la acción directiva, ya que supone el establecimiento de sub-objetivos y los cursos de acción para alcanzarlos.

➤ Etapa de Implementación (Gestión):

En su significado más general, se entiende por gestión la acción y efecto de administrar. En un contexto empresarial, se refiere a la dirección que toman las decisiones y las acciones para alcanzar los objetivos trazados. Las acciones que se toman para llevar adelante un propósito, se sustentan en los mecanismos o instrumentos administrativos (estrategias, tácticas, procedimientos, presupuestos, etc.), que están sistémicamente relacionados y que se obtienen del proceso de planificación.

➤ Etapa de Control:

El control es una función administrativa esencialmente reguladora, que permite verificar (o también constatar, palpar, medir o evaluar), si el elemento seleccionado ya sea la actividad, proceso, unidad o sistema, está cumpliendo sus objetivos o a alcanzando los resultados que se esperan. [2]

1.3 Gestión Académica

La gestión académica es un proceso con características fundamentales en el orden administrativo. En una institución contribuye al proceso de formación integral del estudiante, tiene relación directa con la enseñanza, el aprendizaje, y los procesos de admisión de estudiantes, matrículas, pensiones y la administración de edificios y servicios educativos integrales.

Algunos de los componentes de la Gestión Académica son:

- Procesos de preinscripción.
- Selección y admisión.
- Planes de estudio.
- Modelos pedagógicos.
- Ofertas de programas.
- Información académica y administrativa de los estudiantes y docentes.[3]

La función académica está conformada por una serie de variables, cuya ponderación depende en cada caso del efecto que se le asigne sobre el producto final. En el proceso enseñanza- aprendizaje, entendemos como producto final el egresado. [4]

- Recursos humanos dedicados a la docencia
- Infraestructura y equipamiento para la docencia
- Contexto socio-cultural de los alumnos
- Planes de estudio y programas
- Organización cátedra – departamento
- Proceso enseñanza aprendizaje

1.4 Sistemas de gestión académica

Realizar los procesos docentes de forma manual puede resultar una tarea compleja y tediosa debido al gran volumen de información involucrado. Afortunadamente la incorporación de las tecnologías de la información ha dado la posibilidad de informatizar estos procesos mediando el empleo de sistemas automatizados para la gestión académica. Estos sistemas constituyen una poderosa herramienta de trabajo que permite hacer más eficientes los procesos docentes y optimizar sus recursos.

SIGA: Sistema Integrado de Gestión Académica.

La Universidad Técnica Federico Santa María de Chile posee la importante misión de formar profesionales idóneos según las necesidades actuales del mundo, por ende su sello característico y diferenciador ha sido siempre ser una universidad líder en el área científica y tecnológica. Se ha propuesto mejorar las Tecnologías de la Información y las Comunicaciones (TICs) cada día más, con el fin de propiciar un crecimiento continuo de la universidad y agregar valor a la Organización. En función de lograr los resultados propuestos por dicha universidad se ha implementado el Sistema de Gestión Académica con las siguientes características:

Misión: Contribuir al desarrollo de las Tecnologías de Información con el fin de responder a las necesidades de la comunidad universitaria, a través del desarrollo de sistemas, que permitan mejorar la calidad de los servicios que brinda nuestra institución.

Visión: La comunidad universitaria tendrá acceso a servicios de Tecnologías de Información con calidad, rapidez, seguridad y sostenibilidad en el tiempo, lo que contribuye al cumplimiento de la misión de la universidad.

Objetivos Generales:

- Desarrollar los sistemas institucionales de la universidad.
- Fortalecer la Tecnología de Información en la organización como agente de cambio.
- Dar acceso a la comunidad universitaria, a los recursos de información local y globalmente disponible.

- Aumentar la interacción de la comunidad para el enriquecimiento del conocimiento, la información, las habilidades y la comunicación.
- Generar nuevos servicios.

Servicios: Actualmente se ofrecen los siguientes servicios:

- Clave personal: La clave única de acceso le permite al alumno acceder a todos los servicios ofrecidos con la posibilidad de cambiarla.
- Planes de Carrera: Entrega información sobre siglas y descripción de asignaturas, carga académica asociada a cada asignatura, expresada en número de créditos, duración en semestres y convalidaciones de asignaturas de todas las carreras ofrecidas por la universidad.
- Beca de Alimentación: Los alumnos con beca de alimentación, podrán hacer consultas sobre el total de almuerzos utilizados y disponibles durante el mes en curso.
- Horario Personal: Esta opción despliega en forma gráfica el horario personal del alumno, de acuerdo a las asignaturas inscritas en el periodo académico actual.
- Horario de Asignaturas: Se muestran los horarios y salas de las asignaturas que se dictan en el periodo académico en curso.
- Deudas: Se entrega información sobre el monto aún no cancelado por el alumno, tanto de la cuota de inscripción como de arancel.
- Preinscripción / Inscripción de Asignaturas: Permite al alumno preinscribir/inscribir asignaturas durante el proceso de matrícula de cada periodo académico. Una vez realizado este proceso, podrá consultar sobre las asignaturas inscritas efectivamente, e informarse sobre las inscripciones denegadas.
- Resumen Académico: Contiene información sobre las asignaturas cursadas, notas, reprobaciones, convalidaciones y asignaturas cursadas fuera del plan. Esta información es útil para determinar el avance en la carrera del alumno.
- Ficha Personal: Consiste en el despliegue de la ficha personal del alumno, permitiendo la actualización de la información.
- Encuesta Docente: Consiste en una serie de preguntas que tienen como finalidad contribuir al perfeccionamiento de la docencia.

- Acta de Notas: Los profesores pueden ingresar las notas finales de los alumnos. [5]

Este sistema presenta limitaciones como que está desarrollado en software propietario y es comercializado a un costo elevado. Además no responde a las necesidades que presenta la vice dirección docente de la facultad 3 de una mejor organización y control de los procesos docentes. Tampoco presenta ningún módulo dedicado a la gestión de alumnos ayudantes, y plan de estudio que se hacen necesarios para la gestión docente integral de dicha facultad.

SDI: Sistema Docente Integral.

Este sistema se creó en el Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE) y surgió debido a la vital importancia que tiene para este centro el control, planificación y resultado de los procesos docentes. El SDI permite registrar, procesar y crear mecanismos de recuperación de la información relacionada con el proceso docente, tanto de pregrado como de postgrado de forma automatizada. Esto constituye una ventaja ya que con su uso se pueden realizar estos procesos de forma eficiente permitiendo optimizar los recursos del centro secretaría de facultades. [6]

Este sistema dentro de los módulos implementados contemplan la recuperación de información, pagos de estipendios, módulos de postgrado pero que aún estos no se corresponden con las necesidades planteadas en el presente trabajo de realizar la gestión de perfiles de tesis, seguimiento de alumnos ayudantes y un mejor control del plan de estudio.

SIGENU: Sistema de Gestión de la Nueva Universidad.

En el 2004 a solicitud de la dirección del Ministerio de Educación Superior (MES) de Cuba, y como necesidad de automatizar los procesos fundamentales de la gestión académica de un Centro de Educación Superior en sus modalidades de estudio surge el proyecto SIGENU. El Sistema de Gestión de la Nueva Universidad es un software que se ha desarrollado con el fin de ser una herramienta que permita la gestión de toda la información académica vinculada con la educación superior en Cuba. En correspondencia con su carácter nacional y la gran diversidad de sistemas de enseñanza superior con que cuenta la universidad cubana, este sistema ha sido concebido de manera tal que sea capaz de brindar gran seguridad e integridad de la información. Así como ser tan flexible que permita ser adaptado a todos los centros de educación superior del país.

SIGENU está compuesto por cuatro elementos fundamentales:

- Base de Datos: Donde se almacena la información del sistema.

- Servidor de Aplicaciones: Actúa como intermediario entre la Base de Datos y las aplicaciones clientes y permite dar servicio a través de la red.
- Aplicación Cliente de Administración: Permite la inserción y actualización de los usuarios y contiene las funcionalidades que permiten a los administradores monitorear el sistema.
- Aplicación Cliente (SIGENU): Aplicación Desktop que permite la inserción y actualización de toda la información que se registra en el sistema y la obtención de reportes.

Consta de los siguientes módulos:

- **Codificadores:** Contiene toda la información con que debe contar el sistema y que es provista por el Ministerio de Educación Superior (MES).
- **Matrícula:** Permite realizar el proceso de matrícula a través del cual los estudiantes pasarán a ser registrados en el sistema como estudiantes de Educación Superior.
- **Control de estudiantes:** Permite buscar un estudiante registrado en el sistema, modificar los datos de un estudiante tanto personales como docentes, ubicar a un estudiante en un grupo o cambiarlo de grupo, realizar el pase de estudiantes a otros años de estudio y definir los que serán repitentes así como dar baja a un estudiante del centro ya sea por licencia de matrícula, resolución o traslado.
- **Plan de Estudio:** Permite la creación de los planes de estudio para las diferentes carreras del centro así como realizar ajustes y modificaciones a un plan de estudio específico.
- **Evaluaciones:** Permite registrar, modificar o eliminar las evaluaciones de los estudiantes registrados en el sistema así como premios y bonificaciones.
- **Reportes:** Permite obtener diversos reportes con los que se puede recuperar toda la información necesaria del sistema. [7]

En este sistema se reflejan varios de los procesos relacionados con los estudiantes como son; la matrícula, bajas, licencias especiales, repitencias, entre otros. Pero aún así no satisface las necesidades de la facultad 3 ya que no contempla la gestión de alumnos ayudantes y la planificación de tesis por lo que se puede decir que no resuelve el problema planteado en el presente trabajo.

Sistema de Gestión Académica de la Universidad (Akademos)

En la universidad se han utilizados varios sistemas para la gestión académica que han cubierto con la necesidad de un grupo de personas, pero aún así no ha cumplido con todas las expectativas del sistema docente empleado en la UCI. En el año 2005 se crea el Sistema de Gestión Académica de la Universidad (Akademos), con el objetivo de automatizar los procesos que intervienen en la labor académica del centro de estudios y que pueda, de manera constante, enfrentarse a los cambios y adaptarse de forma natural a los nuevos contextos o formas de hacer.

El sistema está dividido en siete módulos, que abarcan la mayoría de los procesos involucrados en la gestión docente. Los mismos interactúan entre sí para llevar a cabo cada una de las tareas que automatiza el sistema.

A continuación se nombra y describen algunos de los módulos:

Módulo Matrícula: Permite el control de los datos de los estudiantes que van a comenzar a ser parte de la Universidad, así como la gestión de los movimientos a que estos son sometidos en su paso por la misma. En los centros que aceptan cada año miles de estudiantes, la tarea de matricularlos puede convertirse en una labor titánica, si no se involucra a una gran cantidad de personas en la misma. Este módulo permite la descentralización del proceso de ingreso en un conjunto de personas tan grande como se decida y también es el encargado de definir la estructura del centro de estudios, la cual se concibe como un conjunto de estructuras de diferentes tipos: facultades, grupos docentes, que se organizan jerárquicamente para agrupar por un criterio determinado, tanto a los estudiantes como a los demás implicados en la labor docente, sean profesores, personal de secretaría o directivos.

Módulo Profesor: Da la posibilidad de planificar la carga docente de los profesores pues es el encargado de asignar al mismo a un departamento y además asignarle grupos. Los profesores son la materia fundamental con que cuenta un centro de estudios para desarrollar con éxito su labor formativa.

Módulo Estudiantes: Brinda a los estudiantes, principales protagonistas del proceso docente, un espacio destinado a brindarles información referida a su desempeño académico en su paso por la Universidad. [8]

Sin embargo, el sistema está desarrollado completamente en software propietario, por lo que no se adecua a las políticas de software tanto de la Universidad como del país, ya que se ha optado por el software libre debido a la serie de ventajas que proporciona su uso. Posee muy poca documentación,

lo que hace muy difícil su mantenimiento y soporte. Además, muchas de las funcionalidades de los módulos matrícula y profesor se encuentran repetidos, perdiendo eficiencia.

Sistema de Gestión Universitaria

El Sistema de Gestión Universitaria es una solución integral para la gestión de los procesos sustantivos de la universidad. Tiene como objetivo informatizar estos procesos partiendo de las debilidades de los sistemas existentes. Para ello se identificaron doce áreas de procesos. Una de ellas es Pregrado o Subsistema de Gestión Académica de Pregrado, el cual es el encargado de gestionar lo relacionado con la formación de estudiantes. Pregrado posee las funcionalidades del actual Sistema de Gestión Académica “Akademos”, solución que está en uso desde el 2005 e implementada con herramientas propietarias, e incorpora nuevos módulos y funcionalidades, entre los que se encuentra el módulo Gestión de Tesis. Este módulo surge a partir de la necesidad de informatizar el proceso de gestión de trabajos de diplomas en la universidad. En la actualidad se encuentra en construcción el cual cuenta con varios módulos y tiene previstas un conjunto de funcionalidades para una gestión integral.

Después del estudio y análisis realizado de los distintos gestores académicos expuestos, se puede concluir que estos sistemas están diseñados para una mejor gestión, organización y control de los procesos docentes. Sin embargo no dan solución a la gestión de repitencias y arrastres, al seguimiento y control de los perfiles de tesis y alumnos ayudantes que necesita la facultad 3. Además de que algunos de estos son software privativos y bajo licencias. Por ello, para dar solución al problema planteado es preciso desarrollar un software que permita mejorar la gestión, seguimiento y control de la actividad académica en el área de formación de la facultad 3.

1.5 Metodología y plataforma tecnológica a utilizar

En la facultad 3 de la Universidad de las Ciencias Informáticas, a partir de la decisión administrativa de informatizar los procesos sustantivos que en ella se desarrollan, surgió la necesidad de establecer una plataforma tecnológica para todas las aplicaciones. Esto permite que haya homogeneidad y una estrecha comunicación entre los diferentes sistemas. A partir de esta decisión se realizó un estudio que permitiera definir la metodología, lenguaje de programación, entorno de desarrollo así como las diferentes herramientas informáticas para su implementación. Esto está recogido en el documento PIN-F03001_2011 Propuesta de informatización. [9]

1.5.1 Metodología

Una buena elección de la metodología a utilizar garantiza la calidad del producto y el buen desarrollo del proceso del software para que se arrojen los mejores resultados.

En el documento Propuesta de informatización se establecen las metodologías ágiles Scrum y XP, por las propiedades y características del sistema a desarrollar se propone a utilizar la metodología SXP.

SXP (Scrum -XP)

Se selecciona la metodología SXP, que es la unión de las ya referidas Scrum y XP por las características del sistema a desarrollar. Para llevar a cabo el proceso de desarrollo del proyecto se tomarán en cuenta las mejores prácticas de ambas.

Se usará Scrum para la planificación del proyecto. De la misma serán tomadas algunas prácticas para su implantación al igual que de la metodología XP, procurando que el proceso sea efectivo y eficiente.

Consta de 4 fases principales:

- **Planificación-Definición:** se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** se realiza la implementación del sistema hasta que esté listo para ser entregado.
- **Entrega:** puesta en marcha.
- **Mantenimiento:** se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las historias de usuario (HU), Diseño, Implementación, Pruebas, de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, que permite mejorar el diseño cada vez que se añade una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo. Esta metodología es aplicada a la presente investigación por la razón de que debe ser desarrollada en un corto tiempo y es un proyecto pequeño.

Scrum se enfoca en las prácticas de organización y gestión, mientras que XP se centra más en las prácticas de programación. Esa es la razón de que funcionen tan bien juntas: tratan de áreas diferentes y se complementan entre ellas [10].

1.5.2 Herramientas utilizadas para el desarrollo del sistema

PHP5 5.3

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma rápido, con una gran librería de funciones y mucha documentación. Es un potente lenguaje de script del lado del servidor, que se utiliza principalmente para generar páginas de forma dinámica. Es tan potente que se utiliza para muchísimas cosas más: generar imágenes, generar PDF, entre otras. La sintaxis gramática y funciones del lenguaje son muy sencillas. Su compatibilidad con otros sistemas se incrementa agregando nuevas funciones. La manera de manejar los datos, las funciones disponibles y la documentación son realmente excepcionales. La mayor parte de su sintaxis ha sido tomada de C, Java y PERL con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML.

Ventajas de PHP5:

- Mejor soporte para la Programación Orientada a Objetos, que en versiones anteriores era extremadamente rudimentario.
- Mejoras de rendimiento.
- Manejo de excepciones.
- Soportado por una gran comunidad de desarrolladores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

- Entre las ventajas de PHP se puede destacar que este lenguaje es completamente expandible, está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código. Muchas interfaces distintas para cada tipo de servidor. [11]

Framework de desarrollo

CodeIgniter

CodeIgniter es un conjunto de herramientas para personas que construyen aplicaciones web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido que si lo escribiese desde cero, proporcionándole un rico juego de librerías para tareas comúnmente necesarias, así como una interfaz simple y una estructura lógica para acceder a esas librerías. Permite creativamente enfocarse en el proyecto minimizando la cantidad de código necesario para una tarea dada. Se encuentra bajo una licencia OpenSource Apache/BSD-style, lo cual facilita el éxito de las tendencias de migración a software libre del país y de la UCI, pudiéndose utilizar en cualquier proyecto que se desee. Está escrito para ser compatible con PHP 4. Aunque corre en PHP5, simplemente no toma ventaja de cualquiera de las características nativas que sólo están disponibles en esa versión.

Es verdaderamente liviano, ya que el núcleo del sistema sólo requiere unas pocas librerías que permiten realizar las tareas de desarrollo web más comunes como acceder a una base de datos, mandar un correo electrónico, validar datos de un formulario, mantener sesiones, manipular imágenes, entre otras, aunque se le pueden agregar librerías cargándolas dinámicamente a pedido, basado en las necesidades de un proceso dado, así que el sistema base es muy delgado y bastante rápido. [12]

Symfony 2.0

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. A continuación se muestran algunas de sus características.

Características de Symfony

- Fácil de instalar y configurar en la mayoría de las plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Symfony está desarrollado completamente con PHP 5, enfocado al desarrollo de aplicaciones en el mismo lenguaje de programación y está basado en un patrón clásico del diseño web conocido como arquitectura Modelo Vista Controlador (MVC). [13]

Sistema Gestor de Base de Datos (SGBD):

MySQL Server

MySQL Server es la base de datos de código abierto más usada del mundo desarrollado y proporcionado por MySQL AB, empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos.

Los fundadores buscaban la forma de crear un manejador de bases de datos que fuera “rápido”, todavía más rápido que el sistema final. Así surgió MySQL, primero como un producto de la empresa y después como software de dominio público.

Este servidor fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, éste ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad, y seguridad hacen que sea un servidor bastante apropiado para acceder a bases de datos en Internet. [14]

PostgreSQL 9.1

PostgreSQL es el motor de base de datos que se utilizará para almacenar toda la información de la aplicación, por ser un servidor de base de datos relacional libre, liberado bajo la licencia BSD.

Dentro de las características de PostgreSQL, están:

- Alta concurrencia
- Mediante un sistema denominado Acceso concurrente multiversión (MVCC) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.
- Multiplataforma.
- Herramientas gráficas de diseño y administración de bases de datos.
- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial). [15]

Object Relational Mapper

Doctrine es un potente y completo sistema Object Relational Mapper (ORM) para PHP 5.2 o superior, con un Database Abstraction layer (DBAL) incorporado. Entre muchas otras cosas tiene la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

Su principal ventaja radica en poder acceder a la base de datos utilizando la programación Orientada a Objetos (POO) debido a que doctrine utiliza el patrón Active Record para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrable a los principales framework de desarrollos utilizados actualmente. [16]

Como framework se selecciona Symfony ya que está bien documentado, es maduro y con una gran comunidad que lo apoya. Su estructura modelo vista controlador facilita su estudio. Como sistema gestor de base datos por sus características se utilizará PostgreSQL, se encuentra disponible en cualquier plataforma. PostgreSQL es un gestor magnífico, que posee una gran escalabilidad, haciéndolo idóneo para sitios web. PHP5 actualmente, es el lenguaje de software libre más utilizado en

la Internet con una amplia documentación, posee una de las comunidades más grandes, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente. Es multiplataforma y soporta POO (Programación Orientada a Objetos).

1.5.3 Servidor Web y el entorno integrado de desarrollo (IDE)

Servidor Web Apache 2.2.1

Apache es el servidor web por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Ventajas del servidor Web Apache

- Facilita la integración como "*plugins*" de lenguajes de programación de páginas web dinámicas.
- Provee interfaz a todas las bases de datos.
- Servidor altamente configurable de diseño modular: Se pueden escribir módulos para realizar determinadas funciones.
- Es flexible, rápido y eficiente.
- Se desarrolla de forma acelerada estimulando la retroalimentación desde sus usuarios a través de nuevas ideas, reportes de errores y parches.
- Multiplataforma.
- Es una tecnología gratuita de código fuente abierta. Esto le da una transparencia a este software de manera que si se quiere ver que es lo que se está instalando como servidor, se puede saber, sin ningún secreto. [17]

NetBeans v7.1

Se propone NetBeans como herramienta para el desarrollo del sistema, pues contiene las condiciones necesarias para que los desarrolladores de software puedan crear aplicaciones Web y móviles, a través de numerosos lenguajes de programación. Es un producto gratis y de código abierto que ofrece a los desarrolladores numerosas ventajas en la creación de nuevas aplicaciones multiplataforma. Ofrece características de soporte PHP ampliado que expande el soporte de los lenguajes dinámicos con apoyo para PHP 5. Esta plataforma permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos, los cuales dan la posibilidad de ser extendidos agregándole otros. Su flexibilidad entre plataformas, el cumplimiento de UML y la

capacidad de administrar la complejidad, ayudan a garantizar que las aplicaciones cumplan con los requerimientos específicos del negocio.

NetBeans es un entorno de desarrollo (IDE) sumamente completo, fácil de usar, cómodo, de excelente calidad y es completamente gratis. Es una aplicación de código abierto diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas. [18]

Se selecciona como servidor web, Apache porque es multiplataforma, rápido, eficiente, de código abierto con una fácil integración con los lenguajes de programación. Se decidió utilizar como entorno de desarrollo NetBeans porque es el único que da soporte para Symfony, es un IDE multilenguaje, completo y modular. Tiene gran cantidad de módulos de terceros (plugins), desarrollo intuitivo arrastrar y soltar, y completa código. Es gratis y de código abierto y posee una gran comunidad de usuarios y desarrolladores.

1.5.4 Herramientas y lenguajes de modelado

Visual Paradigm v6.4

Visual Paradigm es una herramienta que utiliza el lenguaje unificado de modelado (UML), el cual facilita la rápida construcción de aplicaciones de calidad y a un menor coste. Logra la integración con NetBeans IDE/Sun ONE, entre otros. Esta herramienta está diseñada para desarrollar software con programación orientada a objeto, cubre el ciclo de vida completo del desarrollo brindando ayuda a arquitectos, analistas, diseñadores y desarrolladores. Permite representar diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Ventajas

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.
- Producto de calidad.

- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones. [19]

UML Lenguaje Unificado de Modelado

UML es el lenguaje de modelado de sistemas de software más usado en la actualidad, y un estándar internacional aprobado por la Objeto de administración de grupos (OMG). Cuando se define UML no se puede hablar de métodos o guías, pues simplemente es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema intensivo de software. Este lenguaje ofrece una forma estándar para escribir un plano del sistema, incluyendo cosas conceptuales tales como procesos de negocios y funciones del sistema, así como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Características de UML

- Estandarización del lenguaje.
- Visualiza, especifica, construye y documenta los artefactos de un sistema que involucra una gran cantidad de software.
- Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y preside de reglas para combinar tales elementos los elementos.
- Permite la modelación de sistemas con tecnología orientada a objetos. [20]

Notación de gestión de los procesos del negocio. (BPMN)

Es un nuevo estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. Es una notación que modela los procesos del negocio, basada en diagramas de flujo fácil de entender. Está diseñada para cubrir varios tipos de modelado, permite la creación tanto de segmentos de procesos, como procesos de negocio de comienzo a fin en diferentes niveles de representatividad. BPMN es gráficamente más rico, con menos símbolos fundamentales, pero con más variaciones de éstos, lo que facilita su comprensión por parte de personas no expertas.

Ventajas

- Provee una notación que es fácilmente entendida por todos los usuarios, desde el analista de negocio, el desarrollador técnico y hasta las propias personas del negocio.
- Crea un puente estandarizado para el vacío existente entre el diseño del proceso de negocio y su implementación.
- Asegura que los lenguajes para la ejecución de los procesos de negocio puedan ser visualizados con una notación común.
- Es usado para comunicar una amplia variedad de información a una amplia variedad de audiencias.
- BPMN es un lenguaje gráfico para representar el negocio, facilitando a las organizaciones la habilidad para comunicar esos procedimientos de una manera estándar. [21]

Para realizar el modelado se utilizará la herramienta case Visual Paradigm Enterprise Edition (EE_VP_UML), usando la notación UML, excelente lenguaje de modelado visual y BPMN un lenguaje gráfico para representar el negocio, además Visual Paradigm soporta el ciclo de vida completo de un software y la Universidad de las Ciencias Informática paga la licencia de su uso.

1.6 Conclusiones del capítulo

A partir de la revisión bibliográfica realizada se pudo constatar que los sistemas de gestión académica existentes incorporan como principales funcionalidades la gestión de matrículas, control de evaluaciones pero no dan respuesta a las necesidades de la facultad 3 de llevar el seguimiento y control de alumnos ayudantes, plan de estudio, perfiles de tesis, arrastres y repitencias. Por estas razones se concluye que es necesario implementar un sistema que sea capaz de satisfacer a los requerimientos de la secretaría docente y la vice dirección de dicha facultad.

Para la implementación del sistema se realizó un previo estudio del documento PIN-F03001_2011 Propuesta de informatización que norma los requerimientos tecnológicos para los sistemas informáticos de la facultad. [9]

Se realizó un análisis crítico de la metodología, lenguaje de programación y herramientas propuestas y se pudo comprobar que es idóneo para el desarrollo del sistema de gestión académica para la facultad 3 teniendo en cuenta las principales características y requerimientos del sistema.

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

El presente capítulo aborda las principales características del sistema. Con este fin se realiza el modelado de los principales procesos del negocio, se especifican las personas relacionadas con el sistema, además son identificadas las funcionalidades que debe brindar a partir de la Lista de Reserva del Producto (LRP) y las descripciones de las historias de usuarios.

2.2 Modelo de Historia de Usuario del Negocio

Se utilizará Modelo de Proceso del Negocio (BPMN), para una mejor comprensión entre la interacción cliente, usuarios y procesos. Se incluye dentro de uno de los artefactos que genera la metodología implantada Plantilla Modelo de Historia de usuario del Negocio.

Actores del negocio

Los actores del negocio son aquellas personas que interactúan con el negocio para beneficiarse de sus resultados:

Decana: Es la principal instancia en la facultad y quien tiene el conocimiento de todos los reportes conformados en la misma.

Vicedecano de formación: Es la principal autoridad en los temas docentes.

Jefes de centro: Lleva el control de todas las informaciones referidas al centro al que pertenece.

Trabajadores del negocio

Los trabajadores del negocio son las personas que realizan una o varias actividades en el mismo, interactúan con otros trabajadores manipulando las entidades del negocio para que funcione.

Vicedecano de formación: Es la persona encargada de registrar, actualizar y controlar lo referido a los temas docentes.

Subdirectores de formación: Se encargan de llevar el control de los perfiles de tesis.

Jefes de departamentos: Es quien lleva el control de las evaluaciones de los alumnos ayudantes.

Comités revisores: Son los encargados de revisar los perfiles de tesis después de creados y actualizar su estado.

Jefe de tribunal: Se encarga de actualizar las evaluaciones e informar el estado de la tesis.

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Secretaria Docente: Es quien actualiza y gestiona la información de los estudiantes a la vicedecana referida al las evaluaciones y plan de estudio.

2.2.1 Modelo del negocio

Tabla 1 Proceso repitencias y arrastres.

Ficha de proceso	
Proceso:	Repitencias y arrastres
Descripción :	El proceso se inicia cuando la secretaria docente realiza un reporte de las notas de los estudiantes en cada convocatoria (ordinario, extraordinario y mundial) a partir de la consulta de las actas de exámenes. Esta información se le envía a la vicedecana de formación, filtra por notas para obtener los estudiantes desaprobados y se analizan para tomar decisiones en caso de arrastres o repitencias.
Entradas:	Actas de exámenes.
Salidas:	Registro de estudiantes repitencias y arrastres.
Actividades principales	
Actividad 1	Registrar evaluaciones. (Son insertadas las notas al estudiante de cada asignatura, en sus respectivas convocatorias).
Actividad 2	Enviar reporte de las evaluaciones. (Se envía el reporte de notas a la dirección de la facultad).
Actividad 3	Buscar desaprobados. (Luego de tener el listado de estudiantes suspensos se analizan).
Actividad 4	Definir condición de arrastres o repitencias. (Después de analizados se define la condición a asignar).

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

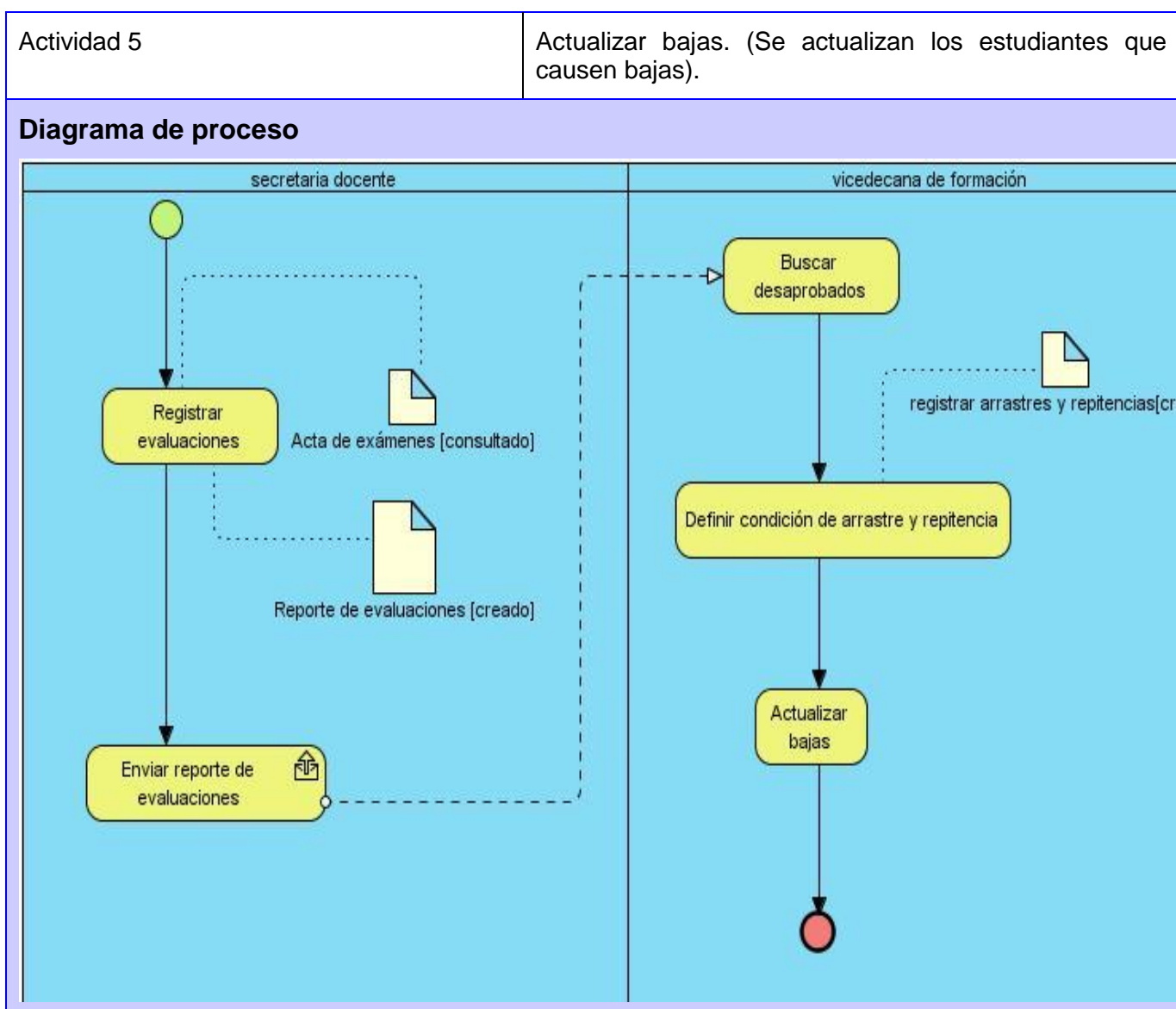


Tabla 2 Proceso Plan de estudio.

Ficha de proceso	
Proceso:	Plan de estudio

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Descripción:	El proceso inicia cuando se registran a los estudiantes las siguientes observaciones: (traslados, reincorporados por licencias, ingresos, repitencias y arrastres). La secretaria docente se encargada de realizar las convalidaciones de las asignaturas que cursarán estos estudiantes teniendo en cuenta las que ya recibió y el plan de estudio actual. Se les asigna grupos a los diferentes estudiantes en dependencia de la capacidad de las matrículas por asignaturas.
Entradas:	Registro de estudiantes.
Salidas:	Convalidaciones de asignaturas y grupos donde serán insertados.
Actividades principales	
Actividad 1	Registrar estudiantes con observaciones. (Todos los estudiantes que son: incorporados por licencia, traslados, ingresos y repitencias se identifican).
Actividad 2	Realizar convalidaciones de las asignaturas. (Luego se realizan las equivalencias entre asignaturas teniendo en cuenta las cohortes y se realizan las convalidaciones).
Actividad 3	Conformar grupos. (En el caso de que la matrícula por asignaturas sea menor que la cantidad de estudiantes a asignar se crea un nuevo grupo).
Actividad 4	Asignar grupo a estudiantes con observaciones. (Se asigna el grupo).
Diagrama de proceso	

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

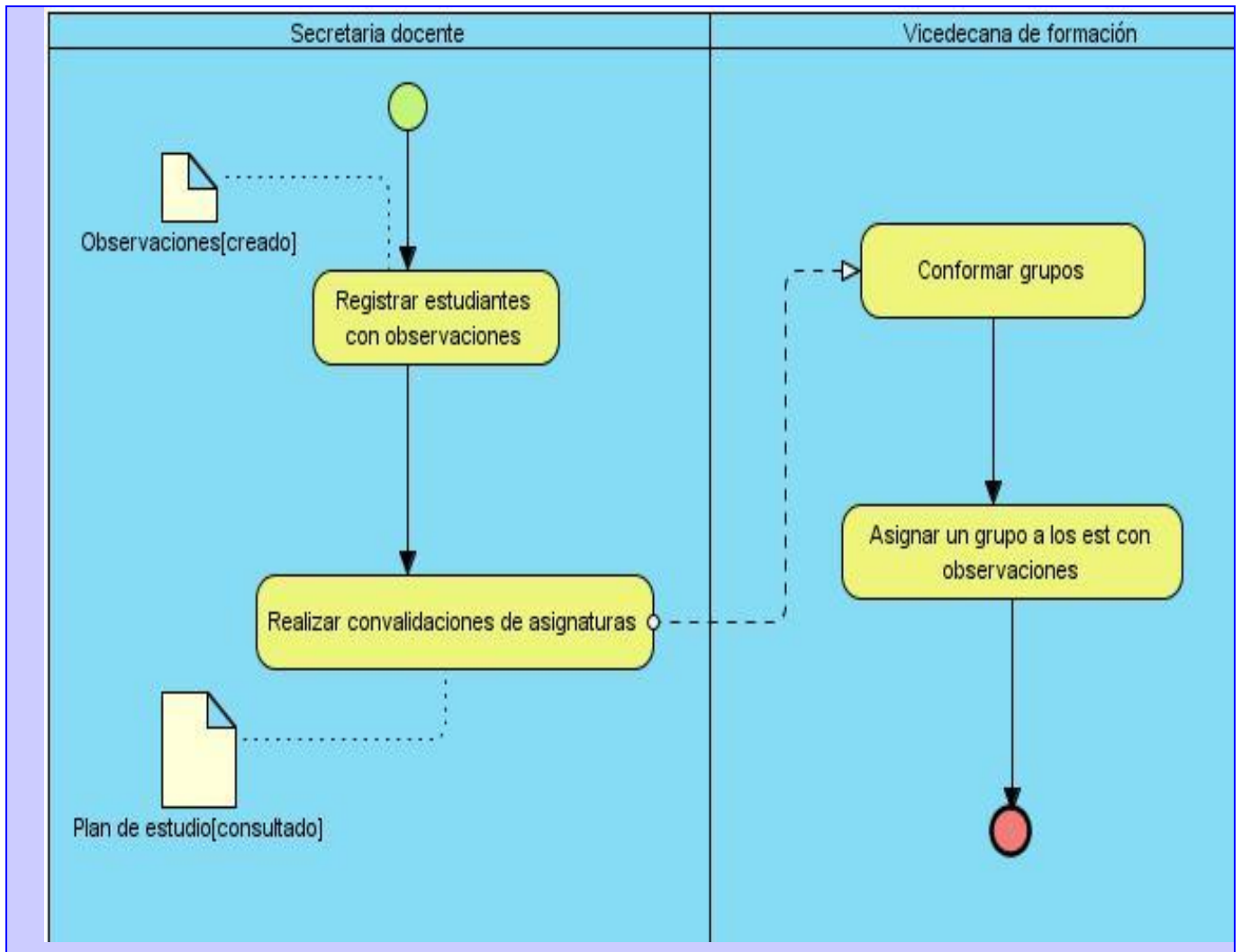


Tabla 3 Proceso Perfiles de tesis.

Ficha de proceso	
Proceso:	Perfiles de tesis

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Descripción:	El proceso inicia cuando el perfil de tesis es generado por el banco de problemas, los subdirectores de formación se encargan de la gestión. Se conforman los comités revisores del centro que revisarán los perfiles elaborados, luego de revisados por el centro, los aprobados son enviados a los comités revisores a nivel de facultad que se encargan después de revisados y aprobados hacerlos oficial, actualizan los estado ya sea aprobado o rechazado. Se les asigna un tribunal a cada uno de los aprobados y el jefe de tribunal será el encargado de llevar el control de las evaluaciones en cada taller que se realice hasta la defensa.
Entradas:	Perfiles de tesis.
Salidas:	Registro de evaluaciones.
Actividades principales	
Actividad 1	Crear perfiles de tesis.(Son creados a partir del banco de problemas)
Actividad 2	Conformar comités revisores del centro. (Se conforman los comités que revisarán los perfiles.)
Actividad 3	Revisar perfiles de tesis a nivel de centro. (Se verifica el alcance y la validez del perfil).
Actividad 4	Actualizar estado de los perfiles.
Actividad 5	Enviar perfiles revisados a comités de la facultad. (Son enviados los perfiles para que sean revisados a nivel de facultad).
Actividad 6	Asignar estado a los perfiles (aprobado o rechazado).
Actividad 7	Conformar comité revisores de la facultad. (Serán los encargados de revisar luego del comité del

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

	centro)
Actividad 8	Controlar y organizar el proceso de revisión. (Se encarga del control de las revisiones a nivel de la facultad).
Actividad 9	Enviar perfiles de tesis aprobados. (Se les asigna un tribunal de tesis).
Actividad 10	Realizar talleres de tesis. (Se efectúan los talleres de tesis).
Actividad 11	Registrar evaluaciones de los talleres. (Son registradas las evaluaciones de cada taller, predefensa y defensa).
Actividad 12	Enviar evaluaciones. (Se envían las evaluaciones para ser registradas por la subdirección docente).

Diagrama de proceso

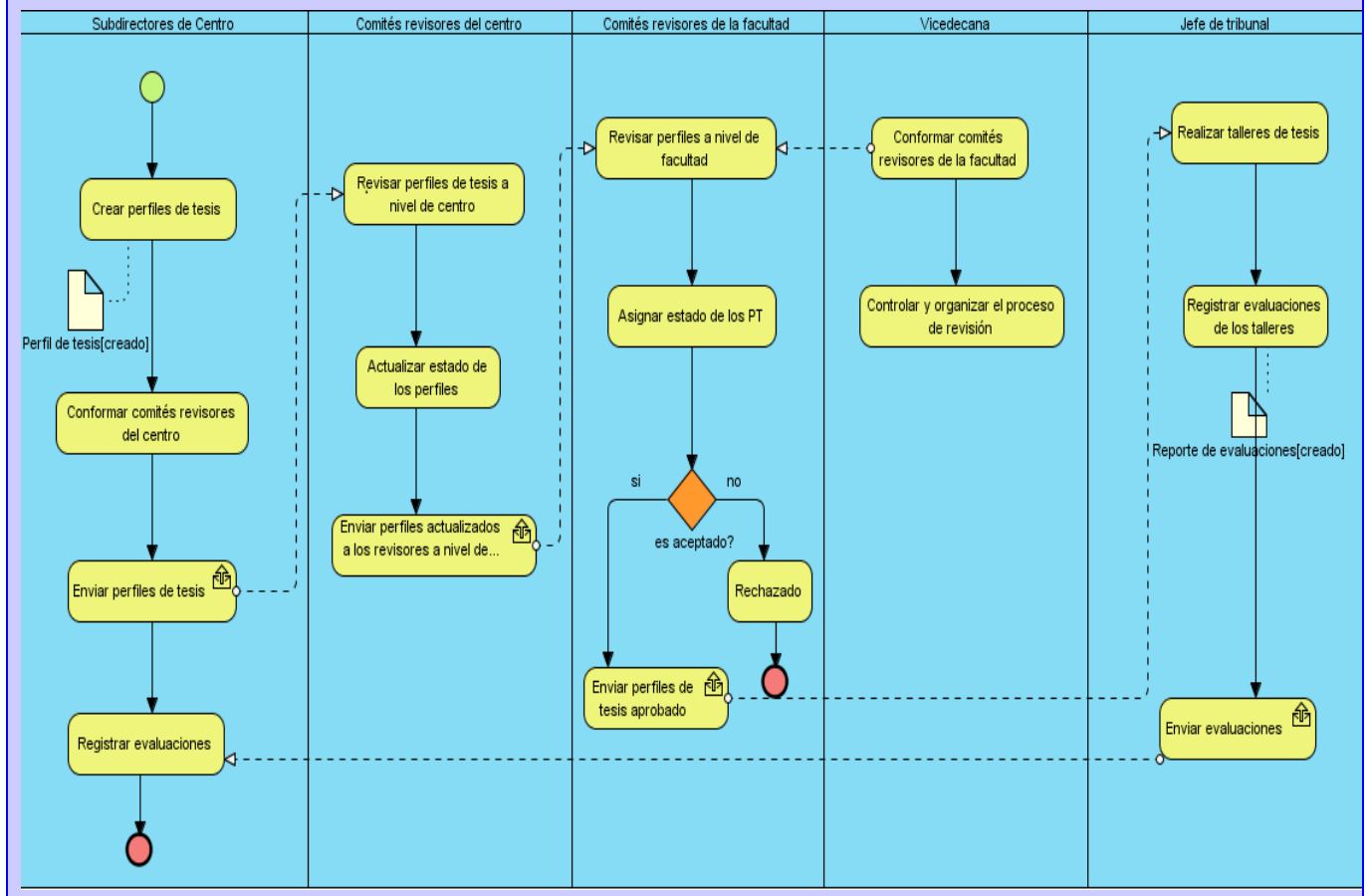


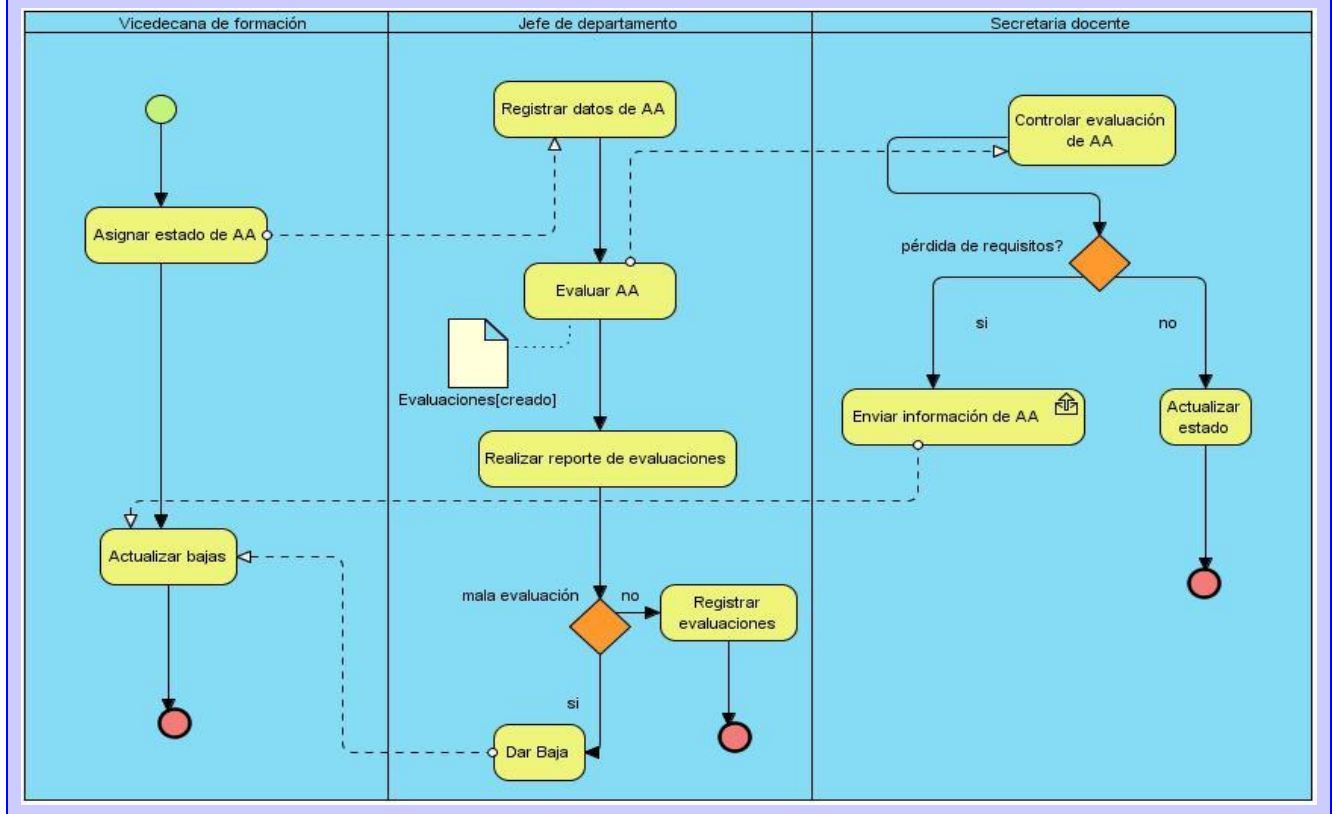
Tabla 4 Proceso Seguimiento y Control de Alumnos Ayudantes.

Ficha de proceso	
Proceso:	Alumnos ayudantes
Descripción:	El proceso se inicia cuando se actualiza el estado de alumnos ayudantes a los estudiantes teniendo en cuenta los que son nuevos ingresos los ratificados o inactivos. El jefe de departamento actualiza los datos de los estudiantes que pertenezcan al movimiento AA: tutor, asignatura que imparte, departamento al que pertenece y evaluación. Se registran las evaluaciones con periodicidad mensual, dos evaluaciones con R al igual que una con M causan bajas. La secretaria docente puede poner pérdida de requisitos por mundial o por sanción disciplinaria. La vicedecana docente lleva el seguimiento y control de los AA y actualiza las bajas.
Entradas:	Registro de estudiantes
Salidas:	Registro de alumnos ayudantes.
Actividades principales	
Actividad 1	Asignar estado de AA. (Se incorporan los estudiantes al movimiento).
Actividad 2	Registrar datos de AA. (Son registrados los datos correspondientes del AA).
Actividad 3	Evaluar AA. (Se lleva el control de las

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

	evaluaciones)
Actividad 4	Controlar evaluación de AA. (Se analizan los estudiantes por pérdidas de requisitos).
Actividad 5	Realizar reporte de evaluaciones.
Actividad 6	Registrar evaluaciones.
Actividad 7	Dar baja. (Se da de baja del movimiento de AA por pérdidas de requisitos, mala evaluación, y mundiales)
Actividad 8	Actualizar bajas.

Diagrama de proceso



2.4 Lista de reserva del producto

Es una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes. Esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente.

2.4.1 Requisitos Funcionales

A continuación se listan una serie de requisitos funcionales con los que el sistema debe cumplir:

Tabla 5 Requisitos funcionales.

Código	Descripción	Prioridad
RFN1	Autenticar Usuario	Alta
RFN2	Definir el nivel de acceso a la información de cada usuario que interactúa con el sistema.	Alta
Control de repitencias y arrastres		
RFN3	Registrar asignaturas.	Media
RFN4	Mostrar asignaturas.	Media
RFN5	Registrar notas de las convocatorias.	Alta
RFN6	Mostrar notas.	Media
RFN7	Mostrar estudiantes desaprobados.	Media
RFN8	Actualizar bajas.	Alta
Ajuste del plan de estudio		
RFN9	Registrar estudiantes con observaciones.	Alta
RFN10	Realizar convalidaciones de asignaturas.	Alta
RFN11	Asignar grupos a estudiantes.	Media

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Perfiles de tesis		
RFN12	Insertar perfiles de tesis.	Media
RFN13	Conformar tribunales de centro.	Media
RFN14	Conformar tribunales de facultad.	Media
RFN15	Asignar estado al perfil.	Media
RFN16	Registrar evaluaciones de los talleres de tesis.	Alta
Seguimiento y control de alumnos ayudantes		
RFN17	Asignar estado de alumnos ayudantes.	Media
RFN18	Registrar datos de los alumnos ayudantes.	Media
RFN19	Evaluar alumnos ayudantes.	Media
RFN20	Actualizar bajas de alumnos ayudantes.	Media

2.4.2 Requisitos No Funcionales

Como requisitos no funcionales se definen a continuación un conjunto de cualidades con las que debe cumplir el sistema.

Usabilidad

Facilidad de uso por parte de los usuarios: El módulo debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.

Fiabilidad

- Servicios Web restringidos: Los servicios Web que brinde el módulo deben estar restringidos a grupos de usuarios definidos y aprobados previamente.

- Políticas de seguridad por usuarios y roles: El módulo debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado.
- Alta protección de los datos: Al estar trabajando con información sensible, se hace necesario una alta protección de los datos a nivel de aplicación y de tráfico de red, para tal fin se ha definido además la seguridad en varios niveles dentro de la aplicación: nivel de interfaz, nivel de acceso a datos y nivel de base de datos.

Interfaz

Interfaz Web:

- La interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes.
- Combinar correctamente los colores, tipo de letra y tamaño.
- Mantener el mismo formato en todas las páginas.
- Los íconos deben estar en correspondencia con lo que representan.

Interfaces Hardware

- Para el desarrollo: PC Intel Pentium 4 o superior, CPU 3GHZ o superior, 512 MB RAM o superior, 160 GB HDD o superior.
- Para explotación del cliente: PC Pentium 3 o superior, CPU 133 MHZ o superior, 256 RAM mínimo 512 RAM recomendada o superior.
- Para explotación del servidor: CPU Dual Core 2.0 GHZ o superior, memoria RAM de 4 GB, 250 GB HDD.

Soporte

- Realizar distintas pruebas al software una vez concluido para comprobar su funcionalidad.
- Ofrecer los servicios de instalación y configuración de la aplicación una vez terminado el software.
- Prestar servicios de mantenimiento del software.
- Para el servidor de aplicaciones se requiere que esté instalado un intérprete de ficheros PHP y con las últimas actualizaciones del lenguaje.

- Para el servidor de base de datos se requiere que esté instalado un gestor de base de datos que soporte grandes volúmenes de datos y velocidad de procesamiento.

Eficiencia

- Se desea un sistema eficiente con gran nivel de precisión, con tiempo de procesamiento de información y tiempo de respuestas rápidos, y que mantenga siempre la consistencia de los datos.
- El sistema deberá operar con grandes cantidades de información, por lo que se necesita de alta capacidad de concurrencia del servidor de datos y del servidor web donde se encuentre alojada la aplicación.

2.5 Historias de usuario (HU)

Al utilizar la metodología SXP se generaron diferentes artefactos. Unos de los más importantes son las HU, pues son la forma en que se especifican los requisitos del sistema, lo que equivaldría a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el módulo debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. A continuación se describen las principales historias de usuario, las cuales tienen como entrada principal la Lista de reserva del producto descrita anteriormente. Se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos. Reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

Tabla 6 HU Autenticar usuario.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Autenticar usuario
Usuario: Usuario	Iteración Asignada: 1
Descripción: Se autentican los usuarios del sistema.	
Observaciones: Podrán ser registrados después de autenticados, los profesores.	

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Tabla 7 HU Definir el nivel de acceso a la información de cada usuario que interactúa con el sistema.

Historia de Usuario	
Número: <i>HU_2</i>	Nombre Historia de Usuario: Definir el nivel de acceso a la información de cada usuario que interactúa con el sistema.
Usuario: Usuario	Iteración Asignada: 1
Descripción: Se les dará los permisos a acceder a las funcionalidades a las que les corresponde.	
Observaciones: Se debe definir un grupo de roles a los que se les define el permiso correspondiente.	

Tabla 8 HU Registrar asignaturas.

Historia de Usuario	
Número: <i>HU_3</i>	Nombre Historia de Usuario: Registrar asignaturas.
Usuario: Secretaria docente	Iteración Asignada: 1
Descripción: El sistema debe permitir insertar y listar todas las asignaturas correspondientes al plan de estudio.	
Observaciones:	

Tabla 9 HU Mostrar asignaturas.

Historia de Usuario	
Número: <i>HU_4</i>	Nombre Historia de Usuario: Mostrar asignaturas.
Usuario: Secretaria docente	Iteración Asignada: 1

Descripción: El sistema debe listar todas las asignaturas correspondientes al plan de estudio.

Observaciones: Se deben tener en cuenta las cohortes de cada año.

Tabla 10 HU Registrar notas.

Historia de Usuario	
Número: HU_5	Nombre Historia de Usuario: Registrar notas
Usuario: Secretaria docente	Iteración Asignada: 1
Descripción: La secretaria debe ingresar las notas de cada asignatura por semestre cursada por el estudiante.	
Observaciones: Serán insertadas las notas de las distintas convocatorias: ordinario, extraordinario y mundial.	

Tabla 11 HU Mostrar notas.

Historia de Usuario	
Número: HU_6	Nombre Historia de Usuario: Mostrar notas
Usuario: Secretaria docente	Iteración Asignada: 1
Descripción: Se mostrarán las notas insertadas a cada estudiante.	
Observaciones:	

Tabla 12 HU Mostrar estudiantes desaprobados.

Historia de Usuario	
Número: HU_7	Nombre Historia de Usuario: Mostrar estudiantes desaprobados.

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Usuario: Secretaria docente	Iteración Asignada: 1
<p>Descripción: El sistema deberá filtrar por las distintas convocatorias de las notas y obtener los estudiantes desaprobados para determinar los arrastres y repitencias.</p> <p>Permitirá que cuando se le otorgue al estudiante la repitencia o arrastre, se registre.</p>	
<p>Observaciones: En el caso de los arrastres se puede desaprobar un mundial por semestre o sea puede tener dos asignaturas desaprobadas pero en distintos semestres.</p> <p>En el caso de las repitencias en el primer semestre más de 4 asignaturas desaprobadas, o asignaturas de arrastre invalidadas por asistencias. A final de curso, más de 5 asignaturas desaprobadas, o arrastre suspenso. En el caso de 5to año será repitencia, en el caso de una asignatura desaprobada o invalidada por asistencia, tesis aplazadas por desaprobar la predefensa o asignatura desaprobada en el 2do semestre</p>	

Tabla 13 HU Actualizar bajas.

Historia de Usuario	
Número: HU_8	Nombre Historia de Usuario: Actualizar bajas
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Vicedecano de formación	Iteración Asignada: 1
Descripción: El sistema deberá permitir actualizar los estudiantes que hayan causado bajas.	
Observaciones:	

Tabla 14 HU Registrar estudiantes con observaciones.

Historia de Usuario	
Número: HU_9	Nombre Historia de Usuario: Registrar estudiantes con observaciones.

Modificación de Historia de Usuario Número: Ninguna	
Usuario: Secretaria docente	Iteración Asignada: 1
Descripción: El sistema debe permitir actualizar y registrar los estudiantes reingresos, incorporados por licencias, arrastres, traslados y repitencias. Además de poder llevar el control de las asignaturas que tendría que recibir cada uno de ellos.	
Observaciones: Según la capacidad de la matrícula de los grupos, insertar a lo estudiantes reingresos, incorporados por licencia y repitencia.	

Tabla 15 HU Realizar convalidaciones de asignaturas.

Historia de Usuario	
Número: HU_10	Nombre Historia de Usuario: Realizar convalidaciones de asignaturas.
Usuario: Secretaria docente	Iteración Asignada: 1
Descripción: Se le harán las convalidaciones de las asignaturas según la cohorte del año en que ingresó, o sea las asignaturas que tendría que recibir el estudiante teniendo en cuenta las que ya recibió y las del plan docente actual.	
Observaciones: Para poder realizar las convalidaciones se tienen que hacer las equivalencias entre las asignaturas.	

Tabla 16 HU Asignar grupos a estudiantes.

Historia de Usuario	
Número: HU_11	Nombre Historia de Usuario: Asignar grupos a estudiantes.
Usuario: Secretaria docente	Iteración Asignada: 1

Descripción: Según la matrícula en los grupos por asignaturas, insertar estudiantes arrastres.

Observaciones:

Tabla 17 HU Insertar perfiles de tesis.

Historia de Usuario	
Número: HU_12	Nombre Historia de Usuario: Insertar perfil de tesis
Usuario: Subdirectores de formación	Iteración Asignada: 1
Descripción: El sistema debe registrar los perfiles de tesis insertados con los siguientes datos: tutor, título, clasificación, revisiones, cantidad de estudiantes, tribunales por nombres, integrantes de los tribunales.	
Observaciones:	

Tabla 18 HU Conformar comités revisores del centro.

Historia de Usuario	
Número: HU_13	Nombre Historia de Usuario: Conformar comités revisores del centro
Usuario: Subdirectores de formación	Iteración Asignada: 1
Descripción: El sistema debe permitir registrar los comités revisores del centro conformados por el subdirector de formación.	
Observaciones: Los comités revisores del centro serán los encargados de revisar los perfiles de tesis elaborados y luego lo que estén aprobados enviarlos a los comités de revisores de la facultad.	

Tabla 19 HU Conformar comités revisores de la facultad.

Historia de Usuario

Número: HU_14	Nombre Historia de Usuario: Conformar comités revisores de la facultad.	
Usuario: Vicedecano de formación	Iteración Asignada: 1	
Descripción: El sistema debe permitir registrar los comités revisores de tesis conformados por el vicedecano de formación.		
Observaciones: El comité de tesis de la facultad es el encargado de aprobar (hacer oficial el trabajo de diploma) o rechazar los perfiles de trabajos de diploma e informar al tutor en ambos casos.		

Tabla 20 HU Asignar estado del perfil.

Historia de Usuario		
Número: HU_15	Nombre Historia de Usuario: Asignar estado del perfil	
Usuario: Presidente de comités revisores.	Iteración Asignada: 1	
Descripción: El objetivo de esta historia de usuario es que después de revisado el perfil de tesis se le asigne el estado correspondiente.		
Observaciones: Después de la revisión de los perfiles los posibles estados son: aprobado o rechazado. A cada uno de los perfiles aprobados se le asignarán los tribunales de tesis.		

Tabla 21 HU Registrar evaluaciones.

Historia de Usuario		
Número: HU_16	Nombre Historia de Usuario: Registrar evaluaciones	
Usuario: Jefe del tribunal	Iteración Asignada: 1	

Descripción: El sistema debe permitir registrar las evaluaciones de las distintas convocatorias.

Observaciones:

Las convocatorias son taller 1, taller 2, taller 3, predefensa y defensa.

En el caso de los talleres la nota es cualitativa o cuantitativa.

Si es aplazado mostrar la opción.

La evaluación de la predefensa es satisfactorio o insatisfactorio.

Tabla 22 HU Asignar estado de alumnos ayudantes.

Historia de Usuario	
Número: HU_17	Nombre Historia de Usuario: Asignar estado de alumnos ayudantes
Usuario: Vicedecano de formación	Iteración Asignada: 1
Descripción: Identificar los estudiantes que pertenezcan al movimiento de alumnos ayudantes.	
Observaciones: El sistema debe permitir actualizar el estado de los alumnos ayudantes, ya sea ratificados, nuevos ingresos o inactivos.	

Tabla 23 HU Registrar datos de alumnos ayudantes.

Historia de Usuario	
Número: HU_18	Nombre Historia de Usuario: Registrar datos de alumnos ayudantes
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Jefe de Dpto	Iteración Asignada: 1
Descripción: El jefe de departamento debe registrar los datos a los alumnos ayudantes: tutor, asignatura que imparte, dpto y evaluación.	

Observaciones:

Tabla 24 HU Evaluar alumnos ayudantes.

Historia de Usuario	
Número: HU_19	Nombre Historia de Usuario: Evaluar alumnos ayudantes
Usuario: Vicedecano de formación	Iteración Asignada: 1
Descripción: Las evaluaciones son con periodicidad mensual. Dos evaluaciones con R causan baja de forma automática al igual que una con M.	
Observaciones:	

Tabla 25 HU Actualizar bajas de alumnos ayudantes.

Historia de Usuario	
Número: HU_20	Nombre Historia de Usuario: Actualizar bajas de alumnos ayudantes
Usuario: Vicedecano de formación	Iteración Asignada: 1
Descripción: El sistema debe permitir actualizar los estudiantes que sean bajas del movimiento de alumnos ayudantes.	
Observaciones: Las bajas del movimiento de alumnos ayudantes pueden ser voluntarias, por pérdidas de requisitos, por mundiales o por mala evaluación.	

2.6 Patrones de diseño

Patrón Arquitectónico Modelo-Vista-Controlador (MVC)

Los patrones son soluciones a problemas recurrentes que ocurren una y otra vez en nuestro entorno. Se pueden considerar como recetas para solucionar varias veces un problema del mismo tipo. Los desarrolladores lo usan como una forma de reutilizar la experiencia, clasificando las soluciones con términos de común denominación. En el diseño de la propuesta de solución se tiene en cuenta un

patrón clásico del diseño web conocido como arquitectura modelo-vista-controlador (MVC) que implementa Symfony MVC está formado por tres niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Mantiene aislado al modelo y a la vista de los detalles del protocolo utilizado para las peticiones.

De esta forma se separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

Una aplicación que utilice la arquitectura MVC separa el código del programa en tres capas, según su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. Se pueden utilizar otros patrones de diseño de forma tal que las capas puedan ser subdivididas en otras.

- Capa del modelo: Se puede dividir en la capa de acceso a los datos y en la capa de abstracción de la base de datos. De esta forma se logra crear una independencia entre las funciones para el acceso a los datos y el sistema gestor de base de datos, de manera tal que si el sistema cambia solo sea necesario modificar la capa de abstracción de la base de datos.
- Capa de la vista: Generalmente en una aplicación web las páginas suelen contener elementos que se muestran a lo largo de toda la aplicación de la misma forma y solo cambia el interior de las mismas. Esto permite que se pueda dividir la vista en un layout¹ y en una plantilla, de forma tal que el layout sea global para toda la aplicación o a para un grupo de páginas específicas y la plantilla solo muestre las variables que define el controlador.
- Capa del controlador: En una aplicación web el controlador suele tener una gran carga de trabajo ya que debe encargarse del manejo de las peticiones del usuario, la seguridad, cargar

¹ Plantillas donde se realizan las llamadas a funciones y variables de PHP.

la configuración de la aplicación y otras tareas similares. Por este motivo el controlador se puede dividir en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página. Utilizar un controlador frontal permitirá tener un único punto de entrada para toda la aplicación lo que hace más fáciles las configuraciones de acceso al sistema. [10]

Patrones GOF

Decorador: Este patrón permite añadir funcionalidad a una clase dinámicamente. Symfony lo implementa a través de la utilización de una plantilla global que en este caso es la plantilla base, que almacena el código HTML, es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. De esta forma el contenido de las plantillas específicas de un módulo se integra en la base, el layout decora la plantilla.

Patrones GRASP

Experto: Este patrón se evidencia en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del componente, como por ejemplo las clases controladoras y las del modelo.

Creador: En las clases controladoras de cada uno de los módulos se encuentran definidas las acciones del sistema y se ejecutan cada una de ellas. En las acciones se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase controladora es "creador" de dichas entidades.

Alta cohesión: Este patrón fue utilizado en el diseño del componente de manera general, los paquetes son organizados en la manera cada clase debe implementar las operaciones que estén sobre la misma área funcional. La clase controladora tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.

Controlador: Las peticiones son manejadas por una clase controladora que es el punto de entrada a alguna funcionalidad de la aplicación. Cuando el controlador recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

Bajo Acoplamiento: Este patrón es evidenciado en cada uno de los módulos del sistema, cada clase controladora maneja la información de las entidades que están relacionadas con las funcionalidades que se especifican en dicha clase controladora para lograr un bajo acoplamiento de clases.

2.7 Estructura de las clases del sistema

Para una mejor comprensión de la aplicación a continuación se muestran algunos elementos generales principales que se tuvieron en cuenta para el diseño. En la aplicación existen diferentes elementos: las clases entidades que representan el modelo, las clases controladoras y las plantillas para las vistas. En la generalidad de los requisitos o funcionalidades, la forma en que interactúan es la misma. La vista hace una petición a la controladora, esta toma la petición la procesa interactuando con las clases entidades que esta relacionado, construye una respuesta y la devuelve a la vista. A continuación se muestra el diagrama de clases del diseño de forma genérica. (Ver figura 1)

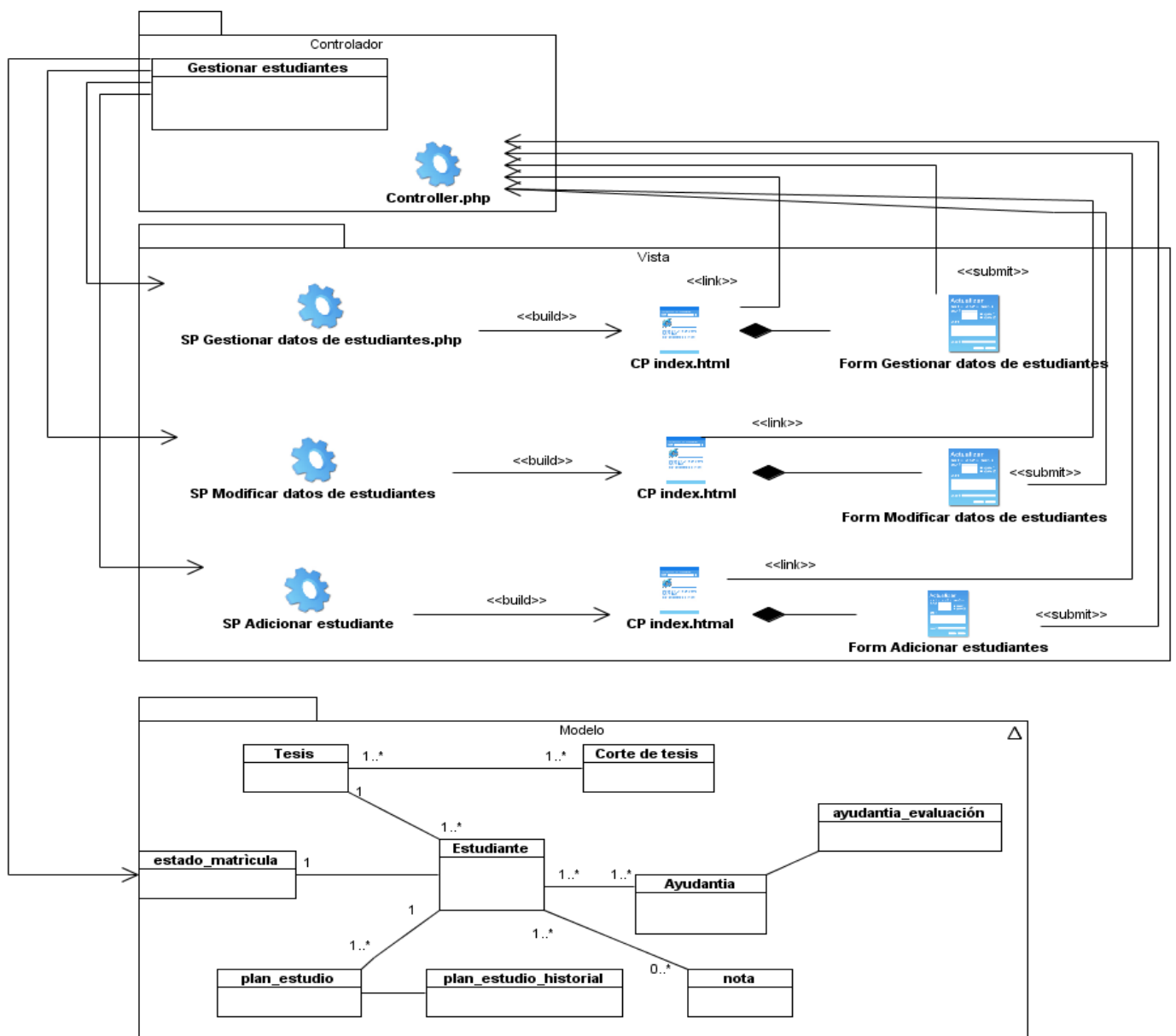


Figura 1 Diagrama de clases del diseño genérico Gestionar estudiantes.

2.8 Diseño de la Base de Datos

En general, el objetivo del diseño de una base de datos relacional es generar un conjunto de esquemas de relaciones que permitan almacenar la información con un mínimo de redundancia, pero que a la vez faciliten la recuperación de la información. Una de las técnicas para lograrlo consiste en diseñar esquemas que tengan una forma normal adecuada. El modelo de la base de datos está conformado por 31 tablas de las cuales 16 son nomencladores. (Ver figura 2).

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

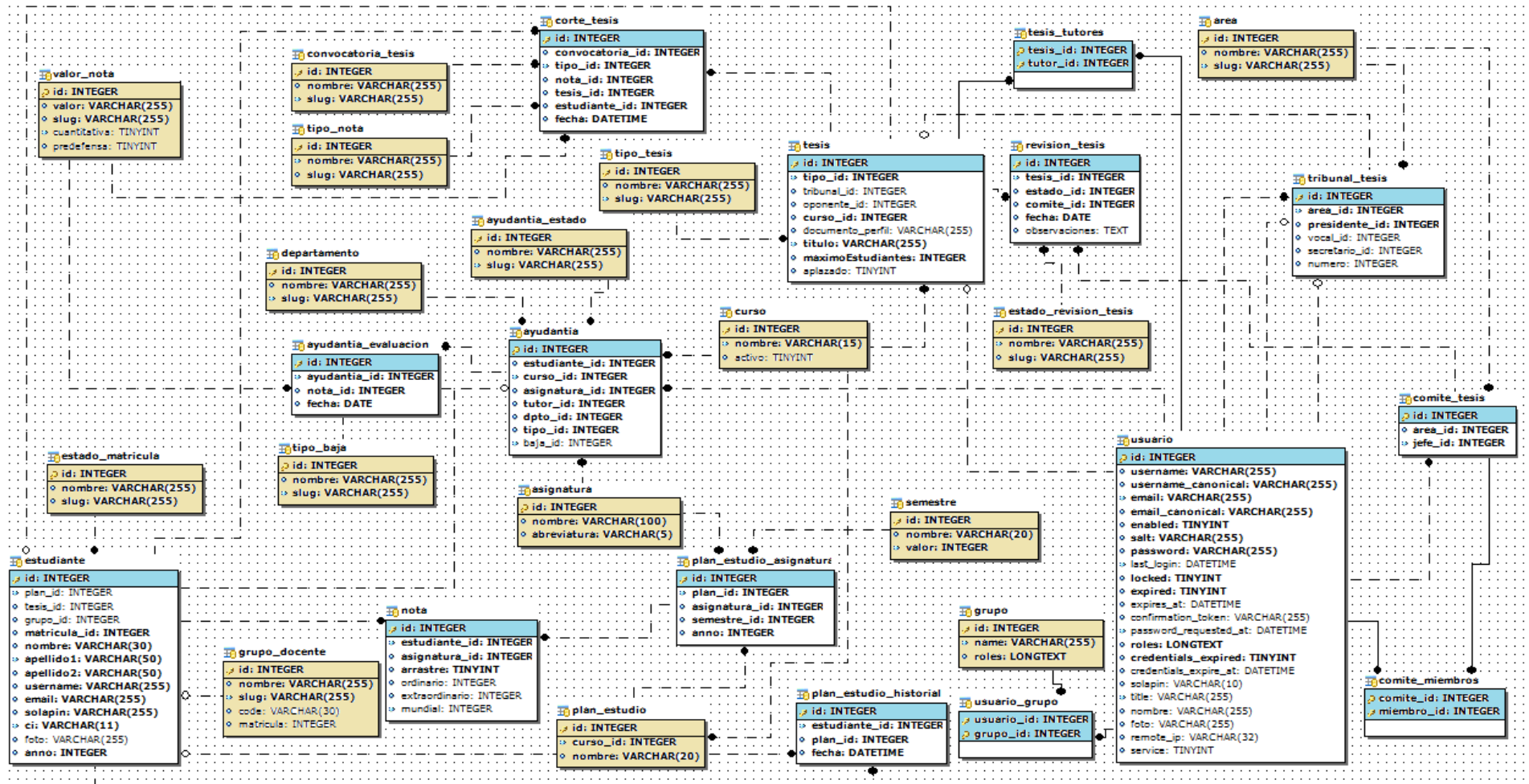


Figura 2 Modelo Físico de la BD.

2.9 Tareas de ingeniería

Las tareas de ingeniería son actividades que los programadores conocen que el sistema debe hacer. Deben ser estimables, y poder ser implementadas entre uno y tres días ideales. La mayoría de estas tareas se derivan directamente de las HU. Existen dos tipos de tareas de ingeniería las que provienen de las historias de usuario y las técnicas.

Las tareas de ingeniería técnicas son aquellas que no son resultado del análisis de ninguna historia de usuario pero deben ser realizadas para que el sistema funcione. Cada tarea de ingeniería será comprobada a través de los casos de prueba y no tienen por qué ser comprendidas por el cliente.

Tabla 26 Tarea de ingeniería 1.

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU_1,HU_2
Nombre Tarea: Implementar código para autenticar usuario.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: Beatriz Molina Legrá	
Descripción: El usuario accede al sistema a través de la web entra al sistema mediante un formulario que posee un campo para usuario y otro para la contraseña, si es usuario del sistema pues accede a la página de inicio con las funcionalidades que les corresponden.	

Tabla 27 Tarea de ingeniería 2

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: HU_3,HU_4
Nombre Tarea: Implementar código para registrar asignaturas.	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Programador Responsable: *Beatriz Molina Legrá*

Descripción: En el menú del usuario aparece la opción registrar asignaturas. Cuando se hace “click” en el vínculo aparece un listado con las asignaturas existentes y las opciones de eliminar, adicionar o editar una asignatura en específico. Teniéndose en cuenta todos los datos asociados a la misma, por semestres y por años.

Tabla 18 Tarea de ingeniería 3.

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: HU_5,HU_6
Nombre Tarea: Implementar código para registrar notas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: Beatriz Molina Legrá	
<p>Descripción: En el menú del usuario aparece la opción registrar notas. Cuando se hace “click” sobre el vínculo aparece una página donde se escoge el grupo y la asignatura y el plan de estudio (que carga por defecto el último, es decir el que se está aplicando). A partir de aquí se carga un formulario con el nombre de los estudiantes del grupo seleccionado y un “select” para escoger la nota de la asignatura por cada estudiante. Luego de terminado de poner las notas se da aceptar y aparece un listado con los nombres de los estudiantes y las notas. Se permite modificar las notas del estudiante haciendo “click” encima de la nota. En esta pantalla hay un botón aceptar que regresa al usuario a la página inicial de la opción.</p>	

Tabla 29 Tarea de ingeniería 4.

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: HU_7
Nombre Tarea: Implementar código para determinar repitencias y arrastres.	

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: Beatriz Molina Legrá	
Descripción: Cuando se accede al vínculo aparece un listado de los estudiantes suspensos, se muestra la información de en que convocatoria suspendió y que asignatura. Por cada estudiante aparece la opción de asignarle la repitencia o arrastre.	

Tabla 30 Tarea de ingeniería 5.

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: HU_8
Nombre Tarea: Implementar código para actualizar bajas	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: Beatriz Molina Legrá	
Descripción: Se muestra el listado de estudiantes con la opción “dar baja”, en el caso que se le de baja el sistema permite registrar estos estudiantes.	

Tabla 31 Tarea de ingeniería 5.

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: HU_9
Nombre Tarea: Implementar código para registrar estudiantes con observaciones (reingresos, tralados, reincorporados por licencias y repitencias).	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: Beatriz Molina Legrá	

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Descripción: Cuando se da “click” en esta opción aparece un listado con los estudiantes registrados. Aparece un botón para registrar un nuevo estudiante y cuando se pincha en este aparece un formulario con los datos que se registran de un estudiante. Luego se procede a llenar los datos correspondientes especificando además si es reingresos, traslados, reincorporados por licencia y repitencias; se da aceptar y se regresa a la página inicial de la opción, es decir el listado. Se le puede asignar el grupo también.

Tabla 32 Tarea de ingeniería 7.

Tarea de Ingeniería	
Número Tarea:7	Número Historia de Usuario:HU_10
Nombre Tarea: Implementar código para realizar convalidaciones de asignaturas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: <i>Beatriz Molina Legrá</i>	
Descripción: Cuando se va a esta opción aparece un listado con los estudiantes reingresos, traslados, reincorporados por licencia y repitencias. Previamente a estos estudiantes se les deben haber registrado las notas de las asignaturas cursadas en su respectivo plan de estudio. Luego se accede en el botón convalidar y automáticamente se le asignan las notas del plan de estudio viejo anterior al actual.	

Tabla 33 Tarea de ingeniería 8.

Tarea de Ingeniería	
Número Tarea:8	Número Historia de Usuario:HU_11
Nombre Tarea: Implementar código para asignar grupos.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: Beatriz Molina Legrá	

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Descripción: Aparece un listado con los grupos que hay en el sistema. Cuando se accede al grupo aparece un listado con los estudiantes asignados al grupo y un botón para adicionar un nuevo estudiante. Cuando se accede este botón aparece un formulario para buscar un estudiante, se escoge el mismo y se pincha en el botón aceptar para que asigne al grupo teniendo en cuenta que no se pase de la matrícula. En el listado, por cada estudiante aparece un vínculo que permite eliminar al estudiante del grupo.

Tabla 34 Tarea de ingeniería 9.

Tarea de Ingeniería	
Número Tarea:9	Número Historia de Usuario:HU_12
Nombre Tarea: Implementar código para insertar perfiles de tesis.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: Beatriz Molina Legrá	
Descripción: Aparece un listado con los perfiles insertados y un botón nuevo perfil, que permite agregar más perfiles. Cuando se pincha en este botón aparece un formulario que permite insertar los datos de un perfil de tesis.	

Tabla 35 Tarea de ingeniería 10.

Tarea de Ingeniería	
Número Tarea:10	Número Historia de Usuario: HU_13,HU_14
Nombre Tarea: Implementar código para conformar tribunales.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: <i>Beatriz Molina Legrá</i>	

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Descripción: Aparece un listado con los tribunales creados especificando si es del centro o de la facultad. Cuando se va a la opción detalle de un tribunal se pueden ver los miembros y la función de cada uno dentro del tribunal, y también se puede editar el mismo pinchando sobre la opción editar en la cual se pueden cambiar los datos y los miembros. Aparece además un botón adicionar que permite conformar un nuevo tribunal.

Tabla 36 Tarea de ingeniería 10.

Tarea de Ingeniería	
Número Tarea:10	Número Historia de Usuario:HU_13,HU_14
Nombre Tarea: Implementar código para asignar revisión de perfiles	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Beatriz Molina Legrá</i>	
Descripción: Aparece un listado con los tribunales y los perfiles asignados, con un botón editar por cada tribunal. Cuando se va al botón editar aparece un formulario para escoger el perfil, y al dar "click" en aceptar se asigna el tribunal, o se puede eliminar algún perfil previamente asignado.	

Tabla 37 Tarea de ingeniería 11.

Tarea de Ingeniería	
Número Tarea:11	Número Historia de Usuario:HU_15
Nombre Tarea: Implementar código para asignar estado del perfil.	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Beatriz Molina Legrá</i>	

CAPÍTULO 2 ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

Descripción: Aparece un listado de los perfiles de tesis con el estado asignado (sin estado, aprobado o rechazado). Por cada perfil aparece un botón editar que llama a un formulario que permite modificar el estado de un perfil, aceptar el cambio y luego regresar a la página del listado.

Tabla 38 Tarea de ingeniería 12.

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: HU_16
Nombre Tarea: Implementar código para las evaluaciones de tesis.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Programador Responsable: <i>Beatriz Molina Legrá</i>	
Descripción: Aparece un listado con las tesis (perfiles de tesis) y las evaluaciones en cada una de las convocatorias (taller 1, taller 2, taller 3, predefensa y defensa) y un botón editar por cada tesis. Al pinchar en el botón editar se llama a un formulario que permite modificar las evaluaciones.	

Tabla 29 Tarea de ingeniería 13.

Tarea de Ingeniería	
Número Tarea: 13	Número Historia de Usuario: HU_17,HU_18,HU_20
Nombre Tarea: Implementar código para alumnos ayudantes.	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable:: <i>Beatriz Molina Legrá</i>	

Descripción: Aparece un listado con los alumnos ayudantes y los botones adicionar y editar AA. Cuando se pincha en el adicionar aparece un formulario donde se escoge un nuevo estudiante para que sea AA. Luego se selecciona el dpto, la asignatura y tutor, se presiona aceptar y se regresa al listado. Cuando se accede al botón editar se puede dar baja a un AA voluntaria, por mundial o por sanción. También se puede actualizar el estado del AA (ratificados, nuevos ingresos o inactivos), se da aceptar y se regresa al listado.

Tabla 40 Tarea de ingeniería 14.

Tarea de Ingeniería	
Número Tarea: 14	Número Historia de Usuario: HU_19
Nombre Tarea: <i>Implementar código para registrar evaluaciones de alumnos ayudantes.</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Beatriz Molina Legrá</i>	
Descripción: Aparece un listado con los AA y las evaluaciones por meses y un botón agregar evaluación para cada AA. Cuando se pincha en el botón, se llama a un formulario que permite registrar la evaluación del AA en un mes determinado, se da aceptar y se regresa al listado inicial.	

2.10 Conclusiones del capítulo

En este capítulo se describieron las principales características del sistema a través de los artefactos definidos por la metodología. Se realizó el modelado y descripción de los procesos de negocio. Además se realizó la captura de los requisitos funcionales y no funcionales del software a través de la Lista de Reserva del Producto. Fueron creadas las historias de usuario a implementar y la descripción de cada una de ellas. A partir de todos los artefactos descritos se puede dar paso a la construcción del sistema, que incluye el diseño, la arquitectura y la implementación. Se construyó el sistema a través de la realización del diseño de clases y la obtención del modelo físico de la base de datos. Se describió la arquitectura de desarrollo MVC utilizada mediante el marco de trabajo así como también se describieron las tareas de ingenierías.

CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

3.1 Introducción

En el presente capítulo se lleva a cabo la implementación del sistema y una valoración de los principales artefactos de la fase de prueba. Se muestran los diagramas de despliegue y componentes, los estándares de codificación, se describen las pruebas unitarias y de aceptación que son las que utiliza la metodología SXP. Además, son realizadas las plantillas de las pruebas de aceptación para cada una de las HU implementadas, ya que son las seleccionadas para la validación de la propuesta de solución.

3.2 Estándares de codificación

La mayoría de los paquetes de código abierto y librerías utilizadas por Symfony2 siguen las mismas pautas, la principal ventaja de los estándares es que cada pieza de código se vea y se sienta familiar. [22]

3.2.1 Estructura

- Nunca utilizar las etiquetas cortas (<?);
- No terminar los archivos de clase con la etiqueta de cierre habitual?>;
- La sangría se hace con pasos de cuatro espacios (las tabulaciones no están permitidas);
- Utilizar el carácter de salto de línea (0x0A) para terminar las líneas;
- Añadir un solo espacio después de cada delimitador coma;
- No poner espacios después de un paréntesis de apertura ni antes de uno de cierre;
- Añadir un solo espacio alrededor de los operadores (==, &&, ...);
- Añadir un solo espacio antes del paréntesis de apertura de una palabra clave de control (if, else, for, while, ...);
- Añadir una línea en blanco antes de las declaraciones return, a menos que el valor devuelto solo sea dentro de un grupo de declaraciones (tal como una declaración if);
- No agregar espacios en blanco al final de las líneas;
- Usar llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga;
- Colocar las llaves en su propia línea en la declaración de clases, métodos y funciones;

- Separar las declaraciones condicionales (if, else, ...) y la llave de apertura con un solo espacio y sin ninguna línea en blanco;
- Declarar expresamente la visibilidad para clases, métodos y propiedades (el uso de var está prohibido);
- Usar minúsculas para escribir las constantes nativas de PHP: false, true y null. Lo mismo ocurre con array();
- Usar cadenas en mayúsculas para constantes con palabras separadas por subrayados;
- Definir una clase por archivo;
- Declarar las propiedades de clase antes que los métodos;
- Declarar los métodos públicos en primer lugar, a continuación, los protegidos y finalmente los privados. [22]

3.2.2 Convenciones de nomenclatura

- Usar mayúsculas intercaladas, no subrayados, para variables, funciones y nombres de métodos.
- Usar subrayados para opciones, argumentos, nombre de parámetros.
- Utilizar espacios de nombres para todas las clases.
- Sufijar interfaces con Interface.
- Utilizar caracteres alfanuméricos y subrayados para los nombres de archivo. [22]

3.3 Diagrama de despliegue

El diagrama de despliegue permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes de la aplicación. En este caso la aplicación se encuentra hospedada en un servidor Web (apache) y maneja los datos persistentes sobre un sistema gestor de base de datos (PostgreSQL) y se accede desde alguna PC Cliente. (Ver figura 3).

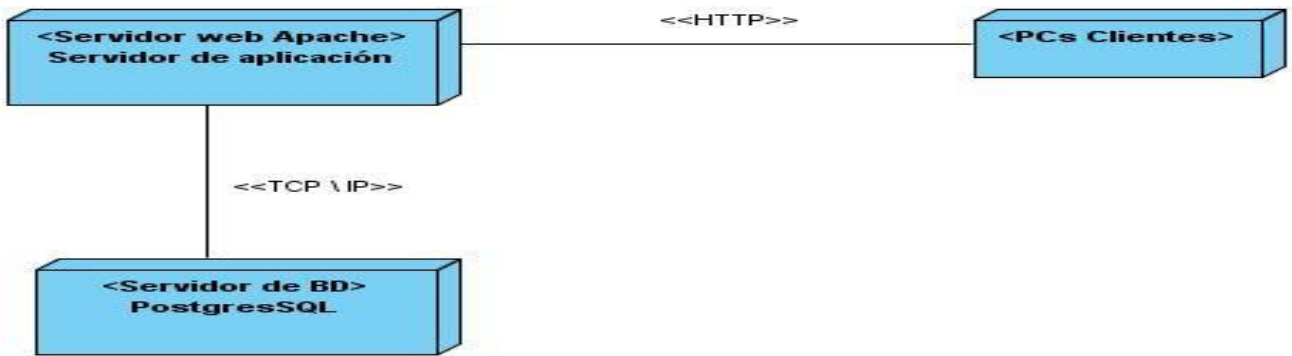


Figura 3 Diagrama de despliegue.

3.4 Diagrama de componentes

El diagrama de componentes muestra los componentes físicos y lógicos utilizados en el sistema así como sus interrelaciones. En la figura se muestran los paquetes por lo que está compuesta la aplicación. (Ver figura 3)

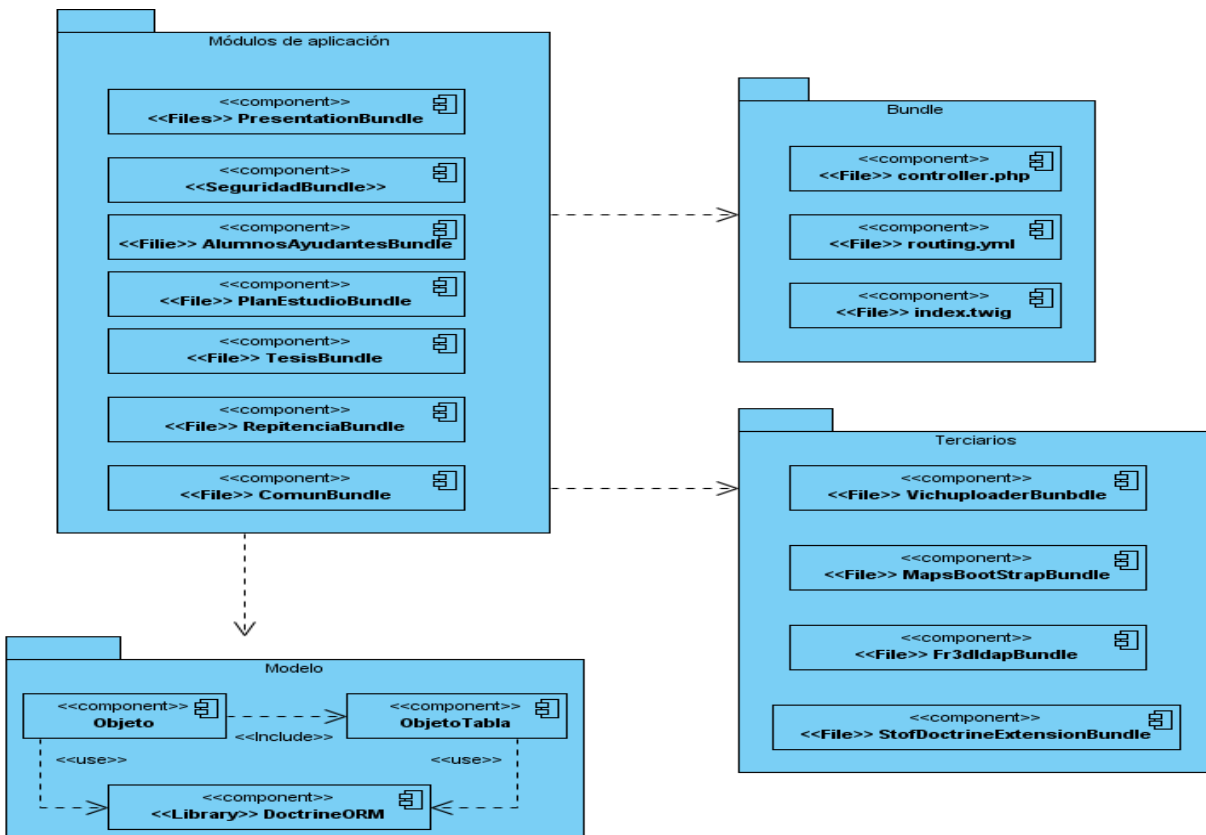


Figura 4 Diagrama de componentes.

3.5 Interfaz de usuario

En este epígrafe serán mostradas las pantallas principales con sus funcionalidades, de los módulos implementados para darle solución al problema planteado.

3.5.1 Interfaz del módulo Repitencias y arrastres

Grupo	Estudiante	Cohorte	Acciones
3101	Yunier Romero Soler	2007-2008	Ver Registrar nota
3201	Beatriz Fernandez Lopez	2007-2008	Ver Registrar nota
3201	Maria Lopez Soler	2008-2009	Ver Registrar nota
3202	Josefa Fernandez Suarez	2008-2009	Ver Registrar nota
3301	Yaniel Fabian Montenegro Martinez	2008-2009	Ver Registrar nota

Figura 5 Interfaz Módulo Repitencias y arrastres.

3.5.2 Interfaz del módulo Plan de estudio

Nombre y apellidos	Cohorte	Año	Grupo	Matrícula	Acciones
Beatriz Fernandez Lopez	2007-2008	2	3201	Repitencia	Convalidaciones
Yunier Romero Soler	2007-2008	1	3101	Arrastre	Convalidaciones

Mostrando de 1 a 2 de 2 filas

[Anterior](#)
[Siguiente](#)

Figura 6 Interfaz Módulo Plan de de estudio.

3.5.3 Interfaz del módulo Tesis

© Universidad de las Ciencias Informáticas 2012 - Gestacad

3.5.4 Interfaz del módulo Alumno ayudantes

Figura 7 Interfaz Módulo Tesis

Figura 8 Interfaz Módulo Alumnos Ayudantes.

3.6 Pruebas

En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos definidos en el análisis y diseño. La prueba no puede asegurar la ausencia de defectos, sólo puede demostrar que existen errores. [23]

En la metodología SXP las pruebas de funcionalidad se realizan a través de los casos de prueba. La metodología divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores y que se encargan de verificar el código de forma automática y las pruebas de aceptación que son las destinadas a comprobar que la funcionalidad implementada sea la esperada por el cliente. Las pruebas funcionales se escriben para cada historia de usuario que deba validarse.

3.6.1 Plan de iteraciones de pruebas.

Se llevaron a cabo pruebas al Sistema de Gestión Académica de la facultad 3, para verificar que las funcionalidades implementadas se correspondan con las historias de usuario diseñadas, donde los resultados obtenidos fueron satisfactorios. Para estos se realizaron tres iteraciones entre el desarrollador y el cliente.

De las catorce tareas de ingeniería a implementar en la primera iteración se realizaron satisfactoriamente nueve, en la segunda doce y en la última trece. La gráfica que se presenta a continuación muestra los resultados obtenidos. (Ver figura 9)

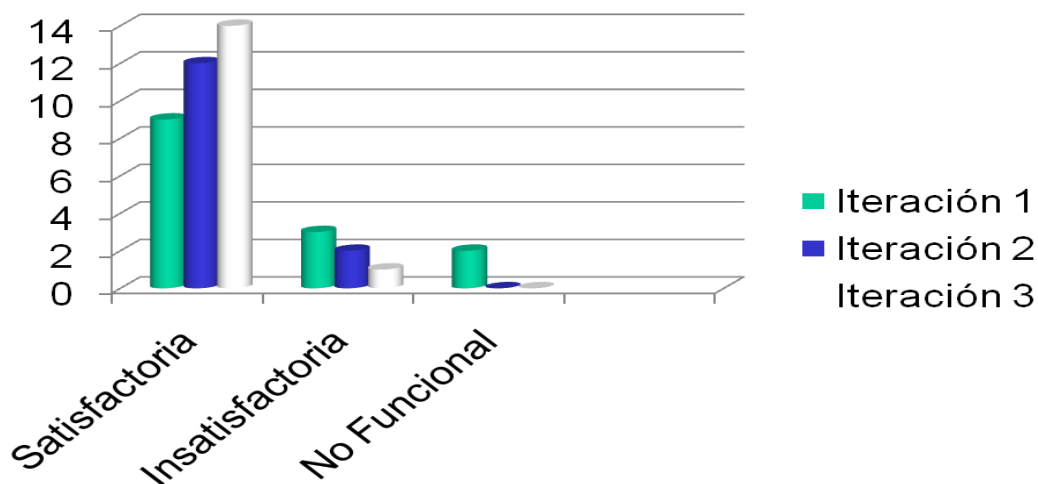


Figura 9 Resultado de pruebas.

3.6.2 Pruebas de aceptación

El objetivo de estas pruebas es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema determinar su aceptación, desde el punto de vista de su funcionalidad y rendimiento. [24]

Las pruebas de aceptación son escritas conjuntamente con el cliente y el equipo de desarrollo, aunque la aprobación final corresponde al usuario. Dicho plan está diseñado para asegurar que se satisfacen todos los requisitos funcionales especificados por el usuario, teniendo en cuenta también los requisitos no funcionales relacionados con el rendimiento, seguridad de acceso al sistema, a los datos, procesos y los distintos recursos del sistema. (Ver anexos1)

3.7 Plantillas de casos de prueba de aceptación

Tabla 41 Caso de Prueba de Aceptación HU1_P1.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU1_P1	Nombre Historia de Usuario: Probar la funcionalidad para autenticar usuario.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que el usuario autenticado acceda a las funcionalidades que le corresponde.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada / Pasos de ejecución: Se inserta usuario y contraseña.	
Resultado Esperado: Se autentica correctamente.	
Evaluación de la Prueba: Satisfactoria	

Tabla 42 Caso de Prueba de Aceptación HU2_P2.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU3,HU4_P2	Nombre Historia de Usuario: Probar la funcionalidad registrar asignaturas.

CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Nombre de la persona que realiza la prueba: Yalice Gámez Batista
Descripción de la Prueba: Probar que se muestren las asignaturas con los datos asociados a la misma, por semestre y año.
Condiciones de Ejecución: Se adicionan asignaturas y sus respectivas abreviaturas, se muestran y se da la opción de modificar.
Entrada / Pasos de ejecución: En el menú del usuario aparece la opción registrar asignaturas. Aparece un listado con las asignaturas existentes y las opciones de eliminar, adicionar o editar una asignatura en específico por semestres y por años.
Resultado Esperado: Se muestran las asignaturas cursadas por cada estudiante por semestre y año.
Evaluación de la Prueba: Satisfactoria

Tabla 43 Caso de Prueba de Aceptación HU3_P3.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU5_P3	Nombre Historia de Usuario: Probar la funcionalidad registrar notas.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que se registren las notas correctamente.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Seleccionar grupo y asignatura del plan de estudio. Mostrar nombres de los estudiantes del grupo seleccionado, y seleccionar la nota a insertar.	
Resultado Esperado: Mostrar los estudiantes con las notas.	
Evaluación de la Prueba: Satisfactoria	

Tabla 44 Caso de Prueba de Aceptación HU3_P4.

Caso de Prueba de Aceptación	
Código Caso de Prueba:	Nombre Historia de Usuario: Probar la funcionalidad

CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

HU6_P4	modificar notas.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que se modifiquen las notas correctamente.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Acceder a la nota y modificar	
Resultado Esperado: Modificar las notas.	
Evaluación de la Prueba: Satisfactoria	

Tabla 45 Caso de Prueba de Aceptación HU3_P5.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU7_P5	Nombre Historia de Usuario: Probar la funcionalidad para determinar repitencias y arrastres.
Nombre de la persona que realiza la prueba: Yalice Gámes Batista	
Descripción de la Prueba: Probar que después de obtener los estudiantes desaprobados se le pueda asignar el arrastre o la repitencia.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Filtrar por notas para obtener los estudiantes suspensos. Asignar repitencias o arrastre.	
Resultado Esperado: Asignar repitencias o arrastres.	
Evaluación de la Prueba: Satisfactoria	

Tabla 46 Caso de Prueba de Aceptación HU4_P6.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU8_P6	Nombre Historia de Usuario: Probar la funcionalidad actualizar bajas
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que se pueda actualizar cuando un estudiante cause baja.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	

Entrada / Pasos de ejecución:
Se busca al estudiante y se va a la opción “dar baja” y se elimina al estudiante.
Resultado Esperado: Actualizar bajas.
Evaluación de la Prueba: Satisfactoria

Tabla 47 Caso de Prueba de Aceptación HU5_P7.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU9_P7	Nombre Historia de Usuario: Probar la funcionalidad registrar estudiantes con observaciones.
Nombre de la persona que realiza la prueba: Yalice Gámes Batista	
Descripción de la Prueba: Probar que sean identificados los estudiantes que son reingresos, traslados, reincorporados por licencias y repitencias.	
Condiciones de Ejecución: Los estudiantes deben estar previamente registrados. El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Aparece un listado con los estudiantes registrados y un botón para registrar un nuevo estudiante. Se accede a un formulario con los datos que se registran de un estudiante se llenan los datos correspondiente especificando además si es reingresos, traslados, reincorporados por licencia y repitencias. Se le puede asignar el grupo también.	
Resultado Esperado: Registrar estudiantes con observaciones.	
Evaluación de la Prueba: Satisfactoria	

Tabla 48 Caso de Prueba de Aceptación HU5_P8.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU10_P8	Nombre Historia de Usuario: Probar la funcionalidad para realizar convalidaciones de asignaturas.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	

CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Descripción de la Prueba: Probar que se realicen las convalidaciones de las asignaturas, teniendo en cuenta el plan de estudio actual y las asignaturas cursadas.
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad. Se deben haber insertado previamente las notas de su respectivo plan de estudio
Entrada / Pasos de ejecución: Aparece un listado con los estudiantes reingresos, traslados, reincorporados por licencia y repitencias. Se accede en el botón convalidar y se le asignan las notas del plan de estudio viejo anterior al actual.
Resultado Esperado: Que se realicen las convalidaciones de las asignaturas.
Evaluación de la Prueba: Satisfactoria

Tabla 49 Caso de Prueba de Aceptación HU5_P9.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU11_P9	Nombre Historia de Usuario: Probar la funcionalidad para asignar grupos.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que a los estudiantes con algún tipo de observación (ingresos, reincorporados por licencias, repitencias, traslados y arrastres) se les pueda asignar un grupo.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Aparece un listado con los grupos que hay en el sistema. Aparece un listado con los estudiantes asignados al grupo y un botón para adicionar un nuevo estudiante. Se busca un estudiante, se escoge el mismo para asignarlo al grupo. Aparece un vínculo que permite eliminar al estudiante del grupo.	
Resultado Esperado: Asignar grupos a estudiantes con observaciones.	
Evaluación de la Prueba: Satisfactoria	

CAPÍTULO 3 IMPLEMENTACIÓN Y VALIDACIÓN DEL SISTEMA

Evaluación de la Prueba: Satisfactoria
Resultado Esperado: Listado de alumnos ayudantes y sus respectivos datos.
Evaluación de la Prueba: Satisfactoria

3.8 Conclusiones del capítulo

En el presente capítulo se ha realizado la implementación del sistema y la descripción del el flujo de pruebas propuesto por SXP de las funcionalidades descritas en las HU. Los casos de prueba son el mecanismo usado para asegurar que el sistema cumple con los requerimientos funcionales, asegurando así la calidad del software. De manera general las pruebas fueron satisfactorias, contribuyendo a obtener un producto de mayor calidad.

CONCLUSIONES GENERALES

- Después de haber realizado un estudio bibliográfico de la situación actual y las tendencias de los sistemas de gestión académica existentes, se llegó a la conclusión de desarrollar un sistema que sea capaz de mejorar los procesos de planificación, organización y control de la actividad académica en la facultad 3.
- El sistema fue representado por la metodología SXP. La modelación del negocio, la captura de requerimientos funcionales y no funcionales, así como los artefactos generados durante la etapa de análisis y diseño, sentaron las bases para la posterior implementación.
- A partir del diseño del esquema de base de datos y la ejecución de las historias de usuarios en conjunto con las tareas de ingeniería, fue construido el sistema, cumpliéndose el principal objetivo del presente trabajo.
- Partiendo del diseño y la aplicación de los casos de prueba, se pudo comprobar que las funcionalidades descritas, satisfacen las necesidades de la facultad en cuanto al proceso de gestión docente garantizando así resultados satisfactorios.

RECOMENDACIONES

- Continuar el estudio con el objetivo de añadir nuevas funcionalidades al sistema.
- Realizar la integración con la Plataforma Informativa SO3.
- Realizar los ajustes necesarios de manera que pueda ser aplicado en el resto de las facultades de la universidad.

REFERENCIAS BIBLIOGRÁFICAS

- [1]. In Artículos, Proceso docente educativo [En línea] 200
http://bvs.cld.cu/revistas/spu/vol32_3_06/spu14306.htm
- [2]. in Artículos, Destacados, Gestión, Management, Mejora Tu Gestion, Sistemas de Gestión [En línea] 2009. [Citado el: 31 marzo] <http://mejoratugestion.com/mejora-tu-gestion/que-es-un-sistema-de-gestion/>
- [3]. BELAMIRO CARO VARGAS. Guía pedagógica del seminario: Gestión Académica, Disponible en: <http://api.ning.com/files/zniINCG9rk6jPjqia8NjAnxQokk0swCI0TF7HP2U9q5cF4rjEPpTnr3V9ZEKZNKiiTAqhNzVttGaDbHBYoReXyOi0GyWQPT/GUIASEMINARIOgestinacadmica.pdf>
- [4]. ZAMBONI, GORGONE. Propuestas innovadoras en la gestión académica.
- [5]. Disponible en: < <https://www.siga.utfsm.cl/jmc/pag/index01.htm> >
- [6]. NOEL MIÑO, E. P. Análisis y diseño del Módulo Estudiante para el sistema de gestión Académica-Akados. 2007, Disponible en: http://bibliodoc.uci.cu/TD/TD_0325_07.pdf.
- [7]. MARRERO, D. P. C. J. L. A. O. Y. S. R. H. E. Manual de Usuario del SIGENU Sistema de Gestión de la Nueva Universidad. Abril del 2007.
- [8]. MIREYDIS, YURISLEIS. Análisis y Diseño de los Procesos de Gestión de Personal para Akados v2.0. 2009, Disponible en: http://bibliodoc.uci.cu/TD/TD_1934_09.pdf
- [9]. Elaborado por Facultad 3. Derechos Reservados. Propuesta de Informatización Facultad 3. Habana: Política Editorial Universidad de las Ciencias Informáticas., 2011.
- [10]. in Metodologías Ágiles en el Desarrollo de Software. Disponible
En :<<http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- [11]. in Índice del manual de lenguajes del Web. Disponible
En: <<http://www.desarrolloweb.com/articulos/709.php>
- [12]. Ellis, Rick y Burdick, Paul. 2009. MANUAL DE CodeIgniter. [En línea] 2009.
http://lax.franhp.net/CodeIgniter_Spanish_UserGuide.pdf.
- [13]. POTENCIER, F. Z. Y. F. Symfony, la guía definitiva. 2008, Disponible en:
<http://www.librosweb.es/symfony/>.

- [14]. González Castellanos, Ma. Arnenis y Rojas Pabón, Wilson. 2005. TRABAJO DE GRADO: Comparación entre sistemas de gestión de bases de datos (SGBD) bajo licenciamiento libre y comercial (pdf). 2005.
- [15]. Andrés Marqués, M. M. ((mayo - 2005)). Componentes de un sistema de gestión de bases de datos.
- [16]. Arias, Yuniel Eliades Proenza. *Diseño Avanzado de Aplicaciones Web. EXT, Zen Framework y Doctrine*. nueve, Ciudad de La Habana, Cuba. : s.n., 9 de marzo de 2009.
- [17]. Foundation, Apache Software. The Apache Software Foundation. [En línea] 2009. [Citado el: 24 de febrero de 2010.] <http://www.apache.org>.
- [18]. NetBeans. (2010). Recuperado el 8 de enero de 2010, de http://netbeans.org/index_es.html
- [19]. Paradigma visual para UML. Free Download Manager. [Online] marzo 5, 2007. [Cited: febrero 18, 2010.]http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
- [20]. Tutorial de UML. Disponible en: <<http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>> [Fecha de consulta 10 febrero 2007].
- [21]. In Business Process Modeling Notation (BPMN).2008, Disponible en:
<http://www.slideshare.net/gugarte/bpmn-estandar-para-modelamiento-de-procesos-presentation>
- [22]. Disponible en: <<http://sf2-es.net16.net/contributing/code/standards.html>>
- [23].El único instrumento adecuado para determinar el status de la calidad. s.l. : Asociación de Técnicos, 2008.
- [24].La prueba de aceptación es la prueba más importante para los productos software. [En línea] 2010. <http://pruebasdesoftware.com/pruebadeaceptacion.htm>.

BIBLIOGRAFÍA CONSULTADA

ANEXOS

1 Anexo 1 Pruebas de aceptación.

2 Tabla 50 Caso de Prueba de Aceptación HU6_P10.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU12_P10	Nombre Historia de Usuario: Probar la funcionalidad insertar perfiles de tesis.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que se puedan insertar los perfiles de tesis.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Se va a la opción insertar nuevo perfil. Se registran los perfiles de tesis insertados con sus respectivos datos.	
Resultado Esperado: Registrar perfiles de tesis.	
Evaluación de la Prueba: Satisfactoria	

3
4 Tabla 51 Caso de Prueba de Aceptación HU7, 8_P12.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU13,14_P12	Nombre Historia de Usuario: Probar la funcionalidad para conformar tribunales de tesis.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que se puedan conformar los tribunales de tesis.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Aparece un listado con los tribunales creados especificando si es del centro o de la facultad. Se muestran los miembros y la función de cada uno dentro del tribunal y también se pueden cambiar los datos y los miembros. Se permite conformar un nuevo tribunal.	

Resultado Esperado: Conformar tribunales de tesis.

Evaluación de la Prueba: Satisfactoria

1
2

Tabla 52 Caso de Prueba de Aceptación HU7, 8_P13.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU13,14_P13	Nombre Historia de Usuario: Probar la funcionalidad para asignar revisión de perfiles de tesis.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que a cada perfil de tesis se le pueda asignar una revisión de cada uno de los comités revisores.	
Condiciones de Ejecución: El perfil de tesis debe estar previamente registrado	
Entrada / Pasos de ejecución: Aparece un listado con los tribunales y los perfiles asignados. Un botón editar por cada tribunal, cuando se va al botón editar aparece formulario para escoger el perfil y se le asigna al tribunal. Se puede eliminar algún perfil previamente asignado.	
Resultado Esperado: Revisar perfiles de tesis.	
Evaluación de la Prueba: Satisfactoria	

3
4

Tabla 53 Caso de Prueba de Aceptación HU9_P14.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU15_P14	Nombre Historia de Usuario: Probar la funcionalidad para asignar estado al perfil.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que después de revisado el perfil de tesis se le pueda asignar un estado.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Aparece un listado de los perfiles de tesis con el estado asignado (sin estado, aprobado o rechazado).	

Modificar el estado de un perfil, aceptar el cambio.

Resultado Esperado:

1
2

Tabla 54 Caso de Prueba de Aceptación HU10_P15.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU16_P15	Nombre Historia de Usuario: Probar la funcionalidad para registrar evaluaciones de tesis.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que se puedan registrar correctamente las evaluaciones de los talleres de tesis.	
Condiciones de Ejecución: El usuario debe tener permiso para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: A los perfiles insertados le aparece la opción de registrar sus evaluaciones. En cada taller se registra su respectiva evaluación.	
Resultado Esperado: Evaluaciones de los diferentes talleres de tesis.	
Evaluación de la Prueba: Satisfactoria	

3
4

Tabla 55 Caso de Prueba de Aceptación HU11, 12,13_P16.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU17,18,20_P14	Nombre Historia de Usuario: Probar la funcionalidad para identificar alumnos ayudantes.
Nombre de la persona que realiza la prueba: Yalice Gámez Batista	
Descripción de la Prueba: Probar que se registren los alumnos ayudantes.	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución: Mostrar un listado con los alumnos ayudantes. Adicionar nuevo AA Editar AA. Se escoge la asignatura de la cual será AA, el tutor y el dpto. Se puede dar baja a un AA voluntaria, por mundial o por sanción. Se puede actualizar el estado del AA (ratificados, nuevos ingresos o inactivos).	

1

Tabla 56 Caso de Prueba de Aceptación HU12_P17.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU19_P15	Nombre Historia de Usuario: Probar la funcionalidad para registrar evaluaciones de alumnos ayudantes.
Nombre de la persona que realiza la prueba: Yalice Gámes Batista	
Descripción de la Prueba:	
Condiciones de Ejecución: El usuario debe tener permisos para ejecutar la funcionalidad.	
Entrada / Pasos de ejecución: Se muestra un listado con los AA y las evaluaciones por meses. Agregar evaluación para cada AA. Registrar al AA una evaluación en un mes determinado.	
Resultado Esperado: Registrar evaluaciones de AA.	
Evaluación de la Prueba: Satisfactoria	

2

3 **Anexos 2 Acta de Aceptación.**

4