

Universidad de las Ciencias Informáticas



FACULTAD 8

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Herramienta Web para la Gestión de la Información de Formación Postgraduada en la Facultad 8

Autores:

Addiel Rodríguez Guadarrama

Ariel Yasel Guillen Bravo

Tutores:

Ing. Yobannys Cabrera González

Lic. Madeleynis Román González

Ciudad de la Habana, Diciembre de 2008.

“Año 50 de la Revolución”

Declaración de Autoría.

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los ___ días del mes de Junio de 2009.

Autores:

Ariel Yasel Guillén Bravo

Addiel Rodriguez Guadarrama

Tutores:

Lic. Madeleynis Román González

Ing. Yobannys Cabrera González



"...Milito en el grupo de los impacientes, y milito en el bando de los apurados, y de los que siempre presionan para que las cosas se hagan, y de los que muchas veces tratan de hacer más de lo que se puede..."

Fidel

Dedicatoria.

A la Revolución, a nuestros Padres, Hermanos, Familiares y Amigos....

Agradecimientos.

A mis padres Mayra María Guadarrama Sierra y Guillermo Rodríguez López por haberme sabido guiar por el camino correcto.

A mi hermano Guillermo Aryam Rodríguez Guadarrama por haberme servido de ejemplo.

A mi familia por haber confiado en mí aún en los momentos más difíciles.

A mis tutores Lic. Madeleynis Román González y Ing. Yobannys Cabrera González por toda la ayuda que nos dieron.

A Yanirys Silva Lemus por ayudarme siempre que lo necesitaba.

A mis compañeros de grupo por los buenos momentos en estos 5 años.

A Arianna Rodríguez Jiménez (Ary) por toda la ayuda brindada y por mandarme al laboratorio cuando no quería ir.

A mis profesores por transmitirme todo el conocimiento necesario para la realización de este trabajo.

A todo el que de una forma u otra me ayudó en la realización de este trabajo.

Addiel Rodríguez Guadarrama.

Agradecimientos.

A mis padres Tania Bravo Murillo y Nicomedes Guillen Paneque por haber forjado en mi todo lo que soy, por la inspiración, por el ejemplo de vida que me han dado y por la guía certera.

A mis tías Yolanda y Kelnis por su dulzura, el apoyo incondicional y por darme fuerzas para continuar.

A mi Indira por todo su amor, ternura y por quererme siempre.

A mi familia por estar siempre ahí cuando lo necesité, a mi abuela, tíos, tías, primos y primas, en fin a todos.

A mis amigos y hermanos Luiso, Curiel, Raúl René, Dina que siempre han estado en los momentos que los he necesitado.

A mis tutores Lic. Madeleynis Roman González y Ing. Yobannys Cabrera González por la guía y la ayuda.

A mis compañeros y amigos de estudio por todo su apoyo y los buenos momentos.

A la mini comunidad de Symphony por la disponibilidad

A todo el que de una forma u otra ha hecho posible la realización de este sueño.

Ariel Yasel Guillén Bravo.

A la Revolución, a la Universidad de las Ciencias Informáticas por darnos la oportunidad de convertirnos en Ingenieros Informáticos.

Muchas Gracias a todos.

Resumen.

En este trabajo se presenta una aplicación web dinámica con el objetivo de optimizar el trabajo y la documentación referente al proceso de gestión de la información de Formación Postgraduada en la Facultad 8. Esta aplicación será independiente del sistema operativo donde se ejecute y presentará un requerimiento de hardware mínimo, brindará una amplia información sobre todo el tema referente a la formación postgraduada, minimizará el papeleo y agilizará el flujo de información, ahorrando así tiempo y recursos a nuestra facultad. El sistema cuenta con una interfaz amigable y fácil de usar, permitiendo así que un mayor número de usuarios puedan disfrutar de sus servicios.

Contenidos.

Introducción.....	16
Situación Problemática	16
Capítulo 1 Fundamentación teórica.....	19
1.1 Introducción.....	19
1.2 Estado del arte.....	20
1.2.1 ¿Qué es la gestión académica?	20
1.2.2 GESTACAD.....	20
1.2.3 Akademos.....	20
1.3 Tendencias y tecnologías actuales.....	21
1.3.1 Lenguajes y plataformas de programación para la Web.....	21
1.3.1.1 JSP.....	22
1.3.1.2 ASP.NET.....	22
1.3.1.3 Python.....	22
1.3.1.4 PHP.....	23
1.3.1.5 ¿Por qué PHP?.....	25
1.3.2 Servidores Web.....	25
1.3.2.1 Internet Information Server.....	25
1.3.2.3 ¿Por qué Apache?.....	27
1.3.3 Sistemas gestores de base de datos.....	27
1.3.3.1 MySQL.....	29
1.3.3.2 Oracle.....	29
1.3.3.3 Microsoft SQL Server.....	29
1.3.3.4 PostgreSQL.....	30
1.3.3.5 ¿Por qué PostgreSQL?	32
1.3.4 Metodologías de Desarrollo de Software.....	33

1.3.4.1 RUP.....	33
1.3.4.2 XP.....	35
1.3.4.3 ¿Por qué XP?	43
1.3.5 Servicios Web.....	43
1.3.6 Frameworks.....	44
1.3.6.1 Prado.....	45
1.3.6.2 Cakephp.....	45
1.3.6.3 Symfony.....	46
1.3.6.4 ¿Por qué Symfony?.....	47
1.3.7 Arquitectura.....	47
1.3.7.1 N Capas.....	47
1.3.7.2 Modelo Vista Controlador (MVC).....	48
1.3.7.3 Arquitectura Seleccionada.....	48
1.4 Conclusiones.....	49
Capítulo 2 Características del sistema.....	50
2.1 Introducción.....	50
2.2 Descripción de los procesos vinculados al campo de acción.....	50
2.2.1 Flujo actual del proceso general.....	50
2.2.2 Objeto de automatización.....	51
2.3 Propuesta del sistema.....	51
2.3.1 Personas relacionadas con el sistema.....	52
2.3.2 Requerimientos funcionales del sistema.....	52
2.3.3 Requerimientos no funcionales del sistema.....	53
2.4 Conclusiones.....	55
Capítulo 3 Exploración y Planificación.....	56
3.1 Introducción.....	56

3.2 Fase de Exploración.....	56
3.2.1 Historias de usuario.....	56
3.3 Fase de Planificación.....	61
3.3.1 Estimación de esfuerzos por historias de usuario.....	61
3.3.2 Plan de iteraciones.....	62
3.3.3 Plan de duración de las iteraciones.....	63
3.3.4 Plan de Entregas.....	64
3.4 Conclusiones	66
Capítulo 4Diseño, Implementación y Pruebas.	67
4.1 Introducción.....	67
4.2 Diseño del Sistema.....	67
4.2.1 Diseño de la Base de Datos.....	70
4.3 Codificación.....	72
4.3.1 Iteración 1.....	72
4.3.2 Iteración 2.....	78
4.3.3 Iteración 3.....	81
4.3.4 Diagrama de Despliegue.....	85
4.4 Pruebas.....	86
4.4.1 Pruebas de Aceptación.....	86
4.5 Conclusiones.....	114
Conclusiones.	115
Recomendaciones	116
Referencias Bibliográficas.....	117
Bibliografía.....	119

Índice de Tablas.

Tabla 1 Personas relacionadas con el sistema.	52
Tabla 2 HU Gestionar Postgrado.	56
Tabla 3 HU Mostrar Postgrados.	56
Tabla 4 HU Gestionar Actividad.....	57
Tabla 5 HU Mostrar Actividades.....	57
Tabla 6 HU Gestionar Curso.	57
Tabla 7 HU Mostrar Cursos.	57
Tabla 8 HU Gestionar Beca.....	58
Tabla 9 HU Mostrar Becas.....	58
Tabla 10 HU Gestionar Diplomado.	58
Tabla 11 HU Mostrar Diplomados.	58
Tabla 12 HU Gestionar Profesor.....	58
Tabla 13 HU Gestionar Polo Productivo.....	59
Tabla 14 HU Gestionar Plan de Superación.	59
Tabla 15 HU Mostrar Plan de Superación.	59
Tabla 16 HU Autenticar.....	59
Tabla 17 HU Guardar Interés del Postgrado.	59
Tabla 18 HU Crear Matrícula de Postgrado.....	60
Tabla 19 HU Publicar Trabajos.	60
Tabla 20 HU Ver Listado de Profesores.....	60
Tabla 21 HU Gestionar Comentario.....	60
Tabla 22 Estimación de esfuerzos por HU.....	61
Tabla 23 Plan de duración de las Iteraciones.	63
Tabla 24 Módulos y HU Asociadas.....	64
Tabla 25 Plan de entregas.....	64

Tabla 26 Tarjeta CRC Módulo admin.	67
Tabla 27 Tarjeta CRC Módulo prof.	68
Tabla 28 Tarjeta CRC Módulo post.	68
Tabla 29 Tarjeta CRC Módulo act.	68
Tabla 30 Tarjeta CRC Módulo polo.	69
Tabla 31 Tarjeta CRC Módulo cursos.	69
Tabla 32 Tarjeta CRC Módulo becas.	69
Tabla 33 Tarjeta CRC Módulo dipl.	69
Tabla 34 Módulos abordados en la Iteración 1.	72
Tabla 35 Tarea 1 del módulo post.	73
Tabla 36 Tarea 2 del módulo post.	73
Tabla 37 Tarea 3 del módulo post.	73
Tabla 38 Tarea 1 del módulo act.	74
Tabla 39 Tarea 2 del módulo act.	74
Tabla 40 Tarea 3 del módulo act.	74
Tabla 41 Tarea 1 del módulo cursos.	75
Tabla 42 Tarea 2 del módulo cursos.	75
Tabla 43 Tarea 3 del módulo cursos.	75
Tabla 44 Tarea 1 del módulo becas.	76
Tabla 45 Tarea 2 del módulo becas.	76
Tabla 46 Tarea 3 del módulo becas.	76
Tabla 47 Tarea 1 del módulo dipl.	77
Tabla 48 Tarea 2 del módulo dipl.	77
Tabla 49 Tarea 3 del módulo dipl.	77
Tabla 50 Tarea 1 del módulo polo.	78
Tabla 51 Tarea 2 del módulo polo.	78

Tabla 52 Módulos abordados en la Iteración 2.	78
Tabla 53 Tarea 1 del módulo prof.	79
Tabla 54 Tarea 2 del módulo prof.	79
Tabla 55 Tarea 3 del módulo prof.	79
Tabla 56 Tarea 1 del módulo admin.	80
Tabla 57 Tarea 2 del módulo admin.	80
Tabla 58 Tarea 3 del módulo admin.	81
Tabla 59 Tarea 4 del módulo admin.	81
Tabla 60 Módulos abordados en la Iteración 3.	82
Tabla 61 Tarea 5 del módulo admin.	82
Tabla 62 Tarea 6 del módulo admin.	83
Tabla 63 Tarea 7 del módulo admin.	83
Tabla 64 Tarea 8 del módulo admin.	83
Tabla 65 Tarea 9 del módulo admin.	84
Tabla 66 Tarea 4 del módulo prof.	84
Tabla 67 Tarea 5 del módulo prof.	84
Tabla 68 Prueba 1 al módulo post.	86
Tabla 69 Prueba 2 al módulo post.	87
Tabla 70 Prueba 3 al módulo post.	87
Tabla 71 Prueba 4 al módulo post.	88
Tabla 72 Prueba 5 al módulo post.	88
Tabla 73 Prueba 6 al módulo post.	89
Tabla 74 Prueba 1 al módulo act.	89
Tabla 75 Prueba 2 al módulo act.	90
Tabla 76 Prueba 3 al módulo act.	90
Tabla 77 Prueba 4 al módulo act.	91

Tabla 78 Prueba 5 al módulo act.	91
Tabla 79 Prueba 6 al módulo act.	92
Tabla 80 Prueba 1 al módulo cursos.	92
Tabla 81 Prueba 2 al módulo cursos.	93
Tabla 82 Prueba 3 al módulo cursos.	93
Tabla 83 Prueba 4 al módulo cursos.	94
Tabla 84 Prueba 5 al módulo cursos.	94
Tabla 85 Prueba 6 al módulo cursos.	95
Tabla 86 Prueba 1 al módulo becas.	95
Tabla 87 Prueba 2 al módulo becas.	96
Tabla 88 Prueba 3 al módulo becas.	96
Tabla 89 Prueba 4 al módulo becas.	97
Tabla 90 Prueba 5 al módulo becas.	97
Tabla 91 Prueba 6 al módulo becas.	98
Tabla 92 Prueba 1 al módulo dipl.	98
Tabla 93 Prueba 2 al módulo dipl.	99
Tabla 94 Prueba 3 al módulo dipl.	99
Tabla 95 Prueba 4 al módulo dipl.	100
Tabla 96 Prueba 5 al módulo dipl.	100
Tabla 97 Prueba 6 al módulo dipl.	101
Tabla 98 Prueba 1 al módulo polo.	101
Tabla 99 Prueba 2 al módulo polo.	102
Tabla 100 Prueba 3 al módulo polo.	102
Tabla 101 Prueba 4 al módulo polo.	103
Tabla 102 Prueba 5 al módulo polo.	103
Tabla 103 Prueba 1 al módulo prof.	104

Tabla 104 Prueba 2 al módulo prof.	104
Tabla 105 Prueba 3 al módulo prof.	105
Tabla 106 Prueba 4 al módulo prof.	105
Tabla 107 Prueba 5 al módulo prof.	106
Tabla 108 Prueba 1 al módulo admin.	106
Tabla 109 Prueba 2 al módulo admin.	107
Tabla 110 Prueba 3 al módulo admin.	107
Tabla 111 Prueba 4 al módulo admin.	108
Tabla 112 Prueba 5 al módulo admin.	108
Tabla 113 Prueba 6 al módulo admin.	109
Tabla 114 Prueba 7 al módulo admin.	109
Tabla 115 Prueba 8 al módulo admin.	110
Tabla 116 Prueba 9 al módulo admin.	110
Tabla 117 Prueba 10 al módulo admin.	111
Tabla 118 Prueba 11 al módulo admin.	111
Tabla 119 Prueba 6 al módulo prof.	112
Tabla 120 Prueba 7 al módulo prof.	112
Tabla 121 Prueba 8 al módulo prof.	113

Introducción.

En la era de la información, de la explosión de sus tecnologías, se vive la etapa en la que la humanidad ha alcanzado un desarrollo imprevisible; cada día son mayores las diferencias sociales, políticas y económicas. Se habla constantemente sobre la sociedad de la información, es visible el paso de las sociedades industriales a las posindustriales y del conocimiento, donde el factor esencial de progreso es el conocimiento. Esta nueva sociedad, con organizaciones basadas en el aprendizaje, cuyo capital máspreciado es el ser humano, se sustenta en un desarrollo tecnológico sin precedentes, es el punto en el cual las grandes compañías planifican sus productos en función de la gestión del conocimiento y de la viabilidad para su obtención.

El poder del conocimiento y disponibilidad de información representan elementos necesarios para desarrollar cualquier actividad en cualquier institución. En la UCI la necesidad de información es constante, lo que determina el surgimiento casi a diario de sistemas automatizados para la gestión y transmisión de la misma.

Situación Problemática

En la Facultad 8 de la Universidad de las Ciencias Informáticas la gestión de la información de postgrado se hace de forma manual, lo que provoca un gran consumo del tiempo de las personas encargadas de dicha actividad y hace demasiado complejo el sistema de control. Esta forma en muchos casos no satisface los requerimientos asociados, es decir, no cumple con los requisitos de seguridad y control, lo que provoca baja confiabilidad y dispersión en el flujo de la información, con la consecuente falta de conocimiento de los profesores acerca de temas referentes a su superación. De dicha situación se puede identificar el **problema** que consiste en que no existe un sistema que gestione de forma eficiente la información de postgrado de los profesores en la Facultad 8 de la UCI, ocasionando retrasos en las búsquedas y obtención de reportes especializados, así como pérdidas de datos. El **objeto de estudio** es el Proceso de gestión de postgrado en la UCI, teniendo un **campo de acción** dirigido al Proceso de gestión de la información de superación postgraduada en la facultad 8.

Para guiar este proceso se plantea como **Idea a Defender**: si se realiza el análisis, diseño e implementación de un sistema que permita la gestión de la información de postgrado perteneciente a la Facultad 8 en la UCI, se logrará un control eficiente por parte de la persona responsable de esta tarea en la Facultad 8.

Con el fin de darle solución a la problemática antes mencionada se define como **objetivo general**: realizar un sistema que gestione la información de la superación

postgraduada en la Facultad 8 de la UCI, del cual se derivan los siguientes **Objetivos Específicos:**

- ❖ Realizar ingeniería de requerimientos a los procesos de Gestión de la Información de Formación Postgraduada en la Facultad 8 y en la UCI de forma general.
- ❖ Realizar el análisis, diseño, implementación del sistema propuesto.
- ❖ Realizar pruebas al sistema.

Para cumplir con los objetivos precedentes se proponen las siguientes **tareas de investigación:**

- ❖ Aplicar técnicas de investigación para conocer todos los procesos que tienen lugar en el área de producción que se relacionan de alguna forma con la superación de los profesores y cómo se realizan estos procesos.
- ❖ Investigar sobre la existencia de otros sistemas similares, que aporten información o servicios para la realización de la herramienta.
- ❖ Estudiar las tendencias y tecnologías actuales para el desarrollo de sistemas similares.
- ❖ Realizar la selección de las herramientas más apropiadas para el desarrollo del sistema, así como del lenguaje y metodología a utilizar.
- ❖ Realizar el análisis, diseño e implementación del sistema.
- ❖ Realizarle pruebas al sistema.
- ❖ Elaborar la documentación del sistema.

Con este trabajo se espera que los profesores de la facultad 8 tengan una aplicación donde puedan consultar su situación en cuanto a superación postgraduada sin necesidad de utilizar a terceras personas, puedan expresar su interés en algún postgrado que se esté ofertando y que el asesor de postgrado de la facultad 8 pueda obtener los reportes que le exige la secretaría de postgrado de la universidad de una forma mucho más rápida y eficiente.

El presente trabajo está estructurado en 4 capítulos, de los cuales a continuación se da una breve descripción:

El **Capítulo1** contiene la fundamentación teórica del trabajo, abordando aspectos importantes de la gestión académica, algunos sistemas similares que se utilizan en nuestro país, se hace un análisis de las tendencias, las tecnologías y las metodologías que se utilizan en la actualidad para el desarrollo de aplicaciones web y se seleccionan cuáles de estas son más apropiadas para el desarrollo de la aplicación.

En el **Capítulo 2** se hace una propuesta del sistema a implementar, describiendo los procesos vinculados al campo de acción, definiendo los requisitos funcionales y no funcionales y las personas que se relacionan con el sistema.

En el **Capítulo 3** se detallan los artefactos generados en la fase de “Planificación” mostrando especial atención en las historias de usuario.

En el **Capítulo 4** se detallan los artefactos generados en las fases de “Diseño”, “Implementación” y “Pruebas”, poniendo gran atención en las tarjetas CRC, las tareas de implementación y las pruebas de aceptación realizadas a pedido del cliente.

Capítulo 1

Fundamentación teórica.

1.1 Introducción.

En este capítulo se brinda un panorama general de los aspectos relacionados con la formación postgraduada en la UCI y en la facultad 8 de manera específica. Además se hace referencia a las tecnologías, lenguajes de programación y herramientas utilizadas en el ámbito internacional para el desarrollo de aplicaciones Web, permitiendo realizar una selección correcta que se adapte a las necesidades del sistema propuesto, lográndose identificar el lenguaje de programación, el sistema gestor de base de datos, la metodología de desarrollo adecuada y las herramientas a utilizar en el desarrollo del sistema.

1.2 Estado del arte.

1.2.1 ¿Qué es la gestión académica?

La gestión académica se origina en las secretarías docentes de los centros educacionales, estas tienen la misión de conducir toda la actividad docente del alumno, desde que ingresa hasta que sale del centro, son las encargadas de organizar y planificar cómo se va a desarrollar el proceso docente de los estudiantes y archivar los resultados de los mismos en este proceso.

A continuación se hará referencia a algunos sistemas de gestión académica que se utilizan en Cuba.

1.2.2 GESTACAD.

Sistema creado por un grupo de jóvenes desarrolladores en un acercamiento a la solución del problema de la gestión de la información docente en las instituciones de la Educación Superior cubana. La versión en ejecución incluye un software cliente elaborado sobre el lenguaje de programación Borland Delphi, que tiene acceso a una base de datos construida sobre InterBase Firebird Server. Sus datos se muestran en la INTRANET de la Universidad de Matanzas en formato Web. Utiliza el lenguaje Server Side PHP para la elaboración de las consultas Web a la Base de Datos.

El sistema brinda funcionalidades que responden a la búsqueda de un alumno, proporciona listados de estudiantes por grupo, reportes dinámicos de la información existente, reporte de notas por asignatura y grupo, tabla con los resultados docentes de un grupo en un semestre y de un estudiante en toda su carrera. Además, genera actas de exámenes de las diferentes asignaturas, un registro de características de un grupo de estudiantes y permite dar Baja a un estudiante. (1)

1.2.3 Akademos.

Es un sistema Web desarrollado en nuestra universidad. Brinda una interfaz común para todos sus usuarios. Además, ofrece la información actualizada en tiempo real de las facultades de la universidad. Akademos realiza la gestión de toda la información referente a la formación de pregrado de un estudiante. Además posee un módulo para profesores donde estos pueden controlar las evaluaciones de sus estudiantes y la asistencia de estos a clases.

Ninguno de estos sistemas cumple las expectativas del cliente ya que lo que este desea no es un sistema que solo gestione la información que resulta de un proceso docente si no que sirva

para brindar información actualizada a los profesores acerca de su situación postgraduada, además de automatizar el proceso de matrícula de los profesores a los diferentes posgrados.

1.3 Tendencias y tecnologías actuales.

Es muy importante para los productores de software estar al tanto de todo lo que ocurre en el mundo que tenga que ver con el desarrollo actual de soluciones informáticas. Se hace muy necesario conocer las ventajas y posibilidades que brindan las tecnologías a utilizar para la realización de un producto.

En este momento existe una marcada tendencia hacia la Web, no sólo por los costos reducidos que esto implica, sino por el alto nivel de flexibilidad y agilidad que permite en la implementación de un sistema. El acceso a una página web es independiente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios. Por esta razón, en vez de crear aplicaciones para Windows, GNU/Linux, u otros Sistemas Operativos, la aplicación es escrita sólo una vez y mostrada a todos los clientes a través del acceso a un servidor.

En las siguientes secciones se describen las principales tecnologías y herramientas que se utilizan en la actualidad para la construcción de aplicaciones web llegando a la conclusión de cuáles serían más factibles a utilizar para el desarrollo de la solución propuesta.

1.3.1 Lenguajes y plataformas de programación para la Web.

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial. Para realizar una página con la programación Web, se deben tener claros tres conceptos fundamentales, los cuales son: el URL (Uniform Resource Locators), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página Web, u otro servicio. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de ella un archivo que indica el camino al recurso que se solicita. El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de internet, por ejemplo HTTP (Hypertext Transfer Protocol). Y por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página Web, este lenguaje es el HTML (Hypertext Markup Language).

En la programación Web, el HTML es el lenguaje que permite codificar o preparar documentos de hipertexto, que es el lenguaje común para la construcción de una página Web.

A continuación se presentan varios lenguajes de programación que se utilizan en la actualidad para la construcción de sitios Web. (2)

1.3.1.1 JSP.

Tecnología destinada a desarrollar páginas Web en Java. Independiente a la plataforma, se ejecuta del lado del servidor muy similar a ASP.Net en la potencia de los desarrollos. Permite separar el código dinámico del estático y tiene como característica que las páginas se compilan en la primera petición y se puede incrustar en código HTML. La extensión del archivo es (.jsp). (3)

Ventajas

- ❖ Ejecución rápida de servlets (Aplicación que opera como un módulo en un servidor).
- ❖ Multiplataforma.
- ❖ Código bien estructurado.
- ❖ Integridad con los módulos de Java.

1.3.1.2 ASP.NET.

ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado para construir sitios web dinámicos, aplicaciones y servicios web. Fue liberado en enero del 2002 con la versión 1.0 del .NET Framework, sustituyendo a la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime (Entorno Común de Ejecución de Lenguajes), permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework (VB.net, C# y JScript.net.). ASP.NET introduce el concepto del code-behind, por el que una misma página se compone de dos ficheros: el de la interfaz de usuario y el de código. Con ello se facilita la programación de aplicaciones en múltiples capas. Los Web Forms o formularios web, son la parte principal de ASP.NET, permiten entre otras cosas, separar la interfaz del usuario de la funcionalidad de la aplicación.

1.3.1.3 Python.

Es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Python es un lenguaje de scripting. Es interpretado, por lo que no es necesario realizar la compilación del código fuente para ejecutarlo. Al mismo tiempo es gratuito (incluso para propósitos empresariales), independiente de plataforma y orientado a

objetos. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Se encuentra preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso páginas Web. En los últimos años este lenguaje se ha hecho muy popular debido a la cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero. Otras de las características que han influido en el auge de Python son su sencillez y velocidad para crear los programas. (7)

Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten un estilo de programación particular:

- ❖ Programación orientada a objetos.
- ❖ Programación estructurada.
- ❖ Programación funcional.
- ❖ Programación orientada a aspectos.

Ventajas:

- ❖ Libre y fuente abierta.
- ❖ Lenguaje de propósito general.
- ❖ Gran cantidad de funciones y librerías.
- ❖ Sencillo y rápido de programar.
- ❖ Multiplataforma.
- ❖ Licencia de código abierto (Opensource).
- ❖ Orientado a Objetos.
- ❖ Portable.

1.3.1.4 PHP.

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Surgió en 1995, desarrollado por PHP Group.

PHP se escribe dentro del código HTML, lo que posibilita que sea realmente fácil de utilizar, al igual que ocurre con ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad. Se puede descargar a través de la página principal de PHP www.php.net y de manera gratuita, un módulo que hace que cualquier servidor web comprenda los scripts realizados en este lenguaje. Es independiente de

plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo. En el caso de estar instalado sobre un servidor Linux o Unix, es más rápido que lenguajes como ASP. Es importante señalar la seguridad, en este punto también es relevante el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como más veloces. Además, PHP permite configurar el servidor de modo que se acepten o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Ventajas:

- ❖ Muy fácil de aprender.
- ❖ Se caracteriza por ser un lenguaje muy rápido.
- ❖ Soporta la orientación a objeto. Clases y herencia.
- ❖ Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- ❖ Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- ❖ Capacidad de expandir su potencial utilizando módulos.
- ❖ Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- ❖ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ❖ Incluye gran cantidad de funciones.
- ❖ No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI (Common Gateway Interface, Interfaz Común de Intercomunicación) que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

1.3.1.5 ¿Por qué PHP?

El estudio de los lenguajes de programación más usados en la actualidad ha demostrado que PHP es la mejor opción y la que más se adecua a las necesidades del sistema a implementar por su rapidez y soporte a la Programación Orientada a Objetos. Además de ser libre lo que es un aspecto muy importante, existe una amplia documentación por lo que resulta fácil de aprender. Es un lenguaje multiplataforma que soporta la integración con la mayoría de los gestores de base de datos

1.3.2 Servidores Web.

Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol), perteneciente a la capa de aplicación del modelo OSI y diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML. Es un programa que se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente a través de un navegador web y que responde a estas peticiones adecuadamente, mostrando la información requerida en una página web que se exhibirá en el navegador o el respectivo mensaje si se detectó algún error. Los servidores web más usados en la actualidad son Internet Information Server (IIS) y Apache.

1.3.2.1 Internet Information Server.

Internet Information Server, desarrollado por Microsoft, constituye una serie de servicios para los ordenadores que funcionan con sistema operativo Windows lo cual convierte un ordenador en un servidor de Internet o Intranet. Esto no es más, que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (Servidor Web). Es rápido y recomendable para la plataforma Windows 2000, dado por la integración que presenta con su Servicio de Directorios que permite el desarrollo de aplicaciones basadas en la Web fiables y escalables.

IIS 5.0 ofrece una administración muy sencilla que se realizará mediante el Administrador de servicios de Internet. La versión 5.0 de IIS permite que el desarrollo de aplicaciones Web sea mucho más robusto y la creación de sitios Web sea más configurable y completa. Ofrece un entorno escalable basado en los componentes cliente/servidor que se pueden integrar dentro de las aplicaciones Web. IIS 5.0 también es capaz de impedir que aquellos usuarios con direcciones Internet Protocol (IP) conocidas obtengan acceso no autorizado al servidor, permitiendo especificar la información apropiada en una lista de restricciones. Una de las formas que tiene IIS de asegurar los datos es mediante Secure Sockets Layer (SSL). Este

método proporciona un método para transferir datos entre el cliente y el servidor de forma segura, permitiendo también que el servidor pueda comprobar al cliente antes de que inicie una sesión de usuario. Otra característica nueva es la autenticación implícita que permite a los administradores autenticar a los usuarios de forma segura a través de servidores de seguridad y proxy. (4)

1.3.2.2 Apache.

Apache, es el servidor Web por excelencia. Presenta características que hacen que cada vez millones de desarrolladores aumenten su confianza en este programa, destacándose: su configurabilidad, robustez y estabilidad. Provee un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Es una tecnología gratuita, de código fuente abierto, que posee una licencia descendiente de la licencias BSD (Distribución de Software Berkeley), la cual permite hacer lo que se quiera con el código fuente, siempre que se les reconozca su trabajo. A continuación se enuncian algunas características de este servidor. (8)

- ❖ Corre sobre todos los sistemas operativos.
- ❖ Facilita la integración como "plugins" de lenguajes de programación de páginas Web dinámicas.
- ❖ Tiene integración en estándar del protocolo de seguridad SSL.
- ❖ Provee interfaz a todas las bases de datos.
- ❖ Tiene una alta configurabilidad en la creación y gestión de logs, ya que permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.
- ❖ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.
- ❖ Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades de este servidor, ya que se pueden escribir módulos para realizar determinadas funciones, lo que implica que haya gran cantidad de ellos disponibles para su utilización.

El Apache posee varios módulos, los cuales pueden clasificarse en tres categorías:

- ❖ Módulos Base: Módulo con las funciones básicas del Apache.
- ❖ Módulos Multiproceso (para manejar las peticiones): Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a

atender a las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y la rapidez del código.

- ❖ **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor. Simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

1.3.2.3 ¿Por qué Apache?

El estudio de los servidores web más usados en la actualidad ha demostrado que Apache es la mejor opción y la que más se adecua a las necesidades del sistema a implementar. Además de su alta configurabilidad y las posibilidades que brinda para que el usuario pueda ampliar sus capacidades, este servidor corre o es compatible con todos los sistemas operativos y provee una interfaz a todas las bases de datos. Indiscutiblemente es el más usado y el que más potencialidades brinda. (9)

1.3.3 Sistemas gestores de base de datos.

Los sistemas de gestión de base de datos (SGBD); (en inglés: DataBase Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos y las aplicaciones que la utilizan. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de estos.

Existen distintos objetivos que deben cumplir los SGBD, dentro de ellos se encuentran:

- ❖ **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- ❖ **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- ❖ **Consistencia.** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea. Por otra parte, la base de datos representa una realidad determinada que

tiene determinadas condiciones, por ejemplo que los menores de edad no pueden tener licencia de conducir. El sistema no debería aceptar datos de un conductor menor de edad. En los SGBD existen herramientas que facilitan la programación de este tipo de condiciones.

- ❖ Seguridad. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- ❖ Integridad. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada. Los SGBD proveen mecanismos para garantizar la recuperación de la base de datos hasta un estado consistente (ver Consistencia, más arriba) conocido en forma automática.
- ❖ Respaldo. Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- ❖ Control de la concurrencia. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
- ❖ Manejo de Transacciones. Una Transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que el estado luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado. Los SGBD proveen mecanismos para programar las modificaciones de los datos de una forma mucho más simple que si no se dispusiera de ellos.
- ❖ Tiempo de respuesta. Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados. (10)

1.3.3.1 MySQL.

MySQL es un sistema de gestión de base de datos desarrollado por MySQL AB (compañía de software) como software libre en un esquema de licenciamiento dual. MySQL AB fundada por David Axmark, Allan Larsson y Michael Widenius en 1995 en Suecia, pertenece a Sun Microsystems desde enero de 2008, quien posee el copyright de la mayor parte del código. Por un lado se ofrece bajo la GNU GPL (Licencia Pública General de GNU) para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden comprar una licencia específica que les permita este uso. MySQL es muy utilizado en aplicaciones Web como MediaWiki o Drupal, en múltiples plataformas, como Linux, Windows y Solaris y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. (11)

1.3.3.2 Oracle.

Oracle es básicamente una herramienta que se basa en la tecnología cliente/servidor para la gestión de bases de datos. Surgió entre finales de los años 70 y principio de los años 80, desarrollado por la compañía "Oracle Corporación" y conocida entonces como "Relational Software". Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- ❖ Soporte de transacciones.
- ❖ Estabilidad.
- ❖ Escalabilidad.
- ❖ Es multiplataforma.

En la actualidad es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas Web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos.

1.3.3.3 Microsoft SQL Server.

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea así como de tener unas ventajas que seguidamente se describen.

- ❖ Soporte de transacciones.

- ❖ Escalabilidad, estabilidad y seguridad.
- ❖ Soporte de procedimientos almacenados.
- ❖ Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL (lenguaje de definición de datos) y DML (lenguaje de manipulación de datos) gráficamente.
- ❖ Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- ❖ Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

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

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows. (12)

1.3.3.4 PostgreSQL.

PostgreSQL es uno de los Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) de código abierto más avanzada del mundo. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation. En 1986 otro equipo dirigido por Michael Stonebraker de Berkeley continuó el desarrollo del código de Ingres para crear un sistema de bases de datos objeto-relacionales llamado Postgres. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, Postgres fue renombrado a PostgreSQL, tras un breve periplo como Postgres95. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

PostgreSQL posee muchas características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente es una breve lista de algunas de esas características, a partir de PostgreSQL 7.1.x.

- ❖ **DBMS Objeto-Relacional:** PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia, y arreglos.
- ❖ **Altamente Extensible:** PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.
- ❖ **Soporte SQL Comprensivo:** PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- ❖ **Integridad Referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- ❖ **API (interfaz de programación de aplicaciones) Flexible:** La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.
- ❖ **Lenguajes Procedurales:** PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- ❖ **MVCC:** MVCC(Control de Concurrencia Multi-Versión Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.
- ❖ **Cliente/Servidor:** PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- ❖ **Write Ahead Logging (WAL):** La característica de PostgreSQL conocida como “Write Ahead Logging” incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos. (13)

1.3.3.5 ¿Por qué PostgreSQL?

El estudio de los sistemas gestores de base de datos más usados en la actualidad ha demostrado que PostgreSQL es la mejor opción y la que más se adecua a las necesidades del sistema a implementar. Por las siguientes características se ha decidido utilizar PostgreSQL como sistema gestor de base de datos para desarrollar nuestro sistema.

- ❖ Altamente Extensible: PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario.
- ❖ Soporte SQL Comprensivo: PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- ❖ Integridad Referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- ❖ API Flexible: La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.
- ❖ Lenguajes Procedurales: PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- ❖ Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
- ❖ Write Ahead Logging (WAL): La característica de PostgreSQL conocida como "Write Ahead Logging" incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos.

1.3.4 Metodologías de Desarrollo de Software.

La calidad en el desarrollo y mantenimiento de los sistemas informáticos se ha convertido en uno de los principales objetivos de las organizaciones, debido a que cada vez más los procesos principales dependen de sistemas informáticos para su buen funcionamiento.

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación que los desarrolladores de sistemas informáticos deben seguir para garantizar la calidad de un producto informático. La metodología que va a guiar el proceso de desarrollo de un software es un paso muy importante ya que esta permite definir los pasos a seguir durante todo el ciclo de desarrollo. En la actualidad existen diversas metodologías de desarrollo con particularidades específicas, entre las más conocidas están: RUP (Rational Unified Process) y XP (Extreme Programming). (14)

1.3.4.1 RUP.

Rational Unified Process es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software. Unifica completamente a un equipo de desarrollo de software y optimiza la productividad de cada uno de los miembros del equipo brindándoles la experiencia de los líderes de la industria y las lecciones aprendidas a través de miles de proyectos. Está fundamentada en un enfoque orientado a modelos de desarrollo basado en componentes, utilizando para ello el Lenguaje de Modelado Unificado (UML, Unified Modeling Language en inglés) el que define técnicas de análisis y diseño que ayudan a la confección de una solución sólida de software. RUP se caracteriza por ser **dirigido por casos de uso** donde los casos de uso definen lo que el usuario desea a partir de la captura de requisitos y la modelación del negocio. Es (Software, 2007) **la arquitectura**, característica que brinda una visión completa del sistema, se describen los procesos del negocio que son más importantes, para comprenderlo, desarrollarlo y producirlo de una forma eficaz. **Iterativo e Incremental** donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo Un proyecto realizado siguiendo RUP se divide en cuatro fases:

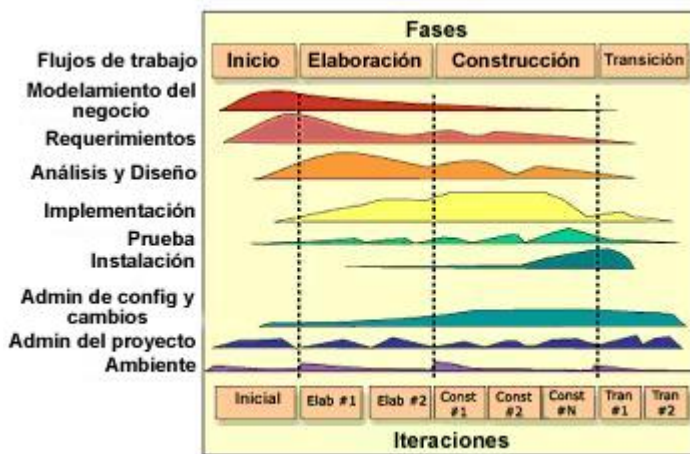
- ❖ Inicio.
- ❖ Elaboración.
- ❖ Construcción.
- ❖ Transición.

En cada fase se ejecutarán una o varias iteraciones de tamaño variable según el proyecto, y dentro de cada una de ellas seguirá un modelo de cascada para los flujos de trabajo. RUP define nueve disciplinas a realizar en cada ciclo del proyecto, donde seis de ellas son flujos de trabajo básicos (Modelamiento del negocio, Requerimientos, Análisis y diseño, Implementación, Prueba e Instalación) y las otras disciplinas de soporte (Administración del proyecto, Administración de configuración y cambios y Ambiente).

Flujos de trabajo:

- ❖ Modelado del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- ❖ Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ❖ Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ❖ Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ❖ Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.
- ❖ Instalación: Produce un entregable del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- ❖ Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ❖ Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. (5)



1.3.4.2 XP.

Extreme Programming es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

Intenta reducir la complejidad del software por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debería estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones, de ahí lo de competente.

El ciclo de trabajo de esta metodología consta de 4 fases de trabajo; Planificación en la cual la tarea más importante es la creación de las Historias de Usuario (HU) las cuales son escritas por el cliente para expresar las funcionalidades que desea y son usadas para estimar el tiempo de desarrollo de la funcionalidad que describen, Diseño en esta fase se crean las Tarjetas CRC las cuales definen una clase expresando las funcionalidades de esta y las otras clases con las que colabora, Codificación donde se definen las tareas de desarrollo para que los

desarrolladores tengan una guía para implementar todas las HU y Pruebas en esta fase se le realizan diferentes test a cada una de las HU para probar que cumplen con las funcionalidades que desea el cliente.

Las principales características de XP son:

- ❖ Desarrollo iterativo e incremental.
- ❖ Pruebas unitarias continuas, frecuentemente repetidas. Se aconseja escribir el código de la prueba antes de la codificación.
- ❖ Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos desarrolladores en un mismo puesto.
- ❖ Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- ❖ Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad sin modificar su comportamiento.
- ❖ Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, promueve que todo el personal pueda corregir y extender cualquier parte del proyecto.

XP se basa en cinco valores, la raíz de los elementos básicos que, a juicio de Beck son los realmente importantes para el éxito de desarrollo de software. Estos valores son la orientación para el propio desarrollo y la inspiración de toda la metodología. Cuatro de ellos son los mismos que XP original, y se añade el respeto como el quinto valor.

Valores

- ❖ Comunicación: la mayoría de los problemas y los errores son causados por la falta de comunicación. Por esta razón, debe existir una gran comunicación entre los miembros del equipo y entre el equipo y los clientes. Además, la forma más eficaz de lograr esto es la comunicación es directa, persona a persona. También la comunicación entre los objetos y las personas que los leen debe ser fácilmente legible y actualizada.
- ❖ Simplicidad: este es el más intelectual de los valores XP. "Las cosas más simples en que podría trabajar". Sin embargo, la simplicidad -no simplista- es muy difícil. Se requiere experiencias, ideas y trabajo duro. Simplicidad favorece la comunicación, reduce la cantidad de código y mejora la calidad. Por otra parte, la simplificación de un sistema, hace más fácil es obtener información sobre él.

- ❖ Comentarios: siempre debe ser capaz de medir el sistema y saber hasta qué punto está terminado a partir de las características necesarias. Los instrumentos son fundamentales comentario estrecho contacto con el cliente y disponibilidad de un conjunto de pruebas automatizadas, que se desarrollan con el propio sistema.
- ❖ Valor: todos los procesos y las metodologías son herramientas para manejar y reducir nuestros miedos. Es más el miedo que tenemos para un proyecto de software, que la más grande y más pesada de las metodologías que necesitamos. Valor por sí solo es peligroso, pero con otros valores es una herramienta de gran alcance para hacer frente a los cambios.
- ❖ Respeto: Si los miembros de un equipo no se preocupan por sí y su trabajo, la metodología no puede funcionar. Usted debe ser respetuoso con sus colegas y sus contribuciones a su organización, así como a personas cuya vida se toca por el sistema que está desarrollando.

Estos cinco valores no dan consejos específicos sobre cómo gestionar un proyecto, o cómo escribir programas. Para este fin, son necesarias prácticas, y antes de las prácticas, se necesitan principios. Los principios son el puente entre los valores, que es lo sintético y abstracto, y las prácticas, que dicen en realidad cómo desarrollar software.

Principios

- ❖ La humanidad: el software es desarrollado por personas, por lo que los factores humanos son la principal clave para ofrecer la calidad del software. XP tiene por objeto abordar los objetivos de las personas y sus organizaciones, por lo que pueden beneficiarse a ambos. Según Beck las necesidades de las personas son las siguientes:
 - Básicas de seguridad - la necesidad de mantener el puesto de trabajo.
 - Realización - el sentido de utilidad para su propio trabajo.
 - Pertenencia - la capacidad de identificarse con un grupo.
 - Crecimiento - la oportunidad de ampliar sus conocimientos y perspectivas.
 - Intimidad - la capacidad de entender y hacerse entender por los demás.
- ❖ Economía: si se produce el software, también debe producir valor de negocio. Dos aspectos de la economía son clave para XP; valor actual y el valor de las opciones. La primera dice que un dólar hoy vale más que un dólar de mañana, por lo que cuanto

antes se desarrolle el software, antes se gana el dinero, más los beneficios que el software crea.

- ❖ **Beneficio mutuo:** cada actividad debe beneficiar a todas las personas y organizaciones interesadas. Esto es quizás el más importante principio de XP, y de los más difíciles de adherir. Siempre hay soluciones fáciles a los problemas, cuando alguien gana y otros pierden. A menudo, estas soluciones son tentadores atajos. Sin embargo, son siempre una pérdida neta, ya que derriban relaciones de trabajo. Por lo tanto, necesita prácticas que beneficien tanto a usted como a su cliente, ahora y en el futuro.
- ❖ **Auto-Similitud:** la naturaleza continuamente utiliza estructuras fractales, que son similares a sí mismos en diversas escalas. El mismo principio debería aplicarse al desarrollo de software, volver a utilizar soluciones similares, en diferentes contextos.
- ❖ **Mejora:** la mejora continua es fundamental para XP. La perfección no existe, pero debemos esforzarnos para la perfección. Cada día debemos esforzarnos por actuar mejor. En la práctica, en cada iteración el sistema es mejorado tanto en calidad como en funcionalidades, utilizando la retroalimentación del cliente.
- ❖ **Diversidad:** los equipos donde todos por igual son cómodos, pero no eficaces. Los equipos deben incluir diferentes conocimientos, habilidades y personajes, para poder descubrir y resolver problemas. Por supuesto, ser diferente lleva a los posibles conflictos, que deben ser gestionados y resueltos. Con diferentes opiniones y proponer diferentes soluciones es muy útil en el desarrollo de software, siempre y cuando sean capaces de gestionar los conflictos y para elegir la mejor alternativa.
- ❖ **Reflexión:** un equipo de trabajo eficaz no sólo hacer su trabajo. Se preguntan cómo se trabaja, y por qué están trabajando de esa manera. Tienen que analizar las razones de éxito o fracaso sin ocultar los errores, tratando de aprender de ellos. Sin embargo, usted no debe pensar demasiado. La reflexión viene después de la acción, y antes de la próxima acción.
- ❖ **Fluidez:** flujo medio útil para desarrollar el software de manera constante, realizando todos juntos el desarrollo de actividades. XP asume practicar un flujo continuo de actividades, no una secuencia de diferentes fases, con versión del software sólo después de la última. Sólo permite el flujo continuo información para garantizar que el

sistema está evolucionando hacia la dirección correcta, y evitar problemas relacionados con la integración final "big-bang".

- ❖ Oportunidad: los problemas deben verse como una oportunidad de mejora. Usted debe experimentar problemas, pero para obtener la excelencia, no puede simplemente corregir los problemas. Necesita convertirlos en oportunidades de aprendizaje y mejora.
- ❖ Redundancia: los problemas críticos y difíciles que deben resolverse de varias maneras diferentes. Por lo tanto, si una solución falla, las demás deben prevenir el desastre. El costo de la redundancia se puede reembolsar en estos casos. Los defectos del software deben ser buscados de muchas maneras. Esto es redundante, porque muchos defectos se encuentran muchas veces. Sin embargo, la calidad no tiene precio.
- ❖ Error: aún si cree que no es capaz de tener éxito, por qué no intentarlo. ¿No sabe cómo poner en práctica una historia? Trate de aplicarlo en tres o cuatro maneras diferentes. Incluso si no todos funcionan, note que ha aprendido algo. Es fracaso es útil si te enseña algo. Es importante no tener miedo al fracaso. Es mejor intentar algo y fracasar, en lugar de demorar demasiado tiempo para una acción, tratando de hacer lo correcto.
- ❖ Calidad: la calidad debe estar siempre al máximo. Aceptar una menor calidad no afecta ni al ahorro, ni al rápido desarrollo. Por el contrario, la mejora de la calidad hace necesariamente una mejora de otras características del sistema, como la productividad y la eficiencia. Por otra parte, la calidad es no sólo un factor económico. Los miembros del equipo deben estar orgullosos de su trabajo porque equipo mejora la autoestima y la eficacia.
- ❖ Pasos de bebé: grandes cambios, preparados en un largo período de tiempo y en un gran tiro, son peligroso. Es mucho mejor proceder iterativamente en pequeños pasos, el paso más corto puede ser apreciado si es en la dirección correcta. Pequeños pasos no significan avanzar lentamente. Un equipo puede proceder mediante pequeños pasos y tomar un montón de ellos en un corto período de tiempo. Si se toma un pequeño paso en la dirección equivocada, se producen pequeños daños, mientras que un gran paso puede perjudicar gravemente el proyecto.
- ❖ Aceptado la responsabilidad: la responsabilidad sólo puede ser aceptada. Es fácil para los desarrolladores "Haz esto", o "hacer que", pero no funciona. Inevitablemente, le pide

menos de lo que podría ser logrado o, probablemente más, más que eso se puede lograr.

Prácticas

XP se basa en trece prácticas primarias, y once corolarios de prácticas. Las prácticas primarias deben aplicarse en primer lugar, y cada uno de ellos puede producir una mejora en el proceso de desarrollo de software. Corolarios de prácticas, por otra parte, requieren prácticas de experiencia en la enseñanza primaria, y son difíciles de aplicar sin las primarias. Las 24 prácticas son muy importantes, y debe aplicarse en su totalidad con el fin de obtener todos los posibles beneficios de XP.

Kent Beck no clasifica las prácticas. Prefiere atribuirles a cuatro categorías:

- ❖ Requisitos de análisis y planificación.
- ❖ Equipo y factores humanos.
- ❖ Diseño.
- ❖ Software, codificación y liberación.

Prácticas primarias.

Requisito de Análisis y Planificación:

- ❖ Niveles: las funcionalidades del sistema se describen mediante historias, breves descripciones del cliente de funcionalidades visibles. Las historias también conducen el desarrollo del sistema.
- ❖ Ciclo semanal: el desarrollo de software se realiza una semana a la vez. Al inicio de cada semana hay una reunión en la que son elegidas las historias que se desarrollan en la semana.
- ❖ Ciclo trimestral: en una escala de tiempo, el desarrollo se ha previsto en una cuarta parte a la vez. Este se compone de reflexiones sobre el equipo, el proyecto y el progreso.
- ❖ Lentitud: no hacer promesas que no pueden cumplir. En cualquier plan, incluir algunas tareas que puedas inclinar hacia atrás. De esta manera, tendrá un margen de seguridad.

Equipo y factores humanos:

- ❖ **Siéntense juntos:** los equipos de desarrollo deben trabajar en un espacio abierto, capaz de acoger todo el equipo, para maximizar la comunicación.
- ❖ **Todo el equipo:** el equipo debe estar compuesto por miembros de todas las aptitudes y las perspectivas necesarias para que el proyecto tenga éxito. Deben tener un fuerte sentido de pertenencia, y de ayuda a los demás.
- ❖ **Información de trabajo:** el área de trabajo deben contar con medios informativos, sobre el estado del proyecto y las tareas a realizar.
- ❖ **Energizar el trabajo:** los desarrolladores deben sentirse libres, de modo que puedan concentrarse en su trabajo y ser productivos. En consecuencia, limitar las horas extraordinarias de trabajo para que todos puedan pasar un tiempo para su vida privada.
- ❖ **Vincular la programación:** el código es siempre escrito por dos programadores en un mismo puesto de trabajo.

Diseño:

- ❖ **Incremento del Diseño:** XP se opone a la elaboración de un diseño completo por adelantado. El equipo de desarrollo produce el código tan pronto como sea posible a fin de obtener información y mejorar el sistema continuamente. XP sugiere diseñar gradualmente durante la codificación.
- ❖ **Prueba, Programación Primera:** antes de actualizar y añadir el código, es necesario escribir los ensayos a fin de verificar el código. Esto resuelve cuatro problemas:
- ❖ **Codificación vaquero:** Es fácil dejarse llevar rápidamente a programar y poner toda la mente en el código. Pero si se escriben las pruebas y hay que ejecutarlas, estas ayudan a centrarse en el problema, y pueden demostrar que el diseño es correcto.
- ❖ **Acoplamiento y cohesión:** si no es fácil escribir un ensayo, esto significa que usted tiene un problema de diseño, no de las pruebas o de codificación. Si su código está débilmente acoplado y altamente cohesionado, puede probarlo fácilmente.
- ❖ **Confianza:** si escribe código que funcione y se documentó con pruebas automatizadas, sus compañeros de equipo confiarán en usted.

- ❖ Ritmo: es fácil perderse y vagar durante horas cuando se está codificando. Si se acostumbra al mismo ritmo “la prueba, el código, refactorizar, la prueba, el código, refactorizar”, esto no va a suceder.

Codificación y liberación de software:

- ❖ Construcción de diez minutos: el sistema debe ser construido y todas las pruebas debe ser terminada en diez minutos, a fin de ejecutar a menudo y obtener retroalimentación.
- ❖ Integración continua: Los desarrolladores deben integrar los cambios cada dos horas con el fin de facilitar la integración total del proyecto.

Corolarios de prácticas.

Requisito de Análisis y Planificación:

- ❖ Participación real de clientes: las personas cuyas vidas se ven afectadas por su sistema se convirtió en un parte del equipo y que pueden contribuir a la planificación semanal y trimestral.
- ❖ Incremento de Despliegue: al sustituir un sistema de legado, empiezan a sustituir algunas funciones inmediatamente y sustituir gradualmente a todos los sistemas. Evitar el enfoque de "todo o nada".
- ❖ Negocio del alcance del contrato: en los contratos de desarrollo de software se han fijado tiempo, costes, y calidad, pero el alcance exacto del sistema tendría que ser negociado durante la misma realización. A la larga es mejor tener una secuencia de contratos a corto plazo, a fin de reducir los riesgos.
- ❖ Pago por uso: el cliente paga por lo general, para cada despacho de los programas informáticos. Esto crea un conflicto entre el proveedor y el cliente, que quiere menos liberaciones, cada una con un montón de funcionalidades. La conexión del dinero directamente al flujo de desarrollo de software proporciona información precisa y oportuna con la que la unidad mejora.

Equipo y factores humanos:

- ❖ Equipo de Continuidad: el desarrollo de equipos debe seguir siendo el mismo en varios proyectos. La relación que comparten en un proyecto son preciosas y no tienen que ser dispersados.

- ❖ La disminución de equipos: el equipo se vuelve más capaz y productivo, manteniendo la carga constante, pero reduciendo gradualmente su tamaño, y se liberan miembros para formar más equipos.

Diseño:

- ❖ Análisis de Causa Raíz: cada vez que se encuentra un defecto, eliminar sus causas. En esta distancia, usted acaba de eliminar el defecto, pero también te evitará cometer el mismo error nuevamente.

Codificación y liberación de software:

- ❖ Código y Pruebas: sólo el código y las pruebas son permanentes y los artefactos que tienen que ser preservados pueden ser generados a partir del código y las pruebas.
- ❖ Código compartido: cualquier persona en el equipo de desarrollo debe ser capaz de cambiar cualquier parte del sistema en cualquier tiempo.
- ❖ Un Código Base: sólo hay una versión oficial del sistema. Usted puede desarrollar una rama temporal, pero no vive más que unas pocas horas.
- ❖ Despliegue diario: cada noche, usted debe poner un nuevo software en la producción. Es arriesgado y costoso tener una diferencia entre la versión de software liberados en la producción y los que tiene en su ordenador.

1.3.4.3 ¿Por qué XP?

El estudio de metodologías de desarrollo más usadas en la actualidad ha demostrado que XP es la mejor opción y la que más se adecua a las necesidades del sistema a implementar ya que es una de las metodologías de desarrollo de software más exitosas utilizadas en la actualidad para proyectos de corto plazo y para equipos de desarrollo pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final.

1.3.5 Servicios Web.

Un Servicio Web (en inglés Web Service) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet. La

interoperabilidad se consigue mediante la adopción de estándares abiertos. También podemos decir que los servicios Web son muy prácticos ya que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más utilizada. (15)

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

1.3.6 Frameworks.

Un framework es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. También puede ser considerado como el conjunto de procesos y tecnologías usados para resolver un problema complejo. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

Los frameworks son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software y entendiendo el negocio, que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Un ejemplo de ello, un equipo que usa Apache Struts para desarrollar un sitio web de un banco puede enfocarse en cómo los retiros de ahorros van a funcionar en lugar de preocuparse de cómo se controla la navegación entre las páginas en una forma segura y libre de errores. Sin embargo, hay quejas comunes acerca de que el uso de frameworks añade código innecesario y que la preponderancia de frameworks competitivos y complementarios

significa que el tiempo que se pasaba un desarrollador programando y diseñando ahora se gasta en aprender a usar frameworks. (16)

1.3.6.1 Prado.

Prado está basado en componentes eventos con el objetivo de acelerar el desarrollo de aplicaciones web usando PHP 5.

El concepto del desarrollo de aplicaciones en Prado es diferente, se utilizan componentes, eventos y propiedades en vez de procedimientos, URL y parámetros.

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

Entre los beneficios que podemos encontrar para el desarrollo de aplicaciones web se encuentran:

- ❖ Reutilización: los códigos y componentes pueden ser reutilizados.
- ❖ Fácil uso: la creación y uso de componentes es fácil.
- ❖ Funcionamiento: utiliza una técnica de caché para asegurar el funcionamiento de aplicaciones basadas en él.
- ❖ Integración: permite la separación del contenido y la presentación. (17)

1.3.6.2 Cakephp.

CakePHP es un framework de desarrollo de aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails. Facilita al usuario la interacción con la base de datos mediante el uso de ActiveRecord. Además hace uso del patrón Modelo Vista Controlador.

Características de Cakephp.

- ❖ Compatible con PHP4 y PHP5.
- ❖ CRUD de la base de datos integrado.
- ❖ URLs amigables.
- ❖ Sistema de plantillas rápido y flexible.
- ❖ Helpers para AJAX, Javascript, HTML, forms y más.
- ❖ Trabaja en cualquier subdirectorio del sitio.
- ❖ Validación integrada.
- ❖ Listas de control de acceso.

- ❖ Componentes de seguridad y sesión. (19)

1.3.6.3 Symfony.

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix, Unix, Linux como en plataformas Windows.

Características de symfony:

- ❖ Fácil de instalar y configurar en la mayoría de las plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y *nix estándares).
- ❖ Independiente del sistema gestor de bases de datos.
- ❖ Sencillo de usar en la mayoría de los 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 es 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 las librerías de otros fabricantes.

(18)

1.3.6.4 ¿Por qué Symfony?

El estudio de los frameworks más usados en la actualidad ha demostrado que Symfony es la mejor opción y la que más se adecua a las necesidades del sistema a implementar, por su facilidad de instalación y configuración en la mayoría de los sistemas operativos. Utiliza una licencia MIT (Licencia de software empleada por el Massachusetts Institute of Technology.), con la que pueden hacerse aplicaciones web comerciales, gratuitas y/o de software libre. Además de su independencia al gestor de bases de datos y ser lo suficientemente flexible como para adaptarse a los casos más complejos.

1.3.7 Arquitectura.

La arquitectura del Software es uno de los grandes temas actuales que definen la construcción de un software. Presenta gran importancia ya que es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. Representa una vista estructural de alto nivel que ocurre muy tempranamente en el ciclo de vida y define los estilos o grupos de estilos adecuados para cumplir con los requerimientos no funcionales del sistema. Además consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del Software.

Un estilo arquitectónico define las reglas generales de organización en términos de un patrón y las restricciones en la forma y la estructura de un grupo numeroso y variado de sistemas de software. En una forma más específica, un estilo determina el vocabulario de componentes y conectores que pueden ser utilizados en instancias de este estilo, con un conjunto de restricciones en las descripciones arquitectónicas.

1.3.7.1 N Capas.

La arquitectura en capas básicamente define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa solo pueden hacer referencia a los componentes de las capas inmediatamente inferiores.

Características de la arquitectura en 3 capas.

- ❖ Implementa la lógica del negocio.
- ❖ Múltiples usuarios pueden estar conectados simultáneamente pues el SGBD no tiene que resolver por si solo la comunicación con los clientes. (20)

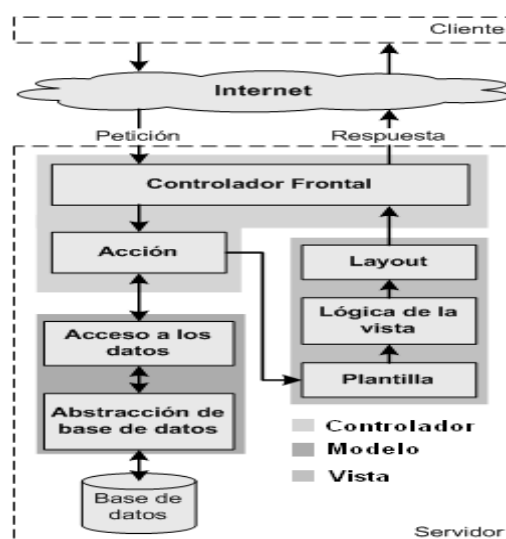
1.3.7.2 Modelo Vista Controlador (MVC).

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

- ❖ El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- ❖ La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- ❖ El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. (21)

1.3.7.3 Arquitectura Seleccionada

El estudio de las arquitecturas de desarrollo de software más usadas en la actualidad ha demostrado que la mejor opción y la que más se adecua a las necesidades del sistema a implementar es una fusión de estos dos estilos arquitectónicos ya que esto es lo que hace el framework symfony el cual juega un papel fundamental en el desarrollo de nuestro sistema. A continuación se muestra como symfony subdivide en capas al modelo, la vista y el controlador.



1.4 Conclusiones.

En este capítulo se realiza un estudio de las tendencias y tecnologías actuales para el desarrollo de aplicaciones Web. Dentro del gran número de herramientas, metodologías y lenguajes existentes de hoy día, se analizan aquellas que gozan de mayor popularidad y se parte del concepto de “independencia tecnológica” que persigue Cuba en el desarrollo de la informática de forma general. Por supuesto que nuestra universidad no se queda detrás en este tema, por eso, la selección realizada se caracteriza por ser software libre y multiplataforma en su mayoría. También se tiene en cuenta el conocimiento previo sobre estas tecnologías, con el objetivo de minimizar el tiempo de desarrollo del sistema a realizar.

Capítulo 2

Características del sistema.

2.1 Introducción.

El presente capítulo tiene como objetivo hacer una valoración de las características principales del sistema a desarrollar, prestando gran atención a la situación problémica que dio origen al mismo. Se detallan las necesidades de los usuarios, describiéndose las funcionalidades que serán objeto de automatización. Por último se presentará una propuesta del software a implementar, especificando detalladamente los requerimientos funcionales y no funcionales del mismo.

2.2 Descripción de los procesos vinculados al campo de acción.

Actualmente el proceso para la publicación y matrícula en una actividad de postgrado en la universidad (Doctorados, Maestrías, Diplomados, Becas, Cursos, y otras Actividades) incluye una serie de flujos y acciones que rigen el comportamiento del mismo, estos flujos permiten la estandarización del entorno de trabajo para la publicación y matrícula de cada una de las actividades de superación que son ofertadas por parte de la Dirección de Postgrado de manera general. Además los profesores no cuentan con la posibilidad de tener información actualizada sobre su situación en cuanto a la superación que desarrollan, así como acceso a documentos de interés, los cuales pueden ser artículos científicos, Tesinas o resúmenes de Doctorados realizados por otros docentes.

2.2.1 Flujo actual del proceso general.

El desarrollo de este proceso se realiza de la siguiente manera: De la oficina de Postgrado de la Universidad le envían al asesor de postgrado en la facultad todos los postgrados disponibles, presentando el programa y la planificación de los mismos. A continuación el asesor le hace llegar a cada Asesor de Postgrado de departamento dichas convocatorias y serán ellos en conjunto a los Jefes de Departamentos los que en dependencia del tipo de postgrado que sea y según las prioridades que presenta el departamento, determinen quienes serán los profesores

que matricularán en las actividades antes mencionadas. Luego se le hace llegar un listado al asesor de los seleccionados y esa información es enviada a la oficina de Postgrado de la Universidad. Al concluir el Postgrado el encargado de impartirlo le entrega al profesor un documento que acredita que la actividad fue realizada de forma satisfactoria, el mismo tendrá que ser previamente entregado a la secretaría de la Dirección de Postgrado. En caso de que el postgrado no sea obligatorio para la categorización del profesor su matrícula se hará en dependencia de los intereses, línea de trabajo, investigación o producción, a no ser que las plazas sean limitadas, lo cual implicaría que el proceso original se repetiría.

2.2.2 Objeto de automatización.

Durante el ciclo de desarrollo de esta actividad existen varios procesos que deben ser automatizados, puesto que su ejecución de forma manual resulta tediosa y propensa a errores, además de consumir una valiosa porción de tiempo de los desarrolladores. Muchos de estos procesos son de vital importancia para lograr un nivel de calidad óptimo en los productos finales. Se automatizarán los procesos de publicación de los cursos mediante una aplicación Web, así como la publicación de artículos de interés, tesis, resúmenes de doctorados, etc. En el sistema se manejará información referente a los cursistas, cursos y otras actividades de postgrado de manera general, se facilitará el plan de superación de los adiestrados y otros materiales de interés para el asesor de postgrado de la facultad.

2.3 Propuesta del sistema.

Se propone la implementación de un sistema que provea las mismas funcionalidades que el flujo de trabajo actual. Dicho sistema debe permitir la gestión (crear, modificar y eliminar) de una cuenta de suscripción al sistema, recogiendo los datos que son de interés para los administradores del sistema. Debe permitir la publicación de convocatorias de cursos incluyendo todos los datos de importancia para los interesados en estos, además debe permitirle a los usuarios emitir comentarios sobre las publicaciones que tengan lugar en el sistema. Los administradores tendrán la posibilidad de crear la matrícula en los diferentes cursos ofertados y serán los encargados de gestionar la publicación de los diferentes materiales de interés tanto para el asesor de postgrado como para los cursistas. El administrador del sistema será el encargado de gestionar toda la información referente al sistema y dará soporte en cada una de las instancias necesitadas. El sistema será una aplicación Web creada y administrada con el framework Symfony.

2.3.1 Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. Además de aquellas que se encuentran involucradas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

Tabla 1 Personas relacionadas con el sistema.

Persona relacionada con el sistema.	Justificación.
Usuario.	Es la persona que puede navegar por el sistema viendo toda la información publicada y su situación personal referente a la superación postgraduada.
Jefes de Departamento y Asesores de Departamento.	Es la persona que además de los privilegios de Usuario, tiene la posibilidad de consultar y editar la situación respecto a la superación postgraduada de los profesores de su departamento.
Asesor y/o Administrador.	Es la persona que además de los privilegios de Usuario, tiene la posibilidad de consultar y modificar toda la información que posee el sistema. Además es el responsable de crear la matrícula de los cursos y darle soporte y mantenimiento a la aplicación.

2.3.2 Requerimientos funcionales del sistema

Una vez conocidos todos los conceptos que rodean al objeto de estudio, se puede analizar qué debe hacer el sistema para que se cumplan los objetivos planteados al inicio de este trabajo. Para ello se enumeran a través de requerimientos funcionales las prestaciones que el sistema será capaz de brindar. Dentro de ellas se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema y las condiciones extremas a determinar por el sistema.

De acuerdo a los objetivos planteados, el sistema debe ser capaz de:

RF1. Gestionar (Insertar, modificar, eliminar) Postgrados (Doctorados, Maestrías).

RF1.1. Mostrar Postgrados (Doctorados, Maestrías).

RF2. Gestionar Actividades.

RF2.1. Mostrar Actividades.

RF3. Gestionar Cursos.

RF3.1. Mostrar Cursos.

RF4. Gestionar Becas.

RF4.1. Mostrar Becas.

RF5 Gestionar Diplomados.

RF5.1. Mostrar Diplomados.

RF6. Gestionar profesores.

RF7. Gestionar Polos Productivos.

RF8. Gestionar Plan de superación.

RF8.1 Mostrar Plan de Superación.

RF9. Autenticar.

RF10. Crear matrícula de un Postgrado.

RF11. Publicar Trabajos.

RF12. Ver listado de profesores.

RF12.1. Por categoría docente o científica.

RF12.2. Por actividad de postgrado (Cursos, Doctorados, Maestrías, Diplomados, Becas, actividades).

RF12.3. Por departamento.

RF12.4 Por Polo Productivo.

RF13. Gestionar Comentario.

RF14. Guardar Interés del Postgrado

2.3.3 Requerimientos no funcionales del sistema.

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.

Diseño e implementación

- ❖ Aplicación Web escrita sobre el lenguaje de programación PHP 5.2.5.
- ❖ Usar el gestor de Base de Datos
- ❖ Utilizar como servidor web Apache 2.4.
- ❖ Desarrollar bajo Framework Symfony.
- ❖ Utilizar como metodología de desarrollo XP.

Apariencia o interfaz externa.

- ❖ Diseño sencillo, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

- ❖ No debe tener animaciones ni imágenes pesadas que obstaculicen la rapidez de las transiciones del ancho de banda.
- ❖ Diseño perfectamente encuadrado para resoluciones de 800x600, pero preparado para verse en otras resoluciones.
- ❖ El diseño garantizará la selección de un esquema de colores a la vez atractivo pero que no resulte agotador, conjugando equilibrio entre los mismos y un contraste suficiente pero no agresivo.

Usabilidad.

- ❖ El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Funcionalidad.

- ❖ Capacidad de búsqueda con un tiempo menor de 5 segundos.
- ❖ Mínima cantidad de páginas para ejecutar todas las funciones posibles, es decir, agrupar funciones afines en las mismas páginas.

Rendimiento.

- ❖ La eficiencia de esta aplicación debe ser óptima en cuanto a la velocidad de procesamiento, disponibilidad, tiempo de respuesta y aprovechamiento de los recursos, entre otros.

Seguridad.

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

Portabilidad.

- ❖ Que corra en cualquier plataforma, es decir, que sea multiplataforma.

2.4 Conclusiones.

En este capítulo se inicia el desarrollo de la propuesta de solución que se desea implementar, tras el análisis de los flujos de trabajos actuales descritos por el cliente. Se obtuvo un listado de funcionalidades que debe tener el sistema, expresados en los requerimientos funcionales. Partiendo de este punto, base de todo proceso de desarrollo, se puede comenzar con la construcción de la propuesta, velando por el cumplimiento de todos los requerimientos y funcionalidades consideradas.

Capítulo 3

Exploración y Planificación.

3.1 Introducción

En este capítulo se hace alusión a las fases de exploración y planificación, propias de la metodología de desarrollo utilizada para la implementación del sistema que se propone. Se exponen además los artefactos generados durante el transcurso de las mismas.

3.2 Fase de Exploración

La metodología de desarrollo Extreme Programming comienza con su fase de exploración. Durante esta etapa se realiza el proceso de identificación de las historias de usuario (UH, del inglés User Histories), así como la familiarización de los equipos de trabajo con las tecnologías y herramientas seleccionadas para la construcción del sistema.

3.2.1 Historias de usuario

Las historias de usuarios son la forma en que se especifican en XP los requisitos del sistema. Estas se redactan desde la perspectiva del cliente aunque los desarrolladores pueden brindar también su ayuda en la identificación de las mismas. El contenido que ellas abarcan debe ser concreto y sencillo. Durante este proceso se identifican 20 historias de usuarios, las cuales se detallan a continuación.

Tabla 2 HU Gestionar Postgrado.

Historia de Usuario	
Número: 1	Nombre: Gestionar Postgrado
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se realiza la acción (previa autenticación) de insertar, eliminar y editar postgrado. Se introducirán los datos de los postgrados.	

Tabla 3 HU Mostrar Postgrados.

Historia de Usuario	
Número: 2	Nombre: Mostrar Postgrados
Usuario: Usuario	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción:	

Se brinda la posibilidad de que cualquier persona que acceda a la aplicación tenga la posibilidad de ver un listado de los Postgrados separados por categorías que se están ofertando, en la lista solo se mostrará el nombre (el cual tendrá el vínculo para poder ver la información referente a dicho postgrado), la matrícula y el estado en que se encuentra (Activo o Expirado), en dependencia del la fecha en la que fue publicado o según características específicas del Postgrado.

Tabla 4 HU Gestionar Actividad.

Historia de Usuario	
Número: 3	Nombre: Gestionar Actividad
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se realizan las acciones (previa autenticación) de insertar, eliminar y editar Actividad. Actividad se refiere a cualquier evento de superación que no se encuentre dentro de la categoría de Cursos, Postgrados, Diplomados ni Becas.	

Tabla 5 HU Mostrar Actividades.

Historia de Usuario	
Número: 4	Nombre: Mostrar Actividades
Usuario: Usuario	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se brinda la posibilidad de que cualquier persona que acceda a la aplicación tenga la posibilidad de ver las Actividades así como toda la información referente a una en particular. En el listado solo se mostrará el nombre de la actividad (el cual será el vínculo para conocer todas sus características) así como la fecha de presentación de la misma.	

Tabla 6 HU Gestionar Curso.

Historia de Usuario	
Número: 5	Nombre: Gestionar Curso
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se realizan las acciones (previa autenticación) de insertar, eliminar y editar Cursos.	

Tabla 7 HU Mostrar Cursos.

Historia de Usuario	
Número: 6	Nombre: Mostrar Cursos
Usuario: Usuario	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se brinda la posibilidad de que cualquier persona que acceda a la aplicación tenga la	

posibilidad de ver un listado de los Cursos así como toda la información referente a uno en particular. En el listado solo se mostrará el nombre del curso (el cual tendrá un vínculo para conocer toda sus características), la matrícula y las fechas de encuentros.

Tabla 8 HU Gestionar Beca.

Historia de Usuario	
Número: 7	Nombre: Gestionar Beca
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se realizan las acciones (previa autenticación) de insertar, eliminar y editar Becas.	

Tabla 9 HU Mostrar Becas.

Historia de Usuario	
Número: 8	Nombre: Mostrar Becas
Usuario: Usuario	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se brinda la posibilidad de que cualquier persona que acceda a la aplicación pueda ver un listado de las Becas así como toda la información referente a una en particular. En el listado solo se mostrará el nombre de la Beca (el cual tendrá un vínculo para conocer todas sus características) y el país donde será impartida.	

Tabla 10 HU Gestionar Diplomado.

Historia de Usuario	
Número: 9	Nombre: Gestionar Diplomado
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se realizan las acciones (previa autenticación) de insertar, eliminar y editar Diplomado.	

Tabla 11 HU Mostrar Diplomados.

Historia de Usuario	
Número: 10	Nombre: Mostrar Diplomados
Usuario: Usuario	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se brinda la posibilidad de que cualquier persona que acceda a la aplicación tenga la posibilidad de ver el listado de los Diplomados así como toda la información referente a uno en particular. En el listado solo se mostrará el nombre (el cual tendrá un vínculo para conocer todas sus características) y el Polo Productivo al que pertenece.	

Tabla 12 HU Gestionar Profesor.

Historia de Usuario	
Número: 11	Nombre: Gestionar Profesor
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación:	Iteración Asignada: 3
Descripción: Se realizan las acciones (previa autenticación) de insertar, eliminar y editar profesor.	

Tabla 13 HU Gestionar Polo Productivo.

Historia de Usuario	
Número: 12	Nombre: Gestionar Polo Productivo
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Bajo
Puntos de Estimación:	Iteración Asignada: 1
Descripción: Se realizan las acciones (previa autenticación) de insertar, eliminar y editar Polo Productivo.	

Tabla 14 HU Gestionar Plan de Superación.

Historia de Usuario	
Número: 13	Nombre: Gestionar Plan de Superación.
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos de Estimación:	Iteración Asignada: 2
Descripción: Se realizan las acciones (previa autenticación) de insertar, eliminar y editar Plan de Superación.	

Tabla 15 HU Mostrar Plan de Superación.

Historia de Usuario	
Número: 14	Nombre: Mostrar Plan de Superación
Usuario: Usuario	
Prioridad en el Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos de Estimación:	Iteración Asignada: 2
Descripción: Se brinda la posibilidad de que cualquier persona que acceda a la aplicación tenga la posibilidad de ver el Plan de Superación.	

Tabla 16 HU Autenticar.

Historia de Usuario	
Número: 15	Nombre: Autenticar
Usuario: Usuario	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación:	Iteración Asignada: 2
Descripción: Se brinda la posibilidad de que la persona que acceda al sistema introduzca sus datos (usuario contraseña) con la finalidad de verificar y otorgarle los permisos según el rol que ocupa dentro de la aplicación.	

Tabla 17 HU Guardar Interés del Postgrado.

Historia de Usuario	
Número: 16	Nombre: Guardar Interés del Postgrado
Usuario: Usuario	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación:	Iteración Asignada: 2
Descripción: Se brinda la posibilidad de que la persona que acceda al sistema (previa autenticación) pueda marcar según los postgrados publicados, cuáles son de su interés.	

Tabla 18 HU Crear Matrícula de Postgrado.

Historia de Usuario	
Número: 17	Nombre: Crear Matrícula de Postgrado
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación:	Iteración Asignada: 2
Descripción: Se brinda la posibilidad de asignarle los integrantes a un Postgrado. Esto se realiza a través de una lista que envían los Asesores de Postgrado de los departamentos.	

Tabla 19 HU Publicar Trabajos.

Historia de Usuario	
Número: 18	Nombre: Publicar Trabajos Personales
Usuario: Asesor y/o Administrador	
Prioridad en el Negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación:	Iteración Asignada: 3
Descripción: Se brinda la posibilidad (previa autenticación) de publicar Trabajos en la aplicación. Estos trabajos pueden ser Tesinas, Doctorados (resúmenes), Maestrías (resúmenes) Diplomados, etc., así como trabajos de otra índole que sean interesantes para los docentes. Los mismos se le hacen llegar al Asesor y/o Administrador previamente y serán publicados en la aplicación, separados por las categorías antes descritas.	

Tabla 20 HU Ver Listado de Profesores.

Historia de Usuario	
Número: 19	Nombre: Ver Listado de Profesores
Usuario: Usuario	
Prioridad en el Negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación:	Iteración Asignada: 3
Descripción: Se realizan las acciones de Listar profesores según los siguientes parámetros: <ol style="list-style-type: none"> 1. Por categoría docente o científica. 2. Por actividad de postgrado. 3. Por Departamentos. 4. Por Polos Productivos. 	

Tabla 21 HU Gestionar Comentario.

Historia de Usuario	
Número: 20	Nombre: Gestionar Comentario
Usuario: Asesor y/o Administrador	

Prioridad en el Negocio: Media	Riesgo en Desarrollo: Medio
Puntos de Estimación:	Iteración Asignada: 3
Se brinda la posibilidad de que cualquier persona que acceda a la aplicación pueda emitir un comentario sobre algún artículo u otro tipo de documento publicado en la misma.	

3.3 Fase de Planificación.

Durante la fase de planificación se realiza una estimación del esfuerzo que costará implementar cada historia de usuario. Esto se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario, por ejemplo, las pruebas unitarias, la integración y refactorización del código, y la preparación y ejecución de las pruebas de aceptación.

3.3.1 Estimación de esfuerzos por historias de usuario

Tabla 22 Estimación de esfuerzos por HU.

Historia de Usuario	Puntos de Estimación
Gestionar Postgrado	0.5
Mostrar Postgrado	0.5
Gestionar Actividad	0.5
Mostrar Actividades	0.5
Gestionar Curso	1
Mostrar Cursos	0.5
Gestionar Beca	0.5
Mostrar Becas	0.5
Gestionar Diplomado	0.5
Mostrar Diplomados	0.5
Gestionar Profesor	1
Gestionar Polo Productivo	0.2
Guardar Interés de Postgrado	1
Gestionar Plan de Superación	0.2
Mostrar Plan de Superación	0.2
Autenticar	2
Crear Matrícula de Postgrado	1
Publicar Trabajos	0.2
Ver Listado de Profesores	1
Gestionar Comentario	1

3.3.2 Plan de iteraciones

Después de ser descritas e identificadas las historias de usuario y estimado el esfuerzo propuesto para la realización de cada una de ellas, se procede a la planificación de la etapa de implementación del sistema. Este plan especifica exactamente cuáles historias de usuario serán implementadas para cada iteración del sistema y las posibles fechas para estas liberaciones.

En base a lo antes mencionado se decide realizar el sistema en 3 iteraciones, las cuales se detallan a continuación:

Iteración 1.

Esta iteración tiene como objetivo la implementación de las Historias de Usuario de importancia en el negocio pero que a su vez tengan baja prioridad en el negocio. Al finalizar se tendrá la implementación de las funcionalidades descritas en las historias de usuario 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12 y 13 las cuales se refieren principalmente a la gestión y visualización de las entidades Postgrado, Actividad, Curso, Beca y Diplomado.

Iteración 2.

Esta iteración tiene como objetivo la implementación de las Historias de Usuario con alta prioridad para el negocio, así como algunas de media prioridad. Al finalizar se tendrá la implementación de las funcionalidades de alta y media prioridad que no generen dependencias con otras historias de usuario, las cuales se pueden comprobar una vez finalizada su implementación si depender de otras implementaciones. Las mismas refieren a la autenticación, la creación de las matrículas de postgrado, guardar el interés de postgrado así como la visualización y gestión del Plan de Superación.

Iteración 3.

En esta iteración se implementarán las HU de alta y media prioridad que dependen de la implementación de las descritas en la iteración 2. Además se refinarán requisitos no funcionales que le permitirán al usuario tener comodidad en la gestión y otras tareas asociadas, además dichas actividades tributarán a la navegabilidad y usabilidad del sistema. Como resultado de esta iteración se tendrá la versión 1.0 del producto final, adicionando lo concerniente a la interfaz web. A partir de este momento el sistema será puesto a prueba por un período de tiempo para evaluar el desempeño del mismo.

3.3.3 Plan de duración de las iteraciones.

Como parte del ciclo de vida de un proyecto utilizando la Metodología XP se crea el plan de duración de cada una de las iteraciones, en este caso se hace para el único equipo de desarrollo con el cual se cuenta. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario en cada una de las mismas.

Tabla 23 Plan de duración de las Iteraciones.

Iteraciones	Orden de las Historias de usuario a implementar	Duración total de las iteraciones
1	1- Gestionar Postgrado. 2- Mostrar Postgrados. 3- Gestionar Actividad. 4- Mostrar Actividades. 5- Gestionar Curso. 6- Mostrar Curso. 7- Gestionar Beca. 8- Mostrar Beca. 9- Gestionar Diplomado. 10- Mostrar Diplomados. 11- Gestionar Polo Productivo.	2 Semanas.
2	1- Autenticar. 2- Crear Matrícula de Postgrado. 3- Guardar Interés de Postgrado. 4- Gestionar Profesor Ver. 5- Ver Listado de Profesores.	3 Semanas.
3	1- Publicar Trabajos. 2- Gestionar Comentario. 3- Gestionar Plan de Superación. 4- Mostrar Plan de Superación.	3 Semanas.

3.3.4 Plan de Entregas

A continuación se presenta el plan de entregas elaborado para la fase de implementación. Para facilitar la elaboración de dicho plan se acoplaron las funcionalidades referentes a un mismo tema en módulos, quedando éstos de la siguiente manera:

Tabla 24 Módulos y HU Asociadas.

Módulos	Historias de Usuario Asociadas
admin	Autenticar. Publicar Trabajos. Crear Matrícula de Postgrado. Gestionar Plan de Superación. Mostrar Plan de Superación. Ver Listado de Profesores.
prof	Gestionar Profesor. Guardar Interés de Postgrado. Gestionar Comentario.
post	Gestionar Postgrado. Mostrar Postgrados.
act	Gestionar Actividad. Mostrar Actividades.
becas	Gestionar Beca. Mostrar Beca.
dipl	Gestionar Diplomado. Mostrar Diplomados.
 cursos	Gestionar Curso. Mostrar Curso.
polo	Gestionar Polo Productivo.

Como producto del plan de entregas se harán releases al sistema en las fechas que se indican a continuación:

Tabla 25 Plan de entregas.

Exploración y Planificación.

Módulos	Final 1ra Iteración 4ta semana de Marzo	Final 2da Iteración 2da semana de Abril	Final 3ra Iteración 3ra semana de Mayo
admin	1.0	1.1	-
prof	1.0	1.1	-
post	1.0	Finalizado	Finalizado
act	Finalizado	Finalizado	Finalizado
polo	1.0	Finalizado	Finalizado
cursos	1.0	Finalizado	Finalizado
becas	1.0	Finalizado	Finalizado
dipl	1.0	Finalizado	Finalizado

3.4 Conclusiones

Durante el desarrollo de este capítulo se hace referencia a todo lo concerniente a la fase de Exploración y Planificación del Sistema, haciendo una descripción de cada uno de los artefactos generados en el transcurso de esta fase. Se asume una implementación iteraciones la cual fue concebida y debidamente detallada. En esta fase es muy importante y necesario el permanente diálogo entre el cliente y el equipo de desarrollo.

Diseño, Implementación y Pruebas.

4.1 Introducción

La Metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éste. En el presente capítulo se describe el diseño de la aplicación, se detallan las tres iteraciones llevadas a cabo durante la etapa de construcción, exponiéndose las tareas generadas por cada historia de usuario, así como las pruebas de aceptación efectuadas sobre la aplicación.

4.2 Diseño del Sistema.

Para el diseño de aplicaciones la metodología XP no requiere la realización de diagramas utilizando notación UML, en lugar de estos se utilizan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración), a continuación se muestran las tarjetas CRC generadas en la Fase de Diseño de la aplicación.

Tabla 26 Tarjeta CRC Módulo admin.

Módulo admin.	
Funcionalidades.	Colaboraciones (Módulos).
Autenticar.	prof.
Publicar trabajo.	
Crear Matrícula de Postgrado.	
Insertar Plan de Superación.	
Modificar Plan de Superación.	
Eliminar Plan de Superación.	
Mostrar Plan de Superación.	

Tabla 27 Tarjeta CRC Módulo prof.

Módulo prof.	
Funcionalidades	Colaboraciones (Módulos)
Insertar Profesor.	admin.
Modificar Profesor.	post.
Eliminar Profesor.	cursos.
Mostrar Profesor.	dipl.
Mostar interés de postgrado.	becas.
Enviar Comentario.	act. polo.

Tabla 28 Tarjeta CRC Módulo post.

Módulo post.	
Funcionalidades.	Colaboraciones (Módulos).
Insertar Postgrado.	prof.
Modificar Postgrado.	cursos.
Eliminar Postgrado.	
Mostrar Postgrados.	

Tabla 29 Tarjeta CRC Módulo act.

Módulo act.	
Funcionalidades.	Colaboraciones (Módulos).
Insertar Actividad.	prof.
Modificar Actividad.	
Eliminar Actividad.	

Mostrar Actividades	
---------------------	--

Tabla 30 Tarjeta CRC Módulo polo.

Módulo polo.	
Funcionalidades.	Colaboraciones (Módulos).
Insertar Polo.	prof.
Modificar Polo.	cursos.
Eliminar Polo.	

Tabla 31 Tarjeta CRC Módulo cursos.

Módulo cursos	
Funcionalidades	Colaboraciones (Módulos)
Insertar Curso.	post.
Modificar Curso.	prof.
Eliminar Curso.	dipl.
Mostrar Curso.	polo.

Tabla 32 Tarjeta CRC Módulo becas.

Módulo becas	
Funcionalidades	Colaboraciones (Módulos)
Insertar Beca.	prof.
Modificar Beca.	
Eliminar Beca.	
Mostrar Beca.	

Tabla 33 Tarjeta CRC Módulo dipl.

Módulo dipl

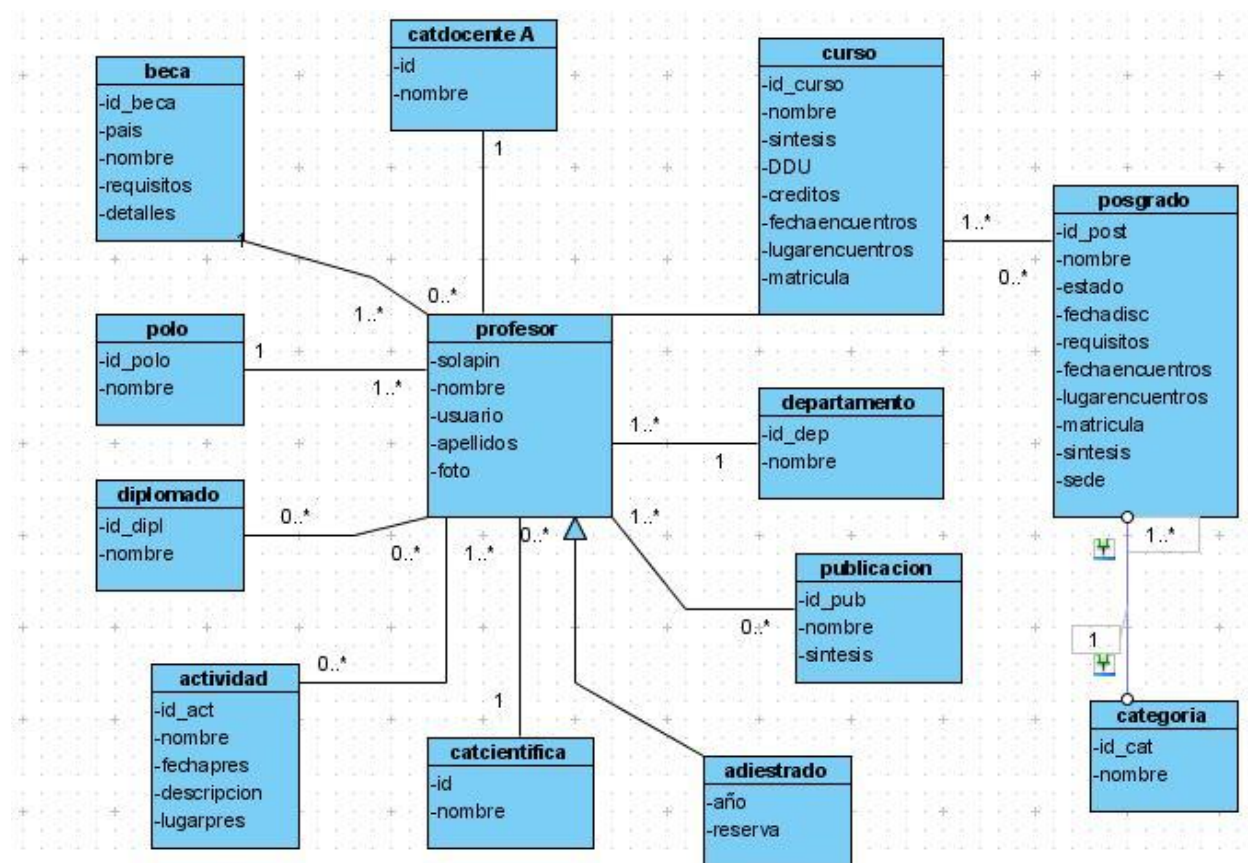
Funcionalidades	Colaboraciones (Módulos)
Insertar Diplomado.	prof.
Modificar Diplomado.	cursos.
Eliminar Diplomado.	
Mostrar Diplomados.	

4.2.1 Diseño de la Base de Datos.

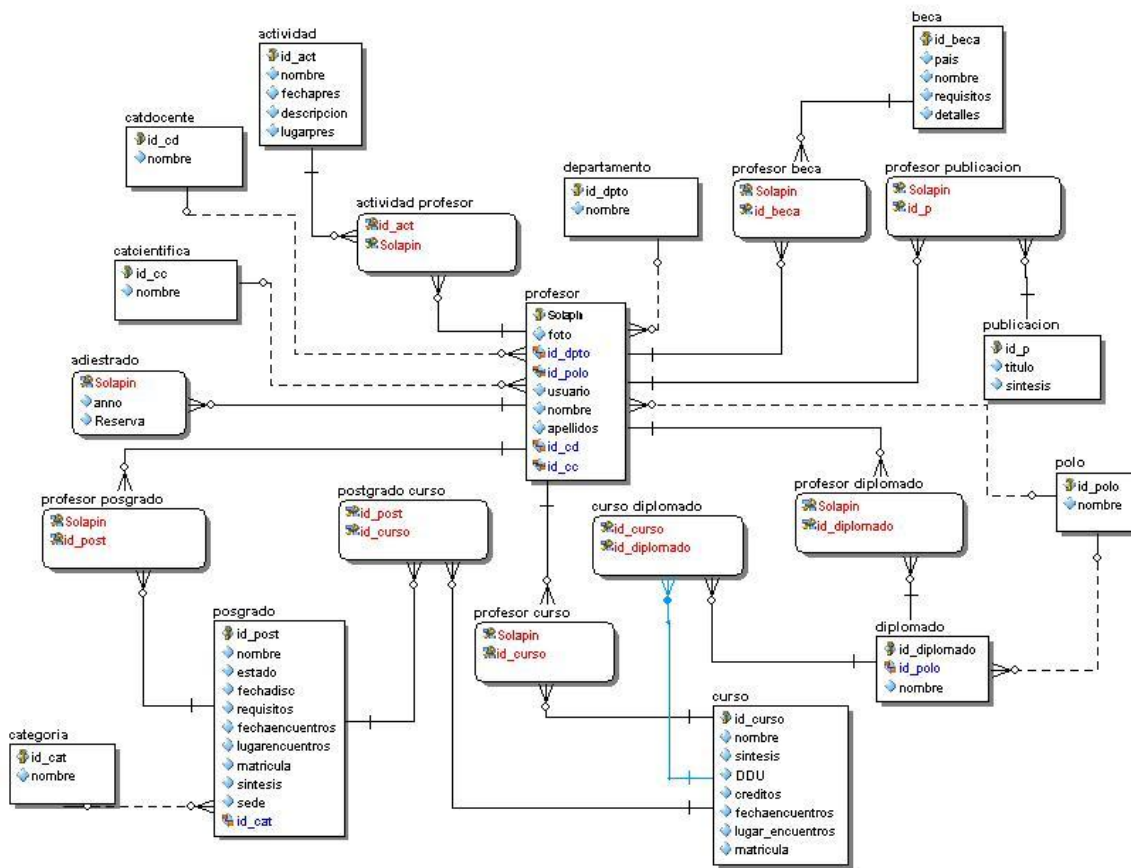
El diseño de la Base de Datos es algo que no se puede pasar por alto, producto de que uno de sus objetivos fundamentales es brindar la persistencia al modelo que se describe en el epígrafe anterior.

A continuación se muestra el diagrama de clases persistentes y el modelo de datos que se utilizó:

Diagrama de Clases Persistentes.



Modelo de Datos.



4.3 Codificación.

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario seleccionadas para ser realizadas en cada una de ellas. Al principio de estas se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando a un grupo de desarrollo (o persona), la responsabilidad de su implementación. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

4.3.1 Iteración 1

En esta iteración se implementan las historias de usuario con una baja complejidad con el fin de obtener algunas funcionalidades para mostrárselas al cliente y obtener información de este, para tomar decisiones con respecto a los pasos a seguir para el desarrollo de las otras iteraciones.

Tabla 34 Módulos abordados en la Iteración 1.

Módulos.	Historias de Usuario.	Tiempo en Implementación (Ptos_Est).	
		Estimación.	Real.
post.	Gestionar Postgrado.	0.5	0.1
	Mostrar Postgrados.	0.5	0.1
act.	Gestionar Actividad.	0.5	0.1
	Mostrar Actividades.	0.5	0.1
cursos.	Gestionar Curso.	1	0.1
	Mostrar Curso.	0.5	0.1
becas.	Gestionar Beca.	0.5	0.1
	Mostrar Beca.	0.5	0.1
dipl.	Gestionar Diplomado.	0.5	0.1
	Mostrar Diplomados.	0.5	0.1
polo.	Gestionar Polo Productivo.	0.2	0.1

A continuación se describen las tareas efectuadas para cada uno de los módulos desarrollados en la iteración 1:

Módulo post.

Tabla 35 Tarea 1 del módulo post.

Tarea	
Número de la tarea: 1	Número de HU: 1
Nombre de la tarea: Creación de la tabla postgrado en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 03-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a postgrado.	

Tabla 36 Tarea 2 del módulo post.

Tarea	
Número de la tarea: 2	Número de HU: 1
Nombre de la tarea: Implementación de Gestionar Postgrado.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-04-2009	Fecha fin: 06-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten adicionar, modificar y eliminar un postgrado.	

Tabla 37 Tarea 3 del módulo post.

Tarea	
Número de la tarea: 3	Número de HU: 2
Nombre de la tarea: Implementación de Mostrar Postgrados.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 06-04-2009	Fecha fin: 07-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten mostrar todos los postgrados y sus datos.	

Módulo act.

Tabla 38 Tarea 1 del módulo act.

Tarea	
Número de la tarea: 4	Número de HU: 3
Nombre de la tarea: Creación de la tabla actividad en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a actividad.	

Tabla 39 Tarea 2 del módulo act.

Tarea	
Número de la tarea: 5	Número de HU: 3
Nombre de la tarea: Implementación de Gestionar Actividad.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 07-04-2009	Fecha fin: 08-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten adicionar, modificar y eliminar una actividad.	

Tabla 40 Tarea 3 del módulo act.

Tarea	
Número de la tarea: 6	Número de HU: 4
Nombre de la tarea: Implementación de Mostrar Actividades.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 08-04-2009	Fecha fin: 09-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten mostrar todas las actividades y sus datos.	

Módulo cursos.

Tabla 41 Tarea 1 del módulo cursos.

Tarea	
Número de la tarea: 7	Número de HU: 5
Nombre de la tarea: Creación de la tabla curso en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a curso.	

Tabla 42 Tarea 2 del módulo cursos.

Tarea	
Número de la tarea: 8	Número de HU: 5
Nombre de la tarea: Implementación de Gestionar Curso.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 09-04-2009	Fecha fin: 10-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten adicionar, modificar y eliminar un curso.	

Tabla 43 Tarea 3 del módulo cursos.

Tarea	
Número de la tarea: 9	Número de HU: 6
Nombre de la tarea: Implementación de Mostrar Cursos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 10-04-2009	Fecha fin: 11-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten mostrar todos los cursos y sus datos.	

Módulo becas.

Tabla 44 Tarea 1 del módulo becas.

Tarea	
Número de la tarea: 10	Número de HU: 7
Nombre de la tarea: Creación de la tabla beca en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a beca.	

Tabla 45 Tarea 2 del módulo becas.

Tarea	
Número de la tarea: 11	Número de HU: 7
Nombre de la tarea: Implementación de Gestionar Beca.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 11-04-2009	Fecha fin: 12-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten adicionar, modificar y eliminar una beca.	

Tabla 46 Tarea 3 del módulo becas.

Tarea	
Número de la tarea: 12	Número de HU: 8
Nombre de la tarea: Implementación de Mostrar Beca.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 12-04-2009	Fecha fin: 13-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten mostrar todas las becas y sus datos.	

Módulo dipl.

Tabla 47 Tarea 1 del módulo dipl.

Tarea	
Número de la tarea: 13	Número de HU: 9
Nombre de la tarea: Creación de la tabla diplomado en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a diplomado.	

Tabla 48 Tarea 2 del módulo dipl.

Tarea	
Número de la tarea: 14	Número de HU: 9
Nombre de la tarea: Implementación de Gestionar Diplomado.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 14-04-2009	Fecha fin: 15-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten adicionar, modificar y eliminar un diplomado.	

Tabla 49 Tarea 3 del módulo dipl.

Tarea	
Número de la tarea: 15	Número de HU: 10
Nombre de la tarea: Implementación de Mostrar Diplomados.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 15-04-2009	Fecha fin: 16-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten mostrar todos los diplomados y sus datos.	

Módulo polo.

Tabla 50 Tarea 1 del módulo polo.

Tarea	
Número de la tarea: 16	Número de HU: 12
Nombre de la tarea: Creación de la tabla polo en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a polo.	

Tabla 51 Tarea 2 del módulo polo.

Tarea	
Número de la tarea: 17	Número de HU: 12
Nombre de la tarea: Implementación de Gestionar Polo.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 16-04-2009	Fecha fin: 17-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten adicionar, modificar y eliminar un polo.	

4.3.2 Iteración 2

En esta iteración se implementan las historias de usuario con un nivel de complejidad alto y se continúan perfeccionando los módulos implementados en la iteración anterior.

Tabla 52 Módulos abordados en la Iteración 2.

Módulos.	Historias de Usuario.	Tiempo en Implementación (Ptos_Est).	
		Estimación.	Real.
prof.	Gestionar Profesor.	1	0.1
	Guardar Interés de Postgrado.*	1	0.5

Diseño, Implementación y Pruebas.

admin.	Autenticar.*	2	0.8
	Crear Matrícula de Postgrado.*	1	0.5
	Ver Listado de Profesores.	1	0.5

A continuación se describen las tareas efectuadas para cada uno de los módulos desarrollados en la iteración 2:

Módulo prof.

Tabla 53 Tarea 1 del módulo prof.

Tarea	
Número de la tarea: 1	Número de HU: 11
Nombre de la tarea: Creación de la tabla profesor en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 18-04-2009	Fecha fin: 19-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a profesor.	

Tabla 54 Tarea 2 del módulo prof.

Tarea	
Número de la tarea: 2	Número de HU: 11
Nombre de la tarea: Implementación de Gestionar Profesor.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha de inicio: 19-04-2009	Fecha fin: 20-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten adicionar, modificar y eliminar un profesor.	

Tabla 55 Tarea 3 del módulo prof.

Tarea	
Número de la tarea: 3	Número de HU: 16
Nombre de la tarea: Implementación de Guardar Interés de Postgrado.	

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 20-04-2009	Fecha fin:
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten que un profesor pueda expresar su interés en un determinado postgrado.	

Módulo admin.

Tabla 56 Tarea 1 del módulo admin.

Tarea	
Número de la tarea: 4	Número de HU: 15
Nombre de la tarea: Creación de la tabla roles en la Base de Datos	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a roles, donde se encuentran las personas relacionadas con el sistema y los permisos con que cuenta cada una de ellas.	

Tabla 57 Tarea 2 del módulo admin.

Tarea	
Número de la tarea: 5	Número de HU: 15
Nombre de la tarea: Implementación de Autenticar.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha de inicio: 21-04-2009	Fecha fin: 22-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades necesarias para que los usuarios puedan autenticarse y navegar por la aplicación de acuerdo con los niveles de acceso que les corresponde.	

Tabla 58 Tarea 3 del módulo admin.

Tarea	
Número de la tarea: 6	Número de HU: 17
Nombre de la tarea: Implementación de Crear Matrícula de Postgrado.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha de inicio: 22-04-2009	Fecha fin:
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades necesarias para que el administrador de la aplicación pueda matricular a los profesores en un postgrado determinado.	

Tabla 59 Tarea 4 del módulo admin.

Tarea	
Número de la tarea: 7	Número de HU: 19
Nombre de la tarea: Implementación de Ver Listado de Profesores.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha de inicio: 24-04-2009	Fecha fin: 25-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades necesarias para que se pueda visualizar el listado de profesores de acuerdo con la categoría que desee el usuario.	

4.3.3 Iteración 3

En esta iteración se implementan las historias de usuario con una complejidad media, se terminan de perfeccionar los módulos implementados en las iteraciones anteriores y se entrega la aplicación al cliente, lista para poner en funcionamiento.

Tabla 60 Módulos abordados en la Iteración 3.

Módulos.	Historias de Usuario.	Tiempo en Implementación (Ptos_Est).	
		Estimación.	Real.
admin.	Gestionar Plan de Superación.	0.2	0.1
	Mostrar Plan de Superación.	0.2	0.1
	Publicar Trabajos.	0.2	0.1
prof.	Gestionar Comentario.	1	0.2

A continuación se describen las tareas efectuadas para cada uno de los módulos desarrollados en la iteración 3:

Módulo admin.

Tabla 61 Tarea 5 del módulo admin.

Tarea	
Número de la tarea: 1	Número de HU: 13
Nombre de la tarea: Creación de la tabla plan_super en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a plan de superación.	

Tabla 62 Tarea 6 del módulo admin.

Tarea	
Número de la tarea: 2	Número de HU: 13
Nombre de la tarea: Implementación de Gestionar Plan de Superación.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 25-04-2009	Fecha fin: 26-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten adicionar, modificar y eliminar un plan de superación.	

Tabla 63 Tarea 7 del módulo admin.

Tarea	
Número de la tarea: 3	Número de HU: 14
Nombre de la tarea: Implementación de Mostrar Plan de Superación.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 27-04-2009	Fecha fin: 28-04-2009
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten mostrar todos los planes de superación y sus datos.	

Tabla 64 Tarea 8 del módulo admin.

Tarea	
Número de la tarea: 4	Número de HU: 18
Nombre de la tarea: Creación de la tabla trabajo en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a trabajos donde se guardan, el nombre del trabajo, una síntesis de este y la dirección para acceder a él.	

Tabla 65 Tarea 9 del módulo admin.

Tarea	
Número de la tarea: 5	Número de HU: 18
Nombre de la tarea: Implementación de Publicar Trabajos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 28-04-2009	Fecha fin:
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten introducir en la aplicación los datos de un trabajo a publicar y mostrar estos datos cuando sean solicitados por un usuario.	

Tabla 66 Tarea 4 del módulo prof.

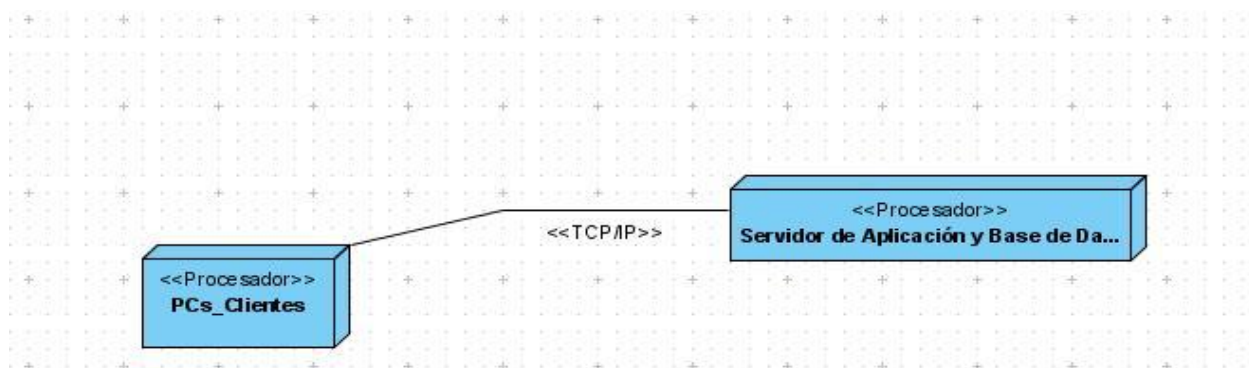
Tarea	
Número de la tarea: 6	Número de HU: 20
Nombre de la tarea: Creación de la tabla comentario en la Base de Datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05-03-2009	Fecha fin: 05-04-2009
Programador responsable: Addiel Rodríguez Guadarrama.	
Descripción: Se crea en la BD la tabla correspondiente a comentarios donde se guarda el texto del comentario y un identificador del usuario que lo emitió.	

Tabla 67 Tarea 5 del módulo prof.

Tarea	
Número de la tarea: 7	Número de HU: 20
Nombre de la tarea: Implementación Gestionar Comentario.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.2
Fecha de inicio: 29-04-2009	Fecha fin:
Programador responsable: Ariel Yasel Guillén Bravo.	
Descripción: Se implementan las funcionalidades que permiten a un usuario emitir un comentario sobre un trabajo publicado y este pueda ser revisado por el administrador de la aplicación para su posterior publicación.	

4.3.4 Diagrama de Despliegue.

El diagrama de despliegue permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes de la aplicación. En este caso la aplicación se encuentra hospedada en un servidor Web (apache) y maneja los datos persistentes sobre un sistema gestor de base de datos (PostgreSQL). El servidor web y el de base de datos están alojados en la misma máquina, ya que la Facultad 8 cuenta con escasos recursos para poner en funcionamiento la aplicación.



4.4 Pruebas.

Uno de los pilares fundamentales de la metodología XP lo constituye el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente. Mediante esta filosofía se minimiza la cantidad de errores no detectados y también el tiempo transcurrido entre la introducción de éste en la aplicación y su detección. Todo esto contribuye a elevar la calidad del producto y a la seguridad de los desarrolladores a la hora de hacer cambios o modificaciones. La metodología XP divide las pruebas en dos grupos:

- ❖ Pruebas unitarias, desarrolladas por los programadores y encargadas de verificar el código de forma automática.
- ❖ Pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

4.4.1 Pruebas de Aceptación.

Las pruebas de aceptación son pruebas que se crean a partir de las historias de usuario. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas pruebas es garantizar que los requerimientos hayan sido cumplidos y que la aplicación es realmente lo que el cliente quería. Una HU no se considera terminada hasta que no ha pasado sus pruebas de aceptación.

Iteración 1:

Tabla 68 Prueba 1 al módulo post.

Caso de Prueba de Aceptación.	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Probar la funcionalidad Insertar Postgrado.	
Descripción: Probar que pueda ser insertado un postgrado en el sistema.	
Condiciones de Ejecución:	
El usuario debe estar registrado y tener permisos de administrador.	

Entrada/Pasos de Ejecución:
Se intenta insertar un postgrado con los datos válidos.
Resultado Esperado:
Se inserta el postgrado correctamente.
Evaluación de la Prueba: Satisfactoria.

Tabla 69 Prueba 2 al módulo post.

Caso de Prueba de Aceptación.	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Probar la funcionalidad Insertar Postgrado.	
Descripción: Probar que no se pueda insertar un postgrado en el sistema con datos no válidos o un campo en blanco.	
Condiciones de Ejecución:	
El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución:	
Se intenta insertar un postgrado con datos no válidos y campos en blanco.	
Resultado Esperado:	
Que el sistema indique qué datos no son válidos y qué campos se dejaron en blanco y la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 70 Prueba 3 al módulo post.

Caso de Prueba de Aceptación.	
Código: HU1_P3	Historia de Usuario: 1
Nombre: Probar la funcionalidad Modificar Postgrado.	
Descripción: Probar que se puedan modificar los datos de un postgrado en el sistema.	
Condiciones de Ejecución:	
El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución:	
Se intentan modificar los datos de un postgrado con los datos válidos.	

Resultado Esperado: Se modifican los datos del postgrado.
Evaluación de la Prueba: Satisfactoria.

Tabla 71 Prueba 4 al módulo post.

Caso de Prueba de Aceptación.	
Código: HU1_P4	Historia de Usuario: 1
Nombre: Probar la funcionalidad Modificar Postgrado.	
Descripción: Probar que se puedan modificar los datos de un postgrado en el sistema con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un postgrado con datos no válidos.	
Resultado Esperado: Que el sistema indique que datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 72 Prueba 5 al módulo post.

Caso de Prueba de Aceptación.	
Código: HU1_P5	Historia de Usuario: 1
Nombre: Probar la funcionalidad Eliminar Postgrado.	
Descripción: Probar que pueda ser eliminado un postgrado del sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta eliminar un postgrado.	
Resultado Esperado: Se elimina el postgrado del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 73 Prueba 6 al módulo post.

Caso de Prueba de Aceptación.	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Probar la funcionalidad Mostrar Postgrado.	
Descripción: Probar que se puedan mostrar los datos de un postgrado.	
Condiciones de Ejecución: El usuario debe estar registrado.	
Entrada/Pasos de Ejecución: Se intenta mostrar los datos de un postgrado.	
Resultado Esperado: Se muestran los datos del postgrado.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 74 Prueba 1 al módulo act.

Caso de Prueba de Aceptación.	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Probar la funcionalidad Insertar Actividad.	
Descripción: Probar que pueda ser insertada una actividad en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar una actividad con los datos válidos.	
Resultado Esperado: Se inserta la actividad correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 75 Prueba 2 al módulo act.

Caso de Prueba de Aceptación.	
Código: HU3_P2	Historia de Usuario: 3
Nombre: Probar la funcionalidad Insertar Actividad.	
Descripción: Probar que no se pueda insertar una actividad en el sistema con datos no válidos o campos en blanco.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar una actividad con datos no válidos y campos en blanco.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y qué campos se dejaron en blanco y la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 76 Prueba 3 al módulo act.

Caso de Prueba de Aceptación.	
Código: HU3_P3	Historia de Usuario: 3
Nombre: Probar la funcionalidad Modificar Actividad.	
Descripción: Probar que se puedan modificar los datos de una actividad en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de una actividad con los datos válidos.	
Resultado Esperado: Se modifican los datos de la actividad.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 77 Prueba 4 al módulo act.

Caso de Prueba de Aceptación.	
Código: HU3_P4	Historia de Usuario: 3
Nombre: Probar la funcionalidad Modificar Actividad.	
Descripción: Probar que se puedan modificar los datos de una actividad en el sistema con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de una actividad con datos no válidos.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 78 Prueba 5 al módulo act.

Caso de Prueba de Aceptación.	
Código: HU3_P5	Historia de Usuario: 3
Nombre: Probar la funcionalidad Eliminar Actividad.	
Descripción: Probar que pueda ser eliminada una actividad del sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta eliminar una actividad.	
Resultado Esperado: Se elimina la actividad del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 79 Prueba 6 al módulo act.

Caso de Prueba de Aceptación.	
Código: HU4_P1	Historia de Usuario: 4
Nombre: Probar la funcionalidad Mostrar Actividad.	
Descripción: Probar que se puedan mostrar los datos de una actividad.	
Condiciones de Ejecución: El usuario debe estar registrado.	
Entrada/Pasos de Ejecución: Se intenta mostrar los datos de una actividad.	
Resultado Esperado: Se muestran los datos de la actividad.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 80 Prueba 1 al módulo cursos.

Caso de Prueba de Aceptación.	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Probar la funcionalidad Insertar Curso.	
Descripción: Probar que pueda ser insertado un curso en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un curso con los datos válidos.	
Resultado Esperado: Se inserta el curso correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 81 Prueba 2 al módulo cursos.

Caso de Prueba de Aceptación.	
Código: HU5_P2	Historia de Usuario: 5
Nombre: Probar la funcionalidad Insertar Curso.	
Descripción: Probar que no se pueda insertar un curso en el sistema con datos no válidos o campos en blanco.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un curso con datos no válidos y campos en blanco.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y qué campos se dejaron en blanco y la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 82 Prueba 3 al módulo cursos.

Caso de Prueba de Aceptación.	
Código: HU5_P3	Historia de Usuario: 5
Nombre: Probar la funcionalidad Modificar Curso.	
Descripción: Probar que se puedan modificar los datos de un curso en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un curso con los datos válidos.	
Resultado Esperado: Se modifican los datos del curso	
Evaluación de la Prueba: Satisfactoria.	

Tabla 83 Prueba 4 al módulo cursos.

Caso de Prueba de Aceptación.	
Código: HU5_P4	Historia de Usuario: 5
Nombre: Probar la funcionalidad Modificar Curso.	
Descripción: Probar que se puedan modificar los datos de un curso en el sistema con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un curso con datos no válidos.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 84 Prueba 5 al módulo cursos.

Caso de Prueba de Aceptación.	
Código: HU5_P5	Historia de Usuario: 5
Nombre: Probar la funcionalidad Eliminar Curso.	
Descripción: Probar que pueda ser eliminado un curso del sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta eliminar un curso.	
Resultado Esperado: Se elimina el curso del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 85 Prueba 6 al módulo cursos.

Caso de Prueba de Aceptación.	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Probar la funcionalidad Mostrar Curso.	
Descripción: Probar que se puedan mostrar los datos de un curso.	
Condiciones de Ejecución: El usuario debe estar registrado.	
Entrada/Pasos de Ejecución: Se intenta mostrar los datos de un curso	
Resultado Esperado: Se muestran los datos del curso.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 86 Prueba 1 al módulo becas.

Caso de Prueba de Aceptación.	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Probar la funcionalidad Insertar Beca.	
Descripción: Probar que pueda ser insertado una beca en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar una beca con los datos válidos.	
Resultado Esperado: Se inserta la beca correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 87 Prueba 2 al módulo becas.

Caso de Prueba de Aceptación.	
Código: HU7_P2	Historia de Usuario: 7
Nombre: Probar la funcionalidad Insertar Beca.	
Descripción: Probar que no se pueda insertar una beca en el sistema con datos no válidos o campos en blanco.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar una beca con datos no válidos y campos en blanco.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y qué campos se dejaron en blanco y la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 88 Prueba 3 al módulo becas.

Caso de Prueba de Aceptación.	
Código: HU7_P3	Historia de Usuario: 7
Nombre: Probar la funcionalidad Modificar Beca.	
Descripción: Probar que se puedan modificar los datos de una beca en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de una beca con los datos válidos.	
Resultado Esperado: Se modifican los datos de la beca.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 89 Prueba 4 al módulo becas.

Caso de Prueba de Aceptación.	
Código: HU7_P4	Historia de Usuario: 7
Nombre: Probar la funcionalidad Modificar Beca.	
Descripción: Probar que se puedan modificar los datos de una beca en el sistema con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de una beca con datos no válidos.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 90 Prueba 5 al módulo becas.

Caso de Prueba de Aceptación.	
Código: HU7_P5	Historia de Usuario: 7
Nombre: Probar la funcionalidad Eliminar Beca.	
Descripción: Probar que pueda ser eliminado una beca del sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta eliminar una beca.	
Resultado Esperado: Se elimina la beca del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 91 Prueba 6 al módulo becas.

Caso de Prueba de Aceptación.	
Código: HU8_P1	Historia de Usuario: 8
Nombre: Probar la funcionalidad Mostar Beca.	
Descripción: Probar que se puedan mostrar los datos de una beca.	
Condiciones de Ejecución: El usuario debe estar registrado.	
Entrada/Pasos de Ejecución: Se intenta mostrar los datos de una beca.	
Resultado Esperado: Se muestran los datos de la beca.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 92 Prueba 1 al módulo dipl.

Caso de Prueba de Aceptación.	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Probar la funcionalidad Insertar Diplomado.	
Descripción: Probar que pueda ser insertado un diplomado en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un diplomado con los datos válidos.	
Resultado Esperado: Se inserta el diplomado correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 93 Prueba 2 al módulo dipl.

Caso de Prueba de Aceptación.	
Código: HU9_P2	Historia de Usuario: 9
Nombre: Probar la funcionalidad Insertar Diplomado.	
Descripción: Probar que no se pueda insertar un diplomado en el sistema con datos no válidos o campos en blanco.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un diplomado con datos no válidos y campos en blanco.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y qué campos se dejaron en blanco y dé la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 94 Prueba 3 al módulo dipl.

Caso de Prueba de Aceptación.	
Código: HU9_P3	Historia de Usuario: 9
Nombre: Probar la funcionalidad Modificar Diplomado.	
Descripción: Probar que se puedan modificar los datos de un diplomado en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un diplomado con los datos válidos.	
Resultado Esperado: Se modifican los datos del diplomado.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 95 Prueba 4 al módulo dipl.

Caso de Prueba de Aceptación.	
Código: HU9_P4	Historia de Usuario: 9
Nombre: Probar la funcionalidad Modificar Diplomado.	
Descripción: Probar que se puedan modificar los datos de un diplomado en el sistema con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un diplomado con datos no válidos.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 96 Prueba 5 al módulo dipl.

Caso de Prueba de Aceptación.	
Código: HU9_P5	Historia de Usuario: 9
Nombre: Probar la funcionalidad Eliminar Diplomado.	
Descripción: Probar que pueda ser eliminado un diplomado del sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta eliminar un diplomado.	
Resultado Esperado: Se elimina el diplomado del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 97 Prueba 6 al módulo dipl.

Caso de Prueba de Aceptación.	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Probar la funcionalidad Mostrar Diplomado.	
Descripción: Probar que se puedan mostrar los datos de un diplomado.	
Condiciones de Ejecución: El usuario debe estar registrado.	
Entrada/Pasos de Ejecución: Se intenta mostrar los datos de un diplomado.	
Resultado Esperado: Se muestran los datos del diplomado.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 98 Prueba 1 al módulo polo.

Caso de Prueba de Aceptación.	
Código: HU12_P1	Historia de Usuario: 12
Nombre: Probar la funcionalidad Insertar Polo.	
Descripción: Probar que pueda ser insertado un polo en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un polo con los datos válidos.	
Resultado Esperado: Se inserta el polo correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 99 Prueba 2 al módulo polo.

Caso de Prueba de Aceptación.	
Código: HU12_P2	Historia de Usuario: 12
Nombre: Probar la funcionalidad Insertar Polo.	
Descripción: Probar que no se pueda insertar un polo en el sistema con datos no válidos o campos en blanco.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un polo con datos no válidos y campos en blanco.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y qué campos se dejaron en blanco y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 100 Prueba 3 al módulo polo.

Caso de Prueba de Aceptación.	
Código: HU12_P3	Historia de Usuario: 12
Nombre: Probar la funcionalidad Modificar Polo.	
Descripción: Probar que se puedan modificar los datos de un polo en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un polo con los datos válidos.	
Resultado Esperado: Se modifican los datos del polo.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 101 Prueba 4 al módulo polo.

Caso de Prueba de Aceptación.	
Código: HU12_P4	Historia de Usuario: 12
Nombre: Probar la funcionalidad Modificar Polo.	
Descripción: Probar que se puedan modificar los datos de un polo en el sistema con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un polo con datos no válidos.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 102 Prueba 5 al módulo polo.

Caso de Prueba de Aceptación.	
Código: HU12_P5	Historia de Usuario: 12
Nombre: Probar la funcionalidad Eliminar Polo.	
Descripción: Probar que pueda ser eliminado un polo del sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta eliminar un polo.	
Resultado Esperado: Se elimina el polo del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Iteración 2:

Tabla 103 Prueba 1 al módulo prof.

Caso de Prueba de Aceptación.	
Código: HU11_P1	Historia de Usuario: 11
Nombre: Probar la funcionalidad Insertar Profesor.	
Descripción: Probar que pueda ser insertado un profesor en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un profesor con los datos válidos.	
Resultado Esperado: Se inserta el profesor correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 104 Prueba 2 al módulo prof.

Caso de Prueba de Aceptación.	
Código: HU11_P2	Historia de Usuario: 11
Nombre: Probar la funcionalidad Insertar Profesor.	
Descripción: Probar que no se pueda insertar un profesor en el sistema con datos no válidos o campos en blanco.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un profesor con datos no válidos y campos en blanco.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y qué campos se dejaron en blanco y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 105 Prueba 3 al módulo prof.

Caso de Prueba de Aceptación.	
Código: HU11_P3	Historia de Usuario: 11
Nombre: Probar la funcionalidad Modificar Profesor.	
Descripción: Probar que se puedan modificar los datos de un profesor en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un profesor con los datos válidos.	
Resultado Esperado: Se modifican los datos del profesor.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 106 Prueba 4 al módulo prof.

Caso de Prueba de Aceptación.	
Código: HU11_P4	Historia de Usuario: 11
Nombre: Probar la funcionalidad Modificar Profesor.	
Descripción: Probar que se puedan modificar los datos de un profesor en el sistema con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un profesor con datos no válidos.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 107 Prueba 5 al módulo prof.

Caso de Prueba de Aceptación.	
Código: HU11_P5	Historia de Usuario: 11
Nombre: Probar la funcionalidad Eliminar Profesor.	
Descripción: Probar que pueda ser eliminado un profesor del sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta eliminar un profesor.	
Resultado Esperado: Se elimina el profesor del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 108 Prueba 1 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU15_P1	Historia de Usuario: 15
Nombre: Probar la funcionalidad Autenticar.	
Descripción: Probar que se pueda autenticar un usuario en el sistema con datos válidos.	
Condiciones de Ejecución: Que se esté ejecutando la aplicación normalmente.	
Entrada/Pasos de Ejecución: Intentar acceder al sistema con un usuario y una clave válidos.	
Resultado Esperado: Que el usuario pueda acceder a las funcionalidades del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 109 Prueba 2 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU15_P2	Historia de Usuario: 15
Nombre: Probar la funcionalidad Autenticar.	
Descripción: Probar que se pueda autenticar un usuario en el sistema con datos no válidos.	
Condiciones de Ejecución: Que se esté ejecutando la aplicación normalmente.	
Entrada/Pasos de Ejecución: Intentar acceder al sistema con un usuario y/o una clave no válidos.	
Resultado Esperado: Que se le informe al usuario que su usuario y/o clave no son válidos, y se le dé la posibilidad de intentarlo nuevamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 110 Prueba 3 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU_P1	Historia de Usuario:
Nombre: Probar la funcionalidad Ver Listado de Profesores.	
Descripción: Probar que se pueda mostrar un listado de profesores de acuerdo a un criterio que desee el usuario.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: El usuario intenta obtener un listado de profesores de acuerdo al criterio que selecciona.	
Resultado Esperado: Que el sistema muestre el listado de los profesores que cumplen con el criterio seleccionado por el usuario.	
Evaluación de la Prueba: Satisfactoria.	

Iteración 3:

Tabla 111 Prueba 4 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU13_P1	Historia de Usuario: 13
Nombre: Probar la funcionalidad Insertar Plan de Superación	
Descripción: Probar que pueda ser insertado un plan de superación en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un plan de superación con los datos válidos.	
Resultado Esperado: Se inserta el plan de superación correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 112 Prueba 5 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU13_P2	Historia de Usuario: 13
Nombre: Probar la funcionalidad Insertar Plan de Superación.	
Descripción: Probar que no se pueda insertar un plan de superación en el sistema con datos no válidos o campos en blanco.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta insertar un plan de superación con datos no válidos y campos en blanco.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y qué campos se dejaron en blanco y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 113 Prueba 6 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU13_P3	Historia de Usuario: 13
Nombre: Probar la funcionalidad Modificar Plan de Superación.	
Descripción: Probar que se puedan modificar los datos de un plan de superación en el sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de un plan de superación con los datos válidos.	
Resultado Esperado: Se modifican los datos del plan de superación.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 114 Prueba 7 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU13_P4	Historia de Usuario: 13
Nombre: Probar la funcionalidad Modificar Plan de Superación.	
Descripción: Probar que se puedan modificar los datos de un plan de superación en el sistema con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intentan modificar los datos de plan de superación con datos no válidos.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 115 Prueba 8 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU13_P5	Historia de Usuario: 13
Nombre: Probar la funcionalidad Eliminar Plan de Superación.	
Descripción: Probar que pueda ser eliminado un plan de superación del sistema.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta eliminar un plan de superación.	
Resultado Esperado: Se elimina el plan de superación del sistema.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 116 Prueba 9 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU14_P1	Historia de Usuario: 14
Nombre: Probar la funcionalidad Mostrar Plan de Superación.	
Descripción: Probar que se puedan mostrar los datos de un plan de superación.	
Condiciones de Ejecución: El usuario debe estar registrado.	
Entrada/Pasos de Ejecución: Se intenta mostrar los datos de un plan de superación.	
Resultado Esperado: Se muestran los datos del plan de superación.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 117 Prueba 10 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU18_P1	Historia de Usuario: 18
Nombre: Probar la funcionalidad Publicar Trabajos.	
Descripción: Probar la funcionalidad publicar trabajos con datos válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta publicar un nuevo trabajo entrando los datos válidos.	
Resultado Esperado: Que se publique el trabajo correctamente.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 118 Prueba 11 al módulo admin.

Caso de Prueba de Aceptación.	
Código: HU18_P1	Historia de Usuario: 18
Nombre: Probar la funcionalidad Publicar Trabajos.	
Descripción: Probar la funcionalidad publicar trabajos con datos no válidos.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Se intenta publicar un nuevo trabajo entrando los datos no válidos.	
Resultado Esperado: Que el sistema indique qué datos no son válidos y de la posibilidad de corregirlos.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 119 Prueba 6 al módulo prof.

Caso de Prueba de Aceptación.	
Código: HU20_P1	Historia de Usuario: 20
Nombre: Probar la funcionalidad Enviar Comentario.	
Descripción: Probar que puede ser enviado un comentario.	
Condiciones de Ejecución: El usuario debe estar registrado.	
Entrada/Pasos de Ejecución: Se intenta enviar un comentario.	
Resultado Esperado: Que el comentario sea almacenado correctamente en la BD.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 120 Prueba 7 al módulo prof.

Caso de Prueba de Aceptación.	
Código: HU20_P3	Historia de Usuario: 20
Nombre: Probar la funcionalidad Revisar Comentario.	
Descripción: Probar que puede ser corregido un comentario.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Intentar corregir un comentario.	
Resultado Esperado: Que el sistema salve los cambios que le hizo el usuario al comentario en la BD.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 121 Prueba 8 al módulo prof.

Caso de Prueba de Aceptación.	
Código: HU20_P3	Historia de Usuario: 20
Nombre: Probar la funcionalidad Publicar Comentario.	
Descripción: Probar que puede ser Publicado un comentario.	
Condiciones de Ejecución: El usuario debe estar registrado y tener permisos de administrador.	
Entrada/Pasos de Ejecución: Intentar publicar el comentario.	
Resultado Esperado: Que el sistema publique el comentario con sus datos correctamente.	
Evaluación de la Prueba: Satisfactoria.	

4.5 Conclusiones.

En este capítulo se hizo alusión a las fases de Diseño, Codificación y Pruebas del proceso de desarrollo de la aplicación exponiendo los artefactos generados en cada una de estas fases y describiendo cada uno de ellos.

Conclusiones.

La gestión de información hoy en día se ha convertido en una tarea de primer orden en cualquier institución. Dado el rápido avance de las tecnologías informáticas y el grado de aceptación de estas en todo el mundo, se hace prácticamente imprescindible gestionar la información de forma digital, debido a las facilidades que brindan estas tecnologías para la gestión y almacenamiento de información.

Durante el desarrollo de este trabajo se realizó un estudio sobre la situación actual y las tendencias de sistemas similares. Se demostró la necesidad de crear un sistema que gestione la información resultante del proceso de superación postgraduada en la Facultad 8 de la UCI presentando una aplicación web que permite la publicación de postgrados, planes, cursos y actividades de superación así como gestionar la información que generan los profesores en el cumplimiento de estos planes de superación. Además de la posibilidad de que un profesor pueda consultar su situación con respecto a la superación en cualquier momento que lo desee.

Recomendaciones.

Como resultado del proceso de investigación y realización de la aplicación han surgido ideas que serían recomendables tener en cuenta para un futuro perfeccionamiento del sistema, a continuación se listan las mismas.

- ❖ Continuar agregándole funcionalidades al sistema, de acuerdo a las expectativas de los usuarios finales del mismo.
- ❖ Integrar este sistema con otros que se desarrollan en la Facultad 8 de la UCI para crear un sistema mayor, capaz de gestionar toda la información de los docentes de esta facultad.

Referencias Bibliográficas.

1. Qué es JSP. [En línea] [Citado el: 12 de 12 de 2008.]
<http://www.desarrolloweb.com/articulos/831.php>.
2. ¿Qué es la programación web? [En línea] [Citado el: 12 de 12 de 2008.]
<http://www.bahiatc.com/programacionweb/programacionweb.php>.
3. **Delgado, Lic. Yanoski Calderín.** GESTACAD. SISTEMA PARA LA GESTIÓN ACADÉMICA . [En línea] [Citado el: 12 de 12 de 2008.]
<http://74.125.47.132/search?q=cache:b3t53aO2SnUJ:www.posgrados.frc.utn.edu.ar/congreso/trabajos/7.doc+GESTACAD%2Bmatanzaz&cd=2&hl=es&ct=clnk&gl=cu>.
4. Internet Information Server. [En línea] [Citado el: 02 de 02 de 2009.]
<http://www.adrformacion.com/curso/intranet/leccion1/InternetInformationServer.htm> .
5. UML y RUP. [En línea] Departamento de Ingeniería de Software. Universidad de las Ciencias Informáticas. [Citado el: 02 de 02 de 2009.] http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_CTP_1/Material_Clase_Teorico_Practica_1.doc.
6. **Software, Departamento de Ingeniería de.** *UML y RUP*. Ciudad de la Habana : s.n., 2007.
7. Introducción al Lenguaje de Programación Python. [En línea] [Citado el: 10 de 1 de 2009.]
<http://gcubo.org/documentacion/charlasirc/python>.
8. The Apache Software Foundation. [En línea] [Citado el: 12 de 01 de 2009.] <http://www.apache.org/>.
9. Una Introducción a APACHE. [En línea] [Citado el: 15 de 12 de 2008.]
http://linux.ciberaula.com/articulo/linux_apache_intro/.
10. Sistemas Gestores de Bases de Datos. [En línea] <http://www.slideshare.net/alexmerono/sistemas-gestores-de-bases-de-datos>.
11. Introducción a MySQL. [En línea] <http://www.webestilo.com/mysql/intro.phtml>.
12. Información general del producto SQL Server 2005. [En línea]
<http://www.microsoft.com/spain/sql/productinfo/overview/default.msp>.
13. PostgreSQL: About. [En línea] <http://www.postgresql.org/about/>.
14. BR Consulting - Extreme Programming - XP . [En línea]
<http://brconsulting.info/portal/articulos/metodologias-de-desarrollo/extreme-programming---xp.html>.
15. Guía Breve de Servicios Web. [En línea] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
16. Atenea tech - Blog Archive Frameworks, frameworks y más frameworks. [En línea]
<http://ateneatech.com/blog/frameworks-frameworks-y-mas-frameworks#more-15>.

17. Los Frameworks de PHP agilizan tu trabajo. [En línea] 2007.
<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.
18. El framework Symfony, una introducción práctica (I parte). [En línea]
<http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.
19. CakePhp Framework. [En línea]
http://www.alltechinfo.biz/index2.php?option=com_content&do_pdf=1&id=28.
20. Arquitectura de Sistemas Informáticos . [En línea]
<http://metodologiasdesistemas.blogspot.com/2007/05/diseo-en-3-capas-fisicas-lgicas-es.html>. 20.
21. Modelo Vista Controlador Un patrón de arquitectura de software que separa los datos de una aplicación... [En línea] <http://buenmaster.com/?a=536>.

Bibliografía.

1. 10 razones por las que usar Symfony en la empresa (Framework PHP). [En línea] [Citado el: 12 de 03 de 2009.] <http://www.bgta.net/2008/04/29/10-razones-por-las-que-usar-symfony-en-la-empresa-framework-php/>.
2. Symfony 1.0, la guía definitiva » Capítulo 2. Explorando el interior de Symfony » 2.1. El patrón MVC. [En línea] [Citado el: 17 de 03 de 2009.] http://www.librosweb.es/symfony_1_0/capitulo2/el_patron_mvc.html.
3. [En línea] [Citado el: 14 de 02 de 2009.] <http://oness.sourceforge.net/proyecto/html/ch05.html>.
4. El Proceso Unificado de Desarrollo de Software (RUP). [En línea] [Citado el: 06 de 02 de 2009.] <http://yaqui.mx/uabc.mx/%7Emolguin/as/RUP.htm>.
5. Herramientas adecuadas para el diseño y desarrollo de un sitio web. [En línea] [Citado el: 24 de 01 de 2009.] <http://www.maestrosdelweb.com/editorial/herramientas-adecuadas-para-el-diseno-y-desarrollo-de-un-sitio-web/>.
6. Una Introducción a Apache. [En línea] 2006. [Citado el: 16 de 02 de 2009.] http://linux.ciberaula.com/articulo/linux_apache_intro/.
7. **Pérez, Javier Eguíluz.** Introducción al AJAX. [En línea] [Citado el: 18 de 03 de 2009.] <http://www.librosweb.es/ajax/index.html>.
8. —. Introducción al JavaScript. [En línea] [Citado el: 20 de 03 de 2009.] <http://www.librosweb.es/javascript/capitulo1.html>.
9. **Quiñones, Ernesto.** Introducción al PostgreSQL. [En línea] [Citado el: 18 de 02 de 2009.] http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf.
10. AJAX un nuevo acercamiento a Aplicaciones Web. [En línea] [Citado el: 19 de 02 de 2009.] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>.