

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



**Título: Sistema para la gestión de la información de
postgrado en la facultad 3.**

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

Autor: Roilán González Guerra.

Tutora: Ing. Virgen Damarys Quevedo.

La Habana. Junio, 2012.



“Al venir a la tierra todo hombre tiene el derecho a que se le eduque y después en pago contribuir a la educación de los demás”.

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.

TUTORA

Ing. Virgen Damarys Quevedo.

AUTOR

Roilán González Guerra.

Para que así conste, firmo la presente a los ____ días del mes de _____ del año 2012.

AGRADECIMIENTOS

Primero que todo quiero agradecer a la Revolución, a Fidel, a Raúl y a la UCL por permitirme durante estos largos años prepararme como profesional y finalmente graduarme como ingeniera en Ciencias Informáticas.

A mis padres por ser siempre la luz que ha guiado mis ojos, y nunca se rindieron apoyándome y estimulándome en cada paso y en cada tropiezo.

A mi hermanita que es lo más bello y tierno que me ha pasado en esta vida.

A mis amigos en la UCL estos 5 años que siempre me han ayudado incondicionalmente Leandro, Mailin, Jorge, Leiser, Ariel, Orlienis, Alkaid, Magui, Hany, Bety, Arletis, Nailet, Darianis, Yirenia a todos ustedes mi agradecimiento y cariño eterno.

A mis compañeros de contienda en la universidad a Jordan, Raidel, Yenekiel y Lian que siempre les estaré agradecido por los buenos momentos que compartimos.

A mis hermanos del alma Modesto, Chávez, Peña, Zenén que aunque no estaban en la universidad siempre me apoyaron y confiaron en mí.

A mi tutora... Virgen que fue quien estuvo conmigo desde el principio y supo guiarme en momentos difíciles y a Julierskys que siempre me ayudó y me apoyó en todo lo que necesité.

DEDICATORIA

Dedico este trabajo a mis padres Antonia Caridad Guerra Ventos y Juan Manuel González Martínez por ser un gran ejemplo de respeto, responsabilidad y amor, por educarme sin descanso, por su preocupación constante, por darme todo a cambio de nada, gracias por existir.

A mi hermana Lissett que es lo más bello que tengo en la vida y por todo el apoyo que me ha brindado en estos 5 años en la Universidad y por el amor y cariño que siente por mí.

A toda mi familia en general por su apoyo incondicional

RESUMEN

El presente trabajo de diploma se centra en la necesidad de desarrollar una aplicación web que permita la gestión eficiente de la información de los postgrados en la facultad 3. Esta aplicación se va a integrar al sistema SO3, encargado de gestionar la información de los procesos fundamentales de la facultad 3.

Para dar cumplimiento al objetivo de la investigación se realizó una fundamentación teórica donde se analizaron los conceptos fundamentales relacionados con el tema para profundizar los conocimientos a la hora de informatizar los procesos a gestionar. Además, se realizó un estudio de las tecnologías, herramientas y lenguajes utilizados en el desarrollo de aplicaciones web, definiéndose el uso de lenguaje PHP, el framework Symfony y como gestor de base de datos PostgreSQL. Se efectuó un análisis de aplicaciones similares existentes en el ámbito nacional e internacional. El proceso de desarrollo de software estuvo guiado por la metodología SXP, indicada para proyectos donde los requisitos cambian con gran facilidad y cuentan con un pequeño equipo de desarrollo.

Se logró un sistema web que cumple con los requisitos establecidos por el cliente, haciendo eficiente la gestión de la información de postgrado en la facultad 3 y permitiéndole al personal autorizado actualizar la información referente a los cursos en dicha facultad.

Tabla de Contenido

TABLA DE CONTENIDO

Introducción.....	1
Capítulo 1. Fundamentación teórica.....	4
1.1 Introducción.....	4
1.2 Postgrado.....	4
1.2.1 Necesidad.....	4
1.2.2 Objetivos.....	4
1.2.3 Características.....	5
1.2.4 Formación Postgraduada.....	5
1.3 Aplicación Web Similares.....	5
1.4 Lenguaje de Modelado.....	7
1.4.1 UML.....	7
1.4.2 BPMN.....	8
1.5 Lenguajes de Programación.....	9
1.5.1 JavaScript.....	9
1.5.2 PHP.....	10
1.6 Metodología de desarrollo.....	11
1.6.1 SXP.....	11
1.7 Herramientas.....	12
1.7.1 Herramientas de Modelado.....	12
1.7.1.1 Visual Paradigm.....	12
1.7.2 Herramientas de Bases de Datos.....	13

Tabla de Contenido

1.7.2.1	<i>PostgresSQL</i>	13
1.7.2.2	<i>pgAdmin</i>	14
1.7.3	<i>Herramientas de Desarrollo</i>	15
1.7.3.1	<i>Netbeans</i>	15
1.7.3.2	<i>Symfony</i>	16
1.7.3.3	<i>ExtJS</i>	18
1.8	<i>Conclusiones</i>	19
Capítulo 2. Análisis y diseño		20
2.1	<i>Introducción</i>	20
2.2	<i>Descripción del negocio actual</i>	20
2.3	<i>Modelo de negocio</i>	20
2.4	<i>Propuesta de solución</i>	21
2.5	<i>Definición de los Requisitos</i>	21
2.6	<i>Requisitos Funcionales</i>	22
2.7	<i>Requisitos no Funcionales</i>	23
2.8	<i>Personal relacionado con el sistema</i>	24
2.9	<i>Lista de Reserva del Producto</i>	25
2.10	<i>Historias de Usuarios</i>	28
2.11	<i>Arquitectura cliente/servidor</i>	47
2.12	<i>Patrones Arquitectónicos</i>	48
2.13	<i>Patrones de Diseño</i>	49
2.14	<i>Tarjetas CRC</i>	51
2.15	<i>Diagrama de Despliegue</i>	58

Tabla de Contenido

2.16	<i>Modelo de Datos</i>	58
2.17	<i>Conclusiones Parciales</i>	59
Capítulo 3. Implementación y prueba		60
3.1	<i>Introducción</i>	60
3.2	<i>Estándar de Programación</i>	60
3.3	<i>Tareas de ingeniería</i>	60
3.4	<i>Pruebas</i>	66
3.4.1	<i>Pruebas Unitarias</i>	67
3.4.2	<i>Pruebas de Aceptación</i>	67
3.5	<i>Caso de Prueba de Aceptación</i>	67
3.6	<i>Realización de las pruebas de aceptación</i>	67
3.7	<i>Conclusiones Parciales</i>	68
<i>Conclusiones Generales</i>		69
<i>Recomendaciones</i>		70
<i>Bibliografía</i>		71

INTRODUCCIÓN

Con el paso del tiempo la información ha sido almacenada, según la necesidad y el desarrollo de la sociedad, pero actualmente, debido a la gran cantidad de información que se acumula, se hace casi imposible la utilización de métodos tradicionales. El avance de la tecnología ha proporcionado el desarrollo de herramientas de software que permiten realizar este trabajo. La mayoría de las empresas e instituciones académicas están haciendo uso de estos avances tecnológicos para difundir información de forma rápida, fácil y a bajo costo. Esto se debe a que Internet proporciona los mecanismos que apoyan el proceso de difusión. La creciente popularidad de la Internet, gracias al surgimiento de la World Wide Web (WWW o Web), generó la necesidad de desarrollar aplicaciones Web. Una de las principales ventajas de este tipo de aplicaciones es permitir que la información difundida a través de ellas se actualice constantemente y pueda ser utilizada por gran cantidad de usuarios.

Cuba, como país en desarrollo, también está informatizando los sistemas digitales de sus empresas, por ello nuestro comandante en jefe Fidel Castro creó la Universidad de Ciencias Informáticas (UCI). La UCI es una institución que vincula los nuevos métodos de enseñanza y aprendizaje, donde además de realizar las labores de institución docente también se desempeña como centro de desarrollo de software para entidades tanto cubanas como extranjeras con el objetivo de contribuir con el desarrollo económico del país. La universidad cuenta con 7 facultades docentes y 3 regionales distribuidas por todo el país, y con una gran cantidad de estudiantes y profesores, por lo que se hace difícil gestionar toda la información competente de los mismos. Cada facultad de la universidad cuenta con personal encargado de llevar un control de la superación de los claustros de profesores. Por tanto, en este documento se abordará el tema de la superación profesional del claustro de profesores de la facultad 3 a fin de elevar su competitividad, desempeño y capacidad innovadora en la esfera de la investigación, la docencia y la producción de software en la UCI.

Hasta el momento, se hace bastante engorroso el trabajo con la información de postgrado en la facultad 3, puesto que el sistema que existe actualmente a nivel UCI para gestionar dicha información no satisface todos los requerimientos que necesita la facultad, como son: el conocimiento, el registro y control cuantitativa y cualitativamente por área y por profesores sobre su información de postgrado. Además, la gestión de la información de los cursos postgrados tiene un ciclo complicado porque cada jefe de área tiene que manipular la información de todos los profesores referente al departamento: cursos impartidos o

recibidos por algún profesor, entre otras tareas; luego esta información tiene que llegar al vicedecano de Investigación y Postgrado de la facultad el cual es el encargado de notificar y archivar la misma. Esta tarea se realiza actualmente de forma manual, provocando un gran consumo de tiempo del personal encargado de efectuar dicha tarea. Como consecuencia de lo anteriormente expuesto, surge el siguiente **problema científico**: ¿Cómo mejorar el registro y control de los datos en el proceso de gestión de postgrado en la facultad 3?

De ahí que el **objeto de estudio** de este trabajo sea: la gestión de la información de postgrado, tomando como **campo de acción**: los procesos que genera la información de postgrado en la facultad 3.

Para resolver el problema planteado se traza como **objetivo general**: Crear un sistema para registrar la información de los postgrados en la facultad 3, de forma tal que facilite el conocimiento, registro y control por áreas y por profesores de dicha información.

La **idea a defender** que se plantea es la siguiente: si se implementa un sistema que permita el manejo de los datos cualitativos de la información de postgrado por profesores, se lograría un registro y control de dichos datos.

Los **objetivos específicos** de este trabajo son:

1. Elaborar el marco teórico de la investigación.
2. Realizar el modelo del negocio.
3. Realizar el levantamiento de requisitos.
4. Implementar el sistema.
5. Validar las mejoras obtenidas con la implantación del sistema.

La investigación quedará estructurada en 3 capítulos, que agruparán los contenidos de la siguiente manera:

Capítulo 1. Fundamentación Teórica: se realiza una fundamentación teórica donde se analizaron los conceptos fundamentales relacionados con el tema para profundizar los conocimientos a la hora de informatizar los procesos a gestionar. Además, se realizó un estudio de las tecnologías, herramientas y lenguajes utilizados en el desarrollo de aplicaciones web y se lleva a cabo un estudio de las aplicaciones existentes en Cuba y en el mundo.

Capítulo 2. Análisis y Diseño de la solución propuesta: se realiza una descripción del negocio actual, teniendo en cuenta los requisitos funcionales y no funcionales y las Historias de Usuarios relacionadas con los mismos, los patrones utilizados a la hora de realizar el sistema, tanto arquitectónicos, como de diseño, y se presenta el modelo de análisis, diseño y el modelo de base de datos. Además, se realiza una descripción de las tareas de ingeniería a través de los diagramas necesarios según plantea la metodología empleada.

Capítulo 3. Implementación y Prueba de la solución propuesta: se describirá el estándar de programación utilizado para la implementación del sistema y se realizan las pruebas al sistema según la metodología definida, ya sean pruebas unitarias o de aceptación y, en el caso de ser pruebas de aceptación, se elaborando los casos de pruebas correspondientes.

Capítulo 1: Fundamentación Teórica

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se realiza un análisis de los conceptos, objetivos y características relacionadas con el procedimiento a tener en cuenta en el momento de informatizar los procesos a gestionar. Además, se hace un estado del arte de la tecnología y las herramientas utilizadas para el desarrollo de la aplicación web, así como las ventajas y desventajas de estas.

1.2 Postgrado

Es un proceso docente–educativo, su fin es formar a aquellos que en él participan. Este proceso puede incluir aspectos de la ciencia o tecnología, o ambos, de una cierta profesión. Se concibe y organiza a partir de los problemas presentes en la sociedad en una rama en cuestión y posee un objetivo a alcanzar, un contenido a desarrollar, un método o modo de actuación, una estructura organizativa o forma, los medios que son necesarios para su desenvolvimiento y el resultado de dicho proceso. En él, está presente lo académico, lo laboral y lo investigativo, sin embargo, la diferencia con el pregrado radica en que en el postgrado, por ser sus estudiantes ya profesionales, el componente laboral-investigativo desempeña un papel más significativo que en el nivel educativo anterior. (1)

1.2.1 Necesidad

La formación postgraduada de todos los universitarios es una consecuencia de los ritmos de desarrollo de la ciencia, la técnica, ya que necesitan mantenerse actualizados en su desempeño profesional. La solución de los problemas a que se enfrenta el profesional requiere del dominio de la lógica de la profesión, de la ciencia, y en especial, de la lógica de la investigación científica. La misma dinámica del desarrollo impone un proceso paulatino de especializaciones que van estrechando el campo de acción del profesional, a lo que se suma la tendencia al surgimiento de contenidos globalizadores, como la cibernética, la ecología, etc. (1)

1.2.2 Objetivos

Los cursos de postgrados tienen como objetivos, la superación profesional del individuo ya sea con intereses personales o por necesidad de la entidad en que se encuentre ajustándose a su permanencia o no en la entidad.

Capítulo 1: Fundamentación Teórica

Los objetivos a lograr en el postgrado son de tipo:

- Instructivo: se precisan las habilidades a formar y los conocimientos a asimilar.
- Desarrollador: se precisan las facultades (capacidades) a alcanzar.
- Educativo: muestran valores éticos y morales que expresan una plena conciencia de la solución de los problemas culturales, ambientales y sociales. (1)

1.2.3 Características

Las características deseables de todo curso de postgrado son:

- Que ofrezca una visión completa de las disciplinas que imparte.
- Que contenga casos prácticos y situaciones reales de trabajo.
- Que posibilite la realización de proyecto final donde se integre todo lo visto a lo largo del curso.
- Que se fomente el networking para potenciar la red de contactos profesionales.
- Que los alumnos con el que comparta el curso se ajusten todos a un perfil parecido.
- Que se use una metodología adaptada a tus objetivos. (2)

1.2.4 Formación Postgraduada

La estrategia de superación profesional postgraduado se elabora en cumplimiento de lo establecido por los Ministerios de Ciencia, Tecnología y Medio Ambiente; Trabajo y Seguridad Social; Educación Superior y otros, quienes regulan el desarrollo de estas actividades. Además, se tiene en cuenta el objeto social así como los objetivos y misión planificados estratégicamente por el Archivo Nacional de la República de Cuba, con el fin de contribuir a la formación y desarrollo constante de todos los trabajadores, sean profesionales universitarios, técnicos, obreros o cuadros. (3)

1.3 Aplicación Web Similares

Una importante cantidad de desarrolladores se encuentra automatizando procesos a través del uso de aplicaciones Web. Una aplicación Web es un software diseñado para automatizar procesos donde su

Capítulo 1: Fundamentación Teórica

resultado se entrega a través de Internet o una Intranet, siendo un sistema Web donde la entrada de datos del usuario y la navegación que realice, afectan al estado del sistema.

Son muchas las aplicaciones destinadas a manipular información con el propósito de la superación postgraduada. En el mundo, cada universidad cuenta con un área destinada a la superación profesional de su claustro de profesores, de ahí que existan grandes cantidades de aplicaciones o sitios informáticos destinados a este fin. En nuestro país, las universidades también cuentan con en este tipo de departamentos destinado a la superación profesional pero a menor escala, ya que no todas tienen sus sistemas digitalizados, aún en un gran número se realizan de forma manual. En la UCI, no existe una aplicación informática que se encuentre actualmente en funcionamiento y que satisfaga las necesidades actuales para realizar un análisis de los datos de los profesores de alguna determinada facultad o área de la misma que haya recibido o impartido algún curso de postgrado.

Ejemplos de sistemas para la gestión de postgrados en el mundo y en Cuba:



Figura 1.1 Sistemas para la gestión de postgrados Universidad Autónoma Barcelona

La Escuela de Postgrado coordina, gestiona e impulsa la oferta formativa de doctorado y de Formación Continua de la Universidad Autónoma Barcelona (UAB). Está concebida como un punto de intercambio entre estudiantes, profesores, investigadores y profesionales de reconocido prestigio, ubicada en el Campus de la UAB.

Los postgrados de la UAB están orientados a crear los profesionales que demandan los sectores económicos y sociales más innovadores donde los estudios de doctorado están destinados a la formación de profesores e investigadores de máximo nivel con una gran cantidad de Maestrías, Diplomas de Postgrado y Cursos al alcance de profesionales, es una herramienta fundamental para la actualización de conocimiento o bien para la superación profesional. (4)

Capítulo 1: Fundamentación Teórica



Este sistema se encuentra en desarrollo actualmente en la Universidad de Ciencias Informáticas (UCI), tiene como fin gestionar gran cantidad de actividades que se realizan en la universidad que actualmente no tienen un sistema informático que las controle, entre estas tareas esta la superación profesional del claustro de profesores de la UCI. Además de no estar en funcionamiento, este sistema no le solucionará los problemas existentes en la facultad debido a que trabajará con los datos de forma general desde el punto de vista de la universidad y no de forma cualitativa como lo necesita la facultad 3.

1.4 Lenguaje de Modelado

El lenguaje de modelado es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar parte de un diseño de software orientado a objetos. Se utiliza en combinación con una metodología de desarrollo para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo el equipo de desarrolladores.

1.4.1 UML



El Lenguaje Unificado de Modelado (UML) es un lenguaje visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. La estructura estática define los tipos de objetos importantes

Capítulo 1: Fundamentación Teórica

para un sistema y para su implementación, así como las relaciones entre los objetos. El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos.

El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos. UML también contiene construcciones organizativas para agrupar los modelos en paquetes, lo que permite a los equipos de software dividir grandes sistemas en piezas de trabajo, para entender y controlar las dependencias entre paquetes, y para gestionar las versiones de las unidades del modelo, en un entorno de desarrollo complejo. Contiene construcciones para representar decisiones de implementación y para elementos de tiempo de ejecución en componentes.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. (7)

Se utilizó UML a la hora de modelar los artefactos del sistema principalmente porque sus diseños se pueden implementar en cualquier lenguaje de programación que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos) esto es debido a que permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos) y tener actualizado la estructura del proyecto con una visión en el diseño de más alto nivel.

1.4.2 BPMN

BPMN (Business Process Modeling Notación, Notación de Modelado de Procesos de Negocio) es el estándar más reciente para modelado de procesos del negocio y servicios web. BPMN es una notación necesaria para expresar los procesos de negocio en un único diagrama de proceso de negocio (Business Process Diagram – BPD) BPMN permite hacer un mejor uso de la gestión de procesos del negocio (BPM), ya que normaliza el método de notación que sirve como ayuda en la automatización de los procesos. (6).

BPMN está dirigido a gerentes, directores, dueños de empresas, ingenieros de procesos, analistas de negocios, analistas de sistemas, administradores de proyectos, responsables de calidad y todo aquel que necesita definir, documentar y hacer más eficientes sus procesos de negocio con el estándar más avanzado y aceptado en el mundo.

Capítulo 1: Fundamentación Teórica

UML (El lenguaje de modelado unificado) toma un perfil orientado a objetos en el modelado de aplicaciones, mientras que BPMN toma un perfil orientado a procesos en el modelado de sistemas. Tiene un enfoque en procesos de negocio, UML se enfoca al diseño de software y por lo tanto ambas notaciones son totalmente compatibles entre sí.

Se utilizó BPMN para el modelado porque proveer una notación estándar que es fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos, con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil entender el flujo y el proceso del mismo.

1.5 Lenguajes de Programación

El lenguaje de programación, es un lenguaje diseñado para describir el conjunto de sintaxis y reglas semánticas que debe ejecutar un determinado equipo. Es una técnica estándar de comunicación para entregarle instrucciones a un ordenador. Un lenguaje le da la capacidad al programador de especificarle al ordenador, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano. (5)

1.5.1 JavaScript 1.6



El lenguaje JavaScript facilita para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems. (8)

Se utilizó JavaScript ya que el código puede incluirse en cualquier documento, ya sea PHP, ASP, JSP y SVG, además la mayoría de los navegadores (Internet Explorer, Netscape Navigator, Opera, Mozilla Firefox, entre otros) soportan el lenguaje. En el desarrollo del sistema se utilizó JavaScript como lenguaje de programación del lado del cliente y PHP como lenguaje del lado del servidor.

Capítulo 1: Fundamentación Teórica

1.5.2 PHP 5.2



Es el acrónimo de Hypertext Pre-Processor. Actualmente existen diferentes lenguajes de programación para desarrollar aplicaciones Web, PHP es un lenguaje de script interpretado por el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor.

La totalidad 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.

El funcionamiento a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor esto permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador, normalmente es una página HTML pero también podría ser una página WML al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP. (9)

PHP Dispone de una gran cantidad de características que lo convierte en la herramienta ideal para la creación de páginas Web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase MySQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) y además analiza código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP cuenta con la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. (10)

Capítulo 1: Fundamentación Teórica

De forma general se puede decir que se utilizó PHP en la confección del sistema ya que la propuesta para informatización de la facultad 3 lo plantea además que la arquitectura que se utilizó en la confección del sistema es cliente-servidor y las principales características de PHP como son rapidez; la facilidad de aprendizaje; el soporte multiplataforma tanto de diversos Sistemas Operativos como servidores HTTP y de bases de datos y el hecho de que se distribuye de forma gratuita bajo una licencia abierta y las políticas de migración de la universidad a software libre ofrece gran crédito a la hora de utilizar este lenguaje.

1.6 Metodología de desarrollo

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales. (11)

1.6.1 SXP

La producción de software incluye aspectos importantes y determinantes, pero sin dudas y es a lo que menos se le presta atención es a una selección adecuada de la metodología a seguir para llevar adelante dicho proceso. En el año 2007 la Ing. Malay Rodríguez Villar atendiendo a esta problemática realizó una investigación realizada en la facultad 7 que arrojó como resultado la propuesta MA-MRV-R1. Puesta en práctica en 6 sistemas escogidos del grupo UNICORNIOS de la facultad 10, de la Universidad de las Ciencias Informáticas.

SXP es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permiten actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo y ayudando al líder del proyecto a tener un mejor control del mismo. SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que se sepa por dónde se anda. XP más bien es una metodología encaminada al desarrollo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Consta de 4 fases principales: planificación-definición, desarrollo, entrega, y por último mantenimiento. En cada una de estas fases se realizan numerosas actividades donde se generan artefactos para documentar todo el proceso.

Capítulo 1: Fundamentación Teórica

Se utilizó esta metodología debido a que la propuesta de informatización de la facultad 3, documento por el que se rige a la hora de desarrollar esta trabajo tiene como metodologías a utilizar en su desarrollo “AUP, SCRUM,XP”, y como ya se detalló en la explicación de SXP anteriormente es la unión de SCRUM y XP, está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda al trabajo en conjunto, 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 puedan ver cómo progresa el trabajo.

1.7 Herramientas

Es importante tener en cuenta a la hora de realizar una aplicación web las herramientas necesarias para la puesta en marcha, es por ello que se explican las herramientas utilizadas en la implantación de este sistema:

1.7.1 Herramientas de Modelado

A la hora de realizar un modelo informático destinado a aumentar la eficiencia y productividad de algún software determinado se tiene que utilizar herramienta de tipo Case (Computer Aided Software Engineering), por sus siglas en inglés o (Ingeniería de Software Asistida por Computadora) en español.

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Como es sabido, los estados en el Ciclo de Vida de desarrollo de un Software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación. (12)

Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida del software en tareas como realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.



1.7.1.1 Visual Paradigm 8.0

En todas las empresas de desarrollo es importantes utilizar diagramas para documentar sus sistemas y aplicaciones para tener un aspecto más profesional. Visual paradigm es

Capítulo 1: Fundamentación Teórica

una herramienta case que permite construir diagramas UML, ya que soporta el ciclo completo de vida del software, análisis y diseño orientado a objeto y dibujar diagramas de clases: código inverso, generar código desde diagramas y generar documentación. Tiene disponibilidad en múltiples plataformas y en múltiples versiones, se integra con los siguientes IDEs javas; NetBeans y Eclipse entre otros. Visual Paradigm es un producto galardonado en la realización de UML, facilita al diagrama visual y al diseño de sus proyectos y les brinda la posibilidad de integrar y desplegar sus aplicaciones empresariales y sus bases de datos subyacentes. (13)

Para el de el desarrollo del modelado se utilizó la herramienta CASE Visual Paradigm ya que además de ser muy fácil su manejo se tiene un gran conocimiento de la misma, además es privativa y favorece la política de migración de la Universidad.

1.7.2 Herramientas de Bases de Datos

Los sistemas de gestión de bases de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan con un propósito general, los sistemas de gestión de bases de datos son los encargados del manejo de manera clara, sencilla y ordenada de un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

1.7.2.1 PostgreSQL 8.4

PostgreSQL Está considerado como el gestor de base de datos de código abierto más avanzado del mundo. PostgreSQL (llamado también Postgres95) fue derivado del proyecto Postgres. Este proyecto lleva más de una década de desarrollo, con diferencia, soportando la gran mayoría de las transacciones SQL, control concurrente, teniendo a su disposición varios "languages bindings" como por ejemplo C, C++, Java, Python, PHP y muchos más.



A continuación se enumeran las principales características de este gestor de bases de datos:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora una estructura de datos array.

Capítulo 1: Fundamentación Teórica

- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. (14)

A modo de conclusión se puede decir que se escogió PostgreSQL a pesar de la propuesta de informatización de la facultad 3, el personal encargado del desarrollo del sistema posee un mayor conocimiento del mismo y es un magnífico gestor de base de datos que cuenta con prácticamente todo lo que necesitan los gestores comerciales y se puede hacer de él una buena alternativa. Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios web de gran cantidad de peticiones, se utiliza principalmente por la condición de que es un software que se puede utilizar en diversos sistemas operativos entre los que destaca Linux y por la política de migración a software libre de la universidad fue la mejor opción.

PGAdmin

El máximo exponente de cliente gráfico de PostgreSQL es el software pgAdmin, que no es más que una herramienta de código abierto para administrar las bases de datos que incluye:

- Interfaz administrativa gráfica
- Herramienta de consulta SQL
- Editor de código procedura.
- Agente de planificación SQL/Shell /batch
- Administración de Slony-I

Una característica interesante de pgAdmin es que, cada vez que realizamos alguna modificación en un objeto, escribe las sentencias SQL correspondientes, lo que hace que, además de una herramienta muy útil, pgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir

Capítulo 1: Fundamentación Teórica

simples consultas SQL hasta desarrollar bases de datos complejas, La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX entre otros. Todas estas características hacen del pgAdmin la única herramienta gráfica que necesitaremos para trabajar con PostgreSQL, tanto desde el punto de vista del usuario como del administrador. (15)

Se utilizó pgAdmin como cliente gráfico para trabajar con el PostgreSQL, además de todo lo que se explica anteriormente, porque es una herramienta que puede ser usada en varios sistemas operativos, entre ellos Linux, lo que favorece la puesta en práctica de las políticas de migración de la universidad hacia esa plataforma.

1.7.3 Herramientas de Desarrollo

Las herramientas de desarrollo son aquellos programas o aplicaciones que tienen cierta importancia en el desarrollo de un programa (programación). Pueden ser de importancia vital (como un ensamblador, un compilador o un editor) o de importancia secundaria, como una IDE (Integrated Development Environment - Entorno de Desarrollo Integrado).

1.7.3.1 Netbeans 7.1



NetBeans es un proyecto exitoso de código abierto con una gran base de usuarios, una comunidad en

constante crecimiento, y con cerca de 100 socios (¡y creciendo!) en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos.

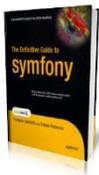
Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extenderlo. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

Capítulo 1: Fundamentación Teórica

También está disponible NetBeans Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. (16)

En el desarrollo del sistema se utilizó el IDE NetBeans ya que es una herramienta de código abierto y gratuito para uso tanto comercial como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL) v1.0 and the GNU General Public License (GPL) v2, además, según la propuesta de informatización de la facultad 3 propone como IDEs de desarrollo “NetBeans, Eclipse, Aptana”.

1.7.3.2 Symfony 2.0



Symfony es un framework que simplifica el desarrollo de una aplicación web mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Está basado en el clásico patrón de diseño web conocido como arquitectura MVC (Modelo-Vista-Controlador). Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. También, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP 5, 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.

Desde su primera versión, publicada en octubre de 2005 bajo la licencia de software libre por su creador Fabien Potencier, desarrolladores de todo el mundo descendieron e instalaron el framework, comenzaron a leer la documentación y construyeron sus primeras aplicaciones con Symfony, aumentando poco a poco la popularidad del mismo.

Los beneficios que brinda esta herramienta atraen a los programadores, que ven en Symfony una poderosa herramienta para la creación de aplicaciones web.

A través del foro de Symfony, las listas de correo y el IRC (canal de mensajería instantánea) se ofrecen otras alternativas válidas para el soporte del framework. De manera que los desarrolladores alrededor del

Capítulo 1: Fundamentación Teórica

mundo están conectados en una comunidad que muestra muchas ganas de colaboración entre sus integrantes.

Symfony cuenta con una gran cantidad de características que lo convierten en un framework ideal para su utilización:

- Fácil de instalar y configurar en la mayoría de 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
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web
- Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- 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. (17)

Se utilizó Symfony además de porque el sistema al que tiene que unirse está elaborado en Symfony y porque es un framework muy completo que se adapta a casi cualquier arquitectura y requerimientos del negocio, en ocasiones es necesario extender su funcionamiento. Es por esta razón que el framework es altamente configurable permitiendo la modificación de casi todas las partes del sistema. Además, es posible reemplazar las clases de su núcleo por clases propias desarrolladas por los usuarios y es un software libre y por las políticas de la universidad de migración a estos software no privativos se hace más conveniente su utilización.

Capítulo 1: Fundamentación Teórica

1.7.3.3 ExtJS 4.0



Es una biblioteca de JavaScript potente para la creación de aplicaciones enriquecidas del lado del cliente y que realmente brilla en la fabricación de aplicaciones web. Cuenta con gran cantidad de widgets, para crear interfaces de usuario complejas. Presenta un sinnúmero de características entre las que se encuentran: componentes de interfaz de usuario personalizables, con buen diseño y documentación. Inicialmente fue construido como una extensión de YUI. Incluye intemporalidad con jQuery (biblioteca o framework de JavaScript) y Prototype (biblioteca o framework de JavaScript). Posee controles para campos de textos, incluyendo áreas de texto. Controladores, selectores de fecha, campos numéricos, radiobox y checkbox. Contiene además componentes para crear y manipular dataGrids donde goza de cierta ventaja sobre otros frameworks. Es posible crear ventanas con barras de herramientas y menús con estilo de aplicaciones de escritorio, diálogos, modales y eventos. ExtJS puede ser adquirido bajo licencias libres y comerciales. La compañía detrás de este framework ofrece cursos de capacitación y un extenso soporte. (18)

Ventajas de EXTJS:

- Código reutilizable
- Independiente o adaptable a frameworks diferentes
- Orientada a la programación de interfaces tipo desktop en el web
- El API es homogeneizado independientemente del adaptador usado. Los controles siempre se verán igual Soporte comercial

EXTJS es una biblioteca de JavaScript muy utilizable en la actualidad por su fácil aprendizaje y que luego de la versión 2.0 su API es muy extenso y recordar todas las funciones, propiedades o configuraciones disponibles es casi imposible, contiene mucho de los métodos y propiedades de DOM (Document Object Model) que se necesitará para proporcionar una interfaz conveniente, unificada y multinavegador, y por las políticas de migración de la Universidad se hace recomendable ya que es un software que se puede adquirir bajo licencia privativa.

Capítulo 1: Fundamentación Teórica

1.8 Conclusiones

En este capítulo se analizaron conceptos necesarios para entender el procedimiento de gestionar la información de postgrado, también se realizó un análisis de las diferentes aplicaciones en el mundo y en Cuba, y la necesidad de realizar la aplicación debido a la inexistencia de una aplicación similar en la UCI y se describió la metodología utilizada para el desarrollo del trabajo SXP, herramientas y tecnologías propuestas para la solución del problema. De forma general se escogieron como lenguajes de programación JavaScript del lado del cliente y PHP del lado del servidor, como lenguajes de modelado se seleccionaron BPMN para modelar el negocio y UML para generar los artefactos relacionados con el sistema, además que se escogió como gestor de base de datos PostgreSQL y como herramienta gráfica para trabajar con el PostgreSQL el pgAdmin, como IDE de desarrollo el Netbeans, Symfony como framework para desarrollar el sistema y EXTJS como librería de JavaScript.

CAPÍTULO 2: ANÁLISIS Y DISEÑO

2.1 Introducción

En este capítulo se realizara un análisis del sistema gestión de la información de los cursos de postgrados de la facultad 3 consiguiendo una mayor comprensión del mismo y sus características. Se modelará el negocio identificando las actividades correspondientes a cada proceso y para una mayor comprensión se describen los requerimientos fundamentales y mediante estos realizar las Historias de Usuarios y la lista de reserva del producto quedando así diseñados los diagramas necesarios según plantea la metodología.

2.2 Descripción del negocio actual

El vicedecano de Investigación y postgrado de la facultad 3 entre otras tareas es el encargado de llevar un control sobre la superación profesional de cada profesor de la facultad. Esto se mide llevando un registro de cada curso recibido por cada profesor ya sea un diplomado, una maestría, un doctorado entre otras cosas, este proceso actualmente se realiza mediante una aplicación con bases Access que fue definido por la universidad de las Villas. Esta tarea se realiza por áreas, todos los profesores están distribuidos por un conjunto de departamentos donde el jefe de departamento es el encargado de llevar un control por el conjunto de profesores de los que son responsables y le hace entrega de forma trimestral al vicedecano toda la información de cada profesor al que él supervisa.

2.3 Modelo de negocio

- **Profesor:** Personal relacionado de alguna forma con la facultad que entrega la información de su superación profesional.
- **VDIP:** Vicedecano de Investigación y Postgrado.
- **Realizar Postgrado:** Es cursar un determinado curso, diplomado, maestría o doctorado.
- **Entregar Resultado:** Entregarle la información al jefe del departamento de que pasó algún determinado curso o lo impartió.
- **Aval:** Es un documento que certifica que un determinado profesor realizó un Postgrado.
- **Jefe de Departamento:** Persona encargada de obtener la información del Profesor procesarla y entregarla al VDIP si es de primer ciclo o al Subdirector de investigación y postgrados si es de segundo ciclo.
- **Primer ciclo:** Si el profesor pertenece a un departamento docente de la facultad es de primer ciclo.

- **Segundo ciclo:** Si el profesor pertenece a un centro relacionado con la facultad es de segundo ciclo.
- **Subdirector IP:** Subdirector de Investigación y Postgrado del centro.

Se validó el modelo de negocio a través del documento para la descripción de procesos de negocio.
Anexo21

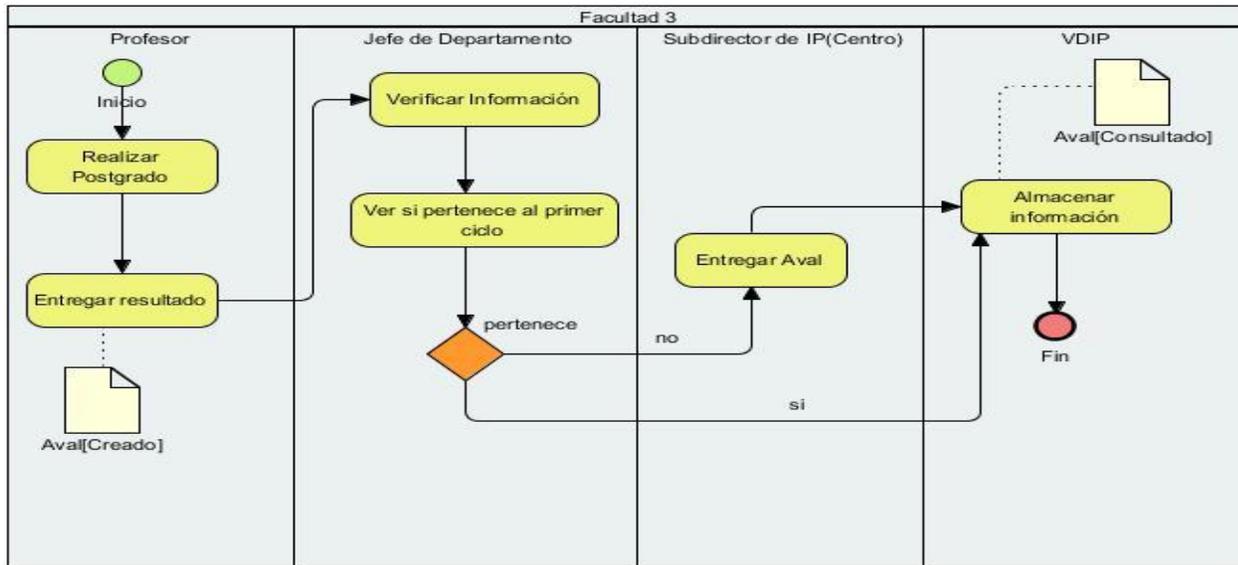


Figura 2.1 Modelo de Negocio

2.4 Propuesta de solución

Para dar solución a la situación problemática, se propone desarrollar un sistema para que la información de los cursos de postgrado de la facultad 3, en el cual se lleve de forma constante un control sobre todos los datos competentes a la superación profesional de todo el personal, permitiéndole a todo jefe de departamento y vicedecano actualizar la información de un determinado profesor así como la información de los cursos que la facultad está impartiendo, ya sea eliminarlos, modificarlos como adicionarlos al sistema y a su vez generar un ranking por área según las políticas de superación de la facultad.

2.5 Definición de los requisitos

La especificación de los requisitos de software se realizó en conjunto con el cliente a través de las técnicas de captura de requisitos: tormenta de ideas y entrevistas realizadas al cliente. Se tuvo en cuenta todas las ideas planteadas, sacando así las más óptimas para realizar el módulo.

2.6 Requisitos Funcionales

RF1. Gestionar área.

- Añadir área.
- Eliminar área.
- Modificar área.

RF2. Gestionar información de cursos.

- Añadir cursos
- Eliminar cursos
- Modificar cursos.

RF3. Gestionar información de diplomados.

- Añadir cursos de diplomados
- Eliminar cursos de diplomados
- Modificar cursos de diplomados

RF4. Gestionar información de doctorado

- Añadir doctorado.
- Eliminar doctorado.
- Modificar doctorado.

RF5. Gestionar información sobre maestrías

- Añadir maestrías.
- Eliminar maestrías.
- Modificar maestrías.

RF6. Generar reportes de profesores por áreas que han impartido cursos.

RF7. General reportes de recurso humanos.

RF8. Generar reportes de profesores que son máster y cursos que imparten.

RF9: Realizar ranking por área de los recursos humanos.

RF10. Generar reportes de profesores por areas.

RF11. Generar reportes de profesores que son doctores y cursos que imparten.

RF12. Generar reportes de la cantidad de egresados.

2.7 Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. (19)

La captura de los requisitos no funcionales se realizó mediante el documento propuesto por el sistema de informatización de la facultad 3 (SO3).

Hardware

Para el desarrollo: PC Intel Pentium 4 o superior, CPU 3GHZ o superior, 512 MB RAM o superior, 40 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 1 GB recomendada o superior, 80 GB HDD.

Usabilidad

Facilidad de uso por parte de los usuarios: El sistema debe presentar una interfaz 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 fácil navegación que posibilite a los usuarios sin experiencia una rápida adaptación.

Emplear perfiles de usuario: Diferenciar las interfaces y opciones para los usuarios que accedan al sistema según los diferentes roles que estos tengan dentro del sistema.

Capítulo 2: Análisis y Diseño

Menús: El sistema debe presentar una serie de menús, tanto laterales como en barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.

Fiabilidad

Seguridad de la base de datos: La base de datos deberá estar fraccionada en esquemas que permitan un mejor uso de la información y la división de forma lógica de las funcionalidades del sistema, trayendo consigo además, la protección de la información al ocurrir un incidente sobre una parte de la base de datos. El SGBD escogido debe presentar facilidades de administración de roles y usuarios restringiendo el acceso a los datos.

Políticas de seguridad por usuarios y roles: El sistema 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.

Seguridad

- Autenticar al usuario antes de que pueda realizar cualquier acción sobre el portal.
- Garantizar diferentes niveles de acceso al Sistema, brindado por la asignación de roles.

2.8 Personal relacionado con el sistema

Se define como persona relacionada con el módulo toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. También son las que se encuentran involucradas en dichos procesos, pues participan en ellos.

Personas relacionadas con el Sistema	Descripción
Profesor	El profesor luego de autenticarse puede ver toda su información archivada en el sistema además de la información de los demás profesores, pero no podrá realizar ningún cambio en el sistema.

Capítulo 2: Análisis y Diseño

Jefe de Área	El jefe de área es el encargado de dirigir un departamento docente el cual tendrá permiso en el sistema para adicionarle, modificarle y eliminarle la información a un profesor determinado mientras pertenezca a su área además de gestionar la información de cualquier curso que sea impartido por su área ya sea diplomado, maestría o doctorados entre otros.
Vicedecano	El vicedecano de Investigación y postgrado de la facultad 3 tendrá permisos además de lo que realiza el jefe de área podrá también generar un ranking por área donde se verá cuál de las áreas es la que mejor está cumpliendo el plan propuesto por la facultad respecto al cambio de categoría y podrá gestionar la información de los jefes de áreas.

2.9 Lista de Reserva del Producto

ID	Estimado (Días)	Requisitos Funcionales	Requisitos Asociados
1	2	Gestionar información del Área	Adicionar la información del Área.
2	2		Modificar la información del Área.
3	1		Eliminar la información del Área.
4	2	Gestionar información de	Adicionar la información de cursos.
5	2		Modificar la información de cursos.

Capítulo 2: Análisis y Diseño

		cursos	
6	1		Eliminar la información de cursos.
7	2		Adicionar información de diplomados.
8	2	Gestionar información de diplomados.	Modificar información de diplomados.
9	1		Eliminar información de diplomados.
10	2		Adicionar información de doctorados.
11	2	Gestionar información de doctorados.	Modificar información de doctorados.
12	1		Eliminar información de doctorados.
13	2		Adicionar información sobre maestrías.
14	2	Gestionar información sobre maestrías	Modificar información sobre maestrías.
15	1		Eliminar información sobre maestrías.
16	2	Generar reportes de profesores por áreas que han recibido cursos.	Gestionar información de cursos.
17	2	Generar reportes de profesores por áreas que han impartido cursos	Gestionar información de cursos.

Capítulo 2: Análisis y Diseño

18	2	Generar reportes de la cantidad de egresados.	Gestionar información de cursos.
19	2	Realizar ranking por área de los recursos humanos.	Realizar ranking por área de los recursos humanos.
20	2	Generar reportes de profesores que son doctores y cursos que imparten.	Gestionar información de cursos.
21	2	RF8. Generar reportes de profesores que son máster y cursos que imparten.	Gestionar información de cursos.
Tiempo total 37 días			Total de requisitos 21.

Prioridad Muy Alta

RF1. Gestionar área.

- Añadir área.
- Eliminar área.
- Modificar área

RF2. Gestionar información de cursos

- Añadir cursos.
- Eliminar cursos.
- Modificar cursos.

RF3. Gestionar información de diplomados.

- Añadir diplomados
- Eliminar diplomados
- Modificar diplomados

RF4. Gestionar información sobre maestrías

- Añadir maestrías.
- Eliminar maestrías.
- Modificar maestrías.

RF5. Gestionar información de doctorado.

- Añadir doctorado
- Eliminar doctorado
- Modificar doctorado

Prioridad Alta

RF6. Generar reportes de profesores por áreas que han impartido cursos.

RF7. Generar reportes de recurso humanos.

RF8. Generar reportes de profesores que son máster y cursos que imparten.

RF9. Generar reportes de profesores que son doctores y cursos que imparten.

RF10. Generar reportes de la cantidad de egresados.

RF 11. Generar reportes de profesores por áreas con cursos.

Prioridad Media

RF12: Realizar ranking por área de los recursos humanos

2.10 Historias de Usuarios

Historia de Usuario

Capítulo 2: Análisis y Diseño

Número: HU_1	Nombre Historia de Usuario: <i>Añadir cursos</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo permitir adicionar un determinado curso por el área referente al mismo y con el profesor encargado de impartir el curso y la matrícula del mismo además de la fecha de inicio del curso como en la que debe terminar el mismo y su programa analítico.</i>	
Observaciones: <ul style="list-style-type: none">➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i>➤ <i>El jefe de área solo podrá adicionar un determinado curso solo en caso que lo esté impartiendo su área.</i>➤ <i>El vicedecano de investigación y postgrado sí podrá adicionar un curso de cualquier área que pertenezca a la facultad.</i>➤ <i>Luego de adicionar el curso debe añadirse automáticamente en la lista de los cursos existentes.</i>	
Prototipo de interfaz: Anexo2	

Historia de Usuario

Capítulo 2: Análisis y Diseño

Número: HU_2	Nombre Historia de Usuario: <i>Modificar cursos</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo permitir cambiar la información de un determinado curso de postgrado ya sea profesor que lo imparte como si cambia la matrícula del curso entre otros.</i>	
Observaciones: <ul style="list-style-type: none">➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i>➤ <i>El jefe de área solo podrá adicionar un determinado curso solo en caso que lo esté impartiendo su área.</i>➤ <i>El vicedecano de investigación y postgrado sí podrá adicionar un curso de cualquier área que pertenezca a la facultad.</i>➤ <i>El sistema debe mostrar los datos que tiene actualmente el curso en su interfaz a la hora de modificarlo.</i>	
Prototipo de interfaz: Anexo3	

Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: <i>Eliminar cursos</i>

Capítulo 2: Análisis y Diseño

Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Usuario: <i>Roilán González Guerra</i>
Prioridad en Negocio: <i>Alta</i>	Prioridad en Negocio: <i>Alta</i>
Riesgo en Desarrollo: <i>Alto</i>	Riesgo en Desarrollo: <i>Alto</i>
Descripción: <i>La Historia de Usuario tiene como objetivo permitir borrar la información de un determinado curso.</i>	
Observaciones: <ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i> ➤ <i>El jefe de área solo podrá adicionar un determinado curso solo en caso que lo esté impartiendo su área.</i> ➤ <i>El vicedecano de investigación y postgrado sí podrá adicionar un curso de cualquier área que pertenezca a la facultad.</i> ➤ <i>Luego de eliminar el curso debe borrarse automáticamente en la lista de los cursos existentes.</i> 	
Prototipo de interfaz: Anexo4	

Historia de Usuario	
Número: <i>HU_4</i>	Nombre Historia de Usuario: <i>Añadir diplomados</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: <i>1</i>

Capítulo 2: Análisis y Diseño

Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
<p>Descripción: <i>La Historia de Usuario tiene como objetivo permitir adicionar un determinado diplomado mostrando sus datos en el sistema y viendo la cantidad de profesores que están matriculados en el diplomado así como la cantidad de cursos con que cuenta el mismo y sus fechas tanto la de inicio como la de fin.</i></p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i> ➤ <i>El jefe de área solo podrá adicionar un determinado diplomado solo en caso que lo esté impartiendo su área.</i> ➤ <i>El vicedecano de investigación y postgrado sí podrá adicionar un diplomado de cualquier área que pertenezca a la facultad.</i> ➤ <i>Luego de adicionar el diplomado debe añadirse automáticamente en la lista de los diplomados existentes.</i> 	
<p>Prototipo de interfaz: Anexo5</p>	

Historia de Usuario	
Número: <i>HU_5</i>	Nombre Historia de Usuario: <i>Modificar diplomados</i>
Modificación de Historia de Usuario Número: <i>Ninguna</i>	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1

Capítulo 2: Análisis y Diseño

Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
<p>Descripción: <i>La Historia de Usuario tiene como objetivo permitir cambiar la información de un determinado diplomado ya sea coordinador del diplomado, como la matrícula del diplomado o la fecha de inicio y de fin del mismo además que la cantidad de cursos que se imparten en el diplomado.</i></p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i> ➤ <i>El jefe de área solo podrá adicionar un determinado diplomado solo en caso que lo esté impartiendo su área.</i> ➤ <i>El vicedecano de investigación y postgrado sí podrá adicionar un diplomado de cualquier área que pertenezca a la facultad.</i> ➤ <i>El sistema debe mostrar los datos que tiene actualmente el diplomado en su interfaz a la hora de modificarlo.</i> 	
<p>Prototipo de interfaz: Anexo6</p>	

Historia de Usuario	
Número: <i>HU_6</i>	Nombre Historia de Usuario: <i>Eliminar diplomado</i>
Modificación de Historia de Usuario Número: <i>Ninguna</i>	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1

Capítulo 2: Análisis y Diseño

Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo permitir borrar la información de un determinado diplomado.</i>	
Observaciones: <ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado</i> ➤ <i>El jefe de área solo podrá adicionar un determinado diplomado solo en caso que lo esté impartiendo su área.</i> ➤ <i>El vicedecano de investigación y postgrado sí podrá adicionar un diplomado de cualquier área que pertenezca a la facultad.</i> ➤ <i>Luego de eliminar el diplomado debe borrarse automáticamente en la lista de los diplomados existentes.</i> 	
Prototipo de interfaz: Anexo7	

Historia de Usuario	
Número: <i>HU_7</i>	Nombre Historia de Usuario: <i>Añadir Maestría</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1

Capítulo 2: Análisis y Diseño

Descripción: La Historia de Usuario tiene como objetivo permitir adicionar una determinada maestría por el área referente al mismo y con el profesor encargado de coordinar la maestría y la matrícula del mismo además de la fecha de inicio de la maestría como en la que debe terminar la misma.

Observaciones:

- Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.
- El jefe de área solo podrá adicionar una determinada maestría solo en caso que lo esté impartiendo su área.
- El vicedecano de investigación y postgrado sí podrá adicionar una maestría de cualquier área que pertenezca a la facultad.
- Luego de adicionar la maestría debe añadirse automáticamente en la lista de las maestrías existentes.

Prototipo de interfaz: Anexo8

Historia de Usuario	
Número: HU_8	Nombre Historia de Usuario: <i>modificar maestría</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: La Historia de Usuario tiene como objetivo permitir cambiar la información de	

Capítulo 2: Análisis y Diseño

una determinada maestría ya sea coordinador de la maestría, como la matrícula de la maestría o la cantidad de maestría que han terminado y su fecha de inicio y de fin de la misma además que la cantidad de cursos que se imparten en la maestría.

Observaciones:

- Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.
- El jefe de área solo podrá adicionar una determinada maestría solo en caso que lo esté impartiendo su área.
- El vicedecano de investigación y postgrado sí podrá adicionar una maestría de cualquier área que pertenezca a la facultad.
- El sistema debe mostrar los datos que tiene actualmente en la interfaz de la maestría a la hora de modificarlo.

Prototipo de interfaz: Anexo9

Historia de Usuario	
Número: HU_9	Nombre Historia de Usuario: Eliminar maestría
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Roilán González Guerra	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: La Historia de Usuario tiene como objetivo permitir borrar la información de una determinada maestría.	

Capítulo 2: Análisis y Diseño

Observaciones:

- *Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.*
- *El jefe de área solo podrá adicionar una determinada maestría solo en caso que lo esté impartiendo su área.*
- *El vicedecano de investigación y postgrado sí podrá adicionar una maestría de cualquier área que pertenezca a la facultad.*
- *Luego de eliminar la maestría debe borrarse automáticamente en la lista de las maestrías existentes.*

Prototipo de interfaz: Anexo10

Historia de Usuario	
Número: <i>HU_10</i>	Nombre Historia de Usuario: <i>Añadir Doctorado.</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo permitir adicionar un determinado doctorado por el área referente al mismo y con el profesor encargado de coordinar el doctorado y la matrícula del mismo además de la fecha de inicio del doctorado como en la que debe terminar el mismo.</i>	

Capítulo 2: Análisis y Diseño

Observaciones:

- *Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.*
- *El jefe de área solo podrá adicionar un determinado doctorado solo en caso que lo esté impartiendo su área.*
- *El vicedecano de investigación y postgrado sí podrá adicionar un doctorado de cualquier área que pertenezca a la facultad.*
- *Luego de adicionar el doctorado debe añadirse automáticamente en la lista de los doctorados existentes.*

Prototipo de interfaz: Anexo11

Historia de Usuario	
Número: HU_11	Nombre Historia de Usuario: <i>Modificar doctorado</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo permitir cambiar la información de un determinado doctorado ya sea coordinador del doctorado, como la matrícula del doctorado entre o la cantidad de doctorado que has terminado o la fecha de inicio y de fin del mismo además que la cantidad de cursos que se imparten en el doctorado.</i>	

Capítulo 2: Análisis y Diseño

Observaciones:

- *Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.*
- *El jefe de área solo podrá adicionar un determinado doctorado solo en caso que lo esté impartiendo su área.*
- *El vicedecano de investigación y postgrado sí podrá adicionar un doctorado de cualquier área que pertenezca a la facultad.*
- *El sistema debe mostrar los datos que tiene actualmente el doctorado en su interfaz a la hora de modificarlo.*

Prototipo de interfaz: Anexo12

Historia de Usuario	
Número: HU_12	Nombre Historia de Usuario: <i>Eliminar doctorado</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo permitir borrar la información de una determinada doctorado.</i>	
Observaciones:	
<ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano</i> 	

Capítulo 2: Análisis y Diseño

de investigación y postgrado.

- *El jefe de área solo podrá adicionar un determinado doctorado solo en caso que lo esté impartiendo su área.*
- *El vicedecano de investigación y postgrado sí podrá adicionar un doctorado de cualquier área que pertenezca a la facultad.*
- *Luego de eliminar el doctorado debe borrarse automáticamente en la lista de los doctorados existentes.*

Prototipo de interfaz: Anexo13

Historia de Usuario	
Número: HU_13	Nombre Historia de Usuario: <i>Añadir Área</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo permitir adicionar una determinada área al sistema y con todos los datos referentes a las mismas, tanto el nombre como la descripción del área.</i>	
Observaciones:	
<ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado vicedecano de investigación y postgrado.</i> 	

Capítulo 2: Análisis y Diseño

- *El vicedecano de investigación y postgrado sí podrá adicionar un área al sistema.*
- *Luego de adicionar el área debe añadirse automáticamente en la lista de las áreas existentes.*

Prototipo de interfaz: Anexo14

Historia de Usuario	
Número: HU_14	Nombre Historia de Usuario: <i>Modificar Área</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>El presente Historial de Usuario tiene como objetivo permitir cambiar la información de un área ya sea el nombre da la misma como su descripción.</i>	
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado vicedecano de investigación y postgrado.</i> ➤ <i>El vicedecano de investigación y postgrado sí podrá modificar un área del sistema.</i> ➤ <i>Luego de adicionar el área debe añadirse automáticamente en la lista de las áreas existentes.</i> 	
Prototipo de interfaz: Anexo15	

Capítulo 2: Análisis y Diseño

Historia de Usuario	
Número: <i>HU_15</i>	Nombre Historia de Usuario: <i>Eliminar Área</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: <i>1</i>
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: <i>2</i>
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: <i>1</i>
<p>Descripción: <i>La Historia de Usuario tiene como objetivo permitir borrar la información de una determinada área.</i></p>	
<p>Observaciones:</p> <ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado vicedecano de investigación y postgrado.</i> ➤ <i>El vicedecano de investigación y postgrado sí podrá adicionar un área al sistema.</i> ➤ <i>Luego de eliminar el área debe eliminarse automáticamente de la lista de las áreas existentes.</i> 	
Prototipo de interfaz: Anexo16	

Historia de Usuario	
Número: <i>HU_16</i>	Nombre Historia de Usuario: <i>Generar reportes de profesores por áreas.</i>
Modificación de Historia de Usuario Número: Ninguna	

Capítulo 2: Análisis y Diseño

Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo mostrar un listado de todos los profesores de la facultad según al área que pertenezcan.</i>	
Observaciones: <ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i> ➤ <i>A la hora de generar el reporte se debe transformar el HTML en un PDF para mostrar el listado de todos los profesores de la facultad que pertenecen a un área determinada.</i> 	
Prototipo de interfaz: Anexo17	

Historia de Usuario	
Número: <i>HU_17</i>	Nombre Historia de Usuario: <i>Generar reportes de profesores por áreas que han impartido cursos.</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo mostrar un listado de todos los</i>	

Capítulo 2: Análisis y Diseño

profesores de la facultad que han impartido un curso.

Observaciones:

- *Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.*
- *A la hora de generar el reporte se debe transformar el HTML en un PDF para mostrar el reporte.*

Prototipo de interfaz: Anexo18

Historia de Usuario	
Número: <i>HU_18</i>	Nombre Historia de Usuario: <i>Generar reportes de profesores que son máster y cursos que imparten</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo mostrar un listado de todos los profesores que son máster y los cursos de postgrados que se encuentran impartiendo.</i>	
Observaciones:	
<ul style="list-style-type: none">➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i>➤ <i>A la hora de generar el reporte se debe transformar el HTML en un PDF para mostrar</i>	

Capítulo 2: Análisis y Diseño

el reporte.

Prototipo de interfaz: Anexo19

Historia de Usuario	
Número: HU_19	Nombre Historia de Usuario: <i>Generar reportes de profesores que son doctores y cursos que imparten</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo mostrar un listado de todos los profesores que son doctores y los cursos de postgrados que se encuentran impartiendo.</i>	
Observaciones: <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i>	
Prototipo de interfaz: Anexo19	

Historia de Usuario	
Número: HU_20	Nombre Historia de Usuario: <i>Generar reportes de la cantidad de egresados</i>
Modificación de Historia de Usuario Número: Ninguna	

Capítulo 2: Análisis y Diseño

Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1
Descripción: <i>La Historia de Usuario tiene como objetivo mostrar un listado de todos los profesores que son egresados de algún curso que impartió la facultad ya sea diplomado, maestría, doctorados, etc...</i>	
Observaciones:	
<ul style="list-style-type: none"> ➤ <i>Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.</i> ➤ <i>A la hora de generar el reporte se debe transformar el HTML en un PDF para mostrar el reporte.</i> 	
Prototipo de interfaz: Anexo19	

Historia de Usuario	
Número: <i>HU_21</i>	Nombre Historia de Usuario: <i>Realizar ranking por área de los recursos humanos</i>
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Roilán González Guerra</i>	Iteración Asignada: 1
Prioridad en Negocio: <i>Alta</i>	Puntos Estimados: 2
Riesgo en Desarrollo: <i>Alto</i>	Puntos Reales: 1

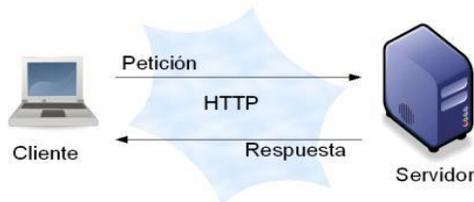
Descripción: *La Historia de Usuario tiene como objetivo mostrar un listado de todas las áreas de la facultad con el objetivo de ver cuál es la que mejor está cumpliendo con lo propuesto por la facultad.*

Observaciones:

- *Esta tarea solo la podrá realizar el personal adecuado, sea jefe de área o vicedecano de investigación y postgrado.*
- *Se debe generar un porcentaje del el área que mejor por ciento tiene en la facultad en lo referente a la cantidad de profesores que están cumpliendo con el plan de superación profesional establecido por la facultad 3 y mostrarse en una interfaz de forma tal que se vea la mejor área de la facultad.*

Prototipo de interfaz: Anexo20

2.11 Arquitectura cliente/servidor



La arquitectura cliente/servidor consiste en varios clientes distribuidos en diferentes nodos, conectados en red a uno o varios nodos servidores, donde el servidor puede atender varios clientes a la vez. En los nodos clientes se encuentran la presentación de usuario y en los nodos servidores la lógica del negocio. La arquitectura cliente/servidor es una forma de dividir y especializar programas y equipos de cómputo de forma que la tarea que cada uno de ellos realiza se efectúa con la mayor eficiencia posible y permita simplificar las actualizaciones y mantenimiento del sistema.

2.12 Patrones Arquitectónicos

Buschmann et al. (1996) define patrón como una regla que consta de tres partes, la cual expresa una relación entre un contexto, un problema y una solución. En líneas generales, un patrón sigue el siguiente esquema:

- Contexto. Es una situación de diseño en la que aparece un problema de diseño
- Problema. Es un conjunto de fuerzas que aparecen repetidamente en el contexto.
- Solución. Es una configuración que equilibra estas fuerzas. Ésta abarca:
 - Estructura con componentes y relaciones
 - Comportamiento a tiempo de ejecución: aspectos dinámicos de la solución, como la colaboración entre componentes, la comunicación entre ellos, etc. (20)

MODELO-VISTA-CONTROLADOR:

Divide una aplicación interactiva en tres componentes. El modelo (model) contiene la información central y los datos. Las vistas (view) despliegan información al usuario. Los controladores (controllers) capturan la entrada del usuario. Las vistas y los controladores constituyen la interfaz del usuario.

- El **Modelo** representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en la base de datos.
- La **Vista** es la información que es presentada al usuario, normalmente, será una página web.
- El **Controlador** sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web. (21)

Symfony es un framework totalmente basado sobre la arquitectura MVC se muestra un flujo de cómo sería el funcionamiento. El cliente envía una señal llamada REQUEST o Petición, ésta es interceptada por el Controlador quien realiza las validaciones necesarias, procesamiento de dichos datos y lógica de negocio asociadas a esa petición del cliente. El Controlador envía datos al Modelado. **(Figura 2.2)**

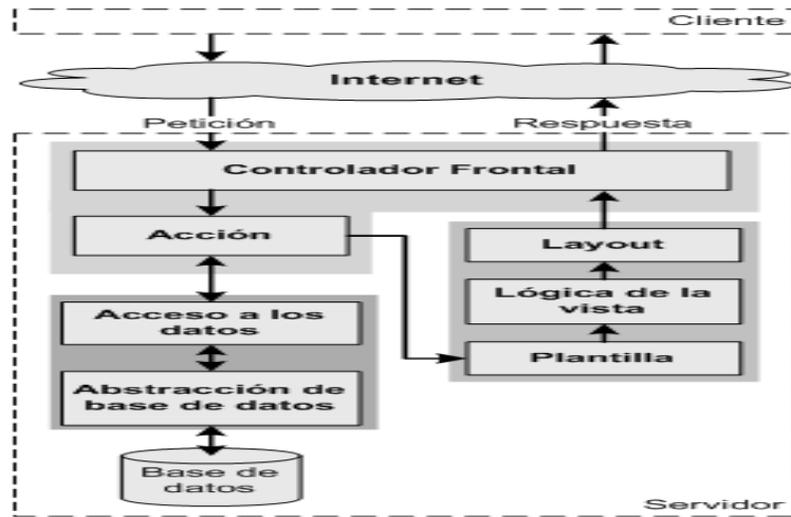


Figura 2.2 Ejemplo del funcionamiento de Symfony basado en el patrón MVC

2.13 Patrones de Diseño

GRASP:

En el diseño de software orientado a objeto el uso de los patrones GRASP acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades) tiene una gran importancia, debido a que solucionan muchos de los problemas que se pueden presentar a la hora de programar.

Para el diseño de la solución se tuvieron en cuenta los 5 patrones GRASP: Experto, Creador, Bajo acoplamiento, Alta cohesión y Controlador.

- **Experto:** Define el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Con la utilización de este patrón se definió dónde colocar en cada clase las funcionalidades que necesitan de esa información.
- **Creador:** permite identificar quién debe ser el responsable de la instanciación de nuevos objetos o clases. Este patrón se utilizó para identificar qué clase A debe crear elementos de una clase B, apoyándose en que la clase A debería: contener, agregar, registrar, utilizar y tener los datos de inicialización de la clase B.
- **Alta Cohesión:** define que la información que almacena una clase debe ser coherente y está en mayor medida relacionada con la clase. En el sistema es necesario controlar la complejidad de

Capítulo 2: Análisis y Diseño

cada clase utilizada para mantener un buen comportamiento de las mismas, por esto, las clases que fueron identificadas con una gran cantidad de funcionalidades se dividieron en otras clases, de manera que se repartiera equitativamente el peso de la complejidad, manteniendo además la coherencia de las clases.

- **Bajo Acoplamiento:** Este patrón se utilizó con la idea de tener las clases lo menos relacionadas entre sí posibles. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto, potenciando la reutilización, y disminuyendo la dependencia entre ellas.
- **Controlador:** se utilizó como intermediario entre cada una de las capas, de forma tal que garantice la comunicación entre los eventos externos del sistema en la capa de presentación y los componentes de la capa de negocio, declara el constructor de clase como privado para que no sea instanciable directamente.

GoF:

- **Singleton (Única instancia):** Es un patrón creacional que garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. A través de la clase `sfContext`: `getInstance()`, se obtiene una referencia a todos los objetos del núcleo del framework, por lo que es posible por ejemplo acceder desde las clases de formularios a métodos definidos en la clase que maneja las llamadas.
- **Front Controller (Controlador Frontal):** Es un patrón de diseño web usado como único punto de entrada a la aplicación. Realiza tareas comunes a todos los controladores: manejo de la seguridad, de las peticiones de los usuarios, carga de la configuración de la aplicación y delega la responsabilidad de responder a las peticiones al módulo específico que tiene la acción enviada en la petición.
- **Decorator (Envoltorio):** Normalmente, en las aplicaciones web existen contenidos que son comunes en todas las páginas que conforman la aplicación. Symfony haciendo uso de este patrón de diseño estructural delimita el código común en todas las páginas definido en un archivo global para todas las vistas denominado `layout` del código HTML generado como respuesta a una petición determinada (plantilla). El contenido de la plantilla se integra en el `layout`, por lo que se puede afirmar que el `layout` decora la plantilla.

2.14 Tarjetas CRC

Estas tarjetas son una técnica de modelado orientado a objeto que permite identificar las clases y sus responsabilidades, se hacen con el objetivo de identificar jerarquías de generalización/especificación, o jerarquías de agregación, de manera que ayude al refinamiento de clases. Se definen las tarjetas CRC con la finalidad de obtener un diseño simple y no incurrir en la implementación de características no necesarias.

Nombre de la Clase: Área	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de los profesores.	Profesores
Es una clase de la que heredan las clases maestría, doctorado y diplomado tiene como función que sea mejor el manejo de los dichos datos de forma más fácil desde un área.	MappedSupperCassBasse

Nombre de la Clase: Profesor	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de las Áreas.	Área
Contiene la lógica de negocio de la gestión de la información de las Maestría.	Maestría
Contiene la lógica de negocio de la gestión	Doctorado

Capítulo 2: Análisis y Diseño

de la información de los Doctorados.	
Contiene la lógica de negocio de la gestión de la información de los Diplomados.	Diplomado
Contiene la lógica de negocio de la gestión de la información de los Cursos.	Curso
Contiene la lógica de negocio de la gestión de la información de la calificación de un doctorado asociado a un profesor.	CalificacionDoctoradoProfesor
Contiene la lógica de negocio de la gestión de la información de la calificación de una maestría asociado a un profesor.	CalificacionMaestríaProfesor
Contiene la lógica de negocio de la gestión de la información de la calificación de un diplomado asociado a un profesor.	CalificacionDiplomadoProfesor
Contiene la lógica de negocio de la gestión de la información de la calificación de un curso asociado a un profesor.	CalificacionCursoProfesor

Nombre de la Clase: Maestría

Responsabilidad

Clases Relacionadas

Contiene la lógica de negocio de la gestión de la información de los profesores.

Profesores

Capítulo 2: Análisis y Diseño

Esta clase es la clase padre de la cual hereda la clase maestría y gestiona su información relacionada con un área.	MappedSupperCassBasse
Contiene la información relacionada con la edición de la maestría.	EdicionMaestría

Nombre de la Clase: Doctorado	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de los profesores.	Profesores
Esta clase es la clase padre de la cual hereda la clase doctorado y gestiona su información relacionada con un área determinada.	MappedSupperCassBasse
Contiene la información relacionada con la edición de los doctorados.	EdicionDoctorado

Nombre de la Clase: Diplomado	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de los profesores.	Profesores
Esta clase es la clase padre de la cual hereda la clase maestría que gestiona su	MappedSupperCassBasse

Capítulo 2: Análisis y Diseño

información relacionada con un área.	
Contiene la información relacionada con la edición de un diplomado.	EdicionDiplomado

Nombre de la Clase: Curso	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de las Áreas.	Área
Contiene la lógica de negocio de la gestión de la información de los profesores.	Profesores
Contiene la información relacionada con la edición de un diplomado.	EdicionCurso
Contiene la lógica de negocio de la gestión de la información de las Maestría.	Maestría
Contiene la lógica de negocio de la gestión de la información de los Doctorados.	Doctorado
Contiene la lógica de negocio de la gestión de la información de los Diplomados.	Diplomado

Capítulo 2: Análisis y Diseño

Nombre de la Clase: MappedSupperCassBasse	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de las Áreas.	Área
Contiene la lógica de negocio de la gestión de la información de las Maestría.	Maestría
Contiene la lógica de negocio de la gestión de la información de los Doctorados.	Doctorado
Contiene la lógica de negocio de la gestión de la información de los Diplomados.	Diplomado

Nombre de la Clase: EdicionDiplomado	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de la calificación de un diplomado asociado a un profesor.	CalificacionDiplomadoProfesor
Contiene la lógica de negocio de la gestión de la información de los Diplomados.	Diplomado

Capítulo 2: Análisis y Diseño

Nombre de la Clase: EdicionMaestría	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de la calificación de un diplomado asociado a un profesor.	CalificacionMaestríaProfesor
Contiene la lógica de negocio de la gestión de la información de las Maestría.	Maestría

Nombre de la Clase: EdicionDoctorado	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de la calificación de un diplomado asociado a un profesor.	CalificacionDoctoradoProfesor
Contiene la lógica de negocio de la gestión de la información de los Doctorados.	Doctorado

Nombre de la Clase: EdicionCurso	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la lógica de negocio de la gestión de la información de la calificación de un diplomado asociado a un profesor.	CalificacionCursoProfesor

Capítulo 2: Análisis y Diseño

Contiene la lógica de negocio de la gestión de la información de los Cursos.	Curso
--	--------------

Nombre de la Clase: CalificacionCursoProfesor	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la información relacionada con la edición de un diplomado.	EdicionCurso
Contiene la lógica de negocio de la gestión de la información de los profesores.	Profesores

Nombre de la Clase: CalificacionMaestríaProfesor	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la información relacionada con la edición de una maestría.	EdicionMaestría
Contiene la lógica de negocio de la gestión de la información de los profesores.	Profesores

Nombre de la Clase: CalificacionDoctoradoProfesor	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la información relacionada con la edición de un doctorado.	EdicionDoctorado

Contiene la lógica de negocio de la gestión de la información de los profesores.	Profesores
--	-------------------

Nombre de la Clase: CalificacionDiplomadoProfesor	
<u>Responsabilidad</u>	<u>Clases Relacionadas</u>
Contiene la información relacionada con la edición de un Diplomado.	EdicionDiplomado
Contiene la lógica de negocio de la gestión de la información de los profesores.	Profesores

2.15 Diagrama de Despliegue

El modelo de despliegue describe la distribución física del sistema, muestra la distribución de los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura de software y la de hardware. **(Figura 2.3)**

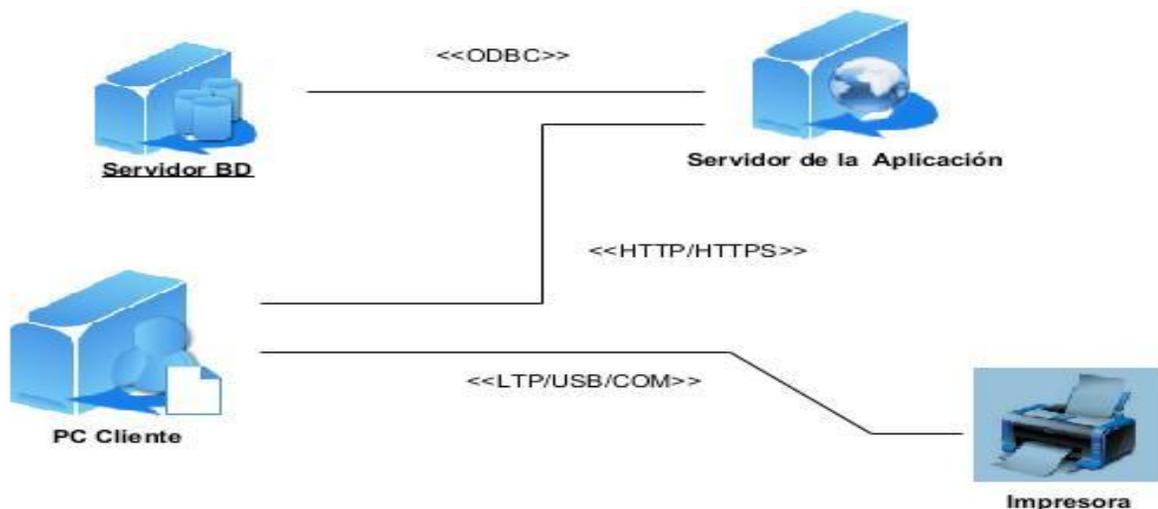


Figura 2.3 Diagrama de Despliegue

Capítulo 3: Implementación y Prueba

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

3.1 Introducción

En el presente capítulo se hace una valoración de los principales artefactos que intervienen en la Fase de Prueba. Se describen las Pruebas Unitarias y de Aceptación que son los requeridos por la metodología SXP. Además, son realizadas las plantillas de las Pruebas de Aceptación para cada una de las Historias de Usuarios implementadas, ya que son las seleccionadas para la validación de la propuesta de solución además del estándar de programación utilizado para desarrollo el sistema.

3.2 Estándar de Programación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez, con vista de realizar una codificación de alta calidad ya que la calidad del software es de suma importancia, por otra parte, a la hora de realizar las pruebas una codificación estándar que facilita esta tarea. Anexo1

3.3 Tareas de ingeniería

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU_1, HU_2, HU_3
Nombre Tarea: <i>Implementar código para gestionar cursos.</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable:: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción gestionar y seguidamente se despliegan todas las actividades a gestionar, cuando se selecciona en el vínculo curso aparece un listado con los cursos existentes y las opciones de eliminar, adicionar o modificar un determinado curso en específico. Teniéndose en cuenta todos los datos asociados al</i>	

Capítulo 3: Implementación y Prueba

mismo.

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: HU_4, HU_5, HU_6
Nombre Tarea: <i>Implementar código para gestionar diplomado.</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción gestionar y seguidamente se despliegan todas las actividades a gestionar, cuando se selecciona en el vínculo diplomado aparece un listado con los diplomados existentes y las opciones de adicionar, modificar y eliminar un determinado diplomado en específico. Teniéndose en cuenta todos los datos asociados a la misma.</i>	

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: HU_7, HU_8, HU_9
Nombre Tarea: <i>Implementar código para gestionar maestría.</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción gestionar y seguidamente se despliegan todas las actividades a gestionar, cuando se selecciona en el vínculo maestría aparece un listado con las maestrías existentes y las opciones de adicionar, modificar y</i>	

Capítulo 3: Implementación y Prueba

eliminar una determinada maestría en específico. Teniéndose en cuenta todos los datos asociados a la misma.

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: HU_10, HU_11, HU_12
Nombre Tarea: <i>Implementar código para gestionar doctorado.</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción gestionar y seguidamente se despliegan todas las actividades a gestionar, cuando se selecciona en el vínculo doctorado aparece un listado con los doctorados existentes y las opciones de adicionar, modificar y eliminar un determinado doctorado en específico. Teniéndose en cuenta todos los datos asociados al mismo.</i>	

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: : HU_13, HU_14, HU_15
Nombre Tarea: <i>Implementar código para gestionar área.</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable:: <i>Roilán González Guerra</i>	

Capítulo 3: Implementación y Prueba

Descripción: *En el menú del usuario aparece la opción gestionar y seguidamente se despliegan todas las actividades a gestionar, cuando se selecciona en el vínculo área aparece un listado con las áreas existentes y las opciones de adicionar, modificar y eliminar una determinada área en específico. Teniéndose en cuenta todos los datos asociados a la misma.*

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: HU_16
Nombre Tarea: <i>Implementar código para generar reportes de profesores por áreas.</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción reportes y seguidamente se despliegan todos los reportes a generar, cuando se selecciona en el vínculo generar reportes de profesores por áreas aparece un listado con los profesores que pertenecen al área referenciada y con la opción de imprimir que permitirá transformar todo la interfaz a un documento PDF con toda la información referente a la misma.</i>	

Tarea de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: HU_17
Nombre Tarea: <i>Implementar código para generar reportes de profesores por áreas que han impartido cursos.</i>	

Capítulo 3: Implementación y Prueba

Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción reportes y seguidamente se despliegan todos los reportes a generar, cuando se selecciona en el vínculo generar reportes de profesores por áreas que han impartido cursos aparece un listado con los profesores que pertenecen al área referenciada que han impartido cursos y con la opción de imprimir que permitirá transformar todo la interfaz a un documento PDF con toda la información referente a la misma.</i>	

Tarea de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: <i>HU_18</i>
Nombre Tarea: <i>Implementar código paragenerar reportes de profesores que son máster y cursos que imparten</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción reportes y seguidamente se despliegan todos los reportes a generar, cuando se selecciona en el vínculo generar reportes de profesores que son máster y cursos que imparten aparece un listado con los profesores que son máster y con los cursos que imparten y con la opción de imprimir que permitirá transformar todo la interfaz a un documento PDF con toda la información referente a la misma.</i>	

Capítulo 3: Implementación y Prueba

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: HU_19
Nombre Tarea: <i>Implementar código para Generar reportes de profesores que son doctores y cursos que imparten.</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 2
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción reportes y seguidamente se despliegan todos los reportes a generar, cuando se selecciona en el vínculo Generar reportes de profesores que son doctores y cursos que imparten aparece un listado con los profesores que son doctores y con los cursos que imparten y con la opción de imprimir que permitirá transformar todo la interfaz a un documento PDF con toda la información referente a la misma.</i>	

Tarea de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: HU_20
Nombre Tarea: <i>Implementar código para generar reportes de la cantidad de egresados</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción reportes y seguidamente se despliegan todos los reportes a generar, cuando se selecciona en el vínculo generar reportes de la cantidad de egresados aparece un listado con los profesores que son egresados de algún curso de la facultad y con los cursos del que son egresados y con la opción de imprimir</i>	

Capítulo 3: Implementación y Prueba

que permitirá transformar todo la interfaz a un documento PDF con toda la información referente a la misma.

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: HU_21
Nombre Tarea: <i>Implementar código para generar ranking por área de los recursos humanos</i>	
Tipo de Tarea : <i>Desarrollo</i>	Puntos Estimados: 1
Programador Responsable: <i>Roilán González Guerra</i>	
Descripción: <i>En el menú del usuario aparece la opción reportes y seguidamente se despliegan todos los reportes a generar, cuando se selecciona en el vínculo generar ranking por área de los recursos humanos aparece un gráfica donde se muestra el estado de los recursos humanos de la facultad sacando el área más destacada según la política de superación de la facultad para los profesores y con la opción de imprimir que permitirá transformar todo la interfaz a un documento PDF con toda la información referente a la misma.</i>	

3.4 Pruebas

Las pruebas son actividades en las cuales se ejecuta un sistema bajo condiciones o requerimientos específicos. Los resultados son observados, registrados y evaluado. Las pruebas verifican la implementación del sistema. La metodología SXP realiza estas pruebas de dos formas: pruebas unitarias y pruebas de aceptación. Los casos de prueba se deben escribir antes de comenzar la implementación y siempre que sea necesario se puede incluir uno nuevo. Se deben escribir casos de prueba hasta que quede claro el objetivo de las Historia de Usuario y se verifique que cumpla con todos los requerimientos, con el objetivo de tener una forma para que el cliente conozca cuando una HU está lista.

Capítulo 3: Implementación y Prueba

3.4.1 Pruebas Unitarias

Son desarrolladas por los programadores, consisten en comprobaciones manuales o automatizadas realizadas para verificar que el código correspondiente a un módulo funciona de acuerdo los requisitos del sistema. Las ventajas de este tipo de prueba son grandes, pues facilitan que ante un error el código sea cambiado por el programador para mejorar su estructura. Son definidas por los programadores antes de comenzar a escribir código, deben contemplar cada módulo del sistema que pueda generar fallas, para poder integrar el código realizado al ya existente.

3.4.2 Pruebas de Aceptación

El cliente con ayuda del probador define las pruebas de aceptación para cada Historia de Usuario a principio de cada iteración. Las pruebas de aceptación se utilizan para validar que cada requerimiento implementado funciona como se había especificado. Tienen como objetivo que las funcionalidades del sistema cumplan con lo que el cliente espera de ellas, permiten al cliente y al grupo de desarrollo conocer los problemas que presenta el sistema dándole la facilidad a los desarrolladores de corregirlas.

3.5 Caso de Prueba de Aceptación

La evaluación del caso de prueba va a estar dada en satisfactoria o insatisfactoria, para que una prueba sea satisfactoria tiene que cumplir con todas las condiciones que define el caso de prueba, que se cumpla con todo lo descrito en la prueba, que se verifiquen las condiciones de ejecución y los pasos de la misma, además que se obtengan todos los resultados esperados. En caso de que alguno de los anteriores aspectos no se cumpla el caso de prueba sería evaluado de insatisfactorio. Anexo22

3.6 Realización de las pruebas de aceptación

En la realización de pruebas de aceptación se llevaron a cabo dos iteraciones. En la primera iteración se evaluó un total de 21 casos de pruebas, de ellos, 15 fueron satisfactorios pues en estos no se detectaron no conformidades, 5 fueron insatisfactorios debido a que presentaban no conformidades y 1 no fue evaluado puesto que no se encontraba en funcionamiento. En la segunda iteración, todos los requisitos fueron evaluados de satisfactorios, quedando culminada las pruebas de aceptación. **(Figura 2.5)**

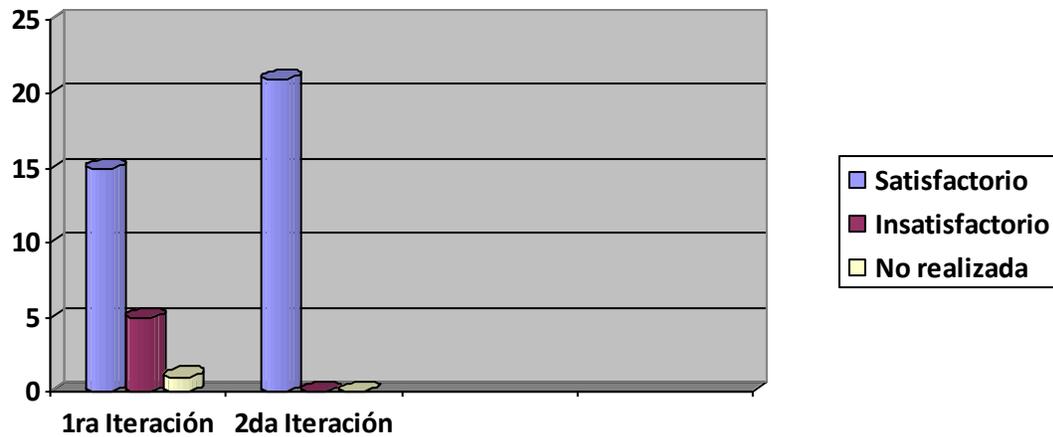


Figura 2.5 Pruebas de Aceptación

3.7 Conclusiones Parciales

En este capítulo se dio fin al flujo de pruebas que describe SXP además de definir el estándar de programación utilizado, se realizó un análisis de todos los casos de pruebas ya que son los mecanismos usados para asegurar que el sistema cumpla con los requerimientos funcionales asegurando así la calidad del software y se corrigieron todos los errores que lanzó el sistema mediante las pruebas de aceptación.

CONCLUSIONES GENERALES

Con la utilización de las herramientas, lenguajes y tecnologías propuestas en el marco teórico se garantizó la puesta en funcionamiento del sistema para la gestión de la información de postgrado en la facultad 3 cumpliéndose que fuese visible en cualquier plataforma y con las políticas de migración de la Universidad.

Una vez realizado el levantamiento de los requisitos funcionales y no funcionales a través de la Lista de Reserva del Producto y la elaboración de todas las Historias de Usuario se pudo diseñar el módulo Sistema Gestión de la Información de Postgrado en la facultad 3, permitiendo establecer sus características y todo lo necesario para comenzar con su construcción.

A partir del diseño del esquemas de la base de datos y la implementación de las historias de usuarios se desplegó el sistema para la gestión de la información de postgrado de la facultad 3 cumpliéndose el objetivo de este trabajo de diploma, partiendo de los tipos de pruebas que se le realizaron al sistema “unitarias y de aceptación” apoyándose en el estándar de programación y los casos de pruebas, se comprobó que las funcionalidades descritas satisfacen las necesidades del cliente obteniéndose los resultados esperados.

RECOMENDACIONES

En el desarrollo del presente trabajo se alcanzaron todos los objetivos propuestos, teniendo en cuenta que se llegaron a implementar todas funcionalidades, sin embargo en el transcurso del mismo surgieron ideas novedosas como son:

- Continuar el estudio con el objetivo de añadir nuevas funcionalidades al módulo.
- Terminar el completamiento de todos los datos de los profesores en el directorio para trabajar con la categoría real.

BIBLIOGRAFÍA

1. EcuRed. [En línea] [Citado el: 05 de 03 de 2012.] <http://www.ecured.cu/index.php/Postgrado>.
2. Ortega, José Manuel. politécnica. [En línea] 23 de 03 de 2011. [Citado el: 28 de 02 de 2012.] revista.eps.ua.es/index.php/noticias/opinion/215-criterios-para-realizar-un-curso-de-postgrado.
3. Archivo Nacional de la republica de cuba. [En línea] 12 de 10 de 2009. [Citado el: 01 de 03 de 2012.] <http://www.arnac.cu/index.php/tema/superacion/estrategia-de-superacion>.
4. UAB. *Escuela de Postgrado*. [En línea] 2012. [Citado el: 22 de 05 de 2012.] <http://www.uab.es/servlet/Satellite/postgrado/la-escuela-de-postgrado/la-escuela-de-postgrado-uab-1096483423465.html>.
5. LENGUAJES DE PROGRAMACIÓN. [En línea] [Citado el: 04 de 03 de 2012.] <http://www.frt.utn.edu.ar/sistemas/paradigmas/lenguajes.htm>.
6. Fajardo, Jorge Ugarte. slideshare. [En línea] 2008. [Citado el: 25 de 5 de 2012.] <http://www.slideshare.net/gugarte/bpmn-estandar-para-modelamiento-de-procesos-presentation..>
7. Anna Dugarte, Denis Aveñaneda, Brigida Contreras. oocities. [En línea] [Citado el: 04 de 03 de 2012.] <http://www.oocities.org/es/annadugarte/ads1/UML.htm>.
8. Pérez, Javier Eguíluz. *Introducción a JavaScript*.
9. Segura, José Luis Ponce. scribd. *Guía N2 - Proyectos Web - PHP*. [En línea] [Citado el: 29 de 05 de 2012.] <http://es.scribd.com/doc/21344724/Guia-N2-Proyectos-Web-PHP>.
10. SERRANO, JOSÉ DAVID CALDERÓN. *SISTEMA INFORMÁTICO DE APOYO PARA LA PLANIFICACIÓN DIDÁCTICA PARA LOS DOCENTES DE EDUCACIÓN BÁSICA, UNIVERSIDAD DE EL SALVADOR*. Salvador : s.n., 6 DE JUNIO DEL 2008.
11. Recopilación, de actores. alarcos. *INGENIERÍA DEL SOFTWARE I*. [En línea] 2006. [Citado el: 04 de 03 de 2012.] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/>.
12. Scribd. [En línea] [Citado el: 03 de 03 de 2012.] <http://es.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE>.

13. Sierra Cruz, Daniel. SlidesShare. [En línea] 15 de 11 de 2007. [Citado el: 03 de 03 de 2012.] <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
14. Salvador, Universidad del. Base de datos. [En línea] 2012. [Citado el: 03 de 03 de 2012.] <http://basesdedatosues.blogspot.mx/>.
15. GUÍA DOCUMENTADA PARA UBUNTU. [En línea] [Citado el: 03 de 03 de 2012.] http://www.guia-ubuntu.org/index.php?title=PgAdmin_III.
16. Corporation, Oracle. NetBeans. [En línea] [Citado el: 03 de 03 de 2012.] http://netbeans.org/index_es.html.
17. Potencier, François Zaninotto y Fabien. *Symfony 2.0, la guía definitiva*. s.l. : Apress (ISBN-13: 978-1590597866), 2011.
18. Tavárez, David. Maestros Del Web. [En línea] [Citado el: 04 de 03 de 2012.] <http://www.maestrosdelweb.com/editorial/comparacion-frameworks-javascript>.
19. Varela, Fèlix. *Ingeniería del Software un enfoque práctico*. La Habana : s.n., 2006. vo I1.
20. NUÑEZ , ERIKA CAMACHO, FABIO CARDESO , GABRIEL. *ARQUITECTURAS DE SOFTWARE, "GUÍA DE ESTUDIO"*. 2004.
21. Lorenzo, Victor. slideshare. *manual de Code IGniteren español*. [En línea] 2002. [Citado el: 29 de 05 de 2012.] <http://www.slideshare.net/combamx/code-igniter-spanishuserguide>.