

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

Título: Aplicación web para la gestión de la información de cursos y entrenamientos de posgrado del ISMMM.

Autor: Nestor Antonio López Arcia

Tutores: Ing. Adriannys Céspedes Ramírez
Ing. Alexei Cala Hinojosa

Moa, 2014

“Año 56 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática del Instituto Superior Minero Metalúrgico de Moa para que hagan el uso que estimen pertinente con el mismo.

Para que así conste firmo la presente a los ____ días del mes de _____del 2014.

Nestor Antonio López Arcia

Firma del Autor

Ing. Adriannys Céspedes Ramírez

Firma del Tutor

Dedicatoria

*A mis padres porque les debo agradecer por la persona que hoy soy,
por el amor y el cariño que me han brindado, a ustedes les dedico
este trabajo.*

*A mi hermano por su incesante motivación de triunfo y todo el
apoyo que me ha dado.*

*A toda mi familia, por su preocupación en los momentos más
difíciles.*

A mis amistades por ser tan especiales.

Agradecimientos

A Yuri y a Jose por ayudarme en la universidad de un modo u otro.

A los profesores porque fueron mi guía en la universidad.

*A mis vecinos Raulito, Javier y Eliober que siempre han estado ahí
apoyándome.*

A mi tutora Adriannys por su apoyo brindado.

A Nápoles por su ayuda en la tesis.

RESUMEN

La Vicerrectoría de Investigación y Posgrado (VRIP) en el Instituto Superior Minero Metalúrgico de Moa (ISMMM) atiende los procesos asociados a la investigación, el posgrado, la superación de cuadros, la comercialización de la actividad científica, la colaboración internacional y la información científica técnica. Además, responde a nivel institucional por las áreas de resultados claves (Posgrado y Superación de cuadros, Ciencia e innovación tecnológica) y coordina las estrategias de informatización y relaciones internacionales.

Debido a esto en el VRIP, se maneja un gran volumen de información referente a los cursos y entrenamientos de posgrado, lo cual provoca que esta tarea se realice de una forma más lenta debido a que las consultas realizadas sobre los datos se deben realizar de forma manual por una o varias personas. Por lo que se propone el desarrollar la informatización de la gestión de la información correspondiente a los cursos y entrenamientos de posgrado a partir de la creación de una aplicación web para controlar toda la información que se maneja de modo eficaz y seguro.

Para la creación de la aplicación web se efectuó un análisis sobre las aplicaciones web y las herramientas para el desarrollo de las mismas.

Abstract

The Vicerectory of Investigation and Postgrad (VRIP) in the Institute of Mining Metallurgical of Moa (ISMMM) assist the processes associated to the investigation, the postgrad, and the growth process of squares, the commercialization of the scientific activity, the international collaboration and the scientific technical information. Also, responds at institutional level for the areas of key (Postgrad and growth process of squares, Science and technological innovation) results and it coordinates the informatization strategies and international relationships.

Due to this in VRIP, a great volume of information is managed with respect to the courses and postgrad trainings, that which provokes that this task is carried out in a slower way because the consultations carried out on the data should be carried out in a manual way for an or several people. For what intends developing the informatization of the management of the information corresponding to the courses and postgrad trainings starting from the creation of an application web to control all the information that is managed in an effective and sure way.

For the creation of the application web an analysis was made on the applications web and the tools for the development of the same ones.

Contenido

Introducción	1
Capítulo 1 – Fundamentación Teórica	5
1.1 Introducción	5
1.2 Estado del Arte	5
1.2.1 Descripción de cursos y entrenamientos de posgrados	5
1.2.2 Soluciones existentes para la gestión de información de posgrados	6
1.2.3 Propuesta de solución	7
1.3 Lenguajes y herramientas utilizados	7
1.3.1 Lenguajes de programación	7
1.3.2 Servidor web	8
1.3.3 Marco de trabajo (Framework)	9
1.3.4 Sistema Gestor de Base de Datos	11
1.3.5 Entorno de Desarrollo Integrado	12
1.3.6 Herramientas CASE	13
1.4 Metodología para el desarrollo de la aplicación	15
1.4.1 Scrum	15
1.4.2 Programación Extrema (Extreme Programming, XP)	20
1.4.3 Metodología propuesta para el desarrollo de la aplicación	26
1.5 Patrón Arquitectónico MVC	27
1.6 Conclusiones	29
Capítulo 2 – Planeación y Diseño	30
2.1 Introducción	30
2.2 Problema y situación problemática	30
2.3 Propuesta de solución	30
2.4 Descripción del proceso del negocio	31
2.5 Personal relacionado con el sistema	32
2.6 Lista de reserva	32
2.7 Historias de usuario	35

2.8 Planificación de entregas.....	36
2.8.1 Estimación de esfuerzo por Historia de Usuario	37
2.3.1 Plan de iteraciones	37
2.4 Tarjetas CRC.....	39
2.5 Conclusiones.....	42
Capítulo 3 – Desarrollo y Pruebas.....	43
3.1 Introducción	43
3.2 Diseño de la base de datos.....	43
3.3 Desarrollo de las iteraciones	44
3.3.1 Tareas por historias de usuario.....	45
3.3.2 Tareas de ingeniería	46
3.4 Pruebas	47
3.4.1 Desarrollo dirigido por pruebas	47
3.4.2 Pruebas de aceptación	48
3.5 Conclusiones.....	49
Capítulo 4 – Estudio de Factibilidad.....	51
4.1 Introducción	51
4.2 Evaluación Costo - Beneficio	51
4.3 Efectos económicos	52
4.3.1 Efectos directos	52
4.3.2 Efectos indirectos	52
4.3.3 Efectos externos	53
4.3.4 Intangibles	53
4.4 Elementos para identificar los Costos y Beneficios del Proyecto	53
4.5 Ficha de costo de un producto informático	54
4.6 Conclusiones.....	57
Conclusiones Generales	58
Recomendaciones.....	59
Referencias Bibliográficas.....	60
Bibliografías	62
Glosario de Términos.....	63

Anexo 1 – Historias de usuarios	68
Anexo 2 – Tarjetas CRC.....	71
Anexo 3 – Tareas de ingeniería	78
Anexo 4 – Pruebas de Aceptación	91

Índice de ilustraciones y tablas

Figura 1.1 Pila del Producto	18
Figura 1.2 Gráfica de progreso	19
Figura 1.3 Patrón Arquitectura Modelo Vista Controlador	28
Figura 3.1 Modelo de la base de datos	44
Figura 4.1 Gráfica de la solución sin el producto y solución con el producto.	57
Tabla 2.1 Personal relacionado con el sistema	32
Tabla 2.10 Historia de usuario gestionar solicitud	69
Tabla 2.11 Historia de usuario gestionar especialidad miembro claustro	70
Tabla 2.12 Historia de usuario autenticar usuario	70
Tabla 2.13 Estimación de esfuerzo por Historia de Usuario	37
Tabla 2.14 Plan de duración de las iteraciones	37
Tabla 2.15 Tarjeta CRC de Posgrado	40
Tabla 2.16 Tarjeta CRC de Profesor	41
Tabla 2.17 Tarjeta CRC de Estudiante	41
Tabla 2.18 Tarjeta CRC de Estudiante	71
Tabla 2.19 Tarjeta CRC de Usuario.....	71
Tabla 2.2 Lista de reserva del producto	32
Tabla 2.20 Tarjeta CRC de Persona.....	72
Tabla 2.21 Tarjeta CRC de Evaluación	72
Tabla 2.22 Tarjeta CRC de Grado científico	73
Tabla 2.23 Tarjeta CRC de Claustro	73
Tabla 2.24 Tarjeta CRC de Especialidad	74
Tabla 2.25 Tarjeta CRC de Categoría docente.....	74

Tabla 2.26 Tarjeta CRC de Solicitud	75
Tabla 2.27 Tarjeta CRC de Modalidad	75
Tabla 2.28 Tarjeta CRC de País	76
Tabla 2.29 Tarjeta CRC del Permiso	76
Tabla 2.3 Historia de usuario gestionar posgrado	35
Tabla 2.30 Tarjeta CRC del Menú	77
Tabla 2.4 Historia de usuario gestionar material	36
Tabla 2.5 Historia de usuario gestionar usuario	68
Tabla 2.6 Historia de usuario gestionar profesor	68
Tabla 2.7 Historia de usuario gestionar estudiante	68
Tabla 2.8 Historia de usuario gestionar evaluación estudiante	69
Tabla 2.9 Historia de usuario gestionar claustro	69
Tabla 3.1 Distribución de tareas por historia de usuario	45
Tabla 3.10 Tarea de programación Buscar material	81
Tabla 3.11 Tarea de programación Eliminar material	81
Tabla 3.12 Tarea de programación Insertar estudiante	81
Tabla 3.13 Tarea de programación Mostrar estudiante	82
Tabla 3.14 Tarea de programación Buscar estudiante	82
Tabla 3.15 Tarea de programación Modificar estudiante	82
Tabla 3.16 Tarea de programación Eliminar estudiante	83
Tabla 3.17 Tarea de programación Insertar profesor	83
Tabla 3.18 Tarea de programación Mostrar profesor	83
Tabla 3.19 Tarea de programación Buscar profesor	84
Tabla 3.2 Tarea de programación Insertar posgrado	47
Tabla 3.20 Tarea de programación Modificar profesor	84
Tabla 3.21 Tarea de programación Eliminar profesor	84
Tabla 3.22 Tarea de programación Insertar especialidad miembro claustro	85
Tabla 3.23 Tarea de programación Mostrar especialidad miembro claustro	85
Tabla 3.24 Tarea de programación Buscar especialidad miembro claustro	85
Tabla 3.25 Tarea de programación Modificar especialidad miembro claustro	86
Tabla 3.26 Tarea de programación Eliminar especialidad miembro claustro	86
Tabla 3.27 Tarea de programación Insertar evaluación	86

Tabla 3.28 Tarea de programación Mostrar evaluación	87
Tabla 3.29 Tarea de programación Modificar evaluación	87
Tabla 3.3 Prueba de aceptación para la HU: Gestionar posgrado	49
Tabla 3.30 Tarea de programación Eliminar evaluación	87
Tabla 3.31 Tarea de programación Insertar claustro	88
Tabla 3.32 Tarea de programación Eliminar claustro	88
Tabla 3.33 Tarea de programación Insertar usuario	88
Tabla 3.34 Tarea de programación Mostrar usuario	89
Tabla 3.35 Tarea de programación Buscar usuario	89
Tabla 3.36 Tarea de programación Modificar usuario	89
Tabla 3.37 Tarea de programación Eliminar usuario	90
Tabla 3.38 Tarea de programación Insertar solicitud	90
Tabla 3.39 Tarea de programación Mostrar solicitud	90
Tabla 3.4 Tarea de programación Mostrar posgrado	79
Tabla 3.40 Tarea de programación Buscar solicitud	91
Tabla 3.41 Tarea de programación Eliminar solicitud	91
Tabla 3.42 Tarea de programación Autenticar usuario	91
Tabla 3.43 Prueba de aceptación para la HU: Gestionar Material	92
Tabla 3.44 Prueba de aceptación para la HU: Gestionar Estudiante	92
Tabla 3.45 Prueba de aceptación para la HU: Gestionar Profesor	93
Tabla 3.46 Prueba de aceptación para la HU: Gestionar Especialidad Miembro Claustro	94
Tabla 3.47 Prueba de aceptación para la HU: Gestionar Evaluación	95
Tabla 3.48 Prueba de aceptación para la HU: Gestionar Claustro	96
Tabla 3.49 Prueba de aceptación para la HU: Gestionar Usuario	97
Tabla 3.5 Tarea de programación Buscar posgrado	79
Tabla 3.50 Prueba de aceptación para la HU: Gestionar Solicitud	97
Tabla 3.51 Prueba de aceptación para la HU: Autenticar Usuario	98
Tabla 3.6 Tarea de programación Modificar posgrado	79
Tabla 3.7 Tarea de programación Eliminar posgrado	80
Tabla 3.8 Tarea de programación Insertar material	80
Tabla 3.9 Tarea de programación Mostrar material	80

Introducción

El Instituto Superior Minero Metalúrgico de Moa (ISMMM), se ha trazado como estrategia la informatización de la mayoría de sus procesos, también se ha hecho necesaria la creación de sistemas que mejoren y faciliten la gestión de la información en los procesos productivos y de servicios. Además se trabaja para incorporar las facilidades y ventajas del uso de las nuevas tecnologías en la formación de profesionales, en las investigaciones científicas y en otras que contribuyan a elevar el nivel técnico y humano.

La Vicerrectoría de Investigación y Posgrado (VRIP) atiende los procesos asociados a la investigación, el posgrado, la superación de cuadros, la comercialización de la actividad científica, la colaboración internacional y la información científica técnica. Además, responde a nivel institucional por las áreas de resultados claves (Posgrado y Superación de cuadros, Ciencia e innovación tecnológica) y coordina las estrategias de informatización y relaciones internacionales.

Particularmente, en el VRIP, se maneja un gran volumen de información referente a los cursos y entrenamientos de posgrado, lo cual provoca que esta tarea se realice de una forma más lenta debido a que las consultas realizadas sobre los datos se deben realizar de forma manual por una o varias personas.

También existen otros problemas que hacen que esta labor no se realice de la manera más adecuada, como la existencia de la mayoría de la documentación en formato duro, por lo que se corre el riesgo de que se deteriore. Otro problema es el relacionado con el acceso a la información por parte de los usuarios interesados en participar en los cursos de posgrados o de entrenamientos debido a que la información no se encuentra centralizada y se desperdicia mucho tiempo buscándola.

Por la situación expuesta anteriormente se identifica como **Problema científico** ¿Cómo favorecer la gestión de información en los cursos y entrenamientos de posgrados de la VRIP del ISMMM? Teniendo en cuenta el problema planteado el

objeto de estudio de este trabajo es el proceso de gestión de información en cursos y entrenamientos de posgrados del ISMMM, **el campo de acción** es la informatización de la gestión de información en cursos y entrenamientos de posgrados del ISMMM.

Para darle solución al problema se plantea el siguiente **objetivo general**:

Desarrollar una aplicación informática que favorezca la gestión de información de los cursos y entrenamientos de posgrados coordinados por la VRIP del ISMMM.

Como **idea a defender** se plantea que si se implementa una aplicación Web orientada a la gestión de información en los cursos y entrenamientos de posgrados, garantizando disponibilidad, control y seguridad de los datos, se favorecerá una de las actividades fundamentales de la VRIP del ISMMM.

Como guía de la investigación se plantean los siguientes **objetivos específicos** y las **tareas de la investigación**:

- A. Definir los aspectos esenciales de la gestión de información en los cursos y entrenamientos de posgrado.
 - A.1. Establecer los elementos teóricos de la gestión de información en los cursos y entrenamientos de posgrados.
 - A.2. Realizar un análisis crítico de los sistemas vinculados con la gestión de información en los cursos y entrenamientos de posgrado.
 - A.3. Analizar y seleccionar las herramientas y tecnologías para el desarrollo de la solución propuesta.
- B. Realizar análisis y diseño de la solución propuesta.
 - B.1. Realizar un análisis crítico de la situación actual del objeto de estudio.
 - B.2. Establecer el diseño de la solución propuesta.
 - B.3. Elaborar la documentación y los artefactos correspondientes a la metodología de desarrollo seleccionada.
- C. Implementar la solución propuesta.
 - C.1. Implementar el sistema informático propuesto.

C.2. Diseñar y realizar las pruebas funcionales al sistema construido.

D. Realizar el estudio de factibilidad.

D.1. Seleccionar la metodología para el estudio de factibilidad.

D.2. Realizar el estudio de factibilidad.

La investigación se realizó basada en los métodos mencionados a continuación:

Métodos teóricos

- **Análisis y síntesis:** Este método se utiliza en los fundamentos teóricos, en el procesamiento de la información y en la descomposición de cada uno de los requerimientos del sistema informático.
- **Histórico-Lógico:** Es utilizado para la comprensión de la evolución del empleo de herramientas para la gestión de la información que es generada por la interacción de dos o más personas.

Métodos empíricos

- **Entrevista:** Se usa con el objetivo de recopilar información, se deberá llevar a cabo un diálogo con las personas involucradas en la materia, esta será la vía fundamental para la determinación de los requerimientos del sistema.
- **La observación:** Es útil para entender el comportamiento del sistema y sus especificaciones.
- **Revisión de documentos:** lo utilizamos para conocer los detalles del funcionamiento del proceso de planificación de los cursos y entrenamientos de posgrado.

El presente trabajo consta de cuatro capítulos:

Capítulo 1 “Fundamentación Teórica”: En este capítulo se ofrece una breve descripción del objeto de estudio, objetivo general, tareas y algunos conceptos fundamentales. Se realiza un estudio acerca de los diferentes sistemas existentes vinculados al campo de acción, además se presenta la metodología y las herramientas a utilizar en el desarrollo del sistema propuesto.

Capítulo 2: “Planeación y Diseño”: Se hace uso de la metodología expuesta en el capítulo anterior para el desarrollo del proyecto, abordando en detalles cada una de sus fases.

Capítulo 3: “Desarrollo y Pruebas”: Se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo. Se describen las pruebas realizadas y sus resultados.

Capítulo 4: “Estudio de factibilidad”: En este se realiza un estudio para ver la factibilidad del producto. Además de un estudio de los esfuerzos requeridos para la realización de la Aplicación Web.

Capítulo 1 – Fundamentación Teórica

1.1 Introducción

En este capítulo se abordarán los conceptos fundamentales para el desarrollo de aplicaciones Web, se hará un estudio de las ventajas y desventajas que estas ofrecen, las características generales de los sistemas de bases de datos y lenguajes de programación Web para la implementación, además se verán las diferentes metodologías existentes para el desarrollo de aplicaciones, y de ellas las adecuadas a utilizar en este trabajo basando su selección en las ventajas que estas ofrecen. También, se hace una descripción de las herramientas y tecnologías utilizadas para el análisis y diseño del software.

1.2 Estado del Arte

En este epígrafe se realiza un estudio acerca de los Trabajos de Diploma, pues para la aplicación de cualquier estrategia es necesario o aconsejable apoyarse en los datos históricos del tema que se está tratando.

1.2.1 Descripción de cursos y entrenamientos de posgrados.

Existen dos tipos de educación de posgrado, la superación profesional y la formación académica.

La superación profesional está formada por los **Cursos, Entrenamiento, Diplomado, Conferencia especializada, Seminario, Taller y Debate científico**; mientras que la formación académica está compuesta por **Maestría, Especialidad de Posgrado y Doctorado**.

Los Cursos posibilitan la formación básica y especializada de los graduados universitarios. Tienen una extensión mínima de un crédito. Cada hora de docencia directa del profesor o tutor (actividad lectiva), implica no menos de tres horas de trabajo independiente del estudiante.

El Entrenamiento por su parte permite la formación básica y especializada, con énfasis en el desarrollo de habilidades, asimilación e introducción de nuevas

tecnologías para complementar, actualizar y perfeccionar conocimientos y habilidades prácticas. Tiene una extensión mínima de un crédito.

1.2.2 Soluciones existentes para la gestión de información de posgrados

En Cuba, el desarrollo de estos sistemas es escaso y principalmente se ha trabajado la presentación de posgrados en la Plataforma MOODLE.

MOODLE es un sistema de gestión de cursos de libre distribución que ayuda a los educadores a crear comunidades de aprendizaje en línea. Tiene una interfaz de navegador de tecnología sencilla, ligera, y compatible. Su arquitectura y herramientas son apropiadas para clases en línea, así como también para complementar el aprendizaje presencial. La instalación es sencilla requiriendo una plataforma que soporte PHP y la disponibilidad de una base de datos. También tiene una capa de abstracción de bases de datos por lo que soporta los principales sistemas gestores de bases de datos. Todos los formularios son revisados, las cookies cifradas, etc. La mayoría de las áreas de introducción de texto (materiales, mensajes de los foros, entradas de los diarios, etc.) pueden ser editadas usando el editor HTML, tan sencillo como cualquier editor de texto. (1)

Este sistema carece de muchas funcionalidades necesitadas en la aplicación que se desea obtener, por lo tanto no se puede hacer uso del mismo, solamente adquirir conocimiento de las funciones que este brinda.

También existen sistemas con similares propósitos, e incluso contamos en el ISMMM con un Software destinado a la gestión docente, SIGENU, solo que no está a disposición de la carrera, es única y exclusivamente utilizado por la secretaría docente. El proyecto SIGENU surge en Junio de 2004 a solicitud de la dirección del Ministerio de Educación Superior de Cuba (MES), como requisito a las necesidades de informatización de los procesos fundamentales de la gestión académica de una Institución de Educación Superior (IES) en todas sus modalidades de estudio. El sistema está compuesto por varios módulos que gestionan la información de un estudiante desde que se matricula hasta que se gradúa o causa baja definitiva.

1.2.3 Propuesta de solución.

Después de analizar la situación existente con respecto a la gestión de los cursos y entrenamientos de posgrados en el ISMMM, se determinó como propuesta de solución al problema existente, la realización de una Aplicación Web, que sea capaz de gestionar la información en los cursos y entrenamientos de posgrados que se realizan actualmente de manera manual. La realización de esta Aplicación será posible mediante herramientas libres como política del ISMMM. En cumplimiento con esto se utilizará como lenguaje de programación php, para esto la aplicación guardará sus datos en el gestor de base de datos PostgreSQL y se utilizará la metodología de programación extrema XP para llevar a cabo el proceso de desarrollo del software.

1.3 Lenguajes y herramientas utilizados

La humanidad con el desarrollo de las tecnologías lucha por hacer más rápido y seguro el manejo de las comunicaciones y el control de datos. Muchas tecnologías han revolucionado el mundo, en el campo de la informática, la tecnología Cliente-Servidor toma fuerza progresivamente debido a sus ventajas. La Web deja de ser un medio de divulgación de información para convertirse en una forma operativa, eficaz y accesible, desde cualquier parte para el control de datos.

1.3.1 Lenguajes de programación

1.3.1.1 HTML

El Lenguaje HTML es el idioma de la Web. Se basa en el uso de “Etiquetas” para la definición del formato del texto, los distintos elementos que conforman la página, sus propiedades y disposición. Este lenguaje es interpretado por los navegadores, procesado y convertido en una Web tal como la vemos en la pantalla, con imágenes, tablas, texto, videos y toda clase de elementos. El lenguaje está compuesto por etiquetas o marcas, gracias a ellos es posible darles forma a todos los componentes de una página o un documento HTML. Las

etiquetas de HTML están divididas en etiquetas de apertura y de cierre, aunque no siempre existen estas últimas. (2)

1.3.1.2 JavaScript

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 puede 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. (3)

1.3.1.3 Personal Home Page (PHP)

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. (4)

1.3.2 Servidor web

Un **servidor web** o **servidor HTTP** es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web.

Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones.

1.3.2.1 Apache

El **servidor HTTP Apache** es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras. 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. (5)

Características:

- 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. (6)

1.3.3 Marco de trabajo (Framework)

En general, con el término framework, nos estamos refiriendo a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. Un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta. Entre los objetivos principales de un framework se encuentran acelerar el proceso de desarrollo,

reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, es un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. El propósito de un framework es ayudar y facilitar el proceso de desarrollo de aplicaciones. Debe permitir desarrollar la aplicación rápido y fácilmente y debe resultar en una aplicación superior finalizada.

Los Frameworks son importantes en todas las fases de desarrollo, desde el diseño hasta el desarrollo y quizá más en el mantenimiento continuo. (7)

1.3.3.1 Ext JS (versión 3.3.0)

Ext JS es una librería de JavaScript construida para el desarrollo veloz de aplicaciones Web para los diferentes tipos de navegadores actuales, usando técnicas como Ajax, DHTML y manipulación del DOM. (8)

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.

Ventajas

- Una de las grandes ventajas de utilizar Ext JS es que 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, 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 qué datos desea transmitir al servidor y viceversa (Criterio este que puede variar con el uso de aplicaciones de precarga).
- 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. (9)

1.3.4 Sistema Gestor de Base de Datos

Un Sistema de Gestión de Bases de Datos (SGBD) es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizar todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una base de datos nunca se accede o manipula directamente, sino a través del SGBD. Se puede considerar al SGBD como la interfaz entre el usuario y la base de datos. El funcionamiento del SGBD está muy interrelacionado con el del Sistema Operativo, especialmente con el sistema de comunicaciones. El SGBD utilizará las facilidades del sistema de comunicaciones para recibir las peticiones del usuario (que puede estar utilizando un terminal físicamente remoto) y para devolverle los resultados. Las peticiones se realizan generalmente en forma de sentencias SQL (Structure Query Language), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado. (10)

1.3.4.1 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Es una herramienta muy potente para los desarrolladores de sistemas de bases de datos. Está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado

del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

Presenta alta concurrencia ya que permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

PostgreSQL cuenta con bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos muestra, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta.

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés). (11)

1.3.5 Entorno de Desarrollo Integrado

Un **entorno de desarrollo integrado**, llamado también **IDE** es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc.

1.3.5.1 NetBeans 7.3

Es un software, en el cual se pueden crear programas en un lenguaje de programación determinado, de manera rápida y fácil. Es una herramienta libre y gratuita. Permite programar aplicaciones principalmente en Java, pero también admite otros lenguajes como PHP. Algo muy importante de NetBeans es que es compatible con diversos sistemas operativos, tal como lo es Windows, Mac, Linux o Solaris, además de tener una fácil instalación. Actualmente hay disponibles dos productos: el NetBeans IDE y NetBeans Platform.

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. Es un producto libre y gratuito sin restricciones de uso.

NetBeans posee múltiples ventajas entre ellas se destacan las siguientes:

- Soporte a JavaScript, HTML, y PHP
- Interprete de fondo (Background Parser) capaz de identificar errores sintácticos en tiempo de edición.
- Completamiento de código.
- Marcado sintáctico que presenta en diferentes estilos de letras palabras claves, identificadores estándares y literales en general facilitando la claridad del código.
- Soporte a documentación tanto para JavaScript como para PHP.

1.3.6 Herramientas CASE

Las **herramientas CASE** (Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el

desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras

1.3.6.1 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. (12)

1.3.6.2 Visual Paradigm para UML

Es una herramienta que interpreta UML profesional, que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Una de las características más importantes del Visual Paradigm es que es multiplataforma. (13)

Características de Visual Paradigm:

- Licencia: gratuita y comercial.
- Producto de calidad.
- Soporta aplicaciones web.
- Varios idiomas.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.

1.4 Metodología para el desarrollo de la aplicación.

1.4.1 Scrum

Scrum es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software. Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas. (14)

Es un método adaptativo de gestión de proyectos que se basa en los principios ágiles:

- Colaboración estrecha con el cliente.
- Predisposición y respuesta al cambio
- Prefiere el conocimiento tácito de las personas al explícito de los procesos
- Desarrollo incremental con entregas funcionales frecuentes
- Comunicación verbal directa entre los implicados en el proyecto
- Motivación y responsabilidad de los equipos por la auto-gestión, auto-organización y compromiso.
- Simplicidad. Supresión de artefactos innecesarios en la gestión del proyecto.

Roles

En Scrum se definen varios roles, estos están divididos en dos grupos: cerdos y gallinas. El nombre de los grupos está inspirado en la siguiente frase.

De esta forma, los “cerdos” están comprometidos a desarrollar el software de forma regular y frecuente, mientras que todos los demás son 'gallinas' que sólo interesados en el proyecto, y si este falla, ellos no son los cerdos, es decir, ellos no fueron los que se comprometieron a hacerlo.

Las necesidades, deseos, ideas e influencias de los roles “gallina” se tienen en cuenta, pero no de forma que pueda afectar, distorsionar o entorpecer el proyecto Scrum.

Roles "Cerdo"

Propietario del Producto

El Propietario del Producto representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. Sus áreas de responsabilidad son:

- Financiación del producto.
- Requisitos del sistema.
- Retorno de la inversión del proyecto.
- Lanzamiento del proyecto.

Equipo

Responsable de transformar la pila del sprint (Sprint Backlog) en un incremento de la funcionalidad del software.

- Auto-gestionado.
- Auto-organizado.
- Multi-funcional.

Scrum Manager

Responsable del proceso Scrum.

- Formación y entrenamiento del proceso.
- Incorporación de Scrum en la cultura de la empresa.
- Garantía de cumplimiento de roles y responsabilidad.

Roles "Gallina"

Los roles gallina en realidad no son parte del proceso Scrum, pero deben tenerse en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (stakeholders). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

Análisis de la frase "Rol gallina":

La gallina alimenta al proyecto "poniendo huevos", no se ve comprometida como el cerdo que va al matadero.

Usuarios

Es el destinatario final del producto. Como bien lo dice la paradoja, El árbol cae en el bosque cuando no hay nadie ¿Hace ruido? Aquí la definición sería Si el software no es usado ¿fue alguna vez escrito?

Stakeholders (Clientes, Proveedores, Inversores)

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que lo justifica. Sólo participan directamente durante las revisiones del sprint.

Managers

Es la gente que establece el ambiente para el desarrollo del producto.

Sprint

Es el periodo de tiempo durante el que se desarrolla un incremento de funcionalidad. Constituye el núcleo de Scrum, que divide de esta forma el desarrollo de un proyecto en un conjunto de pequeñas “carreras”.

- Duración máxima: 30 días.
- Durante el sprint no se puede modificar el trabajo que se ha acordado en el Backlog.
- Sólo es posible cambiar el curso de un sprint, abortándolo, y sólo lo puede hacer el Scrum Master si decide que no es viable por alguna de las razones siguientes:
 - La tecnología acordada no funciona.
 - Las circunstancias del negocio han cambiado.
 - El equipo ha tenido interferencias.

Artefactos

Pila de producto (Product Backlog)

Listado con los requisitos del sistema

- Es responsabilidad del dueño del producto
 - Contenido
 - Priorización
 - Disponibilidad
- Nunca llega a ser una lista completa y definitiva.
- El empleado para planificar el proyecto es sólo una estimación inicial de requisitos.
- Es un documento dinámico que incorpora constantemente las necesidades del sistema.
- Se mantiene durante todo el ciclo de vida (hasta la retirada del sistema).

Pila del producto

Product Backlog		Estimación inicial	Complejidad	Estim. ajustada	Trabajo pendiente			
ID	Elemento				Sprint			
					1	2	3	4
1	Nuevo formulario para peticiones de clientes	2	0,2	2,4	2,4	0	0	0
2	Configuración de respuestas automáticas	3	0,2	3,6	3,6	0	0	0
3	Envío automático de respuestas	1	0,2	1,2	1,2	0	0	0
4	Consulta para los clientes de peticiones enviadas	1	0,2	1,2	1,2	0	0	0
5	Modificación del cliente de sus peticiones enviadas	2	0,2	2,4	2,4	0	0	0
6	Acceso a peticiones sólo para clientes del portal jurídico	5	0,2	6	6	0	6	0
7	Consulta de peticiones por parte del staff	1	0,2	1,2	1,2	0	0	0
SPRINT 1		15		18	18	0	0	0
8	Inserción de comentarios y reasignación a peticiones (staff)	2	0,2	1,2	1,2	1,2	0	0
9	Consultas por clientes, fechas y temas	3	0,2	3,6	3,6	3,6	0	0
10	[Continúa]....							

FIGURA 1.1 PILA DEL PRODUCTO

Pila del Sprint (Sprint backlog)

Trabajo o tareas determinadas por el equipo para realizar en un sprint y lograr al final del mismo un incremento de la funcionalidad. Se recomienda que las tareas reflejadas tengan una duración comprendida entre las 4 y las 16 horas de trabajo. Las de mayor duración deben intentar descomponerse en sub-tareas de ese rango de tiempo.

Gráfica de progreso

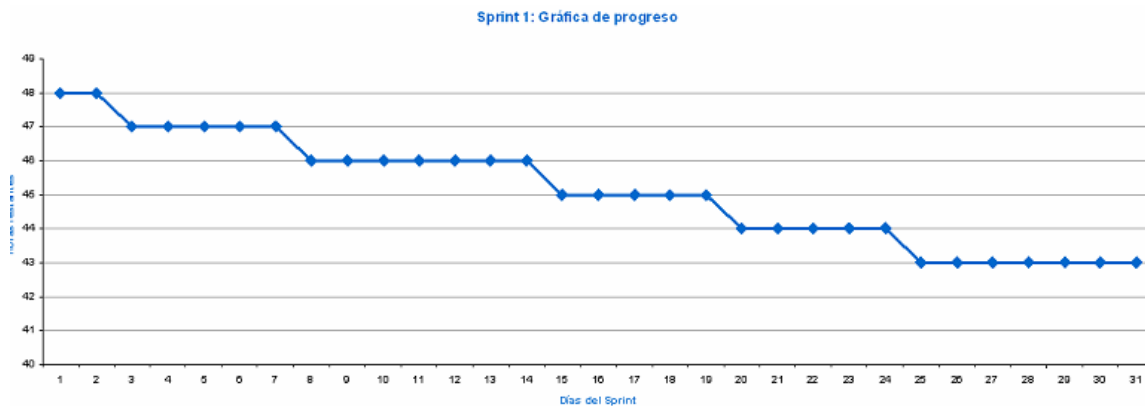


Figura 1.2 Gráfica de progreso

Burn down

La burn down chart es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros.

Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. A continuación se analizan varias metodologías de desarrollo de software escogiendo la más factible para el desarrollo de la herramienta a construir.

1.4.2 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. Se basa en realimentación 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. 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. (15)

1.4.2.1 Valores que promueve XP

Cuatro valores que promueven la metodología XP:

Simplicidad: XP propone el principio de hacer las cosas más simple que pueda funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana.

Comunicación: Algunos problemas en los proyectos tienen su origen en que alguien no dijo algo importante en algún momento. XP hace imposible la falta de comunicación.

Retroalimentación: Retroalimentación concreta y frecuente del cliente, del equipo y de los usuarios finales da una mayor oportunidad de dirigir el esfuerzo eficientemente.

Coraje: El coraje (valor) existe en el contexto de los otros 3 valores.

1.4.2.2 El ciclo de vida ideal de XP

El ciclo de vida ideal de XP consta de seis fases:

- Exploración.
- Planificación de la Entrega (Reléase)
- Iteraciones.
- Producción.
- Mantenimiento.
- Muerte del Proyecto.

1.4.2.3 Prácticas en las que se fundamenta XP

Planificación incremental

La programación extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de retroalimentación que permitan conocer con precisión dónde se está. Como es lógico, la planificación es iterativa: un representante del negocio decide al inicio de cada iteración qué características concretas se van a implementar.

El objetivo de XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional claro, desde el punto de vista del negocio. A estas versiones se les denomina *releases*.

Un release cuenta con un cierto número de historias. La historia es la unidad de funcionalidad en un proyecto XP, y corresponde a la mínima funcionalidad posible que tiene valor desde el punto de vista del negocio. Durante cada iteración se cierran varias historias, lo que hace que toda iteración añada un valor tangible para el cliente.

Gran parte de la eficacia de este modelo de planificación se deriva de una división clara de responsabilidades, que tiene en cuenta las necesidades del negocio en todo momento. Dentro de esta división, el representante del cliente tiene las siguientes responsabilidades:

- Decidir qué se implementa en cada release o iteración.
- Fijar las fechas de fin de un release, recortando unas características o añadiendo otras.
- Priorizar el orden de implementación, en función del valor de negocio.

Las responsabilidades del equipo de desarrollo son las siguientes:

- Estimar cuánto tiempo llevará una historia de usuario, esto es fundamental para el cliente, y puede llevarle a reconsiderar qué historias se deben incluir en una iteración.
- Proporcionar información sobre el coste de utilizar distintas opciones tecnológicas.
- Organizar el equipo.
- Estimar el riesgo de cada historia.
- Decidir el orden de desarrollo de historias dentro de la iteración.

Testing

La ejecución automatizada de tests es un elemento clave de la metodología XP. Existen tanto tests internos (o tests de unidad), para garantizar que el mismo es correcto, como tests de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los tests de aceptación, no necesariamente de implementarlos. Él es la persona mejor calificada para decidir cuál es la funcionalidad más valiosa.

Un efecto lateral importante de los tests, es que dan una gran seguridad a los desarrolladores; es posible llegar a hacer cambios más o menos importantes sin miedo a problemas inesperados, dado que proporciona una red de seguridad. La existencia de tests hace al código muy maleable.

Diseño simple

Una práctica fundamental de la programación extrema es utilizar diseños tan simples como sea posible. El principio es "utilizar el diseño más sencillo que consiga que todo funcione". La metodología XP nos pide, que no se viva bajo la ilusión de que un diseño puede resolver todas o gran parte de las situaciones futuras.

XP define un "diseño tan simple como sea posible" aquel que pasa todos los tests, no contiene código duplicado, deja clara la intención de los programadores (enfatisa el qué, no el cómo) en cada línea de código y contiene el menor número posible de clases y métodos.

Propiedad colectiva del código

XP aboga por la propiedad colectiva del código. En otras palabras, todo el mundo tiene autoridad para hacer cambios a cualquier código, y es responsable de ellos. Esto permite no tener que estar esperando a otros cuando todo lo que hace falta es algún pequeño cambio.

Integración continua

En muchos casos la integración de código produce efectos laterales imprevistos, y en ocasiones esta puede llegar a ser realmente traumática, cuando dejan de funcionar cosas por motivos desconocidos. La programación extrema hace que la integración sea permanente, con lo que todos los problemas se manifiestan de forma inmediata.

Clientes en el equipo

Algunos de los problemas más graves en el desarrollo son los que se originan cuando el equipo toma decisiones de negocio críticas. Esto no debería ocurrir, pero en el momento cumbre, con frecuencia no se obtiene retroalimentación del cliente con la fluidez necesaria.

La metodología XP intenta resolver este tipo de problemas integrando un representante del negocio dentro del equipo de desarrollo. Esta persona siempre está disponible para resolver dudas y para decidir qué se hace en cada momento, en función de los intereses del negocio.

Entregas pequeñas

Siguiendo la política de la metodología XP, de dar el máximo valor posible en cada momento, se intenta liberar nuevas versiones de las aplicaciones con frecuencia. Estas deben ser tan pequeñas como sea posible, aunque deben añadir suficiente valor, para que resulten meritorias para el cliente.

Semana de 40 horas

La programación extrema lleva un modo de trabajo en el que el equipo siempre está al 100%. Una semana de 40 horas en las que se dedica la mayor parte del tiempo a tareas que suponen un avance puede dar mucho de sí, y hace innecesario recurrir a sobreesfuerzos, excepto en casos extremos. Además, el esfuerzo continuado pronto lleva a un rendimiento menor y a un deterioro de la moral de todo el equipo.

1.4.2.4 Fases de la Metodología XP

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 (*user storie*) y permita hacer una estimación de tiempo de desarrollo del mismo.
- Se crea un plan de lanzamiento (*release planning*), 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 usa 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 (HU) 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.

1.4.3 Metodología propuesta para el desarrollo de la aplicación.

En este trabajo, se decide utilizar la metodología XP ya que es ideal para grupos de desarrollo relativamente pequeños y donde el tiempo disponible para la entrega del proyecto es corto; como es el caso de la presente investigación. Además de ser de fácil realización y requiere poco papeleo por lo que se hace más cómoda su utilización. Actualmente XP es el método ágil más documentado (hay una colección de libros con XP Series de Addison Wesley) y extendido. Existe una gran comunidad de desarrolladores XP. Otra de las ventajas de XP es que no es necesario adoptarlo en forma completa, sino que pueden utilizarse varias de sus prácticas en forma independiente. Esto hace que el costo de su implementación sea mucho más accesible que el de otras metodologías.

Ventajas de XP

- Apropiado para entornos volátiles.
- Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas) o en forma gradual.
- Puede adaptarse a las necesidades de cualquier equipo de desarrollo.

- Estar preparados para el cambio, significa reducir su coste.
- Permite definir en cada iteración cuáles son los objetivos de la siguiente.
- Exige que se establezca una comunicación más fluida con el cliente y que este tenga mayor participación en el proceso de desarrollo. La consecuencia de esto es que el cliente se involucre más en el desarrollo del producto.
- Permite la retroalimentación.
- Actualmente es la metodología ágil más extendida y documentada. Se realizan pruebas constantemente del sistema.

1.5 Patrón Arquitectónico MVC

Para el diseño de aplicaciones con sofisticadas interfaces se emplea este patrón: **Modelo-Vista-Controlador**. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si se realiza un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces, la consecuencia será que, cuando se necesite cambiar la interfaz, tendrá que modificarse trabajosamente los componentes de negocio, por lo que propiciará mayor trabajo y más riesgo de error. Se trata de realizar un diseño que desacople la vista del modelo, con el fin de perfeccionar la reusabilidad. (16)

Modelo: En estas clases está todo el código que tiene que ver con el acceso a base de datos. En las modelos se mantiene encapsulada la complejidad la base de datos y es aquí donde se crean funciones para recibir, insertar, actualizar o borrar información de las tablas. Al mantenerse todas las llamadas a la base de datos en un mismo código, desde otras partes del programa se invocan a las funciones que se necesiten del modelo y éste se encargará de procesarlas. Las modelos se encargan de cosas como el tipo de base de datos con las que se trabaja, o las tablas y sus relaciones, pero desde las otras partes del programa simplemente se llaman a las funciones de la modelo sin analizar qué tiene esta clase, o qué hace para conseguir realizar las acciones invocadas.

Vista: La vista codifica y mantiene la presentación final de una aplicación de cara al usuario. Es decir, en ella se coloca todo el código HTML, CSS, Javascript y librerías como JQuery etc. que se tiene que generar para producir la página tal cual se desea que la vea el usuario. En la práctica la vista no sólo sirve para producir páginas web, sino también para cualquier otra salida que se quiera enviar al usuario, en formatos o lenguajes distintos, como pueden ser archivos JSON, XML, etc.

Controlador: El controlador se puede decir que es la parte más importante, porque hace de enlace entre el modelo, la vista y cualquier otro recurso que se tenga que procesar en el servidor para generar la página web. En el controlador se guarda la lógica de las páginas y se realizan todas las acciones que sean necesarias para generarlas, ayudados del modelo o la vista. Con el objetivo de lograr una mejor integridad de los datos, la base de datos está dividida por esquemas, cada uno de estos representa a un módulo, solo se puede acceder directamente a las tablas del esquema al que se pertenezca. La persona que necesite algún dato que se encuentre en la tabla de otro modulo, debe obtenerlo mediante el IOC acrónimo que significa “Inverion of Control (Inversión de Control)”. El IOC es un archivo XML en el cual cada módulo publica métodos que devuelven datos solicitados por otros y su vez accede a los métodos que necesita.

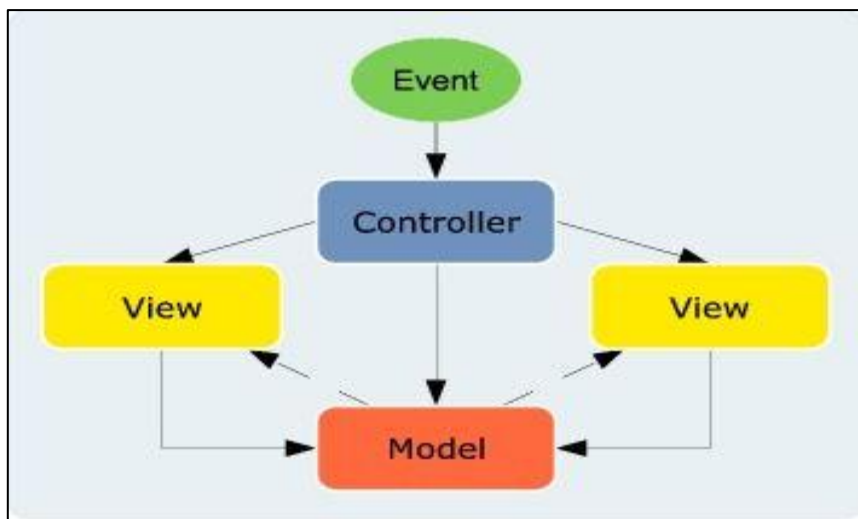


Figura 1.3 Patrón Arquitectura Modelo Vista Controlador

1.6 Conclusiones

En este capítulo se abordaron elementos necesarios para la comprensión y fundamentación de la solución propuesta. Las tendencias y tecnologías actuales relacionadas con el tema, se hizo una valoración del lenguaje de programación, el sistema gestor de bases de datos que se pretenden utilizar en la futura implementación, y la metodología de desarrollo que se utilizó. Una vez conocidas las herramientas y conceptos a usar se puede proseguir con la propuesta de solución.

Capítulo 2 – Planeación y Diseño

2.1 Introducción

En el presente capítulo se tratan los temas relacionados con el análisis y construcción de la aplicación propuesta. Para ello se expone la propuesta de solución, describiéndose el proceso del negocio, los requisitos funcionales y no funcionales con los que contará el sistema y, se da a conocer las historias de usuarios.

2.2 Problema y situación problemática

A los especialistas y profesores de los cursos y entrenamientos de posgrados se les dificulta registrar todas las actividades y datos que se generan con la realización de los distintos cursos y entrenamientos de posgrado. Debido a que es muy diversa y cuantiosa la información generada, profesores y especialistas, necesitan un sistema que les ayude a realizar su trabajo de una manera fácil, con mayor rapidez y organización y a la vez les proporcione la opción de obtener de forma digital todos los documentos que se obtienen, ya que los mismos en copia dura se van deteriorando o extraviando, también porque se les hace muy difícil realizar reportes o actualizaciones de datos, actividades que se realizan de manera muy sencilla con la ayuda de un sistema informático, esto hace que se determine la gestión de información de los cursos y entrenamientos de posgrados como una situación problemática que debe resolverse en el ISMMM.

2.3 Propuesta de solución

Teniendo en cuenta la situación existente con respecto a los cursos y entrenamientos de posgrados convocados en el centro, se determinó como propuesta de solución, la implementación de una Aplicación web para la Gestión de Información de los cursos y entrenamientos de posgrados realizados en el ISMMM.

Se pretende que la aplicación incluya: el registro y gestión de los datos de los cursos y entrenamientos de posgrados y claustro de profesores, permitiendo que se realicen búsquedas rápidas, corrección de los datos en caso de que al introducirlos se cometa algún error, debe brindar la opción de ver resúmenes de toda la información de los estudiantes que se seleccione, igualmente que permita saber todos los integrantes que conforman un determinado claustro.

También debe garantizar que se adjunten los materiales bibliográficos de los cursos y entrenamientos de posgrados, documentos donde se recojan resúmenes e información de vital importancia sobre los estudiantes y estos deben ser conservados de forma digital. De la misma manera podrán ser informatizados los datos de las solicitudes, estas son realizadas por los interesados a cursar dichos cursos y entrenamientos de posgrados.

2.4 Descripción del proceso del negocio

El proceso comienza cuando se crea uno o varios cursos y entrenamientos de posgrados, esta actividad es realizada por el comité académico de la facultad encargada de los cursos y entrenamientos de posgrados. Posteriormente el comité define el claustro de profesores que los impartirán. Terminadas estas tareas se lanza una convocatoria de cada curso con sus respectivas descripciones.

Los aspirantes a cursar dichos cursos y entrenamientos de posgrados envían o entregan personalmente una solicitud de matrícula. Esta solicitud es aprobada o no por el comité académico. De no ser aprobada la solicitud, se guardan todos los datos de esta, dejándole claro al aspirante que no fue aprobado. De ser aprobada esta solicitud el estudiante deberá presentarse en el centro para matricular y recibir las asignaturas personalmente. Este estudiante matriculado también tendrá acceso a los materiales de los cursos y entrenamientos de posgrados cuyas respectivas bibliografías que serán gestionadas a través de la aplicación, la cual controlará además las evaluaciones que se le realicen al estudiante, culminando el proceso con una evaluación final.

2.5 Personal relacionado con el sistema

Tabla 2.1 Personal relacionado con el sistema

Personal relacionado	Justificación
Administrador del sistema	Responsable de administrar el sistema.
Profesor de un claustro	Es el encargado de poner las notas a los estudiantes y poner a disposición de estos los materiales de la asignatura.
Estudiante	Usuario que recibe el curso de maestría.
Solicitante	Persona que solicita recibir un curso de maestría.

2.6 Lista de reserva

La Lista de Reserva del Producto es el primer artefacto generado en la etapa de Captura de Requisitos. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto, debido a eso la metodología XP nos brinda la posibilidad de que esta lista pueda crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente, con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible, para esto la lista debe acompañar los cambios en el entorno y el producto. A continuación, se enumeran las funcionalidades que el sistema debe ser capaz de cumplir.

Tabla 2.2 Lista de reserva del producto

Número	Descripción de las historias de usuarios	Prioridad
RF1	Insertar posgrado	Alta
RF2	Modificar posgrado	Alta
RF3	Buscar posgrado	Alta

RF4	Mostrar posgrado	Alta
RF5	Eliminar posgrado	Alta
RF6	Insertar material	Media
RF7	Mostrar material	Media
RF8	Buscar material	Media
RF9	Eliminar material	Media
RF10	Autenticar usuario	Baja
RF11	Insertar usuario	Alta
RF12	Modificar datos usuario	Alta
RF13	Mostrar usuario	Alta
RF14	Buscar usuario	Alta
RF15	Eliminar usuario	Alta
RF16	Insertar profesor	Alta
RF17	Modificar profesor	Alta
RF18	Buscar profesor	Alta
RF19	Mostrar profesor	Alta
RF20	Eliminar profesor	Alta
RF21	Insertar estudiante	Alta
RF22	Modificar estudiante	Alta
RF23	Buscar estudiante	Alta
RF24	Mostrar estudiante	Alta
RF25	Eliminar estudiante	Alta
RF26	Insertar evaluación estudiante	Media
RF27	Mostrar evaluación estudiante	Media
RF28	Buscar evaluación estudiante	Media
RF29	Eliminar evaluación estudiante	Media

RF30	Insertar claustro	Baja
RF31	Eliminar claustro	Baja
RF32	Insertar miembro claustro	Baja
RF33	Mostrar miembro claustro	Baja
RF34	Modificar datos miembro claustro	Baja
RF35	Eliminar miembro claustro	Baja
RF36	Insertar solicitud	Media
RF37	Mostrar solicitud	Media
RF38	Buscar datos solicitud	Media
RF39	Eliminar solicitud	Media
RF40	Insertar especialidad miembro claustro	Baja
RF41	Mostrar especialidad miembro claustro	Baja
RF42	Buscar especialidad miembro claustro	Baja
RF43	Modificar especialidad miembro claustro	Baja
RF44	Eliminar especialidad miembro claustro	Baja
RNF (Requisitos no funcionales)		
Usabilidad		
RNF1	Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada, y debe ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.	
RNF2	Emplear perfiles de usuario: diferenciar las interfaces y opciones para los usuarios que accedan al sistema según los diferentes roles que estos tengan dentro del sistema.	
Fiabilidad		
RNF3	Seguridad de las bases de datos: la seguridad de la base de datos	

	está a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.
RNF4	Servicios web restringidos: los servicios web que brinde el sistema deben estar restringidos a grupos de usuarios definidos y aprobados previamente.
RNF5	Políticas de seguridad por usuario y rol: el sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado.
Eficiencia	
RNF6	Cantidad de conexiones: el sistema debe soportar una conexión simultánea de más de 30 usuarios.
Interfaz	
RNF7	Interfaz web: la interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.

2.7 Historias de usuario

La HU, es la técnica utilizada en XP para detallar los requisitos del software; es el resultado directo del intercambio entre el cliente y desarrolladores a través de reuniones donde las conocidas tormenta de ideas arrojan no solo los requerimientos, sino también las posibles soluciones; representa una forma rápida de administrar las necesidades del cliente 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. [Anexo 1 – Historias de usuario](#)

Tabla 2.3 Historia de usuario gestionar posgrado

Historia de usuario	
Número: 1	Nombre: Gestionar posgrado
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: Primera
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar, mostrar, buscar, modificar y eliminar los posgrados.	
Observaciones: Confirmado por el cliente.	

Tabla 2.4 Historia de usuario gestionar material

Historia de usuario	
Número: 2	Nombre: Gestionar material
Prioridad en el negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Segunda
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar, mostrar, buscar y eliminar los materiales bibliográficos.	
Observaciones: Confirmado por el cliente.	

2.8 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.

2.8.1 Estimación de esfuerzo por Historia de Usuario

Tabla 2.13 Estimación de esfuerzo por Historia de Usuario

Historias de Usuario	Puntos de Estimación
Gestionar posgrado	2
Gestionar Material	1
Gestionar Usuario	2
Gestionar Profesor	2
Gestionar Estudiante	2
Gestionar Evaluación Estudiante	1
Gestionar Claustro	1
Gestionar Solicitud	1
Gestionar Especialidad Miembro Claustro	1
Autenticar Usuario	1

2.3.1 Plan de iteraciones

Tabla 2.14 Plan de duración de las iteraciones

Iteración	Descripción de la iteración	Orden de la HU a implementar	Duración de cada HU(días)	Duración total (días)
Primera	En esta iteración se	Insertar posgrado	2	
		Modificar posgrado	1	
		Buscar posgrado	1	

	van a implementar las HU que sean de prioridad “Alta” para el proyecto	Mostrar posgrado	1	21 días
		Eliminar posgrado	2	
		Insertar usuario	2	
		Modificar datos usuario	1	
		Mostrar usuario	1	
		Buscar usuario	1	
		Eliminar usuario	2	
		Insertar profesor	2	
		Modificar profesor	1	
		Buscar profesor	1	
		Mostrar profesor	1	
		Eliminar profesor	2	
Segunda	En esta iteración se van a implementar las HU que tengan prioridad “Media”, para el proyecto	Insertar material	2	18 días
		Mostrar material	1	
		Buscar material	1	
		Eliminar material	2	
		Insertar evaluación estudiante	2	
		Mostrar evaluación estudiante	1	
		Buscar evaluación estudiante	1	
		Eliminar evaluación estudiante	2	
		Insertar solicitud	2	
		Mostrar solicitud	1	
		Buscar datos solicitud	1	
		Eliminar solicitud	2	
Tercera	En esta iteración se van a implementar	Autenticar usuario	2	
		Insertar estudiante	2	
		Modificar estudiante	1	
		Buscar estudiante	1	
		Mostrar estudiante	1	

	las HU, que tengan prioridad “Baja” para el proyecto.	Eliminar estudiante	1	21 días
		Insertar especialidad miembro claustro	1	
		Mostrar especialidad miembro claustro	1	
		Buscar especialidad miembro claustro	1	
		Modificar especialidad miembro claustro	1	
		Eliminar especialidad miembro claustro	1	
		Insertar claustro	2	
		Eliminar claustro	1	
		Insertar miembro claustro	2	
		Mostrar miembro claustro	1	
		Modificar datos miembro	1	
		Eliminar miembro claustro	1	
Total			60 días	9 semanas

2.4 Tarjetas CRC

El uso de las tarjetas C.R.C (Clases, Responsabilidades y Colaboración) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación clásica. Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad. Esta nueva técnica de diseño es adoptada como alternativa a los diagramas UML de las clases, pues en estas se plasman las responsabilidades que tienen cada objeto y las clases con las que tienen que interactuar para darles respuesta, brindando así la información que se necesita a la hora de implementar. [Anexo 2 – Tarjetas CRC](#)

Tabla 2.15 Tarjeta CRC de Posgrado

Posgrado	
Descripción:	
Atributos:	
Nombre	Descripción
id_posgrado	identificador del posgrado
id_modalidad	
facultad	
tipo_posgrado	
departamento	
titulo	
creditos	
cant_hrs_pres	
cant_hrs_no_pres	
fecha_inicio	
fecha_fin	
dirigido_a	
profesor_principal	
Responsabilidades:	
Nombre	Collaborator
insertar_posgrado	modalidad
buscar_posgrado	modalidad
modificar_posgrado	modalidad
mostrar_posgrado	modalidad
eliminar_posgrado	modalidad

Tabla 2.16 Tarjeta CRC de Profesor

Profesor	
Descripción: Guarda información del profesor	
Attributes:	
Nombre	Descripción
identidad	Identificador del profesor
id_grado	
id_categ	
id_especialidad	
Responsabilidades:	
Nombre	Collaborator
insertar_profesor	Persona
mostrar_profesor	
modificar_profesor	
eliminar_profesor	
buscar_profesor	

Tabla 2.17 Tarjeta CRC de Estudiante

Estudiante	
Descripción: Guarda los datos del estudiante	
Attributes:	
Nombre	Descripción
identidad	Identificador de estudiante
id_posgrado	
Responsabilidades:	
Nombre	Collaborator
crear_estudiante	Persona, Posgrado
mostrar_estudiante	Persona, Posgrado
modificar_estudiante	Persona, Posgrado
eliminar_estudiante	Persona, Posgrado
buscar_estudiante	

2.5 Conclusiones

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 – Desarrollo y Pruebas

3.1 Introducción

En este capítulo abordaremos lo relacionado con la fase de implementación y pruebas siguiendo la Metodología XP. Se describen cada una de las tareas de ingeniería para cumplir con el desarrollo de cada una de las HU definidas, se presentan las tarjetas clases, responsabilidades y colaboradores, y se mostrarán las pruebas de aceptación elaboradas por el cliente para comprobar que la aplicación funcione correctamente.

3.2 Diseño de la base de datos

Una **base de datos** o **banco de datos** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Desde el punto de vista informático, es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. Cada base de datos se compone de una o más tablas que a la vez están compuestas por filas y columnas, las primeras conforman los objetos almacenados y las columnas guardan parte de la información de cada objeto almacenado en la tabla.

3.3.1 Tareas por historias de usuario

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o 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 3.1 Distribución de tareas por historia de usuario

Historia de Usuario	Tarea
Gestionar Posgrado	<ul style="list-style-type: none">• Insertar posgrado.• Mostrar posgrado.• Buscar posgrado.• Modificar posgrado.• Eliminar posgrado.
Gestionar Material	<ul style="list-style-type: none">• Insertar material.• Mostrar material.• Buscar material.• Modificar material.• Eliminar material.
Gestionar Estudiante	<ul style="list-style-type: none">• Insertar estudiante.• Mostrar estudiante.• Buscar estudiante.• Modificar estudiante.• Eliminar estudiante.
Gestionar Profesor	<ul style="list-style-type: none">• Insertar profesor.• Mostrar profesor.• Buscar profesor.• Modificar profesor.

	<ul style="list-style-type: none"> • Eliminar profesor.
Gestionar Especialidad Miembro Claustro	<ul style="list-style-type: none"> • Insertar especialidad. • Mostrar especialidad. • Buscar especialidad. • Modificar especialidad. • Eliminar especialidad.
Gestionar Evaluación	<ul style="list-style-type: none"> • Insertar evaluación. • Mostrar evaluación. • Modificar evaluación. • Eliminar evaluación.
Gestionar Claustro	<ul style="list-style-type: none"> • Insertar claustro. • Eliminar claustro.
Gestionar Usuario	<ul style="list-style-type: none"> • Insertar usuario. • Mostrar usuario. • Buscar usuario. • Modificar usuario. • Eliminar usuario.
Gestionar Solicitud	<ul style="list-style-type: none"> • Insertar solicitud. • Mostrar solicitud. • Buscar solicitud. • Eliminar solicitud.
Autenticar Usuario	<ul style="list-style-type: none"> • Introducir los datos (nombre de usuario y contraseña) para realizar operaciones con privilegios.

3.3.2 Tareas de ingeniería

En XP generalmente cada historia se divide en tareas de ingeniería o tareas de programación. 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 crean para

obtener una mejor planificación y cumplir con las funcionalidades básicas que luego conformaran las funcionalidades generales de cada historia de usuario.

[Anexo 3 – Tareas de ingeniería](#)

Tabla 3.2 Tarea de programación Insertar posgrado

Tarea de Programación	
No. Tarea: 1	No. Historia: 1
Nombre de la tarea: Insertar posgrado.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 03/03/2014	Fecha fin: 04/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar los posgrados en el sistema.	

3.4 Pruebas

En la metodología XP las pruebas juegan un papel importante, pues estas permiten la comprobación continua del código. El desarrollo constante de las pruebas da lugar a que se desarrolle un software con mayor calidad dando una mayor seguridad de lo que se está haciendo, dividiendo las pruebas en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y las pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñada por el cliente.

3.4.1 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.

3.4.2 Pruebas de aceptación

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.

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

Tabla 3.3 Prueba de aceptación para la HU: Gestionar posgrado

Prueba de Aceptación
HU: Gestionar Curso.
Nombre: Prueba para comprobar la gestión de los posgrados.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de los posgrados.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder a la aplicación y poder insertar, mostrar, buscar, modificar y eliminar los posgrados.
Entrada / Pasos ejecución: El administrador escribe los datos que se piden del posgrado y pulsa el botón Insertar. Después que el posgrado esté insertado se podrá mostrar, buscar, modificar o eliminar de un listado.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Se insertan correctamente los cursos. • Se muestran los cursos correctamente. • Se pueden modificar o eliminar una vez insertados. <p>Se produce un error en caso de que:</p> <ul style="list-style-type: none"> • Se dejen campos obligatorios vacíos. • El posgrado ya está registrado en la base de datos. • Se inserten letras en campos que deben ser numéricos.
Evaluación de la prueba: Aceptada.

Para observar las demás Pruebas de Aceptación ir al [Anexo 4 – Pruebas de Aceptación](#).

3.5 Conclusiones

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 tareas de ingenierías.

Capítulo 4 – Estudio de Factibilidad

4.1 Introducción

En la actualidad es muy importante realizar un estudio de factibilidad antes de empezar a desarrollar un proyecto. En general, ningún producto informático está exento de riesgos durante 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.

4.2 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:** involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados.
- **La efectividad:** 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.

4.3 Efectos económicos

- Efectos directos.
- Efectos indirectos.
- Efectos internos.
- Intangibles.

4.3.1 Efectos directos

Positivos:

- Ahorro de tiempo en la búsqueda de información de un profesor o estudiante.
- Disminución de la acumulación de materiales impresos relacionados con los cursos y entrenamientos de posgrado.
- Los usuarios pueden revisar sus datos en todo momento.
- Seguridad al guardar la información.
- Se cuenta con una herramienta capaz de mantener la seguridad e integridad de los datos que se procesan.

Negativos:

- Para el uso de esta aplicación implementada en plataforma Web se necesitará que la misma sea ejecutada con un navegador compatible, recomendamos para mejor visibilidad el Mozilla Firefox en su versión más actual, porque es con el que se trabajó en la elaboración del producto por lo que el diseño está adaptado a este tipo de navegador.

4.3.2 Efectos indirectos

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

4.3.3 Efectos externos

- Se obtendrá un producto disponible que le facilitará el trabajo a los usuarios encargados de la gestión de la información relacionada a los cursos y entrenamiento de posgrados en el ISMMM.

4.3.4 Intangibles

En la estimación económica siempre hay elementos como perjuicio o beneficio, pero al momento de darle valor en unidades monetarias esto resulta difícil o prácticamente imposible. A fin de medir con precisión los efectos, deberán considerarse dos situaciones: la situación sin proyecto y la situación con proyecto.

4.4 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.

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 (Solución manual):

En el VRIP, se maneja un gran volumen de información referente a los cursos y entrenamientos de posgrado, lo cual provoca que esta tarea se realice de una forma más lenta debido a que las consultas realizadas sobre los datos se deben realizar de forma manual por una o varias personas. Existen otros problemas que hacen que esta labor no se realice de la manera más adecuada, como la existencia de la mayoría de la documentación en formato duro, por lo que se corre el riesgo de que se deteriore. Otro problema es el relacionado con el acceso a la información por parte de los usuarios interesados en participar en los cursos de

posgrados o de entrenamientos debido a que la información no se encuentra centralizada y se desperdicia mucho tiempo buscándola.

Situación con proyecto (Solución automatizada)

Mediante este sistema es posible gestionar y controlar todo el flujo de información concerniente a los cursos y entrenamientos de posgrado. Queda garantizada la centralización de los datos teniendo además una accesibilidad fácil y rápida de todas las partes involucradas en el proceso.

Costos:

- Resistencia al cambio.

Beneficios:

- Mayor comodidad y organización de información para los usuarios.
- Conectividad desde cualquier ordenador que esté enlazado a la red del centro.
- Mayor rapidez a la hora de realizar el proceso.
- Mejora en la accesibilidad y visibilidad de la información.

4.5 Ficha de costo de un producto informático

Para determinar el costo económico del proyecto se utilizará el Procedimiento para elaborar una ficha de costo de un producto informático. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional. (16)

Costos en Moneda Librementemente Convertible:

- **Costos Directos**
 1. Compra de equipos de cómputo: No procede.
 2. Alquiler de equipos de cómputo: No procede.
 3. Compra de licencia de Software: No procede.
 4. Depreciación de equipos: \$ 64.94 mensual
 5. Materiales directos: No procede.

Total: \$ 64.94 CUC

- **Costos indirectos**

1. Formación del personal que elabora el proyecto: No procede.
2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. Know How: No procede.
5. Gastos en representación: No procede.

Total: \$0.00.

- **Gastos de distribución y venta.**

1. Participación en ferias o exposiciones: No procede.
2. Gastos en transportación: No procede.
3. Compra de materiales de propagandas: No procede.

Total: \$0.00.

Total General: \$ 64.93 CUC.

Costos en Moneda Nacional:

- **Costos Directos.**

1. Salario del personal que laborará en el proyecto: \$100.00 mensual.
2. El 12% del total de gastos por salarios se dedica a la seguridad social: No procede.
3. El 0.09% del salario total, por concepto de vacaciones a acumular: No procede.
4. Gasto por consumo de energía eléctrica: 68.64 kW mensual (1 kW = \$ 0.67) $68.64 * 0.67 = \$ 45.99$.
5. Gastos en llamadas telefónicas: No procede.
6. Gastos administrativos: No procede.

- **Costos indirectos**

1. Know How: \$ 0.0.

Total: \$ 145.99.

Total General: \$ 145.99

El análisis de costo-beneficio se basa en un principio muy simple:

Compara los beneficios y los costos de un proyecto particular y si los primeros exceden a los segundos entrega un elemento de juicio inicial, que indica su aceptabilidad. Mientras que el análisis costo-efectividad sigue la misma lógica, compara los costos con las potencialidades de alcanzar más eficientemente los objetivos no expresables en moneda; si no en productos. Para esta técnica es imprescindible definir una variable directa que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado por los especialistas para realizar las actividades de gestión de la información y la variable sería, la complejidad de las pruebas que se desarrollan durante este proceso.

Valores de la variable (Solución manual)

1. Gestionar la información de los cursos y entrenamientos de posgrado (30 min).
2. Gestionar la información de los profesores (20 min).
3. Gestionar la información de los estudiantes (25 min).
4. Gestionar las evaluaciones de los estudiantes (20 min).

Valores de la variable (Solución con el sistema)

1. Gestionar la información de los cursos y entrenamientos de posgrado (5 min).
2. Gestionar la información de los profesores (2 min).
3. Gestionar la información de los estudiantes (3 min).
4. Gestionar las evaluaciones de los estudiantes (2 min).

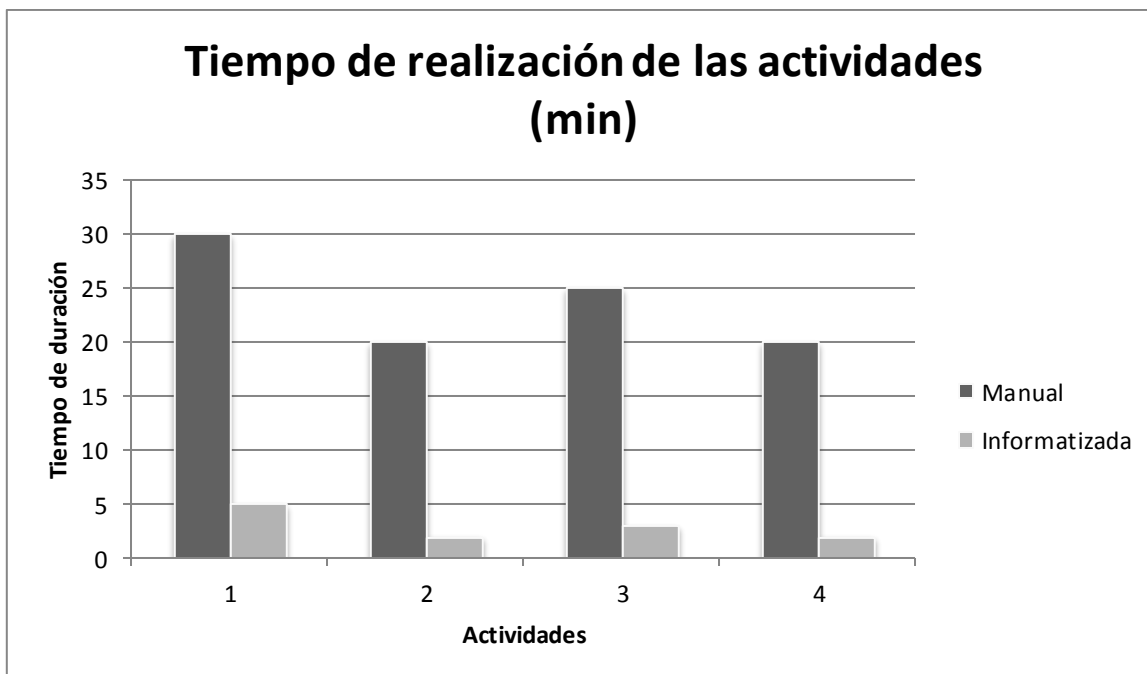


FIGURA 4.1 GRÁFICA DE LA SOLUCIÓN SIN EL PRODUCTO Y SOLUCIÓN CON EL PRODUCTO.

Teniendo en cuenta los resultados reflejados en la gráfica queda demostrada la factibilidad del sistema evidenciado por la relación entre la complejidad del problema (cantidad de variables) y el tiempo que demora la solución del mismo de forma manual y automatizada.

4.6 Conclusiones

En este capítulo se realizó el estudio de factibilidad del proyecto mediante la Metodología Costo Efectividad (Beneficios). Se analizaron todos los factores directos, indirectos, externos e intangibles., se analizaron los efectos económicos, los beneficios y costos, además se calculó el costo de ejecución del proyecto mediante la Ficha de Costo, la cual arrojó como resultado \$ 64.94 CUC y \$ 145.99 MN, además analizando el tiempo empleado para realizar las operaciones sin y con el proyecto se demostró así la factibilidad del producto.

Conclusiones Generales

Con el desarrollo de la aplicación web para la gestión de la información del VRIP del ISMMM se dio cumplimiento a los objetivos trazados en esta investigación, pues se obtuvo como resultado una herramienta informática en la que se aplican los resultados de la investigación realizada arribándose a las siguientes conclusiones:

- Se elaboró el marco teórico metodológico que fundamenta la investigación, esto permitió que quedara identificada la situación problemática existente y las bases para comenzar con el diseño e implementación de la aplicación.
- Se realizó un levantamiento sobre los procesos que se llevan a cabo en el VRIP del ISMMM.
- Durante el desarrollo de la metodología XP se generaron 44 requisitos funcionales, 10 historias de usuario, 16 tarjetas CRC, 10 pruebas de aceptación y 40 tareas de ingeniería.
- Se efectuó un estudio de la factibilidad técnica, económica, y operativa del software en cuestión; arrojando resultados satisfactorios para el proyecto.

Por lo antes expuesto se concluye que los objetivos propuestos fueron cumplidos satisfactoriamente.

Recomendaciones

Como recomendación de la actual investigación se considera ofrecer las siguientes recomendaciones:

- Realizar la implantación del software en el ISMMM.
- Ejecutar la aplicación en el navegador Mozilla Firefox ya que traduce los estilos de diseño con mayor claridad.
- Realizar un estudio más profundo de este sistema en vista a perfeccionarlo en versiones futuras.

Con la puesta en práctica de las recomendaciones se obtendrá una aplicación con mayores funcionalidades. Esto permitirá al especialista del VRIP contar con una herramienta informática que agilice la gestión de información sobre los cursos y entrenamientos de posgrado en el ISMMM.

Referencias Bibliográficas

1. Sitio Web de MOODLE. [En línea] [Citado el: 13 de Febrero de 2014.] <http://docs.moodle.org/>.
2. Lenguaje HTML. [En línea] [Citado el: 13 de Febrero de 2014.] http://www.aulaclie.es/html/t_1_1.htm.
3. ¿Qué es JavaScript? [En línea] [Citado el: 13 de Febrero de 2014.] <http://www.pablin.com.ar/computer/cursos/cursojs/js1.htm>.
4. **Hernán Ruiz, Marcelo.** *Programación Web Avanzada*. La Habana : Editorial Félix Varela, 2006.
5. Servidor httpd (Apache). [En línea] [Citado el: 19 de Febrero de 2014.] <http://www.profesionalhosting.com/servidores-dedicados/definicion/servidor-httpdapache-70.html>.
6. Características Principales. [En línea] [Citado el: 14 de Febrero de 2014.] <http://ocw.uniovi.es/mod/resource/view.php?id=1242>.
7. **Naula, Jorge.** ¿Qué es un framework de desarrollo. [En línea] [Citado el: 25 de Febrero de 2014.] <http://kokiing.blogspot.com/2010/12/que-es-un-framework-de-desarrollo-web.html>.
8. ¿Qué es Ext JS? [En línea] [Citado el: 26 de Febrero de 2014.] <http://www.extjs.mx/2011/11/27/post-con-cursos-generales/>.
9. Ext JS lo bueno, lo malo y lo feo | Desarrollo en Web. [En línea] [Citado el: 26 de Febrero de 2014.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
10. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] [Citado el: 27 de Febrero de 2014.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd>.
11. Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. [En línea] [Citado el: 18 de Febrero de 2014.] <http://www.eaprende.com/gestor-de-basededatos-mysql-postgresql-sqlite.html>.
12. Embarcadero ER/Studio, Software – Herramientas de Desarrollo de Software. [En línea] [Citado el: 4 de Marzo de 2014.]

<http://www.guiadesolucionestic.com/software-del-sistema/herramientas-de-desarrollo/herramientas-de-desarrollo-de-software/972-embarcadero-erstudio>.

13. Visual Paradigm for UML. [En línea] [Citado el: 5 de Marzo de 2014.] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML

.

14. **Bañeres, Juan Palacio.** Gestión ágil de proyectos: Scrum. *Navegapolis.net*. [En línea] [Citado el: 6 de Marzo de 2014.] <http://www.navegapolis.net>.

15. XP - Extreme Programing Ingeniería de Software. [En línea] [Citado el: 17 de Marzo de 2014.] http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html.

16. Definición básica del MVC. [En línea] [Citado el: 23 de Mayo de 2014.] <http://www.lab.inf.uc3m.es/%7Ea0080802/RAI/mvc.html>.

17. Costo - Efectividad. [En línea] [Citado el: 28 de Mayo de 2014.] www.crid.or.cr/digitalizacion/pdf/spa/doc9321-02.pdf.

Bibliografías

1. BECK, 1999]. Kent Beck. *Extreme Programming Explained: Embrace Change*, Pearson Education 1999. Addison-Wesley, 2000.
2. [BEIGHLEY; MORRISON, 2005]. Beighley, Lynn, Morrison, Michael. *Head First PHP & MySQL*. O'Reilly, 2005.
3. [CANOS; LETELIER; PENADES, 2005]. H. Canós, José, Letelier, Patricio, Penadés, M. Carmen. *Metodologías Ágiles en el desarrollo de software*. Universidad Politécnica de Valencia, 2005.
4. Calvo Burke, Aurelio. *Sistema de Gestión de Información sobre Derrames de Hidrocarburos en el puerto de Moa*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, julio 2013.
5. [FERNANDEZ, 2002]. Gerardo Fernández Escribano. *Introducción a Extreme Programming*. Ingeniería de Software II. 9-12-2002.
6. [HERNANDEZ, 2006]. Edelsys Hernández Meléndrez. *Como escribir una tesis*. Escuela Nacional de Salud Pública, 2006.
7. [LETELIER; PENADES, 2005]. Letelier, Patricio; Penadés, M^a Carmen. *Metodologías ágiles para el desarrollo de software: Extreme Programming (XP)*. Universidad Politécnica de Valencia, 2005.
8. [MARTEENS, 1999]. Ian Marteens. *La cara oculta de C++ Builder*. Madrid, 1999.
9. Peñaloza B, Leisewitz T, Bastías G, Zárate V, Depaux R, Villarroel L, et al. Metodología para la evaluación de la relación costo-efectividad en centros de atención primaria de Chile. *Revista Panam. Salud Pública*. 2010.
10. Rodríguez Pérez, Agustín. *SIGEP: Sistema para La Gestión del Presupuesto en el ISMMM*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, junio 2012.

Glosario de Términos

Análisis costo beneficio: El coste-beneficio es una lógica o razonamiento basado en el principio de obtener los mayores y mejores resultados al menor esfuerzo invertido, tanto por eficiencia técnica como por motivación humana. Se supone que todos los hechos y actos pueden evaluarse bajo esta lógica, aquellos dónde los beneficios superan el coste son exitosos, caso contrario fracasan.

Aplicación: Programa preparado para una utilización específica. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

Aplicación web: Una aplicación Web es un sistema informático que los usuarios usan accediendo a un servidor Web a través de los protocolos de Internet. Las aplicaciones Web son populares por su practicidad del navegador Web como cliente ligero.

API: Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

ASP: Active Server Pages, también conocido como ASP clásico, es una tecnología de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS).

Ajax: acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Beneficio: Ganancia económica que se obtiene de un negocio, inversión u otra actividad mercantil.

Bugs: Errores que se presentan en la aplicación que atentan contra su correcto funcionamiento.

Cliente: Persona, organización o grupo de personas que solicita la construcción de un sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas.

Costo: En economía el coste o costo es el valor monetario de los consumos de factores que supone el ejercicio de una actividad económica destinada a la producción de un bien o servicio.

Compatibilidad: Condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente o indirectamente mediante un algoritmo.

Concurrencia: Coincidencia, concurso simultáneo de varias circunstancias. Es la simultaneidad en la ejecución de múltiples tareas interactivas. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa.

Gestión de información: es un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención - por creación o captura, hasta su disposición final - archivada o eliminada. Los procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados.

Herramientas: Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

Historia de usuario: es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario.

HTML: Hyper Text Markup Language, o simplemente HTML, es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web. Si se hace la traducción de su nombre del inglés al castellano, sería “Lenguaje de Marca de Hipertextos”, ya que es justamente un lenguaje que se basa en las marcas para crear los hipertextos.

IDE: Integrated Development Environment / Entorno de Desarrollo Integrado. Entorno de programación que ha sido empaquetado como un programa de

aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

Informatización: Aplicar los conjuntos de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.

Interfaz: Conjunto de representaciones de operaciones públicas.

Iteraciones: En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio.

Múltiples iteraciones contribuyen a crear un producto completamente integrado.

JavaScript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Java: Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador.

Lógica del negocio: Parte de un sistema que se encarga de las tareas relacionadas con los procesos de un negocio, tales como ventas, control de inventario, contabilidad, etc.

Multiplataforma: Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas o sistemas operativos.

Metodología Ágil: Constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a

equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

Metodología de desarrollo: Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

Metodologías tradicionales: Metodologías basadas en procesos.

Navegador web: Un navegador o navegador web (del inglés, web browser) es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que podamos ser capaces de leerla, (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local).

PHP: PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

Posgrado: Se llama estudios de posgrado o postgrado a los estudios de especialización posteriores al título de grado; es decir que es un nivel educativo que forma parte del tipo superior o de tercer ciclo. Es la última fase de la educación formal, tiene como antecedente obligatorio la titulación de pregrado y comprende los estudios de especialización, maestría o magíster, doctorado e Investigación posdoctoral.

Pregrado: Los estudios de pregrado son los estudios superiores hasta el título de Grado. Son necesarios para poder acceder a los estudios de postgrado.

Proceso: Conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial. *INF.* Programa en ejecución.

Proyecto: Esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida.

Programación extrema: Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Requisitos: Son las funciones, servicios y restricciones operativas del sistema.

Release: Versión candidata definitiva de un producto de software y se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan.

Script: En informática un guión, archivo de órdenes o archivo de procesamiento por lotes, es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guión.

Sistema: Delimita el mundo sobre el cual se está construyendo el modelo

SQLite: es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kB) biblioteca en C. SQLite es un proyecto de dominio público

Servidor Web: Un programa que corre sobre el servidor que escucha las peticiones HTTP (Hyper Text Markup Language) que le llegan y las satisface. Dependiendo del tipo de 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

Software: Se refiere a los programas y datos almacenados en un ordenador. Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

Testing: Proceso de pruebas usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Usuario: Persona que interactúa con el sistema.

Anexo 1 – Historias de usuarios

Tabla 2.5 Historia de usuario gestionar usuario

Historia de usuario	
Número: 3	Nombre: Gestionar usuario
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: Primera
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar, mostrar, buscar, modificar y eliminar los usuarios.	
Observaciones: Confirmado por el cliente.	

Tabla 2.6 Historia de usuario gestionar profesor

Historia de usuario	
Número: 4	Nombre: Gestionar profesor
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: Primera
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar, mostrar, buscar, modificar y eliminar los profesores.	
Observaciones: Confirmado por el cliente.	

Tabla 2.7 Historia de usuario gestionar estudiante

Historia de usuario	
Número: 5	Nombre: Gestionar estudiante
Prioridad en el negocio: Baja	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Tercera
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar, mostrar, buscar, modificar y eliminar los	

estudiantes.
Observaciones: Confirmado por el cliente.

Tabla 2.8 Historia de usuario gestionar evaluación estudiante

Historia de usuario	
Número: 6	Nombre: Gestionar evaluación estudiante
Prioridad en el negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: Segunda
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar, mostrar, buscar y eliminar las evaluaciones de los estudiantes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.9 Historia de usuario gestionar claustro

Historia de usuario	
Número: 7	Nombre: Gestionar claustro
Prioridad en el negocio: Baja	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Tercera
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar y eliminar los claustros.	
Observaciones: Confirmado por el cliente.	

Tabla 2.10 Historia de usuario gestionar solicitud

Historia de usuario	
Número: 8	Nombre: Gestionar solicitud
Prioridad en el negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Segunda
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar, mostrar, buscar y eliminar las solicitudes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.11 Historia de usuario gestionar especialidad miembro claustro

Historia de usuario	
Número: 9	Nombre: Gestionar especialidad miembro claustro
Prioridad en el negocio: Baja	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Tercera
Programador asignado: Nestor Antonio López Arcia	
Descripción: La historia de usuario permite insertar, mostrar, modificar y eliminar las especialidades de los miembros del claustro.	
Observaciones: Confirmado por el cliente.	

Tabla 2.12 Historia de usuario autenticar usuario

Historia de usuario	
Número: 10	Nombre: Autenticar usuario
Prioridad en el negocio: Baja	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Tercera
Programador asignado: Nestor Antonio López Arcia	
Descripción: Los usuarios del sistema ingresan sus datos para entrar al sistema (nombre de usuario y contraseña). El sistema verifica que los datos estén correctos, en caso de que no sean correctos la aplicación muestra un mensaje de error.	
Observaciones: Confirmado por el cliente.	

Anexo 2 – Tarjetas CRC

Tabla 2.18 Tarjeta CRC de Estudiante

Material	
Descripción: Guarda información del material	
Attributes:	
Nombre	Descripción
id_material	Identificador del material
descripción	
nombre_material	
Responsabilidades:	
Nombre	Collaborator
insertar_material	
mostrar_material	
eliminar_material	

Tabla 2.19 Tarjeta CRC de Usuario

Usuario	
Descripción: Guarda información del usuario	
Attributes:	
Nombre	Descripción
identidad	Identificador del usuario
nombre_usuario	
password	
Responsabilidades:	
Nombre	Collaborator
insertar_usuario	Persona
mostrar_usuario	Persona
modificar_usuario	Persona
eliminar_usuario	Persona
buscar_usuario	Persona

Tabla 2.20 Tarjeta CRC de Persona

Persona	
Descripción: Guarda información de la persona	
Attributes:	
Nombre	Descripción
identificador	Identificador de persona
nombre	
apellidos	
edad	
sexo	
correo	
Responsabilidades:	
Nombre	Collaborator
insertar_persona	
mostrar_persona	
modificar_persona	
eliminar_persona	

Tabla 2.21 Tarjeta CRC de Evaluación

Evaluación	
Descripción: Guardar los datos de evaluación	
Attributes:	
Nombre	Descripción
id_posgrado	
identidad	
nota	
Responsabilidades:	
Nombre	Collaborator
insertar_evaluacion	Estudiante, Posgrado
mostrar_evaluacion	Estudiante, Posgrado
modificar_evaluacion	Estudiante, Posgrado
eliminar_evaluacion	Estudiante, Posgrado

Tabla 2.22 Tarjeta CRC de Grado científico

Grado_Científico	
Descripción: Guarda los datos del grado científico	
Attributes:	
Nombre	Descripción
id_grado	Identificador del grado científico
nombre_grado	
Responsabilidades:	
Nombre	Collaborator
mostrar_grado_cient	

Tabla 2.23 Tarjeta CRC de Claustro

Claustro	
Descripción: Guarda los datos del claustro	
Attributes:	
Nombre	Descripción
id_posgrado	
id_identidad	
Responsabilidades:	
Nombre	Collaborator
insertar_claustro	Posgrado, Profesor
eliminar_claustro	Posgrado, Profesor

Tabla 2.24 Tarjeta CRC de Especialidad

Especialidad	
Descripción: Guarda información de la especialidad	
Attributes:	
Nombre	Descripción
id_especialidad	Identificador de la especialidad
nombre_especialidad	
Responsabilidades:	
Nombre	Collaborator
insertar_especialidad	
mostrar_especialidad	
modificar_especialidad	
eliminar_especialidad	
buscar_especialidad	

Tabla 2.25 Tarjeta CRC de Categoría docente

Categoría_Docente	
Descripción: Guarda los datos de la categoría docente	
Attributes:	
Nombre	Descripción
id_categ	
nombre_categ	
Responsabilidades:	
Nombre	Collaborator
mostrar_categ_doc	

Tabla 2.26 Tarjeta CRC de Solicitud

Solicitud	
Descripción: Guarda información de la solicitud	
Attributes:	
Nombre	Descripción
identidad	
id_posgrado	
planilla	
aprobado	
Responsabilidades:	
Nombre	Collaborator
insertar_solicitud	Persona
mostrar_solicitud	Persona
modificar_solicitud	Persona
eliminar_solicitud	Persona
buscar_solicitud	Persona

Tabla 2.27 Tarjeta CRC de Modalidad

Modalidad	
Descripción: Guarda los datos de modalidad	
Attributes:	
Nombre	Descripción
id_modalidad	Identificador de modalidad
nombre_modalidad	
Responsabilidades:	
Nombre	Collaborator
listar_modalidad	

Tabla 2.28 Tarjeta CRC de País

País	
Descripción: Guarda los datos del país	
Attributes:	
Nombre	Descripción
id_pais	Identificador del país
nombre_pais	
Responsabilidades:	
Nombre	Collaborator
mostrar_pais	

Tabla 2.29 Tarjeta CRC del Permiso

Permiso	
Descripción: Guarda información del permiso	
Attributes:	
Nombre	Descripción
identidad	
id_menu	
estado	
Responsabilidades:	
Nombre	Collaborator
insertar_permiso	Usuario, Menú
modificar_permiso	Usuario, Menú
modificar_permiso	Usuario, Menú

Tabla 2.30 Tarjeta CRC del Menú

Menú	
Descripción: Guarda los datos del menú	
Attributes:	
Nombre	Descripción
id_menu	Identificador del menú
descripcion	
url	
Responsabilidades:	
Nombre	Collaborator
mostrar_menu	

Anexo 3 – Tareas de ingeniería

Tabla 3.4 Tarea de programación Mostrar posgrado

Tarea de Programación	
No. Tarea: 2	No. Historia: 1
Nombre de la tarea: Mostrar posgrado.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 05/03/2014	Fecha fin: 05/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite mostrar los posgrados.	

Tabla 3.5 Tarea de programación Buscar posgrado

Tarea de Programación	
No. Tarea: 3	No. Historia: 1
Nombre de la tarea: Buscar posgrado.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 06/03/2014	Fecha fin: 06/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite buscar los posgrados.	

Tabla 3.6 Tarea de programación Modificar posgrado

Tarea de Programación	
No. Tarea: 4	No. Historia: 1
Nombre de la tarea: Modificar posgrado.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 07/03/2014	Fecha fin: 07/03/2014
Programador responsable: Nestor Antonio López Arcia	

Descripción: Permite modificar los posgrados.

Tabla 3.7 Tarea de programación Eliminar posgrado

Tarea de Programación	
No. Tarea: 5	No. Historia: 1
Nombre de la tarea: Eliminar posgrado.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 10/03/2014	Fecha fin: 11/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar los posgrados.	

Tabla 3.8 Tarea de programación Insertar material

Tarea de Programación	
No. Tarea: 6	No. Historia: 2
Nombre de la tarea: Insertar material.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 12/03/2014	Fecha fin: 13/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar los materiales bibliográficos.	

Tabla 3.9 Tarea de programación Mostrar material

Tarea de Programación	
No. Tarea: 7	No. Historia: 2
Nombre de la tarea: Mostrar material.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14/03/2014	Fecha fin: 14/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite mostrar los materiales bibliográficos.	

Tabla 3.10 Tarea de programación Buscar material

Tarea de Programación	
No. Tarea: 8	No. Historia: 2
Nombre de la tarea: Buscar material.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/03/2014	Fecha fin: 17/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite buscar los materiales bibliográficos.	

Tabla 3.11 Tarea de programación Eliminar material

Tarea de Programación	
No. Tarea: 9	No. Historia: 2
Nombre de la tarea: Eliminar material.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 18/03/2014	Fecha fin: 19/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar los materiales bibliográficos.	

Tabla 3.12 Tarea de programación Insertar estudiante

Tarea de Programación	
No. Tarea: 10	No. Historia: 3
Nombre de la tarea: Insertar estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 20/03/2014	Fecha fin: 21/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar estudiantes.	

Tabla 3.13 Tarea de programación Mostrar estudiante

Tarea de Programación	
No. Tarea: 11	No. Historia: 3
Nombre de la tarea: Mostrar estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24/03/2014	Fecha fin: 24/10/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite mostrar estudiantes.	

Tabla 3.14 Tarea de programación Buscar estudiante

Tarea de Programación	
No. Tarea: 12	No. Historia: 3
Nombre de la tarea: Buscar estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 25/03/2014	Fecha fin: 25/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite buscar estudiantes.	

Tabla 3.15 Tarea de programación Modificar estudiante

Tarea de Programación	
No. Tarea: 13	No. Historia: 3
Nombre de la tarea: Modificar estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/03/2014	Fecha fin: 26/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite modificar información de los estudiantes.	

Tabla 3.16 Tarea de programación Eliminar estudiante

Tarea de Programación	
No. Tarea: 14	No. Historia: 3
Nombre de la tarea: Eliminar estudiante.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 27/03/2014	Fecha fin: 27/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar estudiantes de la base de datos.	

Tabla 3.17 Tarea de programación Insertar profesor

Tarea de Programación	
No. Tarea: 15	No. Historia: 4
Nombre de la tarea: Insertar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 28/03/2014	Fecha fin: 31/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar profesores.	

Tabla 3.18 Tarea de programación Mostrar profesor

Tarea de Programación	
No. Tarea: 16	No. Historia: 4
Nombre de la tarea: Mostrar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 01/04/2014	Fecha fin: 01/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite mostrar a los profesores.	

Tabla 3.19 Tarea de programación Buscar profesor

Tarea de Programación	
No. Tarea: 17	No. Historia: 4
Nombre de la tarea: Buscar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 02/04/2014	Fecha fin: 02/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite buscar a los profesores.	

Tabla 3.20 Tarea de programación Modificar profesor

Tarea de Programación	
No. Tarea: 18	No. Historia: 4
Nombre de la tarea: Modificar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 03/04/2014	Fecha fin: 03/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite modificar información de los profesores.	

Tabla 3.21 Tarea de programación Eliminar profesor

Tarea de Programación	
No. Tarea: 19	No. Historia: 4
Nombre de la tarea: Eliminar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 04/03/2014	Fecha fin: 05/03/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar profesores de la base de datos.	

Tabla 3.22 Tarea de programación Insertar especialidad miembro claustro

Tarea de Programación	
No. Tarea: 20	No. Historia: 5
Nombre de la tarea: Insertar especialidad miembro claustro.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14/04/2014	Fecha fin: 14/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar la especialidad de los miembros del claustro.	

Tabla 3.23 Tarea de programación Mostrar especialidad miembro claustro

Tarea de Programación	
No. Tarea: 21	No. Historia: 5
Nombre de la tarea: Mostrar especialidad miembro claustro.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 15/04/2014	Fecha fin: 15/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite mostrar la especialidad de los miembros del claustro.	

Tabla 3.24 Tarea de programación Buscar especialidad miembro claustro

Tarea de Programación	
No. Tarea: 22	No. Historia: 5
Nombre de la tarea: Buscar especialidad miembro claustro.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/04/2014	Fecha fin: 16/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite buscar la especialidad de los miembros del claustro.	

Tabla 3.25 Tarea de programación Modificar especialidad miembro claustro

Tarea de Programación	
No. Tarea: 23	No. Historia: 5
Nombre de la tarea: Modificar especialidad miembro claustro.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/04/2014	Fecha fin: 17/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite modificar la especialidad de los miembros del claustro.	

Tabla 3.26 Tarea de programación Eliminar especialidad miembro claustro

Tarea de Programación	
No. Tarea: 24	No. Historia: 5
Nombre de la tarea: Eliminar especialidad miembro claustro.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 18/04/2014	Fecha fin: 18/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar la especialidad de los miembros del claustro.	

Tabla 3.27 Tarea de programación Insertar evaluación

Tarea de Programación	
No. Tarea: 25	No. Historia: 6
Nombre de la tarea: Insertar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 21/04/2014	Fecha fin: 22/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar la evaluación de los estudiantes.	

Tabla 3.28 Tarea de programación Mostrar evaluación

Tarea de Programación	
No. Tarea: 26	No. Historia: 6
Nombre de la tarea: Mostrar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 23/04/2014	Fecha fin: 23/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite mostrar la evaluación de los estudiantes.	

Tabla 3.29 Tarea de programación Modificar evaluación

Tarea de Programación	
No. Tarea: 27	No. Historia: 6
Nombre de la tarea: Modificar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24/04/2014	Fecha fin: 24/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite modificar la evaluación de los estudiantes.	

Tabla 3.30 Tarea de programación Eliminar evaluación

Tarea de Programación	
No. Tarea: 28	No. Historia: 6
Nombre de la tarea: Eliminar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 25/04/2014	Fecha fin: 28/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar la evaluación de los estudiantes.	

Tabla 3.31 Tarea de programación Insertar claustro

Tarea de Programación	
No. Tarea: 29	No. Historia: 7
Nombre de la tarea: Insertar claustro.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 29/04/2014	Fecha fin: 30/04/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar el claustro de profesores.	

Tabla 3.32 Tarea de programación Eliminar claustro

Tarea de Programación	
No. Tarea: 30	No. Historia: 7
Nombre de la tarea: Eliminar claustro.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 05/05/2014	Fecha fin: 05/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar el claustro de profesores.	

Tabla 3.33 Tarea de programación Insertar usuario

Tarea de Programación	
No. Tarea: 31	No. Historia: 8
Nombre de la tarea: Insertar usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/05/2014	Fecha fin: 07/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar usuarios en el sistema.	

Tabla 3.34 Tarea de programación Mostrar usuario

Tarea de Programación	
No. Tarea: 32	No. Historia: 8
Nombre de la tarea: Mostrar usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 08/05/2014	Fecha fin: 08/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite mostrar a usuarios del sistema.	

Tabla 3.35 Tarea de programación Buscar usuario

Tarea de Programación	
No. Tarea: 33	No. Historia: 8
Nombre de la tarea: Buscar usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 09/05/2014	Fecha fin: 09/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite buscar a los usuarios en el sistema.	

Tabla 3.36 Tarea de programación Modificar usuario

Tarea de Programación	
No. Tarea: 34	No. Historia: 8
Nombre de la tarea: Modificar usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 12/05/2014	Fecha fin: 12/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite modificar a los usuarios del sistema.	

Tabla 3.37 Tarea de programación Eliminar usuario

Tarea de Programación	
No. Tarea: 35	No. Historia: 8
Nombre de la tarea: Eliminar usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/05/2014	Fecha fin: 14/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar a los usuarios del sistema.	

Tabla 3.38 Tarea de programación Insertar solicitud

Tarea de Programación	
No. Tarea: 36	No. Historia: 8
Nombre de la tarea: Insertar solicitud.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 15/05/2014	Fecha fin: 16/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite insertar una solicitud de posgrado.	

Tabla 3.39 Tarea de programación Mostrar solicitud

Tarea de Programación	
No. Tarea: 37	No. Historia: 8
Nombre de la tarea: mostrar solicitud.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 19/05/2014	Fecha fin: 19/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite mostrar una solicitud de posgrado.	

Tabla 3.40 Tarea de programación Buscar solicitud

Tarea de Programación	
No. Tarea: 38	No. Historia: 8
Nombre de la tarea: Buscar solicitud.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20/05/2014	Fecha fin: 20/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite buscar una solicitud de posgrado.	

Tabla 3.41 Tarea de programación Eliminar solicitud

Tarea de Programación	
No. Tarea: 39	No. Historia: 8
Nombre de la tarea: Eliminar solicitud.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 21/05/2014	Fecha fin: 22/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite eliminar una solicitud de posgrado.	

Tabla 3.42 Tarea de programación Autenticar usuario

Tarea de Programación	
No. Tarea: 40	No. Historia: 8
Nombre de la tarea: Autenticar usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 25/05/2014	Fecha fin: 26/05/2014
Programador responsable: Nestor Antonio López Arcia	
Descripción: Permite autenticar los usuarios en el sistema.	

Anexo 4 – Pruebas de Aceptación

Tabla 3.43 Prueba de aceptación para la HU: Gestionar Material

Prueba de Aceptación
HU: Gestionar Material.
Nombre: Prueba para comprobar la gestión de los materiales bibliográficos.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de los materiales bibliográficos.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder a la aplicación y poder insertar, mostrar, buscar, modificar y eliminar los materiales bibliográficos.
Entrada / Pasos ejecución: El administrador escribe los datos que se piden para insertar el material bibliográfico y pulsa el botón Insertar. Después que el material bibliográfico esté insertado se podrá mostrar, buscar, modificar o eliminar de un listado.
Resultado esperado: <ul style="list-style-type: none">• Se insertan correctamente los materiales bibliográficos.• Se muestran los materiales bibliográficos.• Se pueden modificar o eliminar una vez insertados. Se produce un error en caso de que: <ul style="list-style-type: none">• Se dejen campos obligatorios vacíos.• El material bibliográfico ya está registrado en la base de datos.• Se inserten letras en campos que deben ser numéricos.
Evaluación de la prueba: Aceptada.

Tabla 3.44 Prueba de aceptación para la HU: Gestionar Estudiante

Prueba de Aceptación
HU: Gestionar Estudiante.

Nombre: Prueba para comprobar la gestión de los estudiantes.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de los estudiantes.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder a la aplicación y poder insertar, mostrar, buscar, modificar y eliminar los estudiantes.
Entrada / Pasos ejecución: El administrador escribe los datos que se piden para insertar el estudiante y pulsa el botón Insertar. Después que el estudiante esté insertado se podrá mostrar, buscar, modificar o eliminar de un listado.
Resultado esperado: <ul style="list-style-type: none"> • Se insertan correctamente los estudiantes. • Se muestran los estudiantes. • Se pueden modificar o eliminar una vez insertados. Se produce un error en caso de que: <ul style="list-style-type: none"> • Se dejen campos obligatorios vacíos. • El estudiante ya está registrado en la base de datos. • Se inserten letras en campos que deben ser numéricos.
Evaluación de la prueba: Aceptada.

Tabla 3.45 Prueba de aceptación para la HU: Gestionar Profesor

Prueba de Aceptación
HU: Gestionar Profesor.
Nombre: Prueba para comprobar la gestión de los profesores.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de los profesores.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder a la aplicación y poder insertar, mostrar, buscar, modificar y eliminar los profesores.

Entrada / Pasos ejecución: El administrador escribe los datos que se piden para insertar el profesor y pulsa el botón Insertar. Después que el profesor esté insertado se podrá mostrar, buscar, modificar o eliminar de un listado.
Resultado esperado: <ul style="list-style-type: none"> • Se insertan correctamente los profesores. • Se muestran los profesores. • Se pueden modificar o eliminar una vez insertados. Se produce un error en caso de que: <ul style="list-style-type: none"> • Se dejen campos obligatorios vacíos. • El profesor ya está registrado en la base de datos. • Se inserten letras en campos que deben ser numéricos.
Evaluación de la prueba: Aceptada.

Tabla 3.46 Prueba de aceptación para la HU: Gestionar Especialidad Miembro Claustro

Prueba de Aceptación
HU: Gestionar Especialidad Miembro Claustro.
Nombre: Prueba para comprobar la gestión de las especialidades de los miembros del claustro.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de las especialidades de los miembros del claustro.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder a la aplicación y poder insertar, mostrar, buscar, modificar y eliminar las especialidades.
Entrada / Pasos ejecución: El administrador escribe los datos que se piden para insertar las especialidades y pulsa el botón Insertar. Después que la especialidad esté insertada se podrá mostrar, buscar, modificar o eliminar de un listado.

Resultado esperado:

- Se insertan correctamente las especialidades.
- Se muestran las especialidades.
- Se pueden modificar o eliminar una vez insertados.

Se produce un error en caso de que:

- Se dejen campos obligatorios vacíos.
- El profesor ya está registrado en la base de datos.
- Se inserten letras en campos que deben ser numéricos.

Evaluación de la prueba: Aceptada.

Tabla 3.47 Prueba de aceptación para la HU: Gestionar Evaluación

Prueba de Aceptación
HU: Gestionar Evaluación.
Nombre: Prueba para comprobar la gestión de las evaluaciones de los estudiantes.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de las evaluaciones.
Condiciones de ejecución: El profesor debe iniciar su sesión para acceder a la aplicación y poder insertar, mostrar, buscar, modificar y eliminar las evaluaciones.
Entrada / Pasos ejecución: El profesor escribe los datos que se piden para insertar las evaluaciones y pulsa el botón Insertar. Después que la evaluación esté insertada se podrá mostrar, buscar, modificar o eliminar de un listado.

<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Se insertan correctamente las evaluaciones. • Se muestran las evaluaciones. • Se pueden modificar o eliminar una vez insertados. <p>Se produce un error en caso de que:</p> <ul style="list-style-type: none"> • Se dejen campos obligatorios vacíos. • Se inserten letras en campos que deben ser numéricos.
<p>Evaluación de la prueba: Aceptada.</p>

Tabla 3.48 Prueba de aceptación para la HU: Gestionar Claustro

Prueba de Aceptación
HU: Gestionar Claustro.
Nombre: Prueba para comprobar la gestión del claustro de profesores.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de los claustros.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder a la aplicación y poder insertar y eliminar los claustros.
Entrada / Pasos ejecución: El administrador escribe los datos que se piden para insertar los claustros y pulsa el botón Insertar. Después de que claustro esté insertado se podrá mostrar, buscar o eliminar de un listado.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> • Se insertan correctamente los claustros. • Se muestran los claustros. • Se pueden eliminar una vez insertados. <p>Se produce un error en caso de que:</p> <ul style="list-style-type: none"> • Se dejen campos obligatorios vacíos. • Se inserten letras en campos que deben ser numéricos.
<p>Evaluación de la prueba: Aceptada.</p>

Tabla 3.49 Prueba de aceptación para la HU: Gestionar Usuario

Prueba de Aceptación
HU: Gestionar Usuario.
Nombre: Prueba para comprobar la gestión de los usuarios.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de los usuarios.
Condiciones de ejecución: El administrador debe iniciar su sesión para acceder a la aplicación y poder insertar, mostrar, buscar, modificar y eliminar los usuarios.
Entrada / Pasos ejecución: El administrador escribe los datos que se piden para insertar los usuarios y pulsa el botón Insertar. Después que el usuario esté insertado se podrá mostrar, buscar, modificar o eliminar de un listado.
Resultado esperado: <ul style="list-style-type: none"> • Se insertan correctamente los usuarios. • Se muestran los usuarios. • Se pueden modificar o eliminar una vez insertados. Se produce un error en caso de que: <ul style="list-style-type: none"> • Se dejen campos obligatorios vacíos. • El usuario ya está registrado en la base de datos. • Se inserten letras en campos que deben ser numéricos.
Evaluación de la prueba: Aceptada.

Tabla 3.50 Prueba de aceptación para la HU: Gestionar Solicitud

Prueba de Aceptación
HU: Gestionar Solicitud.
Nombre: Prueba para comprobar la gestión de las solicitudes.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de las solicitudes de cursos y entrenamientos de posgrado.

Condiciones de ejecución: El administrador debe iniciar su sesión para acceder a la aplicación y poder insertar, mostrar, buscar y eliminar las solicitudes.
Entrada / Pasos ejecución: El administrador escribe los datos que se piden para insertar las solicitudes y pulsa el botón Insertar. Después que la solicitud esté insertada se podrá mostrar, modificar o eliminar de un listado.
Resultado esperado: <ul style="list-style-type: none"> • Se insertan correctamente las solicitudes. • Se muestran las solicitudes. • Se pueden modificar o eliminar una vez insertados. Se produce un error en caso de que: <ul style="list-style-type: none"> • Se dejen campos obligatorios vacíos. • Se inserten letras en campos que deben ser numéricos.
Evaluación de la prueba: Aceptada.

Tabla 3.51 Prueba de aceptación para la HU: Autenticar Usuario

Prueba de Aceptación
HU: Autenticar usuario.
Nombre: Prueba para comprobar la entrada de un usuario al sistema.
Descripción: Validación de entrada de los datos del usuario.
Condiciones de ejecución: El usuario debe entrar a la aplicación y tener el rol de Administrador para poder insertar, modificar, mostrar y eliminar usuarios.
Entrada / Pasos ejecución: El administrador escribe los datos del usuario que se piden para crear el usuario luego presiona el botón Insertar. Después que el usuario esté insertado se podrá eliminar, modificar y acceder a sus datos en el listado de usuarios.

Resultado esperado:

- Si se introducen los datos correctamente se creará el nuevo usuario.
- Si se desea eliminar el usuario se muestra un mensaje de confirmación

Se produce un error en caso de que:

- Se inserten los datos de un usuario ya creado.
- Se inserte en el campo Email un correo no válido.
- Se dejen campos obligatorios vacíos.

Evaluación de la prueba: Aceptada.