



Instituto Superior Minero Metalúrgico

Dr. “Antonio Núñez Jiménez”

Facultad Geología-Minas

Trabajo de Diploma

en opción al título de

Ingeniero Informático

Sistema de Apoyo a la Gestión Académica para la
Enseñanza Pre-Universitaria y Técnica Profesional

Autor: Alexander Garcés Valdés

Tutor: Ing. José Rolando Pérez Sandó

Moa, 2012

“Año 54 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al pre universitario “Antonio López Fernández” para que haga el uso que estime pertinente con este trabajo.

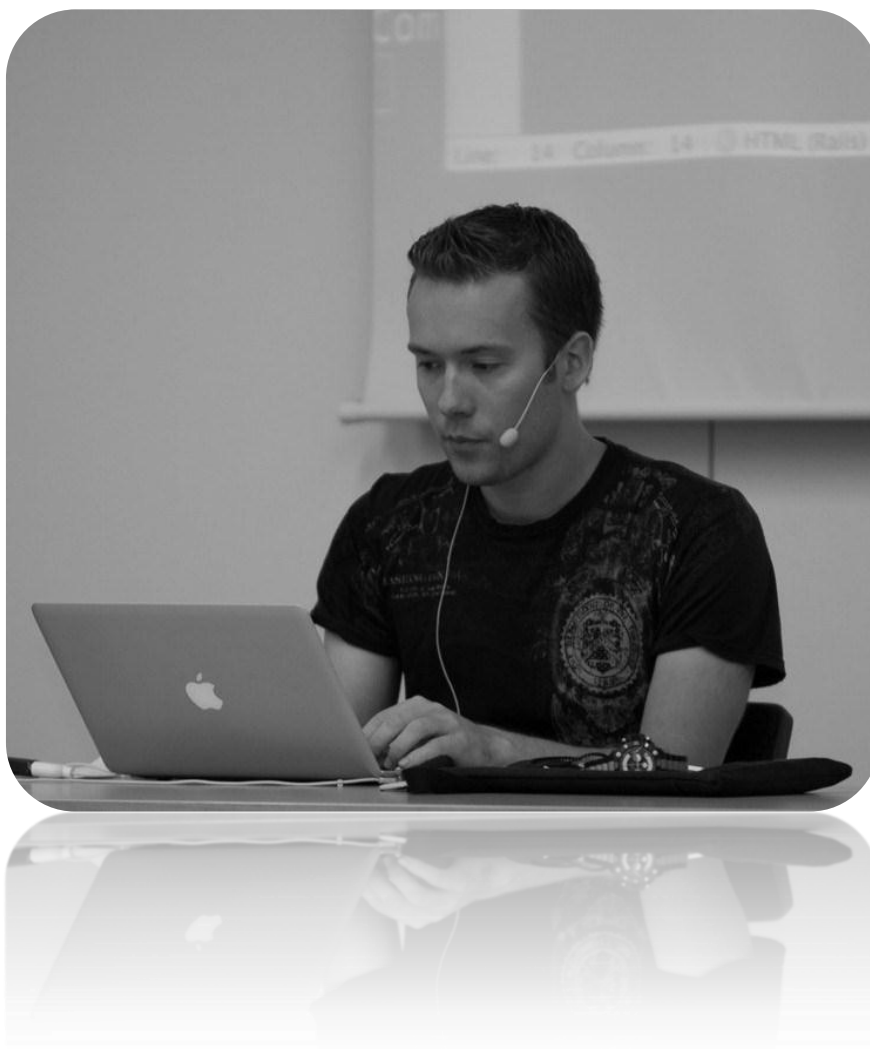
Para que así conste firmamos la presente a los _____ días del mes de _____
del año 2012.

Alexander Garcés Valdés

Firma autor

Ing. José Rolando Pérez Sandó

Firma tutor



La mejor forma de predecir el futuro es implementarlo

David Heinemeier Hansson
Creador de Ruby on Rails

DEDICATORIA

A mami y papí por estar siempre ahí y apoyarme en todo.

A mi hermanita querida.

A mi novia que siempre ha estado a mi lado.

A Frank!!!!, Daniela!!!! y Eloy David mis sobrinos malcriados del alma.

A mis abuelas mami y mamá, saben que las quiero con mi vida.

*A mis abuelos papí y papá que aunque no estén ya físicamente siempre estarán
conmigo.*

A mis suegros Celina y Eloy.

AGRADECIMIENTOS

*A las personas más importantes en mi vida mami, papi y legni, por ellos estoy
aquí. . . .un millón de gracias.*

*A mi novia querida.sin tí este trabajo nunca se hubiese podido realizar. . . tú
lo sabes.*

A mis abuelas y abuelos por mimar a su nieto.

A Celina, Eloy, al cuña Tito y a la familia de Santiago.

A mi hermano Rubenis.

A la gente de Cueto y Vuelta Larga.

A los abu de la universidad: Evelinaga y Charlie “La bala (salva)”.

A Adrian “el cebú”, a pablo lechusay, a eddyto “el guá” o “la aleta”.

A mi grupo de info.

A mi tutor por guiarme en este proyecto.

A mi Cuba, aunque nunca vayan a un mundial de fútbol.

RESUMEN

Los avances en las tecnologías de la información y las comunicaciones están acercando cada vez más la información al usuario final, así como facilitando su procesamiento. Nuestro país se ha ido insertando poco a poco en este mundo tecnológico.

En el pre universitario “Antonio López Fernández” del municipio de Moa, se maneja un gran volumen de información, la cual genera mucha documentación y permite que el proceso de gestión de reportes y académico de manera general sea lento y tedioso. Por esto el siguiente trabajo trae como objetivo el desarrollo del sistema informático “ACADÉMICA”, desarrollado sobre los lenguajes de programación web PHP y JavaScript.

El uso de este sistema agilizará y facilitará la gestión académica en el pre universitario “Antonio López Fernández”. También permitirá realizar diferentes reportes a estudiantes con la seguridad de que no se cometan errores.

SUMMARY

Advances in information technology and communications are getting closer to the end user information, and facilitating processing. Our country has been gradually inserted in this technological world.

In the pre university "Antonio Lopez Fernandez" the municipality of Moa, it handles a large volume of information, which generates a lot of documentation and allows the management process and academic reports generally slow and tedious. For this work brings the following development objective of the computer system "ACADEMIC" developed on web programming languages PHP and JavaScript.

Using this system will expedite and facilitate the academic administration in the pre university "Antonio Lopez Fernandez." Different reports also enable students with the assurance that no mistakes are made.

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	5
1.1 ESTADO DEL ARTE	5
1.1.1 ÁMBITO INTERNACIONAL	5
1.1.2 ÁMBITO NACIONAL.....	6
1.2 TECNOLOGÍAS Y HERRAMIENTAS A UTILIZAR.....	6
1.2.1 APLICACIÓN WEB	6
1.2.2 SERVIDOR WEB	7
1.2.3 LENGUAJES DE PROGRAMACIÓN WEB.....	8
1.2.4 FRAMEWORK DE DESARROLLO WEB.....	10
1.2.5 SISTEMAS GESTORES DE BASES DE DATOS (SGBD) [12]	11
1.2.6 ARQUITECTURA	13
1.2.7 METODOLOGÍA DE DESARROLLO DE SOFTWARE	15
1.2.8 HERRAMIENTAS CASE.....	19
1.2.9 LENGUAJE UNIFICADO DE MODELADO (UML) [23]	21
CAPÍTULO 2 PLANIFICACIÓN Y DISEÑO	23
2.1 LISTA DE RESERVA	23
2.2 HISTORIAS DE USUARIOS (HU) [25]	25
2.3 PLANIFICACIÓN DE ENTREGAS.....	27
2.3.1 ESTIMACIÓN DE ESFUERZO POR HU	28
2.3.2 PLAN DE ITERACIONES	28
2.4 CLASES, RESPONSABILIDADES Y COLABORACIÓN.....	29
CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBAS	31
3.1 MODELO DE DATOS	31
3.2 DESARROLLO DE LAS ITERACIONES.....	32
3.2.1 TAREAS DE INGENIERÍA POR HU	32
3.3 PRUEBAS	35
3.3.1 DESARROLLO DIRIGIDO POR PRUEBAS	35
3.3.2 PRUEBAS DE ACEPTACIÓN (PA)	36
CAPÍTULO 4 ESTUDIO DE FACTIBILIDAD	39
4.1 EVALUACIÓN COSTO-BENEFICIO	39
4.2 EFECTOS ECONÓMICOS	40
4.3 ELEMENTOS PARA IDENTIFICAR LOS COSTOS Y BENEFICIOS DEL PROYECTO	41
4.4 FACTIBILIDAD ECONÓMICA.....	42

4.5 EVALUACIÓN ECONÓMICA	42
4.5.1 BENEFICIOS TANGIBLES	44
4.5.2 BENEFICIOS Y COSTOS INTANGIBLES DEL PROYECTO	44
CONCLUSIONES GENERALES	47
RECOMENDACIONES	48
BIBLIOGRAFÍAS UTILIZADAS	49
OTRAS BIBLIOGRAFÍAS	51
ANEXOS	53
ANEXO 1 HISTORIAS DE USUARIO	53
ANEXO 2 TAREAS DE INGENIERÍA	56
ANEXO 3 TARJETAS CRC	68
ANEXO 4 PRUEBAS DE ACEPTACIÓN.....	74

Introducción

Hoy en día estamos asistiendo a una auténtica revolución. Los avances en las tecnologías de la información y las comunicaciones (TICs) están acercando la información al usuario final, así como facilitando su procesamiento. Uno de los cambios más importantes, tiene que ver con el soporte y canal de transmisión de la información. Internet y las Tecnologías Web, han conseguido que el usuario esté familiarizado con información hipermedia, incluyendo texto, imágenes, audio y vídeo.

World Wide Web, o simplemente Web, constituye uno de los intentos más recientes y a la vez más poderosos de sistematizar y simplificar el acceso a la información. Este nuevo sistema ha revolucionado la forma en que los usuarios se comunican y utilizan los servicios de la llamada "red de redes".

Nuestro país no está ajeno a todo esto y es por ello que las TICs han traído consigo un gran giro en el manejo de la información. Con el nuevo modelo económico que se lleva a cabo en Cuba, la informática y las nuevas TICs juegan un papel muy importante. Evidenciándose en las muchas instituciones que se han sumado a utilizar la informática para beneficio de sus procesos, especialmente en la gestión académica.

La gestión académica es de gran importancia para cualquier institución ya que a través de esta se lleva el control de cada estudiante. Se pueden realizar estudios sobre el comportamiento del aprendizaje en los educandos y planificar mejoras a través de los resultados de estos.

El Ministerio de Educación (MINED) ha realizado cambios en sus estilos de trabajo, respondiendo a las necesidades planteadas en el país. Dentro de estos cambios se encuentra la unión en varias instituciones de la Enseñanza Pre-Universitaria y Técnica Profesional. En el municipio Moa el Pre-Universitario "Antonio López Fernández" es uno de los centros involucrados en este proceso de cambio. El cual tiene como objetivo principal la preparación de los jóvenes en las diferentes materias para su posterior ingreso a la Educación Superior.

Esta institución cuenta con una matrícula considerable de estudiantes, además de los profesores involucrados en el proceso docente educativo. Lo que trae consigo que se maneje un gran volumen de información y se haga difícil llevar el control de la gestión académica de los alumnos.

Esta empieza desde que el estudiante ingresa en la institución hasta que se retira. El sistema de enseñanza está caracterizado por las constantes evaluaciones, las cuales son registradas para realizar reportes de estudiantes suspensos, semanales y mensuales a la dirección municipal y provincial de educación.

El secretario es quién recolecta las notas y los promedios de todos los docentes de forma manual, para luego realizar reportes. Los docentes generalmente no cumplen a tiempo con la entrega de las notas y promedios debido a la lentitud y complejidad del proceso, por lo que se demora además la entrega de los reportes al municipio y la provincia.

Con el aumento imponente en la matrícula de estudiantes se aumenta la carga de trabajo de los profesores lo que hace más complejo el proceso. También existe la posibilidad de que se cometan errores y se pierda información debido al deterioro o pérdida de los documentos.

Además las autoridades no disponen a mano de las notas de las diferentes materias para emitir criterios basados en la estadística de notas adquiridas por los estudiantes.

Por lo expuesto anteriormente se plantea como problema a resolver: ¿Cómo agilizar la gestión académica en el Pre universitario "Antonio López Fernández"?

Para darle solución al problema planteado se identifica como **objeto de estudio** la gestión académica en el pre universitario "Antonio López Fernández ". Como **campo de acción** se tiene la informatización de la gestión académica en el pre universitario "Antonio López Fernández"

Como **objetivo general** de esta investigación se plantea desarrollar una aplicación web que permite agilizar la gestión académica en el pre universitario “Antonio López Fernández”.

Para darle cumplimiento al objetivo planteado se trazaron las siguientes tareas:

- Estudiar las soluciones informáticas vinculadas al campo de acción.
- Valorar las tendencias actuales de las herramientas y tecnologías.
- Seleccionar las herramientas y tecnologías más adecuadas.
- Describir las problemáticas existentes con la gestión académica en el Proceso Docente-Educativo del pre universitario “Antonio López Fernández”.
- Realizar el diseño del sistema informático propuesto.
- Elaborar la documentación y los artefactos correspondientes a la metodología de desarrollo seleccionada.
- Implementar el sistema informático propuesto.
- Diseñar y realizar las pruebas funcionales al sistema construido.

Como **Idea a Defender** se plantea que la creación de una aplicación web como apoyo a la gestión académica permitirá agilizar las actividades realizadas por los profesores y directivos, contribuyendo al control del proceso docente en cualquier etapa del curso.

En la investigación se llevaron a cabo diferentes **Métodos de Investigación**, de los **Métodos Empíricos** fueron utilizados la **observación y análisis**, mediante el análisis de documentos se pudo estar al tanto del funcionamiento actual del proceso de gestión de la información de los estudiantes y a través de la observación se conoció la manera en que funciona la Secretaría Docente y el comportamiento del problema.

Los **Métodos Teóricos** utilizados fueron el **histórico y lógico** el cual permitió la búsqueda de antecedentes del software, las herramientas utilizadas, así como la forma en que se llevaba a cabo la gestión de la información de los estudiantes del pre universitario Antonio López Fernández, el **análisis y síntesis** se utilizó en los fundamentos teóricos, en el procesamiento de la información y en la

descomposición de cada uno de los requerimientos del sistema, el método ***hipotético deductivo*** fue utilizado para la confección de la idea a defender.

El trabajo está estructurado por un resumen, introducción, 4 capítulos, conclusiones generales, recomendaciones, bibliografías utilizadas y los anexos.

En el Capítulo uno **Fundamentación Teórica**, se hace una descripción de los principales conceptos utilizados en el problema, se realiza un análisis crítico del objeto de estudio. Se presenta además un estudio acerca de los sistemas existentes afines al campo de acción y se seleccionan las herramientas y la metodología a utilizar.

En el Capítulo dos **Planificación y Diseño**, se hace uso de la metodología seleccionada en el capítulo primero para el desarrollo del proyecto, emprendiendo en detalles cada una de sus fases.

En el Capítulo tres **Implementación y Prueba**, se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo, además se describen las pruebas realizadas y los resultados que estas arrojaron.

En el Capítulo 4 **Estudio de Factibilidad**, se realiza un estudio de los esfuerzos requeridos para la realización del sistema, se hace referencia a los beneficios tangibles e intangibles, y se valora la sostenibilidad del producto.

Capítulo 1 Fundamentación Teórica

En este capítulo se realizará un estudio de diferentes sistemas existentes, relacionados con la gestión académica tanto a nivel nacional como internacional. También se abordarán conceptos fundamentales sobre el desarrollo de aplicaciones Web, las características generales de los sistemas de bases de datos y de los lenguajes de programación Web, también se analizarán las diferentes metodologías existentes para el desarrollo de aplicaciones, y se seleccionará la más conveniente a utilizar.

1.1 Estado del arte

1.1.1 Ámbito internacional

ACADEMIC XXI [1]

El sistema ACADEMIC XXI ha sido concebido para acoplarse con facilidad al modelo de estudio de todas las instituciones educativas de nivel primario y medio. Esto se ha logrado ya que se le han incluido diferentes características. Es un sistema multiplataforma. Cuenta con un módulo “NOTAS XXI” para el registro de notas desde terminales instaladas para profesores.

El sistema también controla cuentas de usuarios, permitiendo accesos certificados en ambientes de red. Mantiene una ficha completa de cada uno de los estudiantes: datos personales, información médica, datos de los padres y representantes, historial académico, con la opción de presentar la fotografía digitalizada del estudiante. Exporta datos a Word para generar documentos con combinación de correspondencia.

Búho [2]

Búho está diseñado específicamente para resolver los problemas cotidianos de control de estudios de una institución educativa. Mediante el uso de tecnologías Web y arquitectura cliente-servidor, Búho es la plataforma ideal para integrar la información que se genera diariamente en una institución educativa a la Web. Los datos son fácilmente consultados mediante exploradores Web con todas las seguridades del caso bien sea mediante la Internet o mediante acceso a través de la Intranet de la institución.

El sistema consta de tres módulos: El sistema, el cual se sustenta en una base de datos MS SQL Server y una arquitectura cliente-servidor. La intranet, la cual da soporte Web para actividades internas de la organización como lo son la creación de los planes de estudio y el ingreso de las evaluaciones. El Internet, el cual permite el acceso al sistema por parte de los representantes de los alumnos, facilitando información académica y permitiendo la inscripción remota.

1.1.2 Ámbito nacional

GESTACAD [3]

El sistema de gestión académica GESTACAD es un software en explotación en la Universidad de Matanzas “Camilo Cienfuegos”. El mismo está completamente basado en plataforma Web y programado en PHP. EL sistema incluye módulos de administración, de secretaría y de matrícula.

En la actualidad el sistema es capaz de realizar diversas acciones y brindar numerosos reportes. Estos son fruto de los requisitos funcionales del sistema, recogidos en la fase de análisis.

Entre estas acciones podemos encontrar la búsqueda de un alumno, el listado de estudiantes por grupo, una tabla con los resultados docentes de un grupo en un semestre, actas de exámenes de las diferentes asignaturas, dar baja a un estudiante y un módulo Web para la gestión de matrícula. También cuenta con un sitio Web con reportes en línea con la utilidad del registro docente para los profesores.

1.2 Tecnologías y herramientas a utilizar

1.2.1 Aplicación Web

Con la llegada de la Web 2.0, propiciada esta por el desarrollo de las Tecnologías de la Información y las Comunicaciones y los nuevos retos de Internet, se abre paso una nueva idea en la concepción del software; el software orientado a servicios o SaaS (acorde a sus siglas en inglés). SaaS es un modelo de distribución del software que proporciona a los clientes el acceso al mismo a través de la red (generalmente Internet), de manera que les libra del mantenimiento de las aplicaciones, de operaciones técnicas y de soporte.

Las aplicaciones distribuidas en la modalidad SaaS pueden llegar a cualquier tipo de empresa sin importar su tamaño o su ubicación geográfica. Se trata de un modelo que une el producto (software) al servicio, para dotar a las empresas de una solución completa que permita optimizar sus costes y sus recursos. [4]

Estas aplicaciones presentan numerosas ventajas. Entre ellas se pueden mencionar algunas como son la facilidad de manejo; ya que la interacción con el usuario se establece a base de páginas Web, que le permiten conocer la funcionalidad del sistema con poco esfuerzo. También las aplicaciones Web son accesibles desde cualquier punto de la red, lo cual significa que un usuario puede utilizarla si dispone de cualquier conexión a Internet. Son de fácil portabilidad pues los navegadores Web se han desarrollado para todo tipo de máquinas; por lo que cualquier usuario dispone de la herramienta básica para lanzar la aplicación. [5]

Aunque, en principio, todo parece ventajas hay un aspecto muy a tener en cuenta, al que hay que dedicarle una gran atención, el tema de la seguridad, sobre todo si se desea que la aplicación pueda lanzarse desde cualquier ordenador conectado a Internet. En este sentido, es necesaria la adopción de medidas que garanticen su utilización únicamente por usuarios autorizados, sobre todo si las operaciones que se pueden realizar son críticas para el sistema.

1.2.2 Servidor Web

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. [6]

Apache

Apache, sustancialmente, es un proyecto nacido para crear un servidor de Web estable, fiable y veloz para plataformas Unix. Apache nace, por una parte, de un código ya existente y de una serie de parches para mejorar su fiabilidad y sus características; de ahí su nombre. [7]

Características: [8]

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Es una tecnología gratuita con un código fuente disponible. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esta característica le ofrece al software un grado de transparencia tal que es posible determinar en todo momento qué es lo que se está instalando, sin secretos ni puertas traseras.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que se instalen cuando se necesiten. Otra cosa importante es que cualquiera que posea alguna experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- 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.

1.2.3 Lenguajes de programación Web

Personal Home Page (PHP) [9]

Es un lenguaje de programación el cual está completamente orientado al desarrollo de aplicaciones Web dinámicas con acceso a información almacenada en una base de datos. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

Es un lenguaje multiplataforma y de libre distribución, por lo que se representa como una alternativa de fácil acceso para todos. Tiene una amplia capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destacando su conectividad con MySQL. Como otra de sus ventajas se puede decir que permite aplicar técnicas de programación orientada a objetos. Además posee una amplia documentación, entre la cual se destaca que todas las funciones del sistema están expuestas y ejemplificadas en un único archivo de ayuda.

Como desventajas de este lenguaje podemos encontrar que se necesita instalar un servidor web. También el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a la medida que las solicitudes aumenten. Además la programación orientada a objetos es aún muy deficiente para grandes aplicaciones.

Ventajas de PHP

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

JavaScript [31]

Es un lenguaje pensado para agregar interactividad con el usuario a las páginas HTML. Permite ejecutar secuencias de comandos en el mismo navegador del usuario. Con JavaScript se pueden realizar cálculos rápidos y complejos, verificar formularios antes de enviarlos, crear calendarios, convertir divisas. Es un lenguaje que distingue entre minúscula y mayúscula, no exige la declaración explícita de las variables, es posible crear las variables.

Es importante saber que JavaScript no lo soportan todos los navegadores por lo que nos vemos en la situación de probar el código resultante en más de un navegador. La sintaxis es muy parecida a C o C++, por lo que es un lenguaje fácil para el que lo domine.

1.2.4 Framework de desarrollo Web

¿Qué es un Framework? [10]

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

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.

Ext JS [11]

Sencha Ext JS (anterior Ext JS) es una biblioteca de JavaScript para el desarrollo de aplicaciones Web interactivas que además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales. Creado por Jack Slocum, Brian Moeskau, Aaron Conran y Rich Waters.

Esta librería incluye:

- Componentes UI del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open Source (GPL) y comerciales.

Ext JS ha marcado la diferencia en la preferencia de muchos desarrolladores. Y no solo para aplicaciones Web sino que ya es útil para productos de alta demanda en el mercado como iPhone e IPAD.

Ventajas

- Una de las grandes ventajas de utilizar Ext JS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos.
- Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores (Firefox, IE, Safari, Opera etc.).

- El funcionamiento de las ventanas flotantes lo pone por encima de cualquier otro.
- Relación entre Cliente-Servidor balanceado: Se distribuye la carga de procesamiento entre, permitiendo que el servidor pueda atender más clientes al mismo tiempo.
- Eficiencia de la red: Disminuye el tráfico en la red pues las aplicaciones cuentan con la posibilidad de elegir que datos desea transmitir al servidor y viceversa (Criterio este que puede variar con el uso de aplicaciones de pre-carga).
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.

Desventajas

- Necesidad de una plataforma: Pues dependemos del paquete Ext JS para obtener los resultados deseados.
- No contar con una licencia LGPL.

1.2.5 Sistemas gestores de bases de datos (SGBD) [12]

Consiste en un conjunto de programas, procedimientos y lenguajes que nos proporcionan las herramientas necesarias para trabajar con una base de datos. Incorporar una serie de funciones que nos permita definir los registros, sus campos, sus relaciones, insertar, suprimir, modificar y consultar los datos. Actualmente existen muchos sistemas gestores de bases de datos.

MySQL

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de

integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. [32]

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.

Ventajas: [13]

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.

- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

Desventajas:

- Un gran porcentaje de las utilidades de MySQL no están documentadas.

1.2.6 Arquitectura

Arquitectura en capas

Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo o por el contrario, como una de las posibles imágenes de algún estilo envolvente. Definen el estilo en capas como una organización jerárquica tal, que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones de la inmediatamente inferior. [14]

La arquitectura por capas es un estilo de arquitectura en la que el objetivo primordial es la separación de la lógica de negocio de la lógica de diseño, un ejemplo básico es separar la capa de datos, de la capa de presentación al usuario. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles, simplemente es necesario conocer las API que existen entre niveles.

El diseño de sistemas informáticos suele usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables, (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más en boga actualmente es el diseño en tres capas. [15]

Capas o niveles

- Capa de presentación o interface: es la capa de que le permite al usuario interactuar con el sistema, captura y le comunica la información al mismo, dando un mínimo de proceso, (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la del negocio.
- Capa de lógica o de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas después del proceso. Se denomina capa de negocio e incluso lógica del negocio, pues es aquí donde se establecen las reglas que deben cumplirse. Esta capa se comunica con la de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos para solicitar al gestor de bases de datos para almacenar o recuperar datos de él.
- Capa de datos: es donde se ubican los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de los mismos, reciben solicitudes de almacenamiento o de recuperación de información desde la lógica del negocio.

Modelo cliente-servidor de 3 capas

En la arquitectura de 3 capas la presentación, la lógica de aplicación y los elementos de datos están conceptualmente separados. Los componentes de la capa de presentación manejan la interacción con el usuario y realizan las peticiones del cliente a los componentes de la capa intermedia. Los componentes de la capa intermedia, manipulan la lógica de negocio y hacen las peticiones a la base de datos.

A continuación se enumeran algunas ventajas de las aplicaciones de 3 capas:

- Los componentes de la aplicación pueden ser desarrollados en cualquier lenguaje en general lo que posibilita que el grupo de desarrolladores no se centre en el uso de un solo lenguaje.
- Los componentes están centralizados lo que posibilita su fácil desarrollo, mantenimiento y uso.

- Los componentes de la aplicación pueden estar esparcidos en múltiples servidores permitiendo una mayor escalabilidad.
- Los problemas de limitación para las conexiones a las bases de datos se minimizan ya que la base de datos solo es vista desde la capa intermedia y no desde todos los clientes. Además que las conexiones y los drivers de las bases de datos no tienen que estar en los clientes.
- Los componentes de aplicación de la capa intermedia pueden ser asegurados centralmente usando una infraestructura común. Se pueden conceder o denegar los permisos componente a componente simplificando la administración. [16]

1.2.7 Metodología de Desarrollo de Software

Programación Extrema (Extreme Programming, XP)

XP, es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. XP se basa en 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. Kent Beck, el padre de XP, que describe la filosofía de XP en el "Manifiesto Ágil", sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. [17]

Prácticas básicas de la programación extrema

Para que todo esto funcione, la programación extrema se basa en "prácticas básicas" que deben seguirse al pie de la letra. [18]

Equipo completo: Forman parte del equipo todas las personas que tienen algo que ver con el proyecto, incluido el cliente y el responsable del proyecto.

Planificación: Se hacen las historias de usuario y se planifica en qué orden se van a hacer y las mini-versiones. La planificación se revisa continuamente.

Test del cliente: El cliente, con la ayuda de los desarrolladores, propone sus propias pruebas para validar las mini-versiones.

Versiones pequeñas: Las mini-versiones deben ser lo suficientemente pequeñas como para poder hacer una cada pocas semanas. Deben ser versiones que ofrezcan algo útil al usuario final y no trozos de código que no pueda ver funcionando.

Diseño simple: Hacer siempre lo mínimo imprescindible de la forma más sencilla posible. Mantener siempre sencillo el código.

Mejora del diseño: Mientras se codifica, debe mejorarse el código ya hecho con el que nos crucemos y que sea susceptible de ser mejorado. Extraer funcionalidades comunes, eliminar líneas de código innecesarias.

Integración continua: Deben tenerse siempre un ejecutable del proyecto que funcione y en cuanto se tenga una nueva pequeña funcionalidad, debe recompilarse y probarse. Es un error mantener una versión congelada dos meses mientras se hacen mejoras y luego integrarlas todas de golpe. Cuando falle algo, no se sabe qué es lo que falla de todo lo que hemos metido.

El código es de todos: Cualquiera puede y debe tocar y conocer cualquier parte del código.

Normas de codificación: Debe haber un estilo común de codificación (no importa cual), de forma que parezca que ha sido realizado por una única persona.

Metáforas: Hay que buscar unas frases o nombres que definan cómo funcionan las distintas partes del programa, de forma que sólo con los nombres se pueda uno hacer una idea de qué es lo que hace cada parte del programa. Un ejemplo claro es el "recolector de basura" de java. Ayuda a que todos los programadores (y el cliente) sepan de qué estamos hablando y que no haya mal entendidos.

Ritmo sostenible: Se debe trabajar a un ritmo que se pueda mantener indefinidamente. Esto quiere decir que no debe haber días muertos en que no se sabe qué hacer y que no se deben hacer un exceso de horas otros días. Al tener claro semana a semana lo que debe hacerse, hay que trabajar duro en ello para conseguir el objetivo cercano de terminar una historia de usuario o mini-versión.

Características fundamentales de XP. [19]

- *Desarrollo iterativo e incremental*: pequeñas mejoras, unas tras otras.
- *Pruebas unitarias*: continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- *Programación por parejas*: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código esté escrito de esta manera (el código es revisado y discutido mientras se escribe) es más importante que la posible pérdida de productividad inmediata.
- *Frecuente interacción*: del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- *Simplicidad en el código*: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo

Ventajas

- Apropiado para entornos volátiles.
- Estar preparados para el cambio, significa reducir su coste.
- Planificación más transparente para los clientes, ya conocen las fechas de entrega de funcionalidades. Vital para su negocio.
- Permite definir en cada iteración cuales son los objetivos de la siguiente.
- Permite la retroalimentación.

Desventajas

- Delimitar el alcance del proyecto con nuestro cliente.

Fases de la metodología XP [19]

Fase I: Planificación

- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen test de aceptación para historias de usuarios y permita hacer una estimación de tiempo de desarrollo del mismo.
- Se crea un plan de lanzamiento, que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usan como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.
- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.

Fase II: Diseño

- Se eligen los diseños funcionales más simples.
- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.
- Se escriben tarjetas de clase-responsabilidades-colaboración (CRC) para cada objeto, que permitan abstraerse al pensamiento estructurado y que el equipo de desarrollo completo participe en el diseño.

Fase III: Codificación

- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.
- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.

- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro, qué objetivos tiene que cumplir lo que se va a codificar.
- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores que integre código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.
- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.
- Se deja la optimización para el final.
- No se hacen horas extra de trabajo.

Fase IV: Pruebas

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las historias de usuarios elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que la correspondiente historia de usuario se ha completado.

1.2.8 Herramientas CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadoras), son aplicaciones informáticas que tienen como objetivo fundamental solucionar y

afrontar los problemas de mala calidad de software y documentación inadecuada. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo de un determinado proyecto o software, ya que brindan la posibilidad de realizar cálculos de costos, generan código fuente automáticamente de un diseño previamente dado, poseen compilación automática, ayudan con la documentación y juegan un papel importante en la detección de errores.

Objetivos [20]

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Visual Paradigm para UML [21]

Visual Paradigm para UML es una herramienta que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Embarcadero ER/Studio: Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. Simple y fácil al usuario, ayuda a las organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. [22]

Ventajas:

- Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, ER/Studio se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia.
- La creación de diagramas es clara y rápida.
- Tiene la posibilidad de realizar diagramas con desempeño rápido.

1.2.9 Lenguaje Unificado de Modelado (UML) [23]

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Capta la información sobre la estructura estática y el comportamiento dinámico de un sistema, no es un lenguaje de programación.

Las herramientas CASE pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes. UML es un lenguaje de propósito general para el modelado orientado a objetos, es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

Objetivos del UML: [24]

- Es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- No pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso, incluye todos los conceptos que se consideran

necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

En este capítulo se realizó un estudio de los sistemas existentes vinculados al campo de acción tanto en el panorama internacional como en el nacional y debido a que estos sistemas no cumplen con los requerimientos deseados por el cliente, se decidió la implementación de un sistema para apoyar y agilizar la gestión académica en el pre universitario “Antonio López Fernández”.

El sistema GESTACAD está diseñado para ser utilizado en la Educación Superior, por lo que es muy difícil adaptarlo a la Educación Media y Técnica Profesional. Además tanto en el caso del Búho como del ACADEMIC XXI son software propietario, lo que acarrearía grandes costos para el Pre-Universitario “Antonio López Fernández”.

Se realizó un estudio de las herramientas y tecnologías existentes, donde se eligieron como lenguajes de programación PHP y JavaScript, como SGBD se utilizó MySQL y para guiar el proyecto se decidió como metodología a XP.

Capítulo 2 Planificación y Diseño

En este capítulo, se establece la fase de planificación y diseño. En él se especifican las necesidades del cliente. También se figuran las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU). Además se realiza una estimación del esfuerzo necesario para las mismas y se establece un plan de iteraciones necesarias sobre el sistema, para su posterior culminación.

Tabla 2.1 Personal relacionado con el sistema

Personas relacionadas con el sistema	Justificación
Especialista (Secretario Docente)	Esta es la persona que tiene mayor conocimiento en la materia de gestión académica, y está encargada de la gestión de la misma
Administrador	Es la persona encargada de asesorar y dar seguimiento del estado del proceso de desarrollo

2.1 Lista de reserva

Después de conocer el personal relacionado, se procede a realizar el análisis de las funcionalidades que debe cumplir la aplicación para dar respuesta a los mismos. Para ello se enumerarán mediante una lista de reserva, las funcionalidades que el sistema debe ser capaz de cumplir.

Tabla 2.2 Lista de reserva del producto

Código	Descripción del requisito funcional	Prioridad
RFGA1	Insertar miembro	Muy alta
RFGA2	Listar miembros	Muy alta

RFGA3	Modificar miembro	Muy alta
RFGA4	Buscar miembro	Muy alta
RFGA5	Eliminar miembro	Muy alta
RFGA6	Insertar grupo	Muy alta
RFGA7	Proporcionar privilegio	Muy alta
RFGA8	Listar grupos	Muy alta
RFGA9	Modificar grupo	Muy alta
RFGA10	Buscar grupo	Muy alta
RFGA11	Eliminar grupo	Muy alta
RFGA12	Insertar privilegios	Media
RFGA13	Listar privilegios	Media
RFGA14	Modificar privilegios	Media
RFGA15	Buscar privilegios	Media
RFGA16	Eliminar privilegios	Media
RFGA17	Administrar módulos	Media
RFGA18	Insertar tipo de enseñanza	Baja
RFGA19	Modificar tipo de enseñanza	Baja
RFGA20	Crear grupos por enseñanza	Baja
RFGA21	Insertar matrícula de estudiante	Muy alta
RFGA22	Buscar matrícula de un estudiante	Muy alta
RFGA23	Modificar la matrícula de un estudiante	Muy alta
RFGA24	Renovar la matrícula de un estudiante	Muy alta
RFGA25	Insertar Asignatura	Media
RFGA26	Definir sistema de evaluaciones de una asignatura	Media

RFGA27	Registrar nota de un estudiante en una evaluación	Media
RFGA28	Modificar nota de un estudiante en una evaluación	Media
RFGA29	Consultar notas de un estudiante	Media
RFGA30	Cerrar asignatura	Media
RFGA31	Listar informe de matrícula por grado, sexo y edad	Alta
RFGA32	Listar informe de sexo por grado	Alta
RFGA33	Listar informe de color de la piel por grado	Alta
RFGA34	Listar alumnos en movimiento	Alta
RFGA35	Crear reportes de altas y bajas	Alta

2.2 Historias de Usuarios (HU) [25]

Las HU, son la técnica utilizada en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas tormenta de ideas arrojan no solo los requerimientos, sino también las posibles soluciones; representan una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para gestionarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario.

¿Por qué utilizamos las historias de usuario? [26]

Porque siguen los principios básicos de requerimientos ágiles: Potencian la participación del equipo en la toma de decisiones. Se crean y evolucionan a medida que el proyecto avanza. Son peticiones concretas y pequeñas. Contiene la información imprescindible. Menos es más. Apoyan la cooperación, colaboración y conversación entre los miembros del equipo, lo que es fundamental.

¿Qué son las historias de usuarios? [26]

Una manera simple de describir una tarea concisa que aporta valor al usuario o al negocio. No se detalla más hasta el momento que la historias de usuario se vaya a

desarrollar Las historias de usuario pueden ser creadas durante las conversación con los usuarios interesados sobre nuevas funcionalidades o mejoras del proyecto. Las historias de usuario es una invitación a la conversación.

Beneficios

- Al ser muy corta, ésta representa requisitos del modelo de negocio que pueden implementarse rápidamente (días o semanas).
- Necesitan poco mantenimiento.
- Mantienen una relación cercana con el cliente.
- Permite dividir los proyectos en pequeñas entregas.
- Permite estimar fácilmente el esfuerzo de desarrollo.

Tabla 2.3 Modelo de planilla de HU

Historia de Usuario	
Código: Código de la HU	Nombre: Nombre de la HU
Modificación de Historia de Usuario Número: Número de modificación realizada	
Referencia: Referencia a los requisitos de la lista de reserva	
Programador: Nombre del programador	Iteración Asignada: Iteración a la que corresponde.
Prioridad: Importancia: Alta / Media / Baja	Puntos Estimados: Estimación: de 1 a 3 puntos
Riesgo en Desarrollo: Dificultad: Alta /Media / Baja	Puntos Reales: De 1 a 3
Descripción: Se especifican las operaciones por parte del usuario y las respuestas del sistema.	
Observaciones: Algunas observaciones de interés, como glosario, información sobre usuario etc.	

Tabla 2.4 Historia de Usuario Administrar matrícula

Historia de Usuario	
Código: HUGA4	Nombre: Administrar matrícula
Modificación de Historia de Usuario Número: Ninguna	
Referencia: Ítems RFGA14, RFGA15, RFGA16, RFGA17, RFGA18.	
Programador: Alexander Garcés Valdés	Iteración Asignada: Primera
Prioridad: Importancia: Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Dificultad: Alta	Puntos Reales: 3
Descripción: Esta HU permite al administrador del sistema crear las matrículas de los estudiantes del sistema, así como renovarlas, y listar a todos los estudiantes matriculados.	
Observaciones: Confirmado con el cliente.	

2.3 Planificación de entregas

En esta parte se establece la prioridad de cada HU así como una estimación del esfuerzo necesario de cada una de ellas con el fin de determinar un cronograma de entregas. Las estimaciones de esfuerzo asociado a la implementación de las historias se establecen utilizando como medida, el punto. Un punto, equivale a una semana ideal de programación (6 días). Las historias generalmente valen de 1 a 3 puntos. Por otra parte, se mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias.

2.3.1 Estimación de esfuerzo por HU

Tabla 2.5 Estimación de esfuerzo por HU

Historias de Usuario	Puntos de Estimación (semanas)
Administrar Miembros	2
Administrar Grupos	2
Administrar Privilegios	2
Administrar tipo de enseñanza	1
Administrar asignatura	1
Boleta de Movimiento de Alumnos	2
Administrar matrícula	3
Informes de estudiantes matriculados	3

2.3.2 Plan de iteraciones

Tabla 2.6 Plan de iteraciones

Iteraciones	Descripción de la iteración	Orden de la HU a implementar	Duración de cada HU (días)	Duración total (días)
Primera	En esta iteración se van a implementar las HU que sean de prioridad "Muy Alta" para el proyecto.	Administrar miembros	12	42
		Administrar grupos	12	
		Administrar matrícula	18	
Segunda	En esta iteración se van a implementar las	Informes de estudiantes matriculados	18	30

	HU que tengan prioridad "Alta", para el proyecto.	Boleta de Movimiento de Alumnos	12	
Tercera	En esta iteración se van a implementar las HU, que tengan prioridad "Media" para el proyecto.	Administrar privilegios	12	18
		Administrar asignaturas	6	
Cuarta	En esta iteración se van a implementar las HU, que tengan prioridad "Baja" para el proyecto.	Administrar tipos de enseñanza	6	6
Total				96

2.4 Clases, Responsabilidades y Colaboración

Las tarjetas clases, responsabilidades y colaboración (CRC), se realizan para facilitar la comunicación y documentar los resultados. Permiten una total participación y contribución del equipo de desarrollo en el diseño. Cada tarjeta CRC representa clases, donde el nombre de cada clase se ubica en forma de título en la parte superior de la tarjeta; sus atributos y responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente.

Centro	
Description: Guarda los datos del centro	
Attributes:	
Name	Description
id_centro	Identificador del centro
nombre_centro	
Responsibilities:	
Name	Collaborator
insertar centro	

Fig. 2.1 Tarjeta CRC Centro

Con la culminación de este capítulo se han desarrollado las bases con las que se sustentarán las necesidades del cliente, se identificaron las HU con la participación conjunta del cliente y usuarios, destacando la planificación de cada HU por la prioridad de sus iteraciones o sea a partir del esfuerzo de las mismas. También se crearon las tarjetas CRC las cuales van a facilitar y documentar los resultados.

Capítulo 3 Implementación y Pruebas

En este capítulo se desplegará la fase de implementación y pruebas conforme a lo que establece la metodología XP. Se describen cada una de las tareas confeccionadas para cumplir con el desarrollo de cada una de las HU definidas. También se presentan las tarjetas clases, responsabilidades y colaboradores. Igualmente se mostrarán las pruebas de aceptación elaboradas por el cliente para comprobar que la aplicación funcione correctamente.

3.1 Modelo de datos

En este modelo se presentan todas las tablas con la que interactúa el sistema propuesto en esta investigación; se recogen y modelan todos los datos con lo que dispondrá la aplicación.

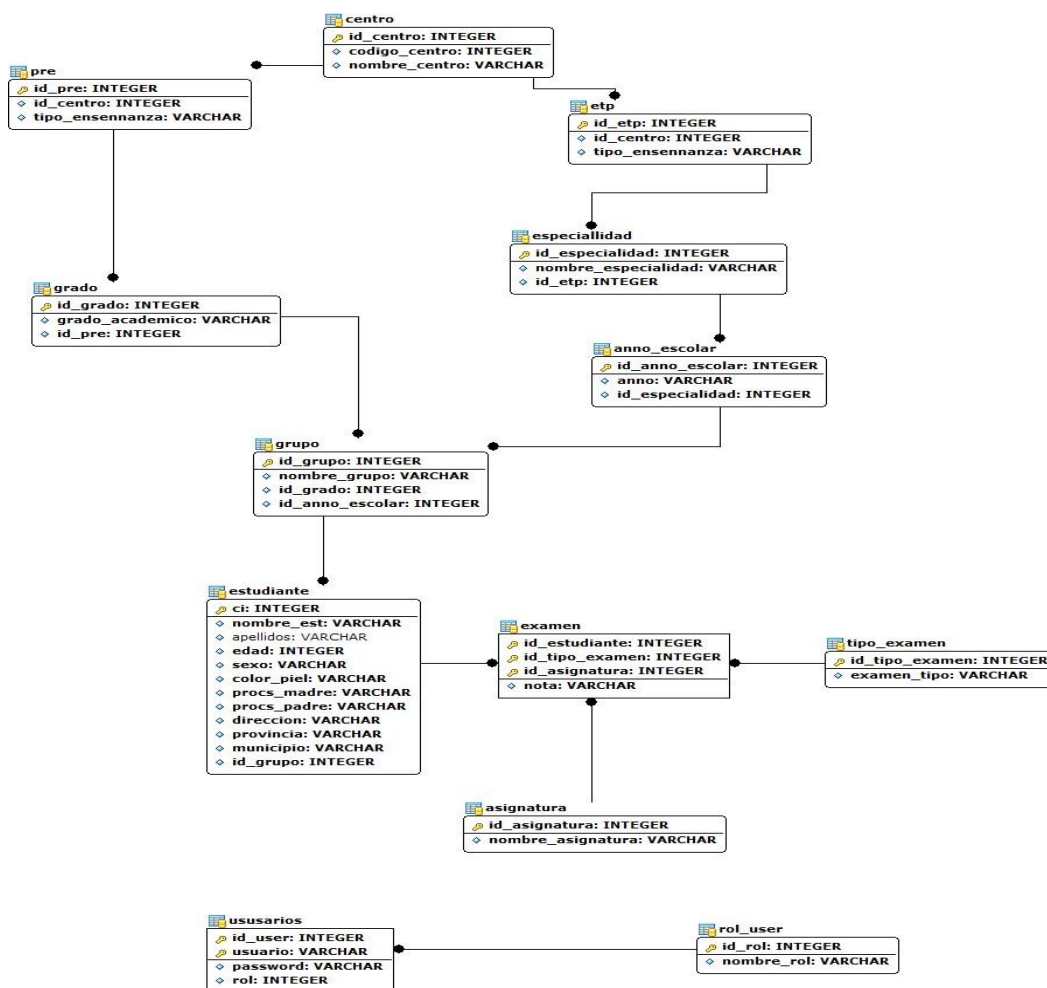


Fig. 3.1 Modelo de datos

3.2 Desarrollo de las iteraciones

Durante la fase planificación y diseño fueron detalladas las historias de usuario correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las prioridades y restricciones de tiempo, previstas por el cliente.

3.2.1 Tareas de ingeniería por HU

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o de ingeniería, y a su vez, estas son asignadas al equipo de desarrollo para su implementación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Estas se representan mediante las tarjetas de tareas.

Tabla 3.2 Tareas de ingeniería por Historias de Usuario

Historias de Usuario	Tarea
Administrar Miembros	<ul style="list-style-type: none">➤ Insertar miembro.➤ Listar miembros.➤ Modificar miembro.➤ Buscar miembros.➤ Eliminar miembro.
Administrar Grupos	<ul style="list-style-type: none">➤ Insertar grupo.➤ Listar grupos.➤ Modificar grupo.➤ Buscar grupos.➤ Eliminar grupo.➤ Proporcionar privilegios.

Administrar Privilegios	<ul style="list-style-type: none">➤ Insertar privilegio.➤ Listar privilegios.➤ Modificar privilegio.➤ Buscar privilegios.➤ Eliminar privilegio.➤ Administrar módulos.
Administrar tipo de enseñanza	<ul style="list-style-type: none">➤ Insertar tipo de enseñanza.➤ Modificar tipo de enseñanza.➤ Crear grupos por enseñanza.
Administrar asignatura	<ul style="list-style-type: none">➤ Insertar asignatura.➤ Definir sistema de evaluaciones de una asignatura.➤ Registrar nota de un estudiante en una evaluación.➤ Modificar nota de un estudiante en una evaluación.➤ Consultar notas de un estudiante.➤ Cerrar asignatura.
Administrar matrícula	<ul style="list-style-type: none">➤ Insertar matrícula de estudiante.➤ Buscar matrícula de un estudiante.➤ Modificar la matrícula de un estudiante.➤ Renovar la matrícula de un estudiante.➤ Listar los estudiantes matriculados.
Informes de estudiantes matriculados	<ul style="list-style-type: none">➤ Listar informe de matrícula por grado, sexo y edad.➤ Listar informe de sexo por grado.➤ Listar informe de color de la piel por grado.➤ Listar reportes de altas y bajas.

Boleta de movimiento de alumnos	<ul style="list-style-type: none">➤ Crear la boleta de movimiento de alumnos➤ Listar alumnos en movimiento.
---------------------------------	--

Tabla 3.3 Historias de Usuario abordadas en la primera iteración.

Historias de Usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Administrar miembros	2	2
Administrar grupos	2	2
Administrar matrícula	3	3

Tabla 3.4 Historias de Usuario abordadas en la segunda iteración.

Historias de Usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Informes de estudiantes matriculados	3	3
Boleta de movimiento de alumnos	2	2

Tabla 3.5 Historias de Usuario abordadas en la tercera iteración.

Historias de Usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Administrar privilegios	2	2

Administrar asignaturas	1	1
-------------------------	---	---

Tabla 3.6 Historias de Usuario abordadas en la cuarta iteración.

Historias de Usuario	Tiempo de estimación (semanas)	
	Estimación inicial	Real
Administrar tipos de enseñanza	1	1

3.3 Pruebas

En la Programación Extrema es esencial el desarrollo de las pruebas, permitiendo probar continuamente el código. Cada vez que se desean implementar las funcionalidades que tendrá el software, XP propone una redacción sencilla de prueba. El proceso constante de las pruebas permite la obtención un producto con mayor calidad ofreciendo a los programadores una mayor certeza en el trabajo que desempeñan.

En la metodología XP hay dos tipos de pruebas; las unitarias o desarrollo dirigido por pruebas (TDD, Test Driven Development), desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación, las cuáles son evaluadas luego de culminar una iteración verificando así que se cumplió la funcionalidad requerida por el cliente. Con estas normas se obtiene un código simple y funcional de manera bastante rápida y eficiente. [27]

3.3.1 Desarrollo dirigido por Pruebas

El desarrollo dirigido por pruebas, se enfoca en la implementación orientada a pruebas. El código debe ser probado paso a paso para lograr un resultado, aunque no con lógica para el negocio, pero si funcional. Algunas personas confunden este término con las llamadas “pruebas de caja blanca” las cuáles se les practican a los métodos u operaciones para medir la funcionalidad del mismo, desde el punto de vista de validez del cliente.

Sin embargo, el TDD se aplica antes de comenzar a implementar cada paso de la tarea en desarrollo, asumiendo que la prueba es insatisfactoria desde un inicio. Sólo una vez que se haya cumplido de la forma más sencilla posible la lógica del código a probar se asume como cumplida.

Luego se realiza un proceso conocido como “refactorización” de código perteneciente a una de las doce prácticas planteadas por la metodología XP, el cual consiste en mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez. Es esencia el TDD, se enfoca en la lógica del negocio y las pruebas de caja blanca en la lógica del negocio.

3.3.2 Pruebas de aceptación (PA)

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en la metodología RUP, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que una historia de usuario ha sido implementada correctamente.

La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas satisfactoriamente.

Para realizar las pruebas de aceptación (PA) el cliente utiliza la siguiente plantilla:

Tabla 3.7 Modelo para pruebas de aceptación.

Prueba de Aceptación: Nombre de la prueba de aceptación
HU: Nombre de la HU que se va a comprobar.
Nombre: Nombre del caso de prueba.
Descripción: Descripción del propósito de la prueba.

Condiciones de ejecución: Precondiciones para que la prueba se pueda realizar.
Entrada / Pasos ejecución: Pasos para probar la funcionalidad.
Resultado: Resultado que se desea de la prueba.
Evaluación de la prueba: Aceptada o denegada.

A continuación se muestra una de las pruebas de aceptación de las historias de usuario.

Tabla 3.8 Prueba de aceptación para la HU Administrar Miembros.

Prueba de Aceptación: Administrar Miembros
HU: Administrar Miembros.
Nombre: Prueba para comprobar la administración de miembros.
Descripción: Con esta prueba de aceptación se persigue validar la administración de miembros.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder al módulo de administración de miembros y poder adicionar, eliminar, mostrar o modificar un miembro.
Entrada / Pasos de ejecución: El administrador hace clic en el botón " Add " e ingresa los datos del miembro que desea insertar, luego presiona el botón " Guardar " y es creado un nuevo miembro. En seguida que el miembro es creado puede ser modificado o eliminado.
Resultado: Se insertan correctamente los miembros. Se visualizan todos los miembros correctamente. Se pueden eliminar o modificar una vez insertados.
Evaluación de la prueba: Aceptada.

En este capítulo se llevó a cabo la fase de implementación y pruebas. Se realizó la implementación de las iteraciones a partir de la distribución de tareas por historias de usuarios y se le hicieron las pruebas de aceptación a estas para verificar que las funcionalidades de la aplicación, siendo todas estas aceptadas por el cliente, además de ver las principales interfaces de la aplicación, así como las tarjetas de ingenierías.

También se presentó el modelo de datos de la futura aplicación, logrando una visión detallada de sus atributos y las relaciones entre sus clases.

Capítulo 4 Estudio de Factibilidad

En la actualidad es imprescindible que a la hora de desarrollar un proyecto, se realice como es debido el estudio de su factibilidad. En general ningún producto informático está libre de posibles riesgos en la concepción del proyecto. Por tanto es necesario minimizar recursos humanos, materiales y financieros; de ahí que es de vital importancia estimar la relación costo–beneficio, así como el esfuerzo, capital humano y el tiempo de desarrollo que se emplea en la ejecución de estos.

En este capítulo se expone el estudio de factibilidad del proyecto, centrado en estimaciones de esfuerzo humano, tiempo de desarrollo para su ejecución y costo. Se estiman los beneficios tangibles e intangibles que representan para el sistema propuesto, un análisis de costos y beneficios.

4.1 Evaluación Costo-Beneficio

La mayoría, por no decir todos los proyectos de informática, son evaluados según el criterio de Costo-Beneficio. Esta Metodología, plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- El costo, que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados.
- La efectividad, que se entiende como la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo). El desarrollo de un producto informático, siempre tiene un costo.

Este puede estar justificado por los beneficios tanto tangibles como intangibles que origina el mismo. En este proceso, se necesita de una selección adecuada de los elementos más convenientes para su evaluación.

4.2 Efectos Económicos

- Efectos directos.
- Efectos indirectos.
- Efectos externos.
- Intangibles.

Efectos Directos

POSITIVOS:

- Se agilizará el proceso de obtención de reportes de los estudiantes matriculados.
- Se facilitará el proceso de impresión de reportes y actas de estudiantes.
- Los usuarios del sistema podrán llevar el control de las notas de cualquier estudiante de manera fácil y sencilla.
- El proceso de cualquier tipo de búsqueda que se desee realizar será de manera sencilla y personalizada.
- El sistema es compatible con todos los navegadores más usados en la actualidad (Mozilla Firefox, Internet Explorer, Google Chrome).

Efectos Indirectos

Los efectos económicos observados que pudieran repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

Efectos Externos

Se obtuvo un producto que apoya y facilita en gran medida la gestión académica en el Pre universitario “Antonio López Fernández”.

Intangibles

En la valoración económica siempre hay elementos como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible.

4.3 Elementos para identificar los Costos y Beneficios del Proyecto

Para la identificación de los costos y beneficios del proyecto que son pertinentes para su evaluación, es necesario definir una situación base o situación sin proyecto; la comparación de lo que sucede con proyecto, versus lo que hubiera sucedido sin proyecto, definirá los costos y beneficios pertinentes del mismo. [28]

Estos escenarios, resultan ser una herramienta determinante, puesto que ayudan en gran medida en la definición de los elementos necesarios para la evaluación.

A continuación se analizan ambas situaciones

SITUACIÓN SIN PROYECTO

En el pre universitario para llevar el control de la gestión académica se realizan los siguientes pasos:

- Realizar la matrícula a cada estudiante al ingresar al centro.
- Realizar reportes semanales y mensuales de estudiantes suspensos.
- Llevar el control de las actas de evaluación y sacar los promedios en cada asignatura.
- Realizar reportes de estudiantes según diferentes parámetros.

En este proceso se acumula un gran volumen de información lo que facilita que se cometan errores continuamente y a su vez el proceso sea más lento.

SITUACIÓN CON PROYECTO

- Se realiza la matrícula de los estudiantes al ingresar al centro.
- Reportes ágiles de estudiantes suspensos.
- Control de evaluaciones y promedio automático en cada asignatura.
- Obtener de forma inmediata las actas de estudios terminados en formato de impresión.
- Realizar reportes al instante de estudiantes según diferentes parámetros.

Expuestas ambas situaciones se procede a continuación con la identificación de los costos y beneficios.

4.4 Factibilidad Económica

El análisis de factibilidad económica identifica los costos y beneficios asociados con el proyecto. El mismo incluye cuatro categorías: [29]

- Costo de desarrollo.
- Costos operacionales.
- Beneficios tangibles.
- Beneficios intangibles.

Esta tesis, englobará los costos de desarrollo y los costos operacionales en Evaluación Económica.

4.5 Evaluación Económica

Los principales factores a considerar para el cálculo de los costos son los relacionados al personal, hardware y software, los que se pueden calcular de diversas maneras, que muchas veces se limitan al buen criterio y a la experiencia.

Para determinar el costo económico del proyecto, se desglosaron los costos en moneda libremente convertible y en moneda nacional

Tabla 4.1 Costos en moneda libremente convertible

Ficha de Costo		
		Precios (\$)
Costos moneda libremente convertible		
Costos directos		
Compra de equipos de cómputo		0.00
Alquiler de equipos de cómputo		0.00
Compra de licencia de software		0.00
Depreciación de equipos		25.00
Materiales directos		7.50
Subtotal		32.50
Costos Indirectos		
Formación del personal que elabora el proyecto		0.00

Gastos en llamadas telefónicas		0.00
Gastos para el mantenimiento del centro		0.00
Know How		0.00
Gastos en representación		0.00
Subtotal		0.00
Gastos de distribución y ventas		
Participación en ferias o exposiciones		0.00
Gastos en transportación		0.00
Compra de materiales de propaganda		0.00
Subtotal		0.00
	Total	32.50

Tabla 4.2 Costos en moneda nacional

Ficha de Costo		
		Precios (\$)
Costos moneda nacional		
Costos directos		
Salario del personal que laborará en el proyecto.		100.00
12,5% del total de gastos por salarios se dedica a la seguridad social.		0.00
9,09% de salario total, por concepto de vacaciones a acumular.		0.00
Gasto por consumo de energía eléctrica.		6.48
Gastos administrativos.		
Subtotal		106.48
Costos Indirectos		
Know How		0.00
Subtotal		0.00
	Total	106.48

La evaluación económica se efectúa conjuntamente con la que se puede llamar evaluación técnica del proyecto, que consiste en cerciorarse de la factibilidad técnica del mismo. En el análisis de la Factibilidad Técnica del proyecto, se pudo apreciar que:

- Se cuenta con hardware mayor a los requisitos mínimos.
- Se cuenta con la disponibilidad de software.

Por lo que se llega a la conclusión de que el proyecto es factible técnicamente.

Normalmente no se encuentran problemas en relación con el mercado o la tecnología disponible que se empleará en la fabricación del producto; por tanto la decisión de inversión casi siempre recae en la evaluación económica.

Los costos de los proyectos de informática son relativamente simples de cuantificar, no así los beneficios, que se presentan como ahorro de costos con respecto a la situación base. [30]

4.5.1 Beneficios tangibles

Los beneficios tangibles son los ingresos adicionales y/o reducción de costos que el nuevo sistema proveerá. Debido a que este proyecto no es construido con el propósito de comercializarse, no posee ingresos monetarios perceptibles, no así con la reducción de costos, pues el sistema proveerá, mejoras en la gestión académica.

Estos elementos son de muy difícil cuantificación, por lo que en esta tesis se consideran sólo como intangibles.

4.5.2 Beneficios y costos intangibles del proyecto

Costos

- Resistencia al cambio.

Beneficios

- Mayor agilidad en la gestión de matrícula y reportes para los trabajadores.
- Menor tiempo empleado en el proceso.

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de la misma, la técnica de punto de equilibrio aplicable a proyectos, donde los beneficios tangibles no son evidentes, el análisis se basa exclusivamente en los costos.

Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado para resolver la gestión académica, y la variable sería la complejidad de las pruebas que se realizan durante este proceso.

Valores de la variable (solución manual)

- Realizar la matrícula a cada estudiante al ingresar al centro. 31 variables (15 min.).
- Realizar reportes de estudiantes. 6 variables (1h).
- Calcular promedio de cada estudiante por asignatura y de manera general. 2 variables (50 min).
- Realizar reportes de estudiantes suspensos. 2 variables (30 min).
- Realizar Boletas de Movimiento de Alumnos. 56 variables (15 min).

Valores de la variable (solución con el sistema)

- Realizar la matrícula a cada estudiante al ingresar al centro. 29 variables (2 min.).
- Cargar reportes de estudiantes. 6 variables (2 min).
- Calcular promedio de estudiantes por asignatura y de manera general. 2 variables (1 min).
- Cargar reportes de estudiantes suspensos. 2 variables (1 min).
- Crear Boletas de movimiento de alumnos. 45 variables (2 min).

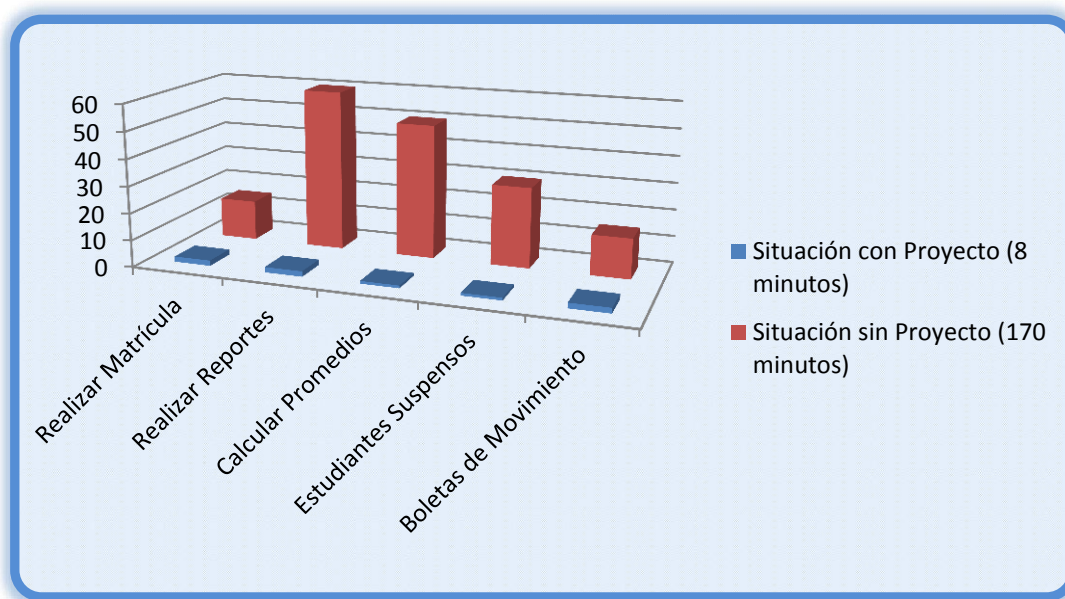


Figura 4.1 Punto de equilibrio

Conclusiones Generales

Durante el desarrollo de la investigación y luego de realizar un estudio sobre los sistemas informáticos existentes en el campo de acción, se concluyó que los mismos no reúnen los requisitos requeridos en el pre universitario “Antonio López Fernández”. Por tanto se propuso y se implementó el sistema ACADÉMICA. Para el desarrollo del mismo se realizó un estudio y se seleccionaron las herramientas y tecnologías más adecuadas.

Se elaboró toda la documentación y los artefactos correspondientes a la metodología seleccionada como la lista de reserva, tarjetas CRC, pruebas de aceptación, historias de usuario y se realizó un manual de usuarios para facilitar el trabajo a los usuarios del sistema. Como resultado se obtuvo un producto de alta calidad, el cual ayuda y agiliza el proceso de gestión académica en el pre universitario “Antonio López Fernández”.

Recomendaciones

- Aprovechar al máximo la aplicación web la cual dará mejor servicio al pre universitario “Antonio López Fernández”, utilizando una herramienta de trabajo básica como es la computadora.
- Continuar con el estudio del sistema, el cual le permita realizar incorporaciones de nuevos módulos a la aplicación base.
- Promover la aplicación en otros centros de educación para que tomen como referencia e implanten en la institución.

Bibliografías Utilizadas

1. <http://www.sistemas.com.ec>
2. lidesoft.com [En Línea] Búho - Sistema de Gestión Académica 12-producto-buho.html.- <http://www.lidesoft.com/12-producto-buho.html>
3. <http://www.monografias.com/trabajos85/aplicaciones-gestion-academica-software-libre/aplicaciones-gestion-academica-software-libre.shtml>.
4. Graham Paul, The Other Road Ahead [Libro]. - 2001.
5. Masternewmedia.org Masternewmedia.org [En línea] // Beneficios De Las Aplicaciones Basadas En Web Y El Anuncio De Microsoft De La Era “En Vivo”. - 2005. - diciembre de 2010. - http://www.masternewmedia.org/es/aplicaciones_web/temas_de_aplicaciones_web/Beneficios_De_Las_Aplicaciones_Basadas_En%20Web_Y_El_Anuncio_De_Microsoft_De_La_Era_En_Vivo.htm.
6. Vegas Jesús El Servidor Web [Libro]. - [s.l.]: Universidad de Valladolid, 2002.
7. Htmlpoint.com [En línea] // ¿Qué es Apache? - 2008. - diciembre de 2010. - <http://www.htmlpoint.com/faq/apache/01.htm>.
8. Ciberaula.com Ciberaula.com [En línea] // Una Introducción a APACHE. - 2005.- diciembre de 2010. - http://linux.ciberaula.com/articulo/linux_apache_intro.
9. HERNÁN RUIZ, Marcelo. Programación Web Avanzada [Libro]. La Habana, Cuba. Editorial Félix Varela, 2006.
10. <http://es.wikipedia.org/wiki/Framework.htm>
11. Learning EXT JS. (2008). [Consultado: 2012-02-5]. Build dynamic, desktop-style user interfaces for your Data-driven web applications.
12. CAVSI. 2004. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] Junio de 2004. <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
13. Eaprende.com Aprende.com. [En línea] // Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. - 2001. - enero de 2011. - <http://www.eaprende.com/gestor-de-basededatos-mysql-postresql-sqlite.html>

14. REYNOSO C y KICCILLOF N. [En línea] // Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. - 2007. - <http://www.willydev.net/descargas/prev/Estiloypatron.pdf>.
15. Cala Hinojosa, Alexei-2011-SISTEMA DE GESTIÓN ENERGÉTICA ASISTIDA PARA EL ISMMM-MÓDULO DE CONTROL DEL CONSUMO ENERGÉTICO.
16. Ronda Amador Yoelys, Cobas Santos Kadir y Marrero Viñas Daymel El modelo cliente-servidor de tres capas para el trabajo en redes de información [Libro]. - 2002.
17. Wesley, A. (2000). "Una explicación de la programación extrema. Aceptar el cambio."
18. Prácticas básicas de la programación extrema, <http://es.wikipedia.org/wiki/extrema.htm>, en inglés, <http://es.xprogramming.com/xpmag/whatisxp.htm>
19. Letelier, P. y P., M. C. (2008). "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Valencia, Universidad Politécnica de Valencia."
20. Herramientas CASE, Objetivos http://es.wikipedia.org/wiki.php/Herramienta_CASE.htm
21. Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) 6.0 http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/
22. ER/STUDIO. "Herramienta para el modelaje de datos Embarcadero." Disponible en: http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&ke.
23. Neuron Corp. [En línea] // ¿Qué es UML? - 2006. - enero de 2011. - http://www.neuronsrl.com.ar/training/uml/uml_intro.html.
24. Vico.org Vico.org [En línea] // Unified Modeling Language. - 2002. - enero de 2011. - <http://www.vico.org/FormMentorOutsourcingUML.pdf>.
25. Mike Cohn, "User Stories Applied", 2004, Addison Wesley.
26. <http://www.slideshare.net/MiquelMora/historias-de-usuario>. [En línea] Historias de usuario ¿Por qué? ¿Qué son? ¿Cómo son? Carbo, Miguel Dec 10, 2010.

27. Beck, K. (2002). "Test-Driven Development By Example, Addison-Wesley Professional."
28. María F. S. Rincondelvago.com [En línea] // "Evaluación de proyectos.". - 2009. - noviembre de 2010. - <http://html.rincondelvago.com/evaluacion-de-proyectos.html>.
29. Torres D. [En línea] // Proyectos Informáticos. - 2005. - enero de 2011. - <http://notasprisma.tripod.com/Proyectos.htm>.
30. Rebaza J. C. V. Sociedad de Estudiantes de Ciencia de la Computación SECC [En línea] // Metodología para la Priorización de Proyectos Informáticos. - 2007. - enero de 2011. - <http://www.seccperu.org/node/302>
31. Flanagan, David (2002). JavaScript: The Definitive Guide (4^a Edición edición).
32. Nieves Borrero Martha y Góngora Rodríguez Asnier Herramienta Informática para automatizar los procesos en el Laboratorio de Calidad de Software: Módulo Gestión de las No Conformidades [Libro]. - Ciudad de La Habana : Universidad de las Ciencias Informáticas, 2007.

Otras Bibliografías

Daniel H. Steinberg and Daniel W. Palmer: Extreme Software Engineering, Pearson Education, Inc., ISBN 0-13-047381-2

Mike Cohn: Agile Estimating and Planning, 2006, Prentice Hall, ISBN 0-13-147941-5

<http://www.computerworld.com.au/index.php?id=760310963>

<http://c2.com/doc/oopsla89/paper.html>

<http://www.quizzpot.com/>

<http://www.forosdelweb.com/>

<http://www.cristalab.com/tutoriales/>

<http://www.sencha.com/forum>

<http://www.genbetadev.com/metodologias-de-programacion/historias-de-usuario-una-forma-natural-de-analisis-funcional>



<http://www.slideshare.net/MiquelMora/historias-de-usuario>

http://es.wikipedia.org/wiki/Historias_de_usuario

Anexos

Anexo 1 Historias de Usuario

Tabla 2.7 Historia de Usuario Administrar miembros

Historia de Usuario	
Código: HUGA1	Nombre: Administrar miembros
Modificación de Historia de Usuario Número: Ninguna	
Referencia: Ítems RFGA1, RFGA2, RFGA3, RFGA4, RFGA5.	
Programador: Alexander Garcés Valdés	Iteración Asignada: Primera
Prioridad: Importancia: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Dificultad: Alta	Puntos Reales: 2,20
Descripción: Permite al administrador del sistema crear, modificar, buscar, listar y eliminar cualquier miembro del sistema.	
Observaciones: Confirmado con el cliente.	

Tabla 2.8 Historia de Usuario Administrar grupos

Historia de Usuario	
Código: HUGA2	Nombre: Administrar grupos
Modificación de Historia de Usuario Número: Ninguna	
Referencia: Ítems RFGA6, RFGA7, RFGA8, RFGA9, RFGA10, RFGA11.	
Programador: Alexander Garcés Valdés	Iteración Asignada: Primera
Prioridad: Importancia: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Dificultad: Alta	Puntos Reales: 2,15
Descripción: Permite al administrador del sistema crear, modificar, buscar, listar y eliminar cualquier grupo del sistema; así como otorgarle los privilegios deseados al grupo.	

Observaciones: Confirmado con el cliente.

Tabla 2.9 Historia de Usuario Administrar privilegios

Historia de Usuario	
Código: HUGA3	Nombre: Administrar privilegios
Modificación de Historia de Usuario Número: Ninguna	
Referencia: Ítems RFGA9, RFGA20, RFGA21, RFGA22, RFGA23, RFGA24.	
Programador: Alexander Garcés Valdés	Iteración Asignada: Cuarta
Prioridad: Importancia: Baja	Puntos Estimados: 1
Riesgo en Desarrollo: Dificultad: Baja	Puntos Reales: 1,40
Descripción: Permite al administrador del sistema crear, modificar, buscar, listar y eliminar cualquier privilegio del sistema; así como otorgarle los módulos deseados a cada privilegio.	
Observaciones: Confirmado con el cliente.	

Tabla 2.10 Historia de Usuario Administrar asignatura

Historia de Usuario	
Código: HUGA5	Nombre: Administrar asignatura
Modificación de Historia de Usuario Número: Ninguna	
Referencia: Ítems RFGA19.	
Programador: Alexander Garcés Valdés	Iteración Asignada: Cuarta
Prioridad: Importancia: baja	Puntos Estimados: 1
Riesgo en Desarrollo: Dificultad: Baja	Puntos Reales: 1
Descripción: Permite al administrador del sistema y a los profesores insertar las asignaturas según su año o especialidad, también permite asignarle las notas a cada estudiante.	

Observaciones: Confirmado con el cliente.

Tabla 2.11 Historia de Usuario Informes de estudiantes matriculados

Historia de Usuario	
Código: HUGA6	Nombre: Informes de estudiantes matriculados
Modificación de Historia de Usuario Número: Ninguna	
Referencia: Ítems RFGA25, RFGA26, RFGA27, RFGA28.	
Programador: Alexander Garcés Valdés	Iteración Asignada: Tercera
Prioridad: Importancia: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Dificultad: Media	Puntos Reales: 2,40
Descripción: Esta HU permite realizar los informes de estudiantes por grado y edad; sexo, año y grupo, etc.	
Observaciones: Confirmado con el cliente.	

Tabla 2.12 Historia de Usuario Administrar tipo de enseñanza

Historia de Usuario	
Código: HUGA7	Nombre: Administrar tipo de enseñanza
Modificación de Historia de Usuario Número: Ninguna	
Referencia: Ítems RFGA25, RFGA26, RFGA27, RFGA28.	
Programador: Alexander Garcés Valdés	Iteración Asignada: Tercera
Prioridad: Importancia: Media	Puntos Estimados: 2
Riesgo en Desarrollo: Dificultad: Media	Puntos Reales: 2,40
Descripción: Permite al administrador del sistema crear los diferentes tipos de enseñanza que existen en el centro, así como eliminarlos, modificarlos o	

listarlos.

Observaciones: Confirmado con el cliente.

Tabla 2.13 Historia de Usuario Boleta de Movimiento de Alumnos

Historia de Usuario	
Código: HUGA8	Nombre: <i>Boleta de Movimiento de Alumnos</i>
Modificación de Historia de Usuario Número: Ninguna	
Referencia: Ítems RFGA34, RFGA35.	
Programador: Alexander Garcés Valdés	Iteración Asignada: Segunda
Prioridad: Importancia: Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Dificultad: Alta	Puntos Reales: 2,20
Descripción: Permite al administrador del sistema crear y listar las boletas de Movimiento de Alumnos.	
Observaciones: Confirmado con el cliente.	

Anexo 2 Tareas de Ingeniería

Tabla 3.9 Tarea de ingeniería Insertar miembro

Tarea de Ingeniería	
Número de la tarea: 1	Número de HU: 1
Nombre de la tarea: Insertar miembro	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10/02/2012	Fecha fin: 12/02/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite insertar un miembro al sistema, es realizada por el administrador del sistema.	

Tabla 3.10 Tarea de ingeniería listar miembros

Tarea de Ingeniería

Número de la tarea: 2	Número de HU: 1
Nombre de la tarea: Listar miembros	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 11/02/2012	Fecha fin: 13/02/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite listar los miembros del sistema, es realizada por el administrador del sistema.	

Tabla 3.11 Tarea de ingeniería modificar miembro

Tarea de Ingeniería	
Número de la tarea: 3	Número de HU: 1
Nombre de la tarea: Modificar miembro	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 13/02/2012	Fecha fin: 14/02/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite modificar los miembros del sistema, es realizada por el administrador del sistema.	

Tabla 3.12 Tarea de ingeniería buscar miembro

Tarea de Ingeniería	
Número de la tarea: 4	Número de HU: 1
Nombre de la tarea: Buscar miembro	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 15/02/2012	Fecha fin: 17/02/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite buscar de manera personalizada los miembros del sistema, es realizada por el administrador del sistema.	

Tabla 3.13 Tarea de ingeniería eliminar miembro

Tarea de Ingeniería	
Número de la tarea: 5	Número de HU: 1
Nombre de la tarea: Eliminar miembro	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 18/02/2012	Fecha fin: 21/02/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite eliminar los miembros del sistema, es realizada por el administrador del sistema.	

Tabla 3.14 Tarea de ingeniería Insertar grupo

Tarea de Ingeniería	
Número de la tarea: 6	Número de HU: 2
Nombre de la tarea: Insertar grupo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 22/02/2012	Fecha fin: 25/02/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite insertar los grupos del sistema, es realizada por el administrador del sistema.	

Tabla 3.15 Tarea de ingeniería Proporcionar privilegios

Tarea de Ingeniería	
Número de la tarea: 7	Número de HU: 2
Nombre de la tarea: Proporcionar privilegios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 25/02/2012	Fecha fin: 29/02/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite dar el privilegio que desee el administrador a cada grupo.	

Tabla 3.16 Tarea de ingeniería Listar grupos

Tarea de Ingeniería	
Número de la tarea: 8	Número de HU: 2
Nombre de la tarea: Listar grupos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 1/03/2012	Fecha fin: 3/03/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite listar los grupos del sistema, es realizada por el administrador del sistema.	

Tabla 3.17 Tarea de ingeniería Modificar grupo

Tarea de Ingeniería	
Número de la tarea: 9	Número de HU: 2
Nombre de la tarea: Modificar grupo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 4/03/2012	Fecha fin: 10/03/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite modificar los grupos del sistema, es realizada por el administrador del sistema.	

Tabla 3.18 Tarea de ingeniería Buscar grupos

Tarea de Ingeniería	
Número de la tarea: 10	Número de HU: 2
Nombre de la tarea: Buscar grupos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/03/2012	Fecha fin: 15/03/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite buscar de manera personalizada los grupos del sistema, es realizada por el administrador del sistema.	

Tabla 3.19 Tarea de ingeniería Eliminar grupos

Tarea de Ingeniería	
Número de la tarea: 11	Número de HU: 2
Nombre de la tarea: Eliminar grupos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/03/2012	Fecha fin: 20/03/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite eliminar los grupos del sistema, es realizada por el administrador del sistema.	

Tabla 3.20 Tarea de ingeniería Insertar privilegios

Tarea de Ingeniería	
Número de la tarea: 12	Número de HU: 3
Nombre de la tarea: Insertar privilegios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20/03/2012	Fecha fin: 23/03/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite insertar los privilegios del sistema, es realizada por el administrador del sistema.	

Tabla 3.21 Tarea de ingeniería Listar privilegios

Tarea de Ingeniería	
Número de la tarea: 13	Número de HU: 3
Nombre de la tarea: Listar privilegios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24/03/2012	Fecha fin: 28/03/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite listar los privilegios del sistema, es realizada por el administrador del sistema.	

Tabla 3.22 Tarea de ingeniería Modificar privilegios

Tarea de Ingeniería	
Número de la tarea: 14	Número de HU: 3
Nombre de la tarea: Modificar privilegios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 28/03/2012	Fecha fin: 31/03/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite modificar los privilegios del sistema, es realizada por el administrador del sistema.	

Tabla 3.23 Tarea de ingeniería Buscar privilegios

Tarea de Ingeniería	
Número de la tarea: 15	Número de HU: 3
Nombre de la tarea: Buscar privilegios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 1/04/2012	Fecha fin: 5/04/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite buscar los privilegios del sistema, es realizada por el administrador del sistema.	

Tabla 3.24 Tarea de ingeniería Eliminar privilegios

Tarea de Ingeniería	
Número de la tarea: 16	Número de HU: 3
Nombre de la tarea: Eliminar privilegios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 6/04/2012	Fecha fin: 8/04/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite eliminar los privilegios del sistema, es realizada por el administrador del sistema.	

Tabla 3.25 Tarea de ingeniería Administrar módulos

Tarea de Ingeniería	
Número de la tarea: 17	Número de HU: 3
Nombre de la tarea: Administrar módulos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 9/04/2012	Fecha fin: 12/04/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite al administrador del sistema otorgarle los módulos necesarios a cada privilegio.	

Tabla 3.26 Tarea de ingeniería Insertar tipo de enseñanza

Tarea de Ingeniería	
Número de la tarea: 18	Número de HU: 4
Nombre de la tarea: Insertar tipo de enseñanza	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20/04/2012	Fecha fin: 25/04/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite insertar los tipos de enseñanza del sistema, es realizada por el administrador del sistema.	

Tabla 3.27 Tarea de ingeniería Modificar tipo de enseñanza

Tarea de Ingeniería	
Número de la tarea: 19	Número de HU: 4
Nombre de la tarea: Modificar tipo de enseñanza	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/04/2012	Fecha fin: 28/04/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite modificar los tipos de enseñanza del sistema, es realizada por el administrador del sistema.	

Tabla 3.28 Tarea de ingeniería Crear grupos por enseñanza

Tarea de Ingeniería	
Número de la tarea: 20	Número de HU: 4
Nombre de la tarea: Crear grupos por enseñanza	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 29/04/2012	Fecha fin: 1/05/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite insertar los grupos por los diferentes tipos de enseñanza, es realizada por el administrador del sistema.	

Tabla 3.29 Tarea de ingeniería Insertar asignatura

Tarea de Ingeniería	
Número de la tarea: 21	Número de HU: 5
Nombre de la tarea: Insertar asignatura	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 2/05/2012	Fecha fin: 5/05/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite insertar las asignaturas al sistema, es realizada por el administrador del sistema.	

Tabla 3.30 Tarea de ingeniería Definir sistema de evaluaciones de una asignatura

Tarea de Ingeniería	
Número de la tarea: 22	Número de HU: 4
Nombre de la tarea: Definir sistema de evaluaciones de una asignatura.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 5/05/2012	Fecha fin: 7/05/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea definir el sistema de evaluaciones de una asignatura, es realizada por el administrador del sistema.	

Tabla 3.31 Tarea de ingeniería Registrar nota de un estudiante en una evaluación

Tarea de Ingeniería	
Número de la tarea: 23	Número de HU: 4
Nombre de la tarea: Registrar nota de un estudiante en una evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 7/05/2012	Fecha fin: 10/05/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite registrar las notas de un estudiante en una asignatura, es realizada por el administrador del sistema y los profesores.	

Tabla 3.32 Tarea de ingeniería Modificar nota de un estudiante en una evaluación

Tarea de Ingeniería	
Número de la tarea: 24	Número de HU: 4
Nombre de la tarea: Modificar nota de un estudiante en una evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/05/2012	Fecha fin: 16/05/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite modificar la nota de un estudiante en una evaluación, es realizada por el administrador del sistema.	

Tabla 3.33 Tarea de ingeniería Consultar notas de un estudiante

Tarea de Ingeniería	
Número de la tarea: 25	Número de HU: 4
Nombre de la tarea: Consultar notas de un estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/05/2012	Fecha fin: 20/05/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite listar las notas de un estudiante, es	

realizada por el administrador del sistema y los profesores.

Tabla 3.34 Tarea de ingeniería Cerrar asignatura

Tarea de Ingeniería	
Número de la tarea: 26	Número de HU: 4
Nombre de la tarea: Cerrar asignatura.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 21/05/2012	Fecha fin: 24/05/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite cerrar una asignatura concluida por un estudiante, es realizada por el administrador del sistema y los profesores.	

Tabla 3.35 Tarea de ingeniería Insertar matrícula de un estudiante

Tarea de Ingeniería	
Número de la tarea: 27	Número de HU: 5
Nombre de la tarea: Insertar matrícula de un estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/05/2012	Fecha fin: 1/06/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite insertar la matrícula de un estudiante, es realizada por el administrador del sistema.	

Tabla 3.36 Tarea de ingeniería Buscar matrícula de un estudiante

Tarea de Ingeniería	
Número de la tarea: 28	Número de HU: 4
Nombre de la tarea: Buscar matrícula de un estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 1/06/2012	Fecha fin: 3/06/2012
Programador responsable: Alexander Garcés Valdés	

Descripción: Esta tarea permite buscar los estudiantes matriculados, es realizada por el administrador del sistema.

Tabla 3.37 Tarea de ingeniería Modificar matrícula de un estudiante

Tarea de Ingeniería	
Número de la tarea: 29	Número de HU: 4
Nombre de la tarea: Modificar la matrícula de un estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 4/06/2012	Fecha fin: 6/06/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite modificar las matricula de un estudiante, es realizada por el administrador del sistema.	

Tabla 3.38 Tarea de ingeniería Renovar matrícula de un estudiante

Tarea de Ingeniería	
Número de la tarea: 30	Número de HU: 4
Nombre de la tarea: Renovar la matrícula de un estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 6/06/2012	Fecha fin: 8/06/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite renovar la matrícula de un estudiante, es realizada por el administrador del sistema.	

Tabla 3.39 Tarea de ingeniería Listar informe de matrícula por grado, sexo y edad

Tarea de Ingeniería	
Número de la tarea: 31	Número de HU: 4
Nombre de la tarea: Listar informe de matrícula por grado, sexo y edad	
Tipo de tarea: Desarrollo	Puntos estimados: 1

Fecha inicio: 12/06/2012	Fecha fin: 15/06/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea listar los informes de matrícula por grado, sexo y edad, es realizada por el administrador del sistema y los profesores.	

Tabla 3.40 Tarea de ingeniería Listar informe de sexo por grado

Tarea de Ingeniería	
Número de la tarea: 32	Número de HU: 4
Nombre de la tarea: Listar informe de sexo por grado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 15/06/2012	Fecha fin: 18/06/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea listar los informes de sexo y grado, es realizada por el administrador del sistema y los profesores.	

Tabla 3.41 Tarea de ingeniería Listar informe de color de la piel por grado

Tarea de Ingeniería	
Número de la tarea: 33	Número de HU: 4
Nombre de la tarea: Listar informe de color de la piel por grado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 18/06/2012	Fecha fin: 20/06/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea listar los informes de color de la piel por grado, es realizada por el administrador del sistema y los profesores.	

Tabla 3.42 Tarea de ingeniería Listar reportes de altas y bajas

Tarea de Ingeniería	
Número de la tarea: 34	Número de HU: 4
Nombre de la tarea: Crear reportes de altas y bajas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1

Fecha inicio: 20/06/2012	Fecha fin: 23/06/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite crear los reportes de altas y bajas de los estudiantes, es realizada por el administrador del sistema.	

Tabla 3.43 Tarea de ingeniería Listar reportes de altas y bajas

Tarea de Ingeniería	
Número de la tarea: 34	Número de HU: 4
Nombre de la tarea: Listar estudiantes en movimiento.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20/06/2012	Fecha fin: 24/06/2012
Programador responsable: Alexander Garcés Valdés	
Descripción: Esta tarea permite listar estudiantes que se encuentren en movimiento de traslado hacia otro centro o en movimiento de alta, es realizada por el administrador del sistema.	

Anexo 3 Tarjetas CRC

Pre	
Description: Guarda los datos del pre	
Attributes:	
Name	Description
id_pre	Identificador del pre
nombre_pre	
id_centro	
tipo_ensennanza	
Responsibilities:	
Name	Collaborator
insertar pre	centro
eliminar pre	
listar pre	

Fig. 2.2 Tarjeta CRC Pre

ETP	
Description: Guarda los datos de etp	
Attributes:	
Name	Description
id_etp	Identificador de etp
nombre_etp	
id_centro	
tipo_ensennanza	
Responsibilities:	
Name	Collaborator
insertar etp	centro
listar etp	
eliminar etp	

Fig. 2.3 Tarjeta CRC ETP

Grado	
Description: Guarda los datos del grado	
Attributes:	
Name	Description
id_grado	Identificador del grado
grado_academico	
id_pre	
Responsibilities:	
Name	Collaborator
Insertar grado	Pre
listar grado	
modificar grado	

Fig. 2.4 Tarjeta CRC Grado

Especialidad	
Description: Guarda los datos de la especialidad	
Attributes:	
Name	Description
id_especialidad	Identificador de la especialidad
nombre_especialidad	
id_etp	
Responsibilities:	
Name	Collaborator
Insertar especialidad	ETP
listar especialidades	
eliminar especialidades	

Fig. 2.5 Tarjeta CRC Especialidad

Grupo	
Description: Guarda los datos del grupo	
Attributes:	
Name	Description
id_grupo	Identificador del grupo
nombre_grupo	
id_grado	
id_anno_escolar	
Responsibilities:	
Name	Collaborator
Insertar grupo	grado
Eliminar grupo	anno_escolar
Listar grupos	

Fig. 2.6 Tarjeta CRC Grupo

Anno_escolar	
Description: Guarda los datos del año escolar	
Attributes:	
Name	Description
id_anno_escolar	Identificador del Anno escolar
anno	
id_especialidad	
Responsibilities:	
Name	Collaborator
insertar anno_escolar	Especialidad
listar anno_escolar	
eliminar anno_escolar	

Fig. 2.7 Tarjeta CRC Anno_Escolar

Estudiante	
Description: Guarda los datos del estudiante	
Attributes:	
Name	Description
id_estudiante	Identificador del estudiante
nombre_est	
apellidos	
edad	
sexo	
color_piel	
procs_padre	
procs_madre	
direccion	
provincia	
municipio	
id_grupo	
Responsibilities:	
Name	Collaborator
Insertar estudiantes	grupo
Listar estudiantes	
Eliminar estudiantes	
Asignar notas	

Fig. 2.8 Tarjeta CRC Estudiante

Asignatura	
Description: Guarda las asignaturas	
Attributes:	
Name	Description
id_asignatura	Identificador de la asignatura
nombre_asignatura	
Responsibilities:	
Name	Collaborator
Insertar asignatura	
Listar asignaturas	
Eliminar asignaturas	

Fig. 2.9 Tarjeta CRC Asignatura

Examen	
Description: Guarda los datos de los exámenes	
Attributes:	
Name	Description
id_estudiante	Identificador del estudiante
id_tipo_examen	
id_asignatura	
nota	
Responsibilities:	
Name	Collaborator
Insertar notas	tipo_examen
	asignatura
	estudiante

Fig. 2.10 Tarjeta CRC Examen

Tipo_examen	
Description: Guarda las los tipos de examen	
Attributes:	
Name	Description
id_tipo_examen	Identificador del examen
examen_tipo	
Responsibilities:	
Name	Collaborator
Insertar tipo de examen	
Listar tipo de examen	
Eliminar tipo de examen	

Fig. 2.11 Tarjeta CRC Tipo_Examen

Usuarios	
Description: Guarda los datos de los usuarios del sistema	
Attributes:	
Name	Description
id_user	
usuario	
contrasenna	
Responsibilities:	
Name	Collaborator
Insertar usuarios	
Eliminar usuarios	
Listar usuarios	

Fig. 2.12 Tarjeta CRC Usuarios

Anexo 4 Pruebas de Aceptación

Tabla 3.9 Prueba de aceptación para la HU Administrar Grupos.

Prueba de Aceptación: Administrar Grupos
HU: Administrar Grupos
Nombre: Prueba para comprobar la administración de grupos.
Descripción: Con esta prueba de aceptación se persigue validar la administración de grupos.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder al módulo de administración de grupos y poder adicionar, eliminar, mostrar o modificar un grupo.
Entrada / Pasos de ejecución: El administrador hace clic en el botón " Add " e ingresa los datos del grupo que desea insertar, luego presiona el botón " Guardar " y es creado un nuevo grupo. En seguida que el miembro es creado el administrador le otorga uno o más privilegios, además puede ser

modificado o eliminado.
Resultado: Se insertan correctamente los grupos. Se visualizan todos los grupos correctamente. Se le atribuyen correctamente los privilegios. Se pueden eliminar o modificar una vez insertados.
Evaluación de la prueba: Aceptada.

Tabla 3.10 Prueba de aceptación para la HU Administrar Privilegios.

Prueba de Aceptación: Administrar Privilegios
HU: Administrar Privilegios
Nombre: Prueba para comprobar la administración de privilegios.
Descripción: Con esta prueba de aceptación se persigue validar la administración de grupos.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder al módulo de administración de privilegios y poder adicionar, eliminar, mostrar o modificar un privilegio.
Entrada / Pasos de ejecución: El administrador hace clic en el botón " Add " e ingresa los datos del privilegio que desea insertar, luego presiona el botón " Guardar " y es creado un nuevo privilegio. En seguida que el privilegio es creado el administrador le da acceso a los módulos necesarios para este privilegio, además puede ser modificado o eliminado.
Resultado: Se insertan correctamente los privilegios. Se visualizan todos los privilegios correctamente. Se le atribuyen correctamente los módulos necesarios. Se pueden eliminar o modificar una vez insertados.

Evaluación de la prueba: Aceptada.

Tabla 3.11 Prueba de aceptación para la HU Administrar Tipos de Enseñanza.

Prueba de Aceptación: Administrar tipo de enseñanza
HU: Administrar tipo de enseñanza
Nombre: Prueba para comprobar la administración de tipos de enseñanza.
Descripción: Con esta prueba de aceptación se persigue validar la administración de tipos de enseñanza.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder al módulo de administración de tipo de enseñanza y poder adicionar, eliminar, mostrar o modificar un privilegio.
Entrada / Pasos de ejecución: El administrador hace clic en el botón " Add " e ingresa los datos del privilegio que desea insertar, luego presiona el botón " Guardar " y es creado un nuevo privilegio. En seguida que el privilegio es creado el administrador le da acceso a los módulos necesarios para este privilegio, además puede ser modificado o eliminado.
Resultado: Se insertan correctamente los privilegios. Se visualizan todos los privilegios correctamente. Se le atribuyen correctamente los módulos necesarios. Se pueden eliminar o modificar una vez insertados.
Evaluación de la prueba: Aceptada.

Tabla 3.12 Prueba de aceptación para la HU Administrar Asignaturas.

Prueba de Aceptación: Administrar asignaturas
HU: Administrar asignatura

Nombre: Prueba para comprobar la administración de asignaturas.
Descripción: Con esta prueba de aceptación se persigue validar la administración de asignaturas.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder al módulo de administración de asignaturas y poder adicionar, eliminar y mostrar una asignatura.
Entrada / Pasos de ejecución: El administrador ingresa el nombre de la asignatura, luego presiona el botón " Insertar asignatura " y es creada una nueva asignatura. En seguida que la asignatura es creada se lista en una tabla y puede ser eliminada si se desea.
Resultado: Se insertan correctamente las asignaturas Se listan todas las asignaturas correctamente. Se pueden eliminar una vez insertadas.
Evaluación de la prueba: Aceptada.

Tabla 3.13 Prueba de aceptación para la HU Administrar Matrícula.

Prueba de Aceptación: Administrar matrícula
HU: Administrar matrícula
Nombre: Prueba para comprobar la administración de matrículas.
Descripción: Con esta prueba de aceptación se persigue validar la administración de matrículas.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder al módulo de administración de matrícula y poder adicionar.
Entrada / Pasos de ejecución: El administrador ingresa los datos de la matrícula, luego presiona el botón " Guardar " y es creada una nueva matrícula, además puede ser modificado o eliminado.
Resultado:

Se inserta correctamente la matrícula.

Evaluación de la prueba: Aceptada.

Tabla 3.14 Prueba de aceptación para la HU Informes de estudiantes matriculados.

Prueba de Aceptación: Informes de estudiantes matriculados
HU: Informes de estudiantes matriculados
Nombre: Prueba para comprobar los informes de estudiantes matriculados
Descripción: Con esta prueba de aceptación se persiguen validar los informes de estudiantes matriculados
Condiciones de ejecución: El administrador o los profesores deben iniciar su sesión para acceder al módulo de reportes de estudiantes matriculados y poder visualizar el reporte deseado.
Entrada / Pasos de ejecución: El administrador o el profesor accede al módulo de reportes y hace clic en el botón que desee según el tipo de reporte que quiera visualizar, si quiere imprimir el reporte presiona el botón para imprimir.
Resultado: Se visualizan correctamente los reportes. Se pueden imprimir correctamente todos los reportes.
Evaluación de la prueba: Aceptada.

Tabla 3.14 Prueba de aceptación para la HU Boleta de estudiantes en movimiento.

Prueba de Aceptación: Boleta de estudiantes en movimiento
HU: Boleta de estudiantes en movimiento

Nombre: Prueba para comprobar las Boletas de estudiantes en movimiento
Descripción: Con esta prueba de aceptación se persiguen validar las boletas de estudiantes en movimiento
Condiciones de ejecución: El administrador o los profesores deben iniciar su sesión para acceder al módulo de reportes de estudiantes matriculados y poder visualizar el reporte deseado.
Entrada / Pasos de ejecución: El administrador accede al módulo de movimiento de estudiantes y hace clic en el botón que desee según el tipo de acción que quiera realizar, si desea listar los estudiantes en movimiento hace clic en el botón “Listar estudiantes en Movimiento”, o si desea crear una boleta de movimiento hace clic en el botón “Nueva Boleta”.
Resultado: Se visualizan correctamente los estudiantes en movimiento. Se insertan correctamente todos los estudiantes.
Evaluación de la prueba: Aceptada.