



ISMMM

INSTITUTO SUPERIOR MINERO
METALURGICO DE MOA
DR. ANTONIO NUÑES JIMENEZ

Ingeniería Informática
Facultad: Geología y Minas

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

**Título: Implementación de una aplicación web
para la gestión de información de las maestrías
en el ISMMM.**

Autor (es): Carlos Fernando Nápoles Ávila

Tutor (es): Ing. Adriannys Céspedes Ramírez

Ing. José Rolando Pérez Sandó

Moa, 2014

“Año 56 de la Revolución”



DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa para que haga el uso que estime pertinente con este trabajo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año 2014.

Carlos Fernando Nápoles Ávila

Firma autor

Ing. Adriannys Céspedes Ramírez

Firma tutor

Ing. José Rolando Pérez Sandó

Firma tutor

Pensamiento.

Si no deseas mucho, hasta las cosas pequeñas
te parecerán grandes.

Dedicatoria.

A mis padres... que se han esforzado mucho para que llegue aquí.

A mi esposa... porque te amo mucho.

A mi familia y amigos que han dedicado recurso y tiempo para la
realización de este proyecto.

Agradecimientos.

Antes que todo -porque siempre he luchado para que esté así en mi vida- agradezco a Dios por estar siempre a mi lado.

Agradezco a mi esposa especialmente por estar conmigo, aguantarme y darme ánimo en todo momento.

A mis padres que desde siempre se han ocupado de que no me falte nada... Por sus palabras de aliento, por comprenderme y confiar en mí.

Gracias por la educación que me han dado...

Agradezco a mis tutores por darme su apoyo.

Gracias a todos los trabajadores del CEETAM por el tiempo que compartí con ustedes.

Especialmente te agradezco a ti, que no he mencionado tu nombre pero haz contribuido para que realice esta tesis.

Resumen

El Instituto Superior Minero Metalúrgico de Moa (ISMMM) es caracterizado entre otras cosas por sus convocatorias de maestrías, cada convocatoria genera grandes cantidades de datos que actualmente se gestionan de forma manual, lo que ha provocado entre otras fallas pérdida de información de vital importancia y atrasos en el trabajo de los especialistas y coordinadores de las maestrías.

El objetivo del presente trabajo es desarrollar una aplicación web que facilite la propagación e intercambio de información de las maestrías convocadas en el ISMMM, con el fin de apoyar la gestión de información.

El uso de esta aplicación le facilitará el trabajo al Coordinador, Comité Académico y al Claustro de la Maestría. Además, permitirá que se realicen búsquedas rápidas, corrección de los datos en caso de que al introducirlos se cometa algún error, brindará la opción de ver resúmenes de toda la información de las ediciones o maestrantes que se seleccione, igualmente que permita saber todos los integrantes que conforman un determinado claustro.

Para el desarrollo de la herramienta, primeramente se hizo un estudio detallado de sistemas de gestión de maestrías en diferentes ámbitos: internacional, nacional y en la universidad. Se utilizó la metodología de desarrollo *eXtreme Programming* (XP), como lenguajes de programación: Java y XML del lado del servidor y HTML y JavaScript del lado del cliente. El trabajo concluye con la implementación de un sistema que cumple con las características necesarias para satisfacer las necesidades y expectativas de los usuarios finales.

Abstract

The Institute Superior Mining Metallurgist of Moa (ISMMM) it is characterized among other things by their convocations of masters, each convocation generates big quantities of data that at the moment are negotiated in a manual way, what has caused between other flaws loss of information of vital importance and back payments in the work of the specialists and coordinators of the masters.

The objective of the present work is to develop an application web that facilitates the propagation and exchange of information of the masters summoned in the ISMMM, with the purpose of supporting the administration of information.

The use of this application will facilitate the work to the Coordinator, Academic Committee and to the Cloister of the Master. Also, it will allow that they are carried out quick searches, correction of the data in case when introducing them some error it is made, it will offer the option of seeing summaries of all the information of the editions or maestranter that it is selected, equally that allows to know all the members that conform a certain cloister.

For the development of the tool, firstly a detailed study of systems of administration of masters was made in different environments: international, national and in the university. The development methodology was used Programming it carries to an extreme (XP), as programming languages: Java and XML on the side of the servant and HTML and JavaScript on the side of the client. The work concludes with the implementation of a system that fulfills the necessary characteristics to satisfy the necessities and the final users' expectations.

Índice

Capítulo 1: Fundamentación Teórica.....	5
1.1 Introducción al capítulo	5
1.2 Estado del Arte.....	5
1.2.1 Soluciones existentes para la gestión de información de postgrado en el ámbito Internacional.....	5
1.2.2 Soluciones existentes para la gestión de información de postgrado en el ámbito nacional.....	6
1.2.3 Soluciones existentes para la gestión de información de las maestrías del ISMMM.....	7
1.3 Lenguajes y herramientas utilizados	8
1.3.1 Lenguajes de programación.....	8
1.3.2 Servidor web	10
1.3.3 Marco(s) de trabajo (Framework).....	11
1.3.4 Sistema Gestor de Base de Datos	12
1.3.5 Entorno de Desarrollo Integrado	13
1.3.6 Herramientas CASE.....	14
1.4 Metodología para el Desarrollo de Software	15
1.4.1 XP (Programación Extrema)	15
1.5 ¿Por qué elegir XP?	16
1.5.1 ¿Qué propone XP?	17
1.5.2 Fases de la metodología XP	17
1.6 Patrón arquitectónico	19
Conclusiones.....	20
Capítulo 2: Planificación y Diseño	21
2.1 Introducción.....	21

2.2 Problema y situación problemática.....	21
2.3 Propuesta de solución.....	21
2.4 Descripción del proceso del negocio.....	22
2.5 Personal relacionado con el sistema.....	23
2.6 Reglas del negocio.....	24
2.7 Lista de reserva del producto	25
2.8 Historias de usuario.....	28
2.9 Planificación de entregas	34
2.9.1 Estimación de esfuerzo por Historia de Usuario	35
2.9.2 Plan de iteraciones.....	35
2.10 Tarjetas CRC (Clase, Responsabilidad, Colaboración).....	37
Conclusiones.....	40
Capítulo 3: Implementación y Pruebas.....	41
3.1 Introducción.....	41
3.2 Patrones de diseño	41
3.3 Diseño de la base de datos.....	44
3.4 Desarrollo de las iteraciones	45
3.4.1 Tareas de ingeniería por Historia de Usuario.....	45
3.5 Pruebas.....	48
3.5.1 Desarrollo dirigido por Pruebas.....	48
3.5.2 Pruebas de aceptación (PA)	49
Conclusiones.....	53
Capítulo 4: Estudio de Factibilidad y Sostenibilidad	54
4.1 Introducción.....	54
4.2 Efectos económicos	54

Índice

4.3 Beneficios y Costos Intangibles en el proyecto	56
4.4 Ficha de costo	56
4.4.1 Costos en Moneda Librementemente Convertible:	56
4.4.2 Costos en Moneda Nacional:	58
Conclusiones.....	59
Conclusiones Generales	60
Recomendaciones.....	61
Referencias Bibliográficas	62
Bibliografía.....	64
Anexos	66
Anexo 1 Historias de Usuario.....	66
Anexo 2 Tarjetas CRC	71
Anexo 3 Pruebas de Aceptación	77

Índice de Tablas y Figuras

Figura 2.1: Diagrama de proceso del negocio	23
Tabla 2.1: Personal relacionado con el sistema	23
Tabla 2.2: Características y Funcionalidades del sistema.....	25
Tabla 2.3: Plantilla de historia de usuario a utilizar.....	29
Tabla 2.4: Historia de usuario gestionar maestría	29
Tabla 2.5: Historia de usuario gestionar edición.....	31
Tabla 2.6: Historia de Usuario Gestionar Profesor	32
Tabla 2.7: Historia de Usuario Gestionar Estudiante	34
Tabla 2.8: Estimación de esfuerzo por Historia de Usuario	35
Tabla 2.9: Plan de iteraciones	36
Fig. 2.2: Tarjeta CRC Edición	37
Fig. 2.3: Tarjeta CRC Estudiante.....	38
Fig. 2.4: Tarjeta CRC Maestría.....	38
Fig. 2.5: Tarjeta CRC Persona	39
Fig. 2.6: Tarjeta CRC Profesor	39
Fig. 2.7: Tarjeta CRC Asignatura.....	40
Figura 3.1: Modelo de la base de datos	45
Tabla 3.1 Tareas de ingeniería por Historia de Usuario	45
Tabla 3.2 Modelo para pruebas de aceptación.....	49
Tabla 3.3 Prueba de aceptación para la Historia de Usuario Gestionar Maestría	50
Tabla 3.4 Prueba de aceptación para la Historia de Usuario Gestionar Edición	51
Tabla 3.5 Prueba de aceptación para la Historia de Usuario Gestionar Módulo	51
Tabla 3.6 Prueba de aceptación para la Historia de Usuario Gestionar Profesor.....	52
Tabla 3.7 Prueba de aceptación para la Historia de Usuario Gestionar Asignatura	53

Tabla 4.1: Costo en Moneda Libremente Convertible	56
Tabla 4.4.2: Costo en Moneda Nacional.	58
Figura 4.1 Gráfica de la solución con el producto y solución sin el producto	59
Tabla 2.10: Historia de usuario gestionar asignatura	66
Tabla 2.11: Historia de usuario gestionar material de asignatura.....	66
Tabla 2.12: Historia de usuario gestionar usuario	67
Tabla 2.13: Historia de usuario gestionar módulo	68
Tabla 2.14: Historia de usuario gestionar especialidad	68
Tabla 2.15: Historia de usuario gestionar solicitud	69
Tabla 2.16: Historia de usuario gestionar evaluación	70
Tabla 2.17: Historia de usuario administrar claustro.....	70
Fig. 2.8: Tarjeta CRC Categoría Docente.....	71
Fig. 2.9: Tarjeta CRC Claustro	72
Fig. 2.10: Tarjeta CRC Especialidad	72
Fig. 2.11: Tarjeta CRC Evaluación	73
Fig. 2.12: Tarjeta CRC Grado Científico.....	73
Fig. 2.13: Tarjeta CRC Material	74
Fig. 2.14: Tarjeta CRC Menú.....	74
Fig. 2.15: Tarjeta CRC Modalidad	75
Fig. 2.16: Tarjeta CRC País.....	75
Fig. 2.17: Tarjeta CRC Permiso.....	76
Fig. 2.18: Tarjeta CRC Solicitud	76
Fig. 2.19: Tarjeta CRC Usuario	77
Tabla 3.8 Prueba de aceptación para la Historia de Usuario Gestionar Especialidad...	77
Tabla 3.9 Prueba de aceptación para la Historia de Usuario Gestionar Material de Asignatura.....	78

Índice de Tablas y Figuras

Tabla 3.10 Prueba de aceptación para la Historia de Usuario Gestionar Usuario..... 78

Tabla 3.11 Prueba de aceptación para la Historia de Usuario Gestionar Solicitud..... 79

Tabla 3.12 Prueba de aceptación para la Historia de Usuario Gestionar Evaluación.... 80

Introducción

En la actualidad el mundo vive dentro de la Sociedad de la Información o Informacional (SI), en la que las condiciones de generación de conocimiento y procesamiento de información han sido sustancialmente alteradas por una revolución tecnológica centrada en el vertiginoso intercambio de información mediante las Tecnologías de la Informática y la Comunicación (TIC). En el marco de estos fenómenos se desarrolla el proceso de Informatización de la Sociedad Cubana, que incluye cada vez más áreas de la vida del país. El Ministerio de Educación Superior (MES) no está al margen de este trabajo, y ha creado numerosos proyectos con el objetivo de lograr la automatización de los procesos fundamentales de la misma. Con esto se pretende garantizar la elevación de la calidad de la gestión de dirección de la Educación Superior a todos los niveles, incluyendo la transformación de los procesos que lo requieran, así como asegurar un eficiente uso, distribución y a la vez centralización de la información, utilizando la interconexión que permiten las redes de computadoras.

En el Instituto Superior Minero Metalúrgico de Moa (ISMMM) se ha hecho necesaria la creación de sistemas informáticos que mejoren y faciliten la informatización de sus procesos sustanciales, favoreciéndose la gestión de la información en los procesos productivos y de servicios. Este objetivo ha sido considerado en una estrategia de informatización que contempla los procesos sustantivos de la institución. Uno de los principales procesos de servicios que se desea informatizar en la Institución es la Gestión de Posgrado Académico, específicamente, las maestrías.

La Vicerrectoría de Investigación y Posgrado (VRIP) del ISMMM es la encargada de atender los procesos asociados a la investigación, el posgrado, la superación de cuadros, la comercialización de la actividad científica, la colaboración internacional, la información científico técnica y 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.

La maestría corresponde al proceso de formación posgraduada que proporciona a los graduados universitarios una amplia cultura científica y conocimientos avanzados en las áreas correspondientes del saber, una mayor capacidad para la actividad docente,

científica, la innovación o la creación artística, en correspondencia con las necesidades del desarrollo económico, social y cultural del país.

Con la ejecución de cada una de las maestrías atendidas por la VRIP se generan grandes cantidades de datos. Actualmente la gestión se realiza de forma manual, provocando entre otras fallas pérdida de información importante, duplicado de datos y atrasos en el trabajo de los profesores, especialistas y coordinadores de las maestrías.

Se evidencia que la disponibilidad de la información está limitada, dañando el lanzamiento de las convocatorias. En muchos casos también existe una notable demora en la gestión de solicitud de las maestrías y postgrados, siendo engorroso de igual manera el intercambio de bibliografías y tareas realizadas, entre profesores y estudiantes. Específicamente, en esta actividad, la Vicerrectoría de Investigación y Postgrado ha propuesto que la solicitud, propagación e intercambio de información de sus maestrías se realicen desde la Web, así como el intercambio de diferentes comunicados, bibliografías, tareas realizadas, evaluaciones, entre otros.

Con este trabajo se pretende dar solución a la problemática anteriormente expuesta, definiéndose como **problema científico**: ¿Cómo favorecer la gestión de la información en las maestrías de la VRIP del ISMMM? Teniendo en cuenta el problema planteado el **objeto de estudio** de este trabajo son los sistemas informáticos para la gestión de la información en maestrías, **el campo de acción** se enmarca en la informatización de la gestión de información en maestrías de la VRIP del ISMMM.

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

Implementar una aplicación que favorezca la gestión de la información en las maestrías de la VRIP del ISMMM.

Como **idea a defender** se plantea que la implementación de una aplicación web, orientada a agilizar el procesamiento e intercambio de la información así como la gestión de las solicitudes de ingreso en las maestrías, favorecerá el funcionamiento de la VRIP del ISMMM.

Para darle paso al objetivo general fue necesario establecer los siguientes **objetivos específicos y tareas de la investigación**:

- A. Definir el marco teórico-conceptual de la investigación.
 - A.1. Establecer los elementos teóricos de la gestión de información en las maestrías de la VRIP del ISMMM.
 - A.2. Estudiar las soluciones informáticas vinculadas a la gestión de información en maestrías.
 - A.3. Valorar las tendencias actuales de las herramientas y tecnología.

- B. Implementar el sistema propuesto.
 - B.1. Determinar las necesidades de funcionamiento de la aplicación a desarrollar.
 - B.2. Implementar la aplicación.
 - B.3. Realizar las pruebas funcionales al sistema construido.

- C. Realizar el estudio de factibilidad
 - C.1. Seleccionar la metodología para el estudio de factibilidad.
 - C.2. Determinar la factibilidad.

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

Métodos teóricos

- **Análisis y síntesis:** Este método se utiliza para desglosar el problema en partes o sub-problemas para de esta forma comprobar el correcto funcionamiento de las mismas, luego integrarlo todo para corroborar las relaciones entre estas y su integración como un todo, llegando así a una mejor solución, también para arribar conclusiones parciales y generales de la investigación.

- **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, 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.

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: “Planificación y Diseño”: Se expone la propuesta de solución, describiendo el proceso del negocio y los requisitos funcionales y no funcionales del software. 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: “Implementación y Pruebas”: se expondrá el resultado de la implementación del sistema, en este capítulo se mostrarán los resultados obtenidos de las pruebas realizadas al sistema como parte del proceso de validación y verificación de los requisitos funcionales.

Capítulo 4: “Estudio de Factibilidad”: se realiza un estudio de los esfuerzos requeridos para la realización del sistema, se hace referencia a los beneficios tangibles e intangibles.

Capítulo 1: Fundamentación Teórica

1.1 Introducción al capítulo

En la actualidad, la informática ha evolucionado considerablemente y junto a ella la producción de diversos software. En este capítulo abordaremos los conceptos fundamentales para el desarrollo de aplicaciones web, así como también ilustraremos un estudio de las ventajas y desventajas que estas ofrecen. Se mostrarán 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 ofrecen. Por último se hace una descripción de las herramientas y tecnologías utilizadas para el análisis y diseño del software.

1.2 Estado del Arte

Con la finalidad de adquirir experiencias para el sistema que se pretende desarrollar se realizó un estudio de sistemas homólogos al que se desea obtener, en el ámbito internacional, nacional y en la universidad. También se analizó en qué medida los sistemas encontrados dan solución a la situación problemática planteada por el cliente.

1.2.1 Soluciones existentes para la gestión de información de postgrado en el ámbito Internacional

Se encontraron sistemas como el Sistema de Gestión de la Calidad en Venezuela el cual es una aplicación que gestiona la mayoría de la información que es necesario almacenar en la VRIP, pero tiene muchas funcionalidades que actualmente no son necesarias, a pesar de que es bastante seguro y eficaz en la gestión de los postgrados requiere de un hardware bastante avanzado para su instalación con el cual no se cuenta actualmente en el ISMMM y sería costoso adquirirlo. Además de ser un software propietario por lo que debería comprarse y de querer modificarse no sería posible ya que no se puede cambiar su código para poder adaptarlo a las necesidades y condiciones de la VRIP. En el caso de que se pudiesen realizar cambios sería más

trabajoso modificarlo que hacer uno nuevo ya que no existe documentación de la realización del mismo, imposibilitando la completa comprensión del código.

1.2.2 Soluciones existentes para la gestión de información de postgrado en el ámbito nacional

En Cuba, el desarrollo de estos sistemas es escaso y principalmente se ha trabajado la presentación de postgrados 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. (8). El moodle gestiona las asignaturas, bibliografías, foros y tareas con sus respectivas evaluaciones, también es fundamental dentro de sus características la gestión de cursos virtuales para sus alumnos (educación a distancia), o la utilización de un espacio en línea que dé apoyo a la presencialidad (aprendizaje semipresencial) (8) sin embargo no contempla aspectos de gran interés como son los registros de maestrantes y profesores, información del claustro, datos de las ediciones de las maestrías, solicitudes de las maestrías. También en entrevistas realizadas a distintos coordinadores de maestrías se resaltan dificultades con el manejo de la interfaz propuesta por el MOODLE.

Por lo analizado anteriormente se decidió tomar como ejemplo el diseño referente a la publicación de los curso, pero se decide no utilizar el mismo dado a la cantidad de características innecesarias que posee para nuestra problemática, además de la insatisfacción de los clientes.

El **SIGENU** es un sistema concebido para la gestión de los distintos procesos concebidos en el marco de la Educación Superior. Surge como sistema en el año 2004, siendo diseñado para desktop y plataforma web, como necesidad del MES para lograr una mayor homogeneidad en los procesos sustantivos de la nueva universidad y hacer una mejor toma de decisiones en cuanto a los recursos económicos que dispone el país para desarrollar cada curso escolar. SIGENU permite registrar los datos de los estudiantes desde el momento de la matrícula hasta que se gradúan o causan baja

definitiva, incluyendo bajas temporales, licencia, repitencia, reportes evaluativos, cambio del lugar de residencia, etc. (2).

Se encuentra como limitación que no posee un módulo para realizar otras funciones como la gestión de información de los profesores. Además no contempla dentro de sus características el manejo de la parte curricular de los estudiantes o maestrantes en el caso de nuestra problemática. Como deficiencias se contempla que no posee un módulo que gestione solicitudes de matrículas a cursos, tampoco maneja información acerca del módulo de asignaturas a cursar por un maestrante, ni los materiales correspondientes a estas, como tampoco permite llevar un registro de evaluaciones de los estudiantes, ni tener control sobre las distintas ediciones de una maestría con su respectivo claustro de profesores.

1.2.3 Soluciones existentes para la gestión de información de las maestrías del ISMMM

En el Instituto Superior Minero Metalúrgico de Moa, en la carrera Ingeniería Informática, en el año 2011 se desarrolló el Trabajo de Diploma titulado “Sistema Automatizado para la Gestión de Información de la Maestría de Electromecánica”, obteniéndose un producto informático para la gestión de algunas actividades referentes a la gestión de maestrías.

Este sistema no está siendo utilizado por los usuarios, ya que no contempla situaciones relevantes del proceso que se debe llevar a cabo. Se detectó además que las actividades informatizadas no se realizan con la calidad requerida, siendo esta queja planteada por varios clientes. En sentido general, opinan que es un sistema de poca usabilidad por lo que prefieren realizar el trabajo de forma manual. Otra de las deficiencias encontradas se orienta a que no ofrece las validaciones necesarias para la entrada de datos al sistema, ni la seguridad que requiere este tipo de aplicación se maneja de forma correcta. Por tanto se concluye que no se puede utilizar, solo tomar como referencia algunos requisitos de este sistema para el desarrollo de la propuesta de solución de esta investigación.

Hasta el momento no existe entre los software encontrados uno que contemple el proceso completo de gestión de maestrías. Algunos de ellos poseen características que

se van más allá del propósito de esta investigación, careciendo de las principales demandadas por los coordinadores de maestrías entrevistados. Por consiguiente es necesario realizar una herramienta que potencie la usabilidad, contemplando las características necesarias y suficientes para facilitar el proceso de desarrollo de una maestría. Para ello se tomarán experiencias de las aplicaciones previamente desarrolladas y estudiadas como parte del estado del arte. Se decide brindar especial atención a la disponibilidad de la información y a su intercambio, minimizando las operaciones manuales.

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. (4)

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 en 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. (3)

1.3.1.4 Structured Query Language (SQL)

El **lenguaje de consulta estructurado** o **SQL** es un lenguaje declarativo de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales, permitiendo así especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella. Gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros —y no a registros individuales— permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. (11)

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. (9)

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 (9).

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.

1.3.3 Marco(s) 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. (19)

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

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.

Esta librería incluye: (19)

- 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.

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. (11) 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 Lenguaje), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado.

1.3.4.1 PostgreSQL

Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS), que ha sido desarrollado de varias formas desde 1977, es una herramienta muy potente para los desarrolladores de sistemas de bases de datos. PostgreSQL tiene transacciones, integridad referencial, vistas, y multitud de funcionalidades. (11)

PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. (1) Posee muchas características, algunas son la alta concurrencia y Amplia variedad de tipos nativos, entre otras.

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. (5)

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. (5)

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: (5)

- 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. (7)

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.

1.3.6.2 Visual Paradigm para UML

Visual Paradigm para UML es una herramienta que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.(20) El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.(13)

1.4 Metodología para el Desarrollo de Software

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. (12)

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. (14) A continuación se analizan varias metodologías de desarrollo de software escogiendo la más factible para el desarrollo de la herramienta a construir.

1.4.1 XP (Programación Extrema)

XP, es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.(12)

XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, que describe la filosofía de XP en el "Manifiesto Ágil", sin cubrir los detalles técnicos y de implantación de las prácticas.

Se centra especialmente en documentar en forma de plantillas, tiene cuatro fases: Planeación, Diseño, Desarrollo o Implementación y Pruebas. En la primera fase se generan como artefactos los usuarios del negocio, las historias de usuarios, la lista de reserva del producto, el plan de iteraciones, entre otros. En la segunda se tiene el modelo de datos, tarjetas CRC. En tercera fase se desarrollaron las tareas de ingeniería y la cuarta fase son efectuadas las pruebas al software para verificar que el mismo cumpla con todas las funcionalidades acordadas, estas pruebas pueden ser aceptadas por el cliente o denegadas por el mismo. (12)

1.5 ¿Por qué elegir XP?

Actualmente XP es la metodología ágil más documentada (hay una colección de libros "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. Un estudio a la bibliografía, muestra las ventajas que tiene XP y que exponemos en los puntos siguientes: (12)

- Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas)
- Puede ser implementado en forma gradual.
- Puede adaptarse a las necesidades de cualquier equipo de desarrollo. De hecho, Kent Beck recomienda a los equipos que lo adapten a sus necesidades.
- 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.
- Se realizan pruebas constantemente del sistema.

1.5.1 ¿Qué propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes de que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

1.5.2 Fases de la metodología XP

1.5.2.1 Fase I: Planificación

1. 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 store) y permita hacer una estimación de tiempo de desarrollo del mismo.
2. 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.
3. 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.
4. Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.

1.5.2.2 Fase II: Diseño

1. Se eligen los diseños funcionales más simples.
2. Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.

3. 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.

1.5.2.3 Fase III: Codificación

1. 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.
2. El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
3. 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.
4. 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.
5. 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.
6. 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.
7. Se deja la optimización para el final.
8. No se hacen horas extra de trabajo.

1.5.2.4 Fase IV: Pruebas

1. Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
2. Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
3. Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las user stories 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 el correspondiente user storie se ha completado.

1.6 Patrón arquitectónico

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. (6)

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 “Inverition 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.

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 de los lenguajes de programación y el sistema gestor de bases de datos que se pretenden utilizar en la 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: Planificación 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 coordinadores, especialistas y profesores de las maestrías del ISMMM se les dificulta registrar todas las actividades y datos que se generan con la realización de las distintas ediciones de las maestrías. Debido a que es muy diversa y cuantiosa la información generada, los involucrados en estas actividades necesitan un sistema que les ayude a realizar su trabajo con mayor rapidez y organización; proporcionando la opción de obtener de forma digital los documentos que se generan, ya que los mismos en copia dura se van deteriorando o extraviando. También se les hace muy difícil realizar reportes o actualizaciones de datos, actividades cuya realización es favorecida con la ayuda de un sistema informático.

2.3 Propuesta de solución

Teniendo en cuenta la situación existente con respecto a las maestrías convocadas en el instituto, se determinó como propuesta de solución, la implementación de una Aplicación web para la Gestión de Información de las maestrías realizadas en el ISMMM.

Se pretende que la aplicación incluya: el registro y gestión de los datos de las maestrías, sus ediciones, maestrantes 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 las ediciones o maestrantes que se seleccione, igualmente que permita saber todos los integrantes que conforman un determinado claustro.

También debe garantizar que se adjunten los materiales de las diferentes asignaturas a impartir en el módulo de una determinada maestría, como también el documento de tesis de maestría, opinión del tutor y oponente entre otros, estos documentos recogen resúmenes e información de vital importancia y deben ser conservadas de forma digital. De la misma manera podrán ser informatizados los datos de las solicitudes, estas son realizadas por los interesados a cursar dichas maestrías.

2.4 Descripción del proceso del negocio

El proceso comienza cuando se crea una nueva edición de una o varias maestrías, esta actividad es realizada por el comité académico de la facultad encargada de dicha maestría. Posteriormente el comité elabora uno o varios módulos que contienen las asignaturas y a la vez define el claustro de profesores que las impartirán. Terminadas estas tareas se lanza una convocatoria de cada maestría con sus respectivas descripciones.

Los aspirantes a cursar dichas maestrías 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 módulos de las asignaturas y sus 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 Tesis de Maestría la cual contribuirá en gran medida para dar una evaluación final al aspirante esta se consolida en un acto de defensa.

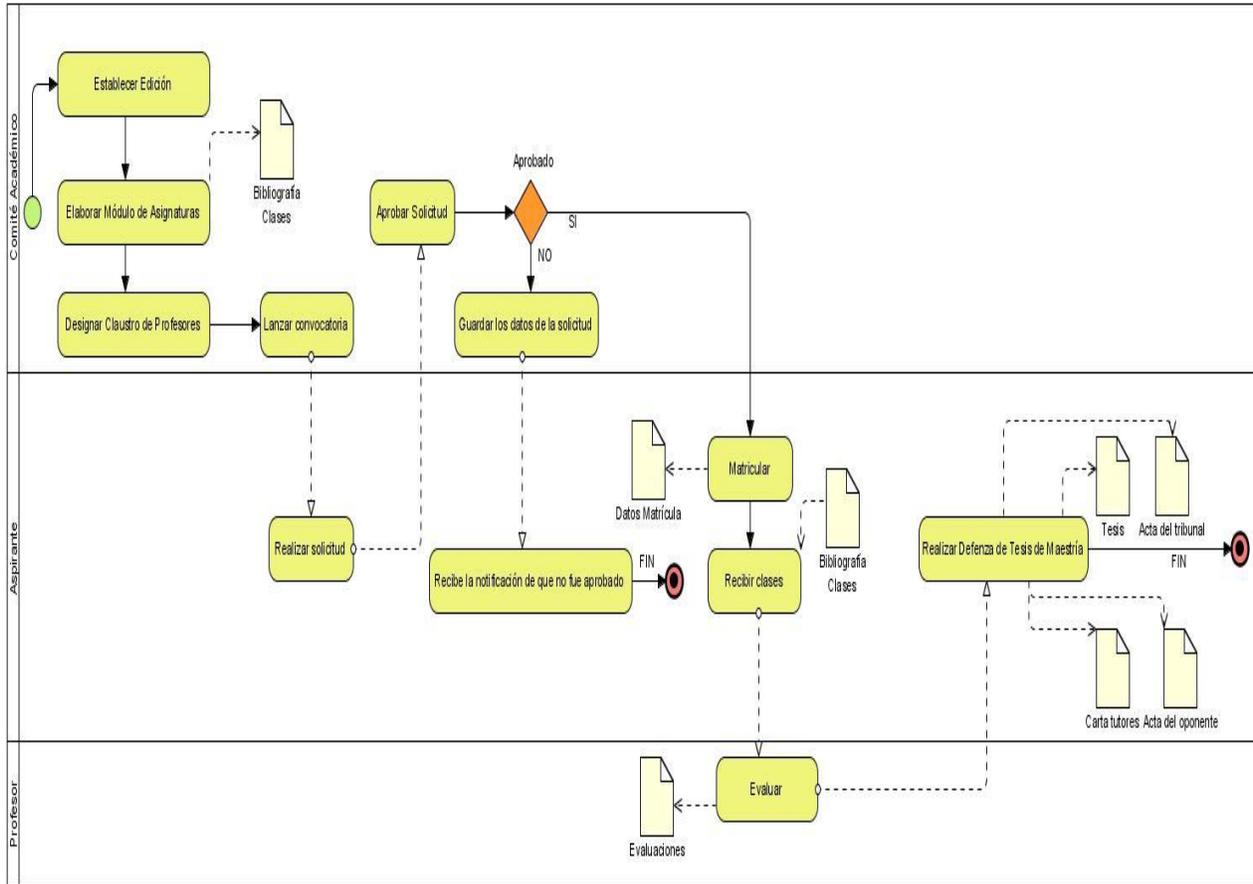


Figura 2.1: Diagrama de proceso del negocio

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.
Coordinador	Usuario que tiene ciertos privilegios para operar sobre 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 Reglas del negocio

Como condiciones de uso del sistema a desarrollar y para su buen funcionamiento se establecieron un conjunto de reglas que deben ser seguidas de manera obligatoria.

1. El comité académico de la maestría aprueba o no la solicitud de un aspirante a la maestría.
2. Una solicitud de matrícula en la maestría puede transitar en los estados: Aprobada o No aprobada.
3. Después que el aspirante es aprobado por el comité académico adquiere el estatus de estudiante, con los deberes y derechos que este implica.
4. Un grado científico puede estar clasificado en: Máster en Ciencias, Doctor en Ciencias.
5. Las evaluaciones finales de las asignaturas a los estudiantes pueden ser: Trabajo final y/o Prueba final.
6. Solo el coordinador de la maestría introducirá los miembros del claustro.
7. Solo el coordinador de la maestría introducirá los datos de los profesores.
8. Solo el profesor de la asignatura podrá añadir evaluaciones al estudiante en esa asignatura.
9. Solo el profesor de la asignatura podrá añadir los materiales obligatorios de la misma.
10. Solo el coordinador de la maestría introducirá los datos de cada edición.
11. Solo el coordinador define la cantidad de módulos a impartir y las asignaturas que conformarán los mismos.
12. Solo el coordinador define que profesor impartirá cada asignatura.
13. Para obtener el Título de Master se debe concluir la etapa con la realización de un trabajo de diploma.

14. Al menos uno de los tutores de la tesis debe pertenecer al claustro de profesores de la maestría.
15. Para defender la tesis de maestría debe tener aprobadas todas las asignaturas obligatorias y un porcentaje de asignaturas optativas.
16. Para obtener el Título de Master el estudiante debe acumular un número mínimo de créditos.

2.7 Lista de reserva del producto

La lista de reserva del producto se realiza una vez esté definido el proceso del negocio, en la misma se pueden observar los requisitos funcionales y no funcionales que el sistema deberá contemplar dentro de sus características, una vez se haya concluido. También incluye la prioridad que tendrá cada requisito a la hora de desarrollarlos, esta prioridad es asignada por el cliente con el fin de obtener en cada entrega mayor valor del negocio. Esta lista se crea con la intención de tener documentadas todas las condiciones que deberá cumplir y las características a tener el sistema una vez esté realizado.

Tabla 2.2: Características y Funcionalidades del sistema

Código	Funcionalidades del Sistema	Prioridad
RF1	Agregar maestría	Alta
RF2	Listar maestrías	Alta
RF3	Modificar información de una maestría	Alta
RF4	Eliminar maestría	Alta
RF5	Buscar maestría	Alta
RF6	Agregar edición	Alta
RF7	Listar ediciones	Alta
RF8	Modificar información de una edición	Alta
RF9	Eliminar edición	Alta
RF10	Buscar edición	Alta
RF11	Agregar módulo	Alta
RF12	Listar módulos	Alta
RF13	Modificar información de un módulo	Alta
RF14	Eliminar módulo	Alta

RF15	Agregar asignatura	Alta
RF16	Listar asignaturas	Alta
RF17	Modificar información de una asignatura	Alta
RF18	Eliminar asignatura	Alta
RF19	Buscar asignatura	Alta
RF20	Agregar profesor	Media
RF21	Listar profesores	Media
RF22	Modificar información de un profesor	Media
RF23	Eliminar profesor	Media
RF24	Buscar profesor	Media
RF25	Agregar estudiante	Media
RF26	Listar estudiantes	Media
RF27	Modificar información de un estudiante	Media
RF28	Eliminar estudiante	Media
RF29	Buscar estudiante	Media
RF30	Agregar claustro	Media
RF31	Eliminar claustro	Media
RF32	Agregar solicitud	Media
RF33	Listar solicitudes	Media
RF34	Eliminar solicitud	Media
RF35	Buscar solicitud	Media
RF36	Agregar especialidad	Media
RF37	Listar especialidades	Media
RF38	Modificar especialidad	Media
RF39	Eliminar especialidad	Media
RF40	Buscar especialidad	Media
RF41	Agregar usuario	Baja
RF42	Listar usuarios	Baja
RF43	Modificar información de un usuario	Baja
RF44	Eliminar usuario	Baja
RF45	Buscar usuario	Baja
RF46	Agregar material de asignatura	Baja
RF47	Listar materiales de asignaturas	Alta

RF48	Eliminar material de asignatura	Alta
RF49	Buscar material de asignatura	Alta
RF50	Agregar evaluación a un estudiante	Media
RF51	Listar las evaluaciones de un estudiante	Media
RF52	Eliminar evaluación a un estudiante	Media
RF53	Modificar evaluación a un estudiante	Media
Características del sistema		
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. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.	
RNF2	Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.	
RNF3	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.	
RNF4	Interfaces intuitivas: Potencialidades de capacitación orientadas a interfaces intuitivas, lo que enaltece la posibilidad de que el usuario aprenda mediante el uso y explotación de la herramienta.	
Fiabilidad		
RNF5	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.	

RNF6	Servicios web restringidos: los servicios web que brinde el sistema deben estar restringidos a grupos de usuarios definidos y aprobados previamente.
RNF7	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.
Interfaz	
RNF8	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.

2.8 Historias de usuario

Luego de estar definidos los requisitos funcionales que deberá tener la aplicación, se describen las historias de usuario (HU), las que son utilizadas en la metodología de desarrollo XP para especificar los requisitos del software. Teniendo en cuenta el esfuerzo asociado a las historias de usuario y las prioridades del cliente se define una versión que sea de valor para el cliente.

Tabla 2.3: Plantilla de historia de usuario a utilizar.

Historia de usuario	
Número: No. Historia de usuario	Nombre: Nombre de la historia de usuario que sirve para identificarla mejor entre los desarrolladores y el cliente.
Prioridad en el negocio: Importancia de la historia para el cliente: Alta / Media / Baja	Riesgo en desarrollo: Dificultad para el programador: Alta / Media / Baja
Puntos estimados: Estimación: de 1 a 3 puntos	Iteración asignada: Iteración a la que corresponde
Programador responsable: Persona responsable de la programación de la HU	
Descripción: Se especifican las operaciones por parte del usuario y las respuestas del sistema.	
Observaciones: Algunas observaciones de interés, como glosario, información sobre usuario, etc.	
Prototipo de interfaz: Se presentará un diseño de lo que va a ser la interfaz en el software de la historia de usuario.	

Tabla 2.4: Historia de usuario gestionar maestría

Historia de usuario	
Número: 1	Nombre: Gestionar Maestría
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: Primera
Programador responsable: Carlos Fernando Nápoles Ávila	
Descripción: La historia de usuario permite mostrar todas las maestrías, eliminarlas, agregar nuevas maestrías, buscarlas por su nombre y modificarlas. Las maestrías pueden ser ordenadas alfabéticamente por el nombre. Las opciones que se mostrarán serán: agregar, eliminar y modificar.	

Observaciones:

Para agregar o modificar una maestría se tendrán presente las siguientes observaciones:

- 1.- En caso que la maestría ya exista se muestra un mensaje de error “La maestría ya existe”.
- 2.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado.
- 3.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.

Prototipo de interfaz

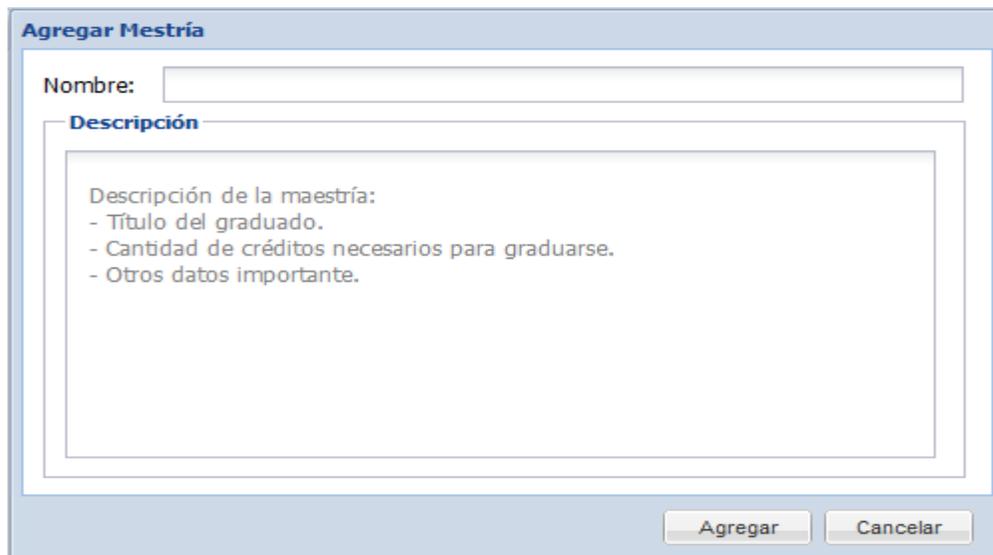
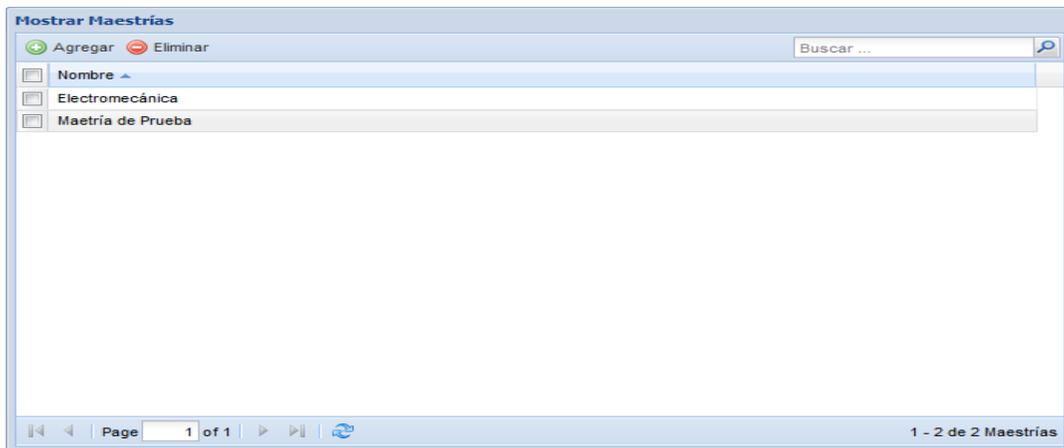


Tabla 2.5: Historia de usuario gestionar edición

Historia de usuario	
Número: 2	Nombre: Gestionar Edición
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: Carlos Fernando Nápoles Ávila	
<p>Descripción:</p> <p>La historia de usuario permite mostrar todas las ediciones, eliminarlas, agregar nuevas ediciones, buscarlas por su nombre y modificarlas.</p> <p>Las ediciones pueden ser ordenadas por su nombre.</p> <p>Las opciones que se mostrarán serán: agregar, eliminar, buscar y modificar.</p>	
<p>Observaciones:</p> <p>Para agregar o modificar una edición se tendrán presente las siguientes observaciones:</p> <ol style="list-style-type: none"> 1.- En caso que la maestría ya exista se muestra un mensaje de error “La edición ya existe”. 2.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado. 3.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos. 	

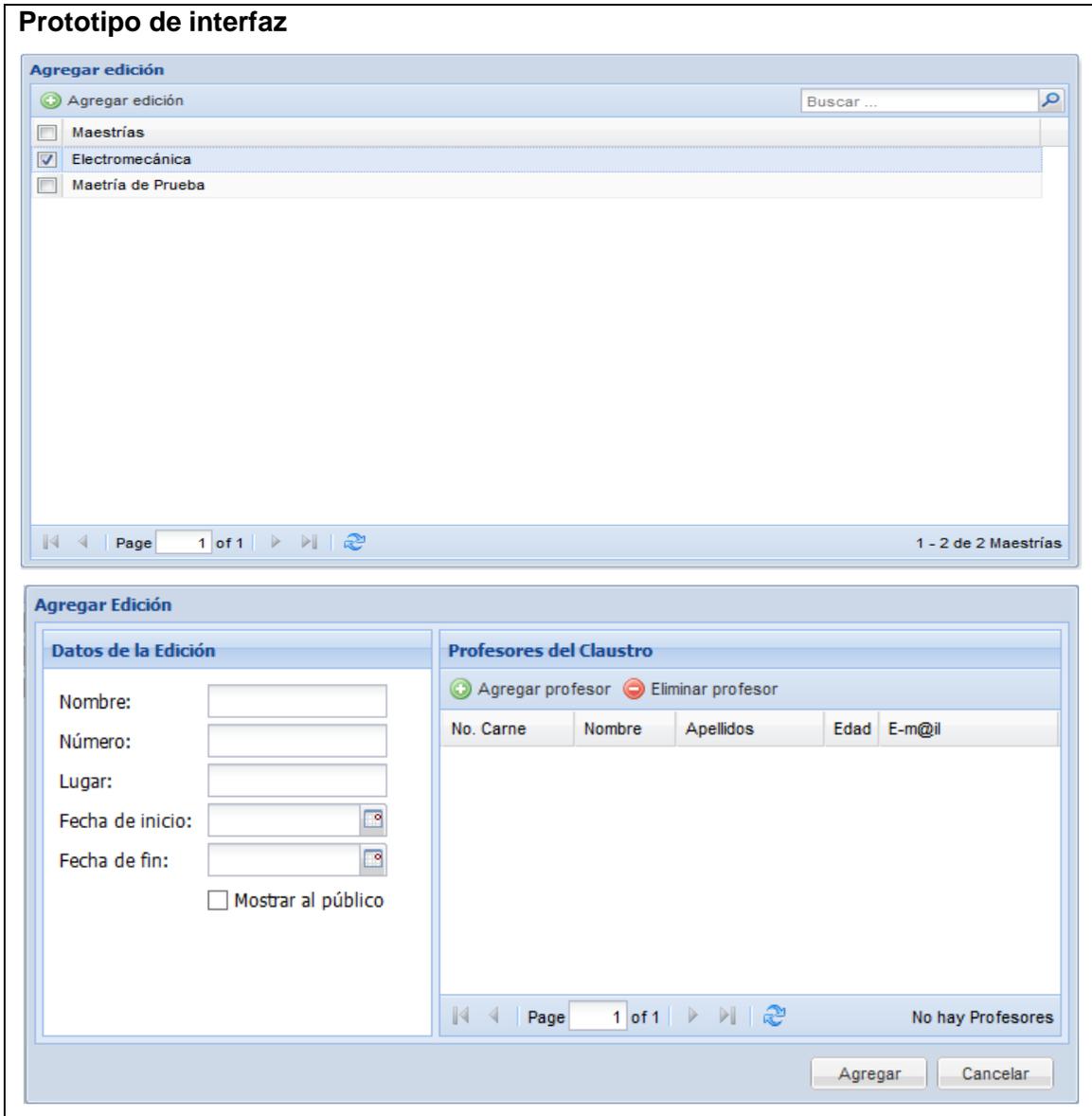


Tabla 2.6: Historia de Usuario Gestionar Profesor

Historia de usuario	
Número: 4	Nombre: Gestionar Profesor
Prioridad en el negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: Carlos Fernando Nápoles Ávila	
Descripción: La historia de usuario permite mostrar todos los profesores, eliminarlos, agregar nuevos profesores, buscarlos por su nombre y modificar sus datos.	

Los profesores pueden ser ordenados por su nombre.

Las opciones que se mostrarán serán: agregar, eliminar, buscar y modificar.

Observaciones:

Para agregar o modificar un profesor se tendrán presente las siguientes observaciones:

- 1.- En caso que el profesor ya exista se muestra un mensaje de error “El profesor ya existe”.
- 2.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado.
- 3.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.

Prototipo de interfaz

No. Carne	Nombre	Apellidos	Edad	E-m@il
-----------	--------	-----------	------	--------

Page 1 of 1 | No hay Profesores

Agregar Cancelar

No. Identidad: Em@il:

Nombre: Grado Cient:

Apellidos: Categoría:

País: Especialidad:

Edad: Teléfono:

Sexo:

Agregar Cancelar

Tabla 2.7: Historia de Usuario Gestionar Estudiante

Historia de usuario	
Número: 5	Nombre: Gestionar Estudiante
Prioridad en el negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: Carlos Fernando Nápoles Ávila	
<p>Descripción:</p> <p>La historia de usuario permite mostrar todos los estudiantes, eliminarlos, agregar nuevos estudiantes, buscarlos por su nombre y modificar sus datos.</p> <p>Los estudiantes pueden ser ordenados por su nombre o por el nombre de la maestría en que participan.</p> <p>Las opciones que se mostrarán serán: agregar, eliminar, buscar y modificar.</p>	
<p>Observaciones:</p> <p>Para agregar o modificar un estudiante se tendrán presente las siguientes observaciones:</p> <ol style="list-style-type: none"> 1.- En caso que el estudiante ya exista se muestra un mensaje de error “El estudiante ya existe”. 2.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado. 	

Las demás historias de usuario están en el Anexo 1.

2.9 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.

2.9.1 Estimación de esfuerzo por Historia de Usuario

Tabla 2.8: Estimación de esfuerzo por Historia de Usuario

Historias de Usuario	Puntos de Estimación (semanas)
Gestionar Maestría	1
Gestionar Edición	2
Gestionar Asignatura	1
Gestionar Profesor	2
Gestionar Estudiante	1
Gestionar Material de Asignatura	1
Gestionar Usuario	1
Gestionar Módulo	2
Gestionar Especialidad	1
Administrar Solicitud	1
Gestionar Evaluación	1
Administrar Claustro	1
Total	15 semanas

2.9.2 Plan de iteraciones

Después de realizar un análisis de las historias de usuarios y la estimación del esfuerzo propuesto para la realización de las mismas se realizó el siguiente plan de iteración, quedando definida la duración de cada una de estas.

Tabla 2.9: Plan de 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 van a implementar las HU que sean de prioridad "Alta" para el proyecto.	Gestionar Maestría	6	42 días
		Gestionar Edición	12	
		Gestionar Módulo	12	
		Gestionar Asignatura	6	
		Gestionar Material de Asignatura	6	
Segunda	En esta iteración se van a implementar las HU que sean de prioridad "Media" para el	Gestionar Estudiante	6	30 días
		Gestionar Profesor	12	
		Gestionar Especialidad	6	
		Administrar Claustro	6	
Tercera	En esta iteración se van a implementar las HU que sean de prioridad "Baja" para el proyecto.	Gestionar Usuario	6	18 días
		Administrar Solicitud	6	
		Gestionar Evaluación	6	

Total			90 días	15 semanas
-------	--	--	---------	------------

2.10 Tarjetas CRC (Clase, Responsabilidad, Colaboración)

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

Edición	
Descripción: Guarda los datos de la edición	
Atributos:	
Nombre	Descripción
id_edicion	Identificador de la edición
id_maestria	
nombre_edicion	
numero_edicion	
lugar_edicion	
matricula_inicio	
matricula_fin	
total_creditos	
fecha_inicio	
fecha_fin	
Responsabilidades:	
Nombre	Colaborador
crear_edicion	Modalidad, Maestria
mostrar_edicion	Modalidad, Maestria
modificar_edicion	Modalidad, Maestria
eliminar_edicion	Modalidad, Maestria
buscar_edicion	Modalidad, Maestria

Fig. 2.2: Tarjeta CRC Edición

Estudiante	
Descripción: Guarda los datos del estudiante	
Atributos:	
Nombre	Descripción
identidad	Identificador de estudiante
id_edicion	
Responsabilidades:	
Nombre	Collaborator
crear_estudiante	Persona, Edición
mostrar_estudinte	Persona, Edición
modificar_estudiante	Persona, Edición
eliminar_estudiante	Persona, Edición
buscar_estudiante	

Fig. 2.3: Tarjeta CRC Estudiante

Maestría	
Descripción: Guarda información de la maestría	
Atributos:	
Nombre	Descripción
id_maestria	Identificador de la maestría
nombre_maestria	
descripcion_maestria	
Responsabilidades:	
Nombre	Collaborator
crear_maestria	
mostrar_maestria	
modificar_maestria	
eliminar_maestria	
buscar_maestria	

Fig. 2.4: Tarjeta CRC Maestría

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

Fig. 2.5: Tarjeta CRC Persona

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

Fig. 2.6: Tarjeta CRC Profesor

Asignatura	
Descripción: Guarda los datos de la asignatura	
Atributos:	
Nombre	Descripción
id_asignatura	Identificador de la asignatura
id_modulo	
identidad	
nombre_asig	
cant_creditos	
cant_hrs_pres	
cant_hrs_no_pres	
fecha_inicio	
fecha_fin	
Responsabilidades:	
Nombre	Collaborator
crear_asignatura	Módulo, Profesor
mostrar_asignatura	Módulo, Profesor
modificar_asignatura	Módulo, Profesor
eliminar_asignatura	Módulo, Profesor
buscar_asignatura	Módulo

Fig. 2.7: Tarjeta CRC Asignatura

Para consultar el resto de las tarjetas CRC ir al Anexo 2

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: Implementación y Pruebas

3.1 Introducción

En este capítulo se inicia la fase de desarrollo y pruebas conforme a la metodología XP. Se presenta el modelo de datos empleado para la aplicación concluyente, y se realiza el desarrollo de las iteraciones a partir del desglose de las historias de usuario en tareas.

3.2 Patrones de diseño

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. (6)

Objetivos de los patrones de diseño (6)

- Formalizar un vocabulario común entre diseñadores
- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Estandarizar el modo en que se realiza el diseño.

Para el diseño del módulo se tuvieron en cuenta los cinco patrones GRASP: experto, creador, bajo acoplamiento, alta cohesión y controlador también se utilizó el patrón GOF “fachada”. El marco de trabajo utilizado en el desarrollo del módulo busca un máximo rendimiento y flexibilidad en sus soluciones y pone en práctica estos patrones para lograr un sistema reusable y flexible.

Los patrones GRASP acrónimo de General Responsibility Assignment Software Patterns (Patrones de Software para la asignación General de Responsabilidad),

describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades.

Bajo acoplamiento: Este patrón permitió que en el sistema se tuvieran pocos componentes que dependieran de otros, probando que mientras menos acoplamiento haya, más reusable y flexible se vuelve el sistema. El patrón establece que debe haber pocas dependencias entre las clases. Si todas las clases dependen de todas ¿cuánto software se puede extraer de un modo independiente y reutilizarlo en otro proyecto? Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Siempre hay que considerar las ventajas de la delegación respecto de la herencia.

Problema: ¿Cómo dar soporte a las bajas dependencias y al incremento de la reutilización? (17)

Solución: Diseñar con el objetivo de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. (10)

Patrón experto: La utilización de este patrón permitió la asignación a las clases de la información necesaria para cumplir sus responsabilidades, lo que conllevó a que el módulo fuera más fácil de entender, mantener y ampliar. La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Hay que tener en cuenta que esto es aplicable mientras se consideran los mismos aspectos del sistema:

- Lógica de negocio
- Persistencia a la base de datos
- Interfaz de usuario

Problema: ¿De qué forma podemos saber qué responsabilidad delegar a cada objeto? (10)

Solución: Asignar una responsabilidad al experto en información; la clase que tiene la información necesaria para llevar a cabo la responsabilidad. (10)

Patrón alta cohesión: Este patrón permitió que a cada clase se le asignara la responsabilidad de realizar labores específicas en el sistema y que no tuvieran información que no estuviesen estrechamente relacionadas con la clase. Ejemplo de una baja cohesión es una clase que hace demasiadas actividades incluyendo las que no debería de realizar porque no están muy relacionadas con esta. En todas las metodologías se considera la refactorización. Ejemplo de elementos a los que se le debe realizar la refactorización son las clases saturadas de métodos.

Problema: ¿Cómo mantener manejable la complejidad? (10)

Solución: Asignar responsabilidades de manera que la información que almacena una clase sea coherente y esté relacionada con la clase. (10)

Patrón creador: Este patrón guió la asignación de responsabilidades de la creación de objetos, el propósito fundamental del mismo es encontrar un creador que deba conectar con el objeto producido en cualquier evento. Se asigna la responsabilidad de que una clase B cree un objeto de la clase C solamente cuando:

1. B contiene a C.
2. B es una agregación (o composición) de C.
3. B almacena a C.
4. B tiene los datos de inicialización de C (datos que requiere su constructor).
5. B usa a C.

La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asigna bien el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

Problema: ¿Quién debería ser responsable de crear una nueva instancia? (10)

Solución (10): Crear una nueva instancia por la clase que:

- Tiene la información necesaria para realizar la creación del objeto.
- Usa directamente las instancias creadas del objeto.
- Almacena o maneja varias instancias de la clase.
- Contiene o agrega la clase.

Patrón controlador: Este patrón ayudó a asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas ya que esta es la función del patrón controlador, esto facilitó la centralización de actividades como las validaciones y seguridad, el controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema.

Problema: ¿Quién gestiona un evento del sistema? (10)

Solución (10): Asignar la responsabilidad de gestionar un mensaje de un evento del sistema a una clase que represente una de estas dos opciones:

- 1- Representa el sistema global, dispositivo o subsistema (controlador de fachada).
- 2- Representa un escenario de caso de uso en el que tiene lugar el evento del sistema (controlador de caso de uso o de sesión).

Patrón GOF “fachada”: Establece proveer el acceso de una interfaz unificada y simple a una interfaz o grupo de interfaces de un subsistema.

3.3 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.

Entre las principales características de los sistemas de base de datos se pueden mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.

- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

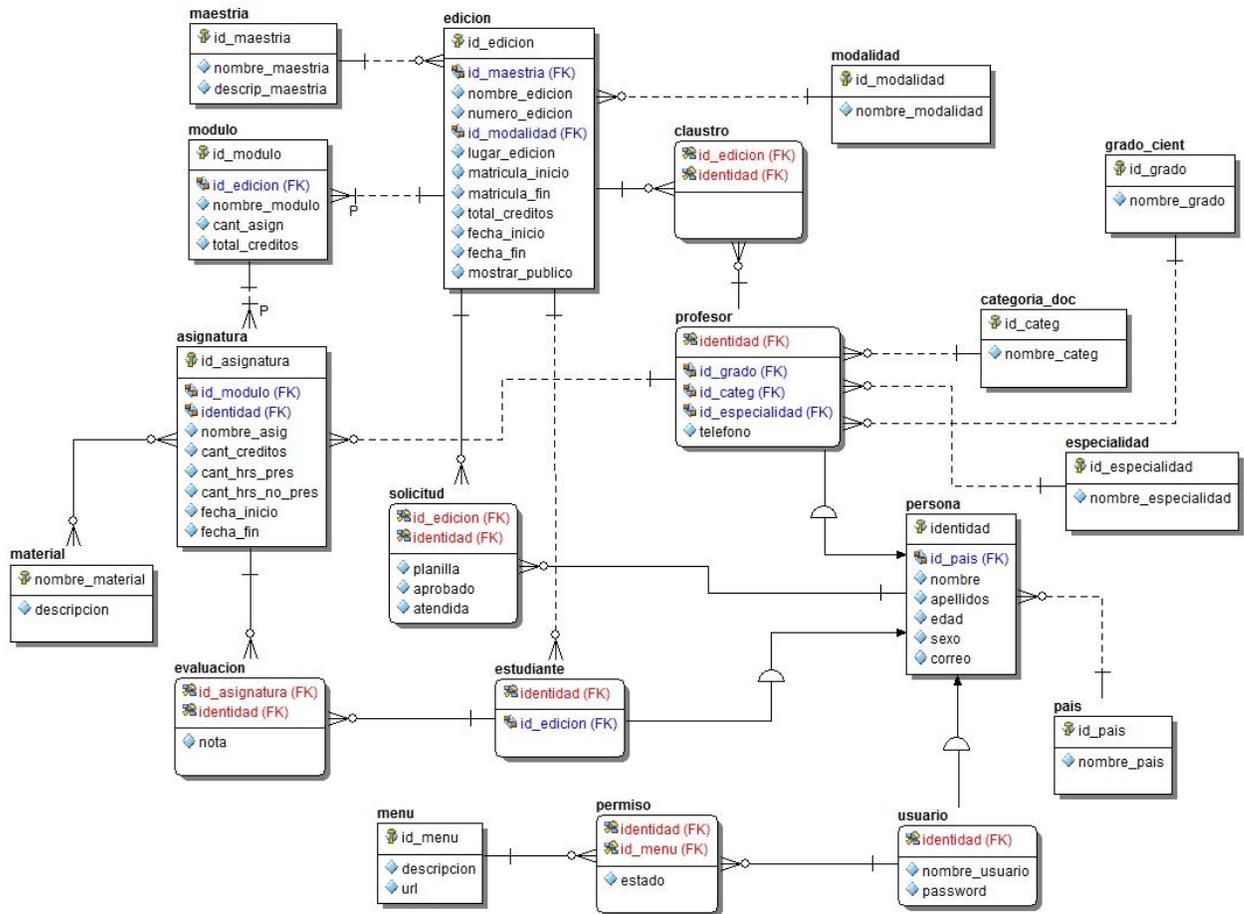


Figura 3.1: Modelo de la base de datos

3.4 Desarrollo de las iteraciones

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

3.4.1 Tareas de ingeniería por Historia de Usuario

Tabla 3.1 Tareas de ingeniería por Historia de Usuario

Historia de Usuario	Tarea
---------------------	-------

Gestionar Maestría	<ul style="list-style-type: none">✓ Agregar maestría.✓ Listar maestrías.✓ Buscar maestría.✓ Modificar maestría.✓ Eliminar maestría.
Gestionar Edición	<ul style="list-style-type: none">✓ Agregar edición.✓ Listar edición.✓ Buscar edición.✓ Modificar edición.✓ Eliminar edición.
Gestionar Módulo	<ul style="list-style-type: none">✓ Agregar módulo.✓ Listar módulo.✓ Modificar módulo.✓ Eliminar módulo.
Gestionar Asignatura	<ul style="list-style-type: none">✓ Agregar asignatura.✓ Listar asignatura.✓ Buscar asignatura.✓ Modificar asignatura.✓ Eliminar asignatura.
Gestionar Material de Asignatura	<ul style="list-style-type: none">✓ Agregar material de asignatura.✓ Listar material de asignatura.✓ Buscar material de asignatura.

	<ul style="list-style-type: none"> ✓ Modificar material de asignatura. ✓ Eliminar material de asignatura.
Gestionar Estudiante	<ul style="list-style-type: none"> ✓ Agregar estudiante. ✓ Listar estudiante. ✓ Buscar estudiante. ✓ Modificar estudiante. ✓ Eliminar estudiante.
Gestionar Profesor	<ul style="list-style-type: none"> ✓ Agregar profesor. ✓ Listar profesor. ✓ Buscar profesor. ✓ Modificar profesor. ✓ Eliminar profesor.
Gestionar Especialidad	<ul style="list-style-type: none"> ✓ Agregar especialidad. ✓ Listar especialidad. ✓ Buscar especialidad. ✓ Modificar especialidad. ✓ Eliminar especialidad.
Gestionar Evaluación	<ul style="list-style-type: none"> ✓ Agregar evaluación. ✓ Listar evaluación. ✓ Modificar evaluación. ✓ Eliminar evaluación.
Administrar Claustro	<ul style="list-style-type: none"> ✓ Agregar claustro

	✓ Eliminar claustro
Gestionar Usuario	✓ Agregar usuario. ✓ Listar usuario. ✓ Buscar usuario. ✓ Modificar usuario. ✓ Eliminar usuario.
Gestionar Solicitud	✓ Agregar solicitud. ✓ Listar solicitud. ✓ Buscar solicitud. ✓ Eliminar solicitud.

3.5 Pruebas

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

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

3.5.1 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.

3.5.2 Pruebas de aceptación (PA)

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

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

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

Tabla 3.2 Modelo para pruebas de aceptación.

Prueba de Aceptación: Nombre de la prueba de aceptación
Historia de Usuario: Nombre de la Historia de Usuario que se va a comprobar.
Nombre: Nombre del caso de prueba.

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

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

Tabla 3.3 Prueba de aceptación para la Historia de Usuario Gestionar Maestría.

Prueba de Aceptación: Nombre de la prueba de aceptación
Historia de Usuario: Gestionar Maestría.
Nombre: Prueba para comprobar la gestión de las maestrías.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de las maestrías.
Condiciones de ejecución: El coordinador debe iniciar su sesión para acceder al módulo de gestión de maestrías y poder adicionar, eliminar, mostrar, buscar por nombre o modificar una maestría.
Entrada / Pasos ejecución: El coordinador hace clic en el botón “Agregar” de la interfaz Mostrar Maestrías y en el formulario Agregar Maestría inserta los datos de la maestría que desea agregar, luego presiona el botón “Agregar” y es creado una nueva maestría. En seguida que la maestría es creada puede ser modificada o eliminada.
Resultado: Se insertan correctamente las maestrías. Se visualizan las maestrías correctamente. Se pueden modificar o eliminar una vez insertados.
Evaluación de la prueba: Aceptada.

Tabla 3.4 Prueba de aceptación para la Historia de Usuario Gestionar Edición.

Prueba de Aceptación: Gestionar Edición
Historia de Usuario: Gestionar Edición.
Nombre: Prueba para comprobar la gestión de las ediciones.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de las ediciones.
Condiciones de ejecución: El coordinador debe iniciar su sesión para acceder al módulo de gestión de edición y poder adicionar, eliminar, mostrar, buscar por nombre o modificar una edición.
Entrada / Pasos ejecución: El coordinador selecciona la maestría a la cual se le agregará la edición y hace clic en el botón “Agregar edición”, en el formulario Agregar Edición inserta los datos de la edición que desea agregar, luego presiona el botón “Agregar” y es creada una nueva edición. En seguida que la edición es creada puede ser modificada o eliminada.
Resultado: Se insertan correctamente las ediciones. Se visualizan las ediciones correctamente. Se pueden modificar o eliminar una vez agregadas.
Evaluación de la prueba: Aceptada.

Tabla 3.5 Prueba de aceptación para la Historia de Usuario Gestionar Módulo.

Prueba de Aceptación: Gestionar Módulo
Historia de Usuario: Gestionar Módulo.
Nombre: Prueba para comprobar la gestión de los módulos.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de los módulos.
Condiciones de ejecución: El coordinador debe iniciar su sesión para acceder al módulo de gestión de módulos y poder adicionar, eliminar, mostrar o modificar un módulo.

<p>Entrada / Pasos ejecución: El coordinador selecciona la edición de la maestría a la cual se le agregará el módulo. En la tabla “Módulos de la Edición” se muestran los módulos que actualmente tiene la edición, las opciones que se muestran son: agregar, modificar, eliminar módulo.</p>
<p>Resultado: Se insertan correctamente los módulos. Se visualizan los módulos correctamente. Se pueden modificar o eliminar una vez agregados.</p>
<p>Evaluación de la prueba: Aceptada.</p>

Tabla 3.6 Prueba de aceptación para la Historia de Usuario Gestionar Profesor.

Prueba de Aceptación: Gestionar Profesor
<p>Historia de Usuario: Gestionar Profesor.</p>
<p>Nombre: Prueba para comprobar la gestión de los profesores.</p>
<p>Descripción: Con esta prueba de aceptación se persigue validar la gestión de los profesores.</p>
<p>Condiciones de ejecución: El coordinador debe iniciar su sesión para acceder al módulo de gestión de asignaturas y poder adicionar, eliminar, mostrar o modificar un profesor.</p>
<p>Entrada / Pasos ejecución: El coordinador hace clic en el botón “Agregar” de la tabla “Mostrar Profesor” y en el formulario “Agregar Profesor” inserta los datos del profesor que desea agregar, luego presiona el botón “Agregar” y es insertado un nuevo profesor. En seguida que el profesor es agregado puede ser modificado o eliminado.</p>
<p>Resultado: Se insertan correctamente los profesores. Se visualizan los profesores correctamente. Se pueden modificar o eliminar una vez agregados.</p>
<p>Evaluación de la prueba: Aceptada.</p>

Tabla 3.7 Prueba de aceptación para la Historia de Usuario Gestionar Asignatura.

Prueba de Aceptación: Gestionar Asignatura
Historia de Usuario: Gestionar Asignatura.
Nombre: Prueba para comprobar la gestión de las asignaturas.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de las asignaturas.
Condiciones de ejecución: El coordinador debe iniciar su sesión para acceder al módulo de gestión de asignaturas y poder adicionar, eliminar, mostrar o modificar una asignatura.
Entrada / Pasos ejecución: El coordinador selecciona el módulo de la edición a la cual se le agregará la asignatura y hace clic en el botón “Gestionar Asignatura”. En la tabla “Mostrar Asignaturas” se muestran las asignaturas que actualmente tiene el módulo, las opciones que se muestran son: agregar, modificar, eliminar asignatura.
Resultado: Se insertan correctamente las asignaturas. Se visualizan las asignaturas correctamente. Se pueden modificar o eliminar una vez agregadas.
Evaluación de la prueba: Aceptada.

Para observar las demás Pruebas de Aceptación ir al Anexo 3.

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

Capítulo 4: Estudio de Factibilidad y Sostenibilidad

4.1 Introducción

En el presente capítulo se muestran los resultados obtenidos en el Estudio de Factibilidad del Proyecto. El mismo se realizó basado en la Metodología Costo Efectividad (Beneficio), la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

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

4.2 Efectos económicos

- Efectos directos
- Efectos indirectos
- Efectos externos
- Intangibles

Efectos directos:

Positivos:

El administrador y los usuarios del sistema (alumnos, profesores y el coordinador de las maestrías) tendrán la posibilidad de realizar de forma integrada sus actividades de matrícula y de acceso a la información docente y metodológica de las asignaturas a estudiar.

- Se establece un mecanismo eficaz del postgrado académico.

- Se obtiene información actualizada, resumida y detallada, del proceso docente de la maestría.

Negativos:

- Para el uso de esta aplicación implementada en plataforma Web se necesitará que la misma sea ejecutada preferentemente con el navegador Mozilla Firefox, porque es con el que se trabajó en la elaboración del producto por lo que el diseño está adaptado a este tipo de navegador.

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.

Efectos externos:

- Se obtendrá un producto disponible que le facilitará gran parte del trabajo a los Coordinadores y Profesores encargados de las maestrías.

Intangibles:

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

A fin de medir con precisión los efectos, deberán considerarse dos situaciones:

Situación sin el producto

Para llevar a cabo las maestrías debe tener en cuenta los siguientes pasos:

1. Los maestrantes deben presentarse y entregar varios documentos que son necesarios para ser aprobados en la maestría, estos pueden ser de distintos países.
2. Los maestrantes deben recibir las clases personalmente o en algunos casos por sistemas no presenciales.
3. El trabajo del coordinador se torna engorroso, debido al gran volumen de información que este maneja.

Situación con el producto

Para llevar a cabo la maestría se realizan los siguientes pasos:

1. Entrar los datos correspondientes del usuario ya sea Administrador, Coordinador, Profesor o Estudiante que hará uso de la aplicación.
2. Según el tipo de usuario podrá realizar operaciones específicas del Sistema.
3. Los maestrantes podrán enviar sus datos utilizando el sistema y una vez que el coordinador los revise podrá aprobar o no la solicitud del maestrante.
4. Los maestrantes encontrarán todos los documentos necesarios para poder realizar la maestría.
5. El trabajo del coordinador se facilitará porque tendrá toda la documentación digitalizada.

4.3 Beneficios y Costos Intangibles en el proyecto

Costos

- Resistencia al cambio.

Beneficios

- Más comodidad para los usuarios.
- Los procesos informativos de las maestrías se realizan con mayor rapidez y fiabilidad.
- Conectividad desde cualquier PC que esté conectada a la red.

4.4 Ficha de costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar una Ficha de Costo de un producto.

Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

4.4.1 Costos en Moneda Libremente Convertible:

Tabla 4.1: Costo en Moneda Libremente Convertible

Ficha de Costo.		
		Precio(s)
Costos Moneda Libremente Convertible		
Costos Directos		
Compra de equipos de cómputo		0,00
Alquiler de equipos de cómputo		0,00
Compra de licencia de Software		0,00
Depreciación de equipos		25,00
Materiales directos		0,00
	Subtotal	25,00
Costos Indirectos		
Formación del personal que elabora el proyecto		0,00
Gastos en llamadas telefónicas		0,00
Gastos para el mantenimiento del centro		0,00
Know How		0,00
Gastos en representación		0,00
	Subtotal	0,00
Gastos de Distribución y Venta		
Participación en ferias o exposiciones		0,00
Gastos en transportación		0,00
Compra de materiales de propagandas		0,00
	Subtotal	0,00

Total	25,00
--------------	-------

4.4.2 Costos en Moneda Nacional:

Tabla 4.4.2: Costo en Moneda Nacional.

Ficha de Costo.		
		Precio(s)
Costos Moneda Nacional		
Costos Directos		
Salario del personal que laborará en el proyecto		100,00
12,5% del total de gastos por salarios se dedica a la seguridad social		0,00
9.09% de salario total, por concepto de vacaciones a acumular		0,00
Gasto por consumo de energía eléctrica		270,70
Gastos en llamadas telefónicas		0,00
Gastos administrativos		0,00
Subtotal		370,70
Costos Indirectos		
Know How		0,00
Subtotal		
Total		370,70

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

que el costo para este proyecto es despreciable, se tomará como costo el tiempo en minutos empleado en la ejecución de algunas de las actividades propias del proceso.

Valores de la variable (Solución manual)

- 1.- Realizar solicitud. (25 min)
- 2.- Gestionar información de los estudiantes. (15 min)
- 3.- Obtener material de una asignatura. (10 min)
- 4.- Gestionar información de los profesores. (13)

Valores de la variable (Solución con el sistema)

- 1.- Realizar solicitud. (3 min)
- 2.- Gestionar información de los estudiantes. (2 min)
- 3.- Obtener material de una asignatura. (1 min)
- 4.- Gestionar información de los profesores. (2)

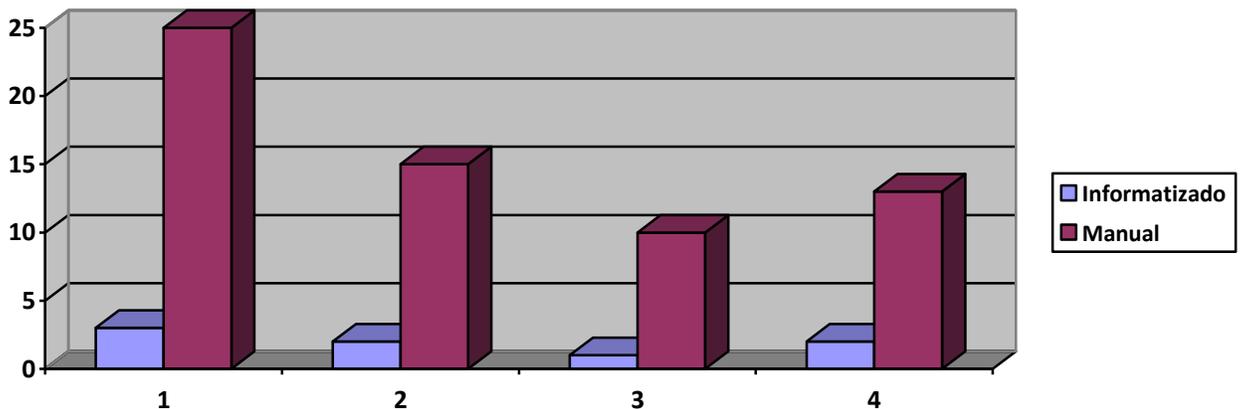


FIGURA 4.1 GRÁFICA DE LA SOLUCIÓN CON EL PRODUCTO Y SOLUCIÓN SIN EL PRODUCTO

Conclusiones

En este capítulo se realizó el estudio del costo real en que se incurrió durante el diseño e implementación de la aplicación del software mediante la Metodología Costo Efectividad (Beneficios), se analizaron todos los factores directos, indirectos, externos e intangibles, así como se calculó el costo de ejecución del producto software mediante la ficha de costo arrojando como resultados (**costo**) demostrándose la conveniencia de la elaboración del sistema.

Conclusiones Generales

La realización del sistema de gestión de maestrías deviene en favorables resultados que pueden concluirse de la siguiente forma:

Se logró establecer los elementos teóricos del proceso de gestión de maestrías, luego de realizar un estudio del proceso de estas en el ISMMM, esto permitió que quedara identificada la situación problémica existente y las bases para comenzar con el diseño del futuro sistema. Luego se realizó el levantamiento de requisitos funcionales y no funcionales del sistema mediante la Lista de Reserva del Producto, se conformaron todas las historias de usuario y se realizó el diseño de la futura aplicación, lo que permitió establecer todo lo necesario para su futura construcción. También se realizó el diseño del esquema de base de datos del sistema. Se construyó el sistema a partir del diseño del esquema de base de datos del mismo y la implementación de sus historias de usuario, dando cumplimiento al objetivo general del trabajo de diploma. Se comprobó que las funcionalidades descritas, satisfacen las necesidades del cliente en cuanto al proceso de gestión de maestrías, partiendo del diseño y ejecución de los casos de prueba, obteniéndose los resultados esperados.

Recomendaciones

Se recomienda:

- ✓ Extender el software a cualquier centro de educación superior de nuestro país.
- ✓ Para futuras versiones del software agregar un foro para la interacción entre profesores y estudiantes
- ✓ Continuar el estudio de los diferentes procesos que conforman la gestión de las maestrías en la universidad con el objetivo de añadir nuevas funcionalidades al sistema.

Referencias Bibliográficas

1. **A., Ernesto Quiñones.** *Introducción a PostgresSQL.*
2. Cubano, P.E. 2009 [cited 2010; Available from: http://www.rimed.cu/index.php?option=com_content&view=article&id=4034:la-nueva-universidad-en-informatica-2009-&catid=343&Itemid=184.
3. *Developer.mozilla.org.* (s.f.). Obtenido de https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript?redirectlocale=en-US&redirectslug=JavaScript%2FAbout_JavaScript
4. *El lenguaje HTML.* (2013). Obtenido de <http://www.ri5.com.ar/ayuda03.php>
5. [En línea] http://netbeans.org/index_es.html.
6. [En línea] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
7. **G. Booch, J. R. Addison Wesley.** 1999. *El Lenguaje Unificado de Modelado*
8. *IEEE Std, IEEE Software Engineering Standard: Glossary of Software Engineering Terminology.* IEEE Computer Society Press. 1993.
9. *Hypertext Transfer Protocol.* (2012). Obtenido de <https://datatracker.ietf.org/wg/httpbis/charter/>
10. **Larman, C.** *UML y Patrones.* Ciudad de la Habana : Editorial Universitaria.
11. **Marc Gibert Ginestà, Oscar Pérez Mora.** *Bases de datos en PostgreSQL, P06/M2109/02152.*
12. **Peñalver, G., Meneses, A., García, S.** *SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE.* Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba : s.n.
13. **POPKIN, S. Y. S.** *Modelado de Sistemas con UML.*
14. **PRESSMAN, R. S.** *Ingeniería del Software. Un enfoque práctico.,* 2005. p.
15. Rodríguez, H., *El movimiento del Software Libre en Cuba,* in *Revista Cubana de Ciencias Informáticas.* 2009. p. 86.
16. **Roxana Giandini, Gabriela Pérez, Claudia Pons.** *Un lenguaje de Transformación específico para Modelos de Proceso del Negocio.*
17. **SENN, J. A.** *Análisis y Diseño de Sistemas de Información.* Segunda. 1992. p.
18. **Torres, I. J. (s.f).** *IEEE (Especificación de requisitos de Ingeniería de Software).* <http://www.ewh.ieee.org/r9/guadalajara/boletin/sep01/requerimientos.htm>
19. *Trasweb Framework php.* (2014). Obtenido de <http://framework.latrasweb.net/>

Referencias Bibliográficas

20. Visual Paradigm. (s.f.). <http://www.visual-paradigm.com/>
21. Villagra, Sergio. (2004). *Diseño de una Metodología Ágil de Desarrollo de Software*. Facultad de Ingeniería. Buenos Aires (Universidad).

Bibliografía

1. **A., Ernesto Quiñones.** *Introducción a PostgreSQL.*
2. ALIAGA, Antonio; Miani, Marcos. Trabajo investigativo sobre postgresQL. (2008). [en línea]. Disponible en: www.iessanvicente.com/colaboraciones/postgreSQL.pdf
3. **ALVAREZ, Miguel Angel.** Revista Digital: Desarrolloweb.com. Manual de CSS, hojas de estilo. [en línea] (2000) Disponible en:
<http://www.desarrolloweb.com/manuales/manual-css-hojas-de-estilo.html>.
4. CASTILLO, A. D. (2007). "Manual PHP. ¿Qué es PHP? una breve introducción." Disponible en: <http://www.lawebera.es/manuales/php/1.php>.
5. Cubano, P.E. 2009 [cited 2010; Available from: http://www.rimed.cu/index.php?option=com_content&view=article&id=4034:la-nueva-universidad-en-informatica-2009-&catid=343&Itemid=184.
6. *Developer.mozilla.org.* (s.f.). Obtenido de https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript?redirectlocale=en-US&redirectslug=JavaScript%2FAbout_JavaScript
7. *El lenguaje HTML.* (2013). Obtenido de <http://www.ri5.com.ar/ayuda03.php>
8. [En línea] http://netbeans.org/index_es.html.
9. [En línea] <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>.
10. **G. Booch, J. R. Addison Wesley.** 1999. *El Lenguaje Unificado de Modelado*
11. *IEEE Std, IEEE Software Engineering Standard: Glossary of Software Engineering Terminology.* IEEE Computer Society Press. 1993.
12. *Hypertext Transfer Protocol.* (2012). Obtenido de <https://datatracker.ietf.org/wg/httpbis/charter/>
13. **Larman, C.** *UML y Patrones.* Ciudad de la Habana : Editorial Universitaria.
14. **Marc Gibert Ginestà, Oscar Pérez Mora.** *Bases de datos en PostgreSQL, P06/M2109/02152.*
15. **Peñalver, G., Meneses, A., García, S.** *SXP, METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE.* Universidad de las Ciencias Informáticas, Ciudad de La Habana, Cuba : s.n.
16. **POPKIN, S. Y. S.** *Modelado de Sistemas con UML.*
17. **PRESSMAN, R. S.** *Ingeniería del Software. Un enfoque práctico., 2005. p.*

18. RATIONAL ROSE ENTERPRISE RATIONAL.COM [en línea] Disponible en:
<http://www.rational.com.ar/herramientas/roseenterprise.html>
19. Rodríguez, H., *El movimiento del Software Libre en Cuba*, in *Revista Cubana de Ciencias Informáticas*. 2009. p. 86.
20. **Roxana Giandini, Gabriela Pérez, Claudia Pons.** *Un lenguaje de Transformación específico para Modelos de Proceso del Negocio*.
21. SCHENONE, Marcelo; Villagra, Sergio. (2004). *Diseño de una Metodología Ágil de Desarrollo de Software*. Facultad de Ingeniería. Buenos Aires (Universidad)
22. **SENN, J. A.** *Análisis y Diseño de Sistemas de Información*. Segunda. 1992. p.
23. SOLÍS, M. C. (2003). "Una explicación de la programación extrema (XP)." from
<http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/explicaxp.pdf>.
24. **Torres, I. J. (s.f.).** *IEEE (Especificación de requisitos de Ingeniería de Software)*.
<http://www.ewh.ieee.org/r9/guadalajara/boletin/sep01/requerimientos.htm>
25. *Trasweb Framework php*. (2014). Obtenido de <http://framework.latrasweb.net/>
26. *Visual Paradigm*. (s.f.). <http://www.visual-paradigm.com/>
27. Villagra, Sergio. (2004). *Diseño de una Metodología Ágil de Desarrollo de Software*. Facultad de Ingeniería. Buenos Aires (Universidad).

Anexos

Anexo 1 Historias de Usuario

Tabla 2.10: Historia de usuario gestionar asignatura

Historia de usuario	
Número: 3	Nombre: Gestionar Asignatura
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: Carlos Fernando Nápoles Ávila	
<p>Descripción:</p> <p>La historia de usuario permite mostrar todas las asignaturas, eliminarlas, agregar nuevas asignaturas, buscarlas por su nombre y modificarlas.</p> <p>Las asignaturas pueden ser ordenadas por el nombre.</p> <p>Las opciones que se mostrarán serán: agregar, eliminar y modificar.</p>	
<p>Observaciones:</p> <p>Para agregar o modificar una asignatura se tendrán presente las siguientes observaciones:</p> <ol style="list-style-type: none"> 1.- En caso que la asignatura ya exista se muestra un mensaje de error “La asignatura ya existe”. 2.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado. 3.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos. 	

Tabla 2.11: Historia de usuario gestionar material de asignatura

Historia de usuario	
Número: 6	Nombre: Gestionar Material de Asignatura
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto

Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: Carlos Fernando Nápoles Ávila	
Descripción: La historia de usuario permite mostrar todos los materiales, eliminarlos, agregar nuevos materiales y buscarlos por su nombre. Los materiales pueden ser ordenados por su nombre. Las opciones que se mostrarán serán: agregar, eliminar y buscar.	
Observaciones: Para agregar un material se tendrán presente las siguientes observaciones: 1.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: "Ha dejado campos obligatorios vacíos" y se muestra en rojo el campo que debe ser llenado. 2.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.	

Tabla 2.12: Historia de usuario gestionar usuario

Historia de usuario	
Número: 7	Nombre: Gestionar Usuario
Prioridad en el negocio: Baja	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: Tercera
Programador responsable: Carlos Fernando Nápoles Ávila	
Descripción: La historia de usuario permite mostrar todos los usuarios, eliminarlos, agregar nuevos usuarios, buscarlos por su nombre y modificar sus datos. Los usuarios pueden ser ordenados por su nombre. Las opciones que se mostrarán serán: agregar, eliminar, buscar y modificar.	
Observaciones: Para agregar un usuario se tendrán presente las siguientes observaciones: 1.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de	

error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado.

2.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.

Tabla 2.13: Historia de usuario gestionar módulo

Historia de usuario	
Número: 8	Nombre: Gestionar Módulo
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: Carlos Fernando Nápoles Ávila	
<p>Descripción:</p> <p>La historia de usuario permite mostrar todos los módulos de una edición, eliminarlos, agregar nuevos módulos y modificar sus datos.</p> <p>Los módulos pueden ser ordenados por su nombre.</p> <p>Las opciones que se mostrarán serán: agregar, eliminar y modificar.</p>	
<p>Observaciones:</p> <p>Para agregar un módulo se tendrán presente las siguientes observaciones:</p> <p>1.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado.</p> <p>2.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.</p>	

Tabla 2.14: Historia de usuario gestionar especialidad

Historia de usuario	
Número: 10	Nombre: Gestionar Especialidad
Prioridad en el negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: Segunda

Programador responsable: Carlos Fernando Nápoles Ávila
<p>Descripción:</p> <p>La historia de usuario permite mostrar todas las especialidades, eliminarlas, agregar nuevas especialidades y modificar sus datos.</p> <p>Las especialidades pueden ser ordenadas por su nombre.</p> <p>Las opciones que se mostrarán serán: agregar, eliminar, buscar y modificar.</p>
<p>Observaciones:</p> <p>Para agregar una especialidad se tendrán presente las siguientes observaciones:</p> <p>1.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado.</p> <p>2.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.</p>

Tabla 2.15: Historia de usuario gestionar solicitud

Historia de usuario	
Número: 11	Nombre: Gestionar Solicitud
Prioridad en el negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: Carlos Fernando Nápoles Ávila	
<p>Descripción:</p> <p>La historia de usuario permite agregar, listar, buscar y eliminar una solicitud.</p> <p>Las opciones que se mostrarán serán: agregar, buscar y eliminar.</p>	
<p>Observaciones:</p> <p>Para agregar una solicitud se tendrán presente las siguientes observaciones:</p> <p>1.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: “Ha dejado campos obligatorios vacíos” y se muestra en rojo el campo que debe ser llenado.</p>	

2.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.

Tabla 2.16: Historia de usuario gestionar evaluación

Historia de usuario	
Número: 13	Nombre: Gestionar Evaluación
Prioridad en el negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: Carlos Fernando Nápoles Ávila	
Descripción: La historia de usuario permite agregar, listar, modificar y eliminar una evaluación. Las opciones que se mostrarán serán: agregar, modificar y eliminar.	
Observaciones: Para agregar una evaluación se tendrán presente las siguientes observaciones: 1.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: "Ha dejado campos obligatorios vacíos" y se muestra en rojo el campo que debe ser llenado. 2.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.	

Tabla 2.17: Historia de usuario administrar claustro

Historia de usuario	
Número: 14	Nombre: Administrar Claustro
Prioridad en el negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: Carlos Fernando Nápoles Ávila	
Descripción:	

La historia de usuario permite agregar, listar y eliminar un claustro.
 Las opciones que se mostrarán serán: agregar y eliminar.

Observaciones:

Para agregar un claustro se tendrán presente las siguientes observaciones:

- 1.- En caso que se dejen campos obligatorios vacío se muestra un mensaje de error: "Ha dejado campos obligatorios vacíos" y se muestra en rojo el campo que debe ser llenado.
- 2.- En caso de que se introduzcan datos incorrectos se marcan en rojo los campos que tienen valores incorrectos.

Anexo 2 Tarjetas CRC

Categoría_Docente	
Descripción: Guarda los datos de la categoría docente	
Atributos:	
Nombre	Descripción
id_categ	
nombre_categ	
Responsabilidades:	
Nombre	Collaborator
listar_categ_doc	

Fig. 2.8: Tarjeta CRC Categoría Docente

Claustro	
Descripción: Guarda los datos del claustro	
Atributos:	
Nombre	Descripción
id_edicion	
id_identidad	
Responsabilidades:	
Nombre	Collaborator
crear_claustro	Edición, Profesor
eliminar_claustro	Edición, Profesor

Fig. 2.9: Tarjeta CRC Claustro

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

Fig. 2.10: Tarjeta CRC Especialidad

Evaluación	
Descripción: Guardar los datos de evaluación	
Atributes:	
Nombre	Descripción
id_asignatura	
identidad	
id_tipo_eval	
nombre_evaluacion	
nota	
Responsabilidades:	
Nombre	Collaborator
crear_evaluacion	Estudiante, Asignatura
Listar_evaluacion	Estudiante, Asignatura
modificar_evaluacion	Estudiante, Asignatura
eliminar_evaluacion	Estudiante, Asignatura

Fig. 2.11: Tarjeta CRC Evaluación

Grado_Científico	
Descripción: Guarda los datos del grado científico	
Atributes:	
Nombre	Descripción
id_grado	Identificador del grado científico
nombre_grado	
Responsabilidades:	
Nombre	Collaborator
listar_grado_cient	

Fig. 2.12: Tarjeta CRC Grado Científico

Material	
Descripción: Guarda información del material	
Atributos:	
Nombre	Descripción
id_material	Identificador del material
descripcion	
nombre_material	
Responsabilidades:	
Nombre	Collaborator
crear_material	
listar_material	
eliminar_material	

Fig. 2.13: Tarjeta CRC Material

Menú	
Descripción: Guarda los datos del menú	
Atributos:	
Nombre	Descripción
id_menu	Identificador del menú
descripcion	
url	
Responsabilidades:	
Nombre	Collaborator
listar_menu	

Fig. 2.14: Tarjeta CRC Menú

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

Fig. 2.15: Tarjeta CRC Modalidad

País	
Descripción: Guarda los datos de país	
Atributes:	
Nombre	Descripción
id_pais	Identificador del país
nombre_pais	
Responsabilidades:	
Nombre	Collaborator
listar_pais	

Fig. 2.16: Tarjeta CRC País

Permiso	
Descripción: Guarda información del permiso	
Atributos:	
Nombre	Descripción
identidad	
id_menu	
estado	
Responsabilidades:	
Nombre	Collaborator
crear_permiso	Usuario, Menú
listar_permiso	Usuario, Menú
modificar_permiso	Usuario, Menú

Fig. 2.17: Tarjeta CRC Permiso

Solicitud	
Descripción: Guarda información de la solicitud	
Atributos:	
Nombre	Descripción
id_solicitud	Identificador de la solicitud
identidad	
id_edicion	
planilla	
aprobado	
Responsabilidades:	
Nombre	Collaborator
crear_solicitud	
listar_solicitud	
modificar_solicitud	
eliminar_solicitud	
buscar_solicitud	

Fig. 2.18: Tarjeta CRC Solicitud

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

Fig. 2.19: Tarjeta CRC Usuario

Anexo 3 Pruebas de Aceptación

Tabla 3.8 Prueba de aceptación para la Historia de Usuario Gestionar Especialidad.

Prueba de Aceptación: Gestionar Especialidad
Historia de Usuario: Gestionar Especialidad.
Nombre: Prueba para comprobar la gestión de las especialidades.
Descripción: Con esta prueba de aceptación se persigue validar la gestión de las especialidades.
Condiciones de ejecución: El coordinador debe iniciar su sesión para acceder al módulo de gestión de especialidades y poder adicionar, eliminar, mostrar o modificar una especialidad.
Entrada / Pasos ejecución: El coordinador hace clic en el botón “Agregar” de la tabla “Mostrar Especialidad” y en el formulario “Agregar Especialidad” inserta los datos de la especialidad que desea agregar, luego presiona el botón “Agregar” y es insertada una nueva especialidad. En seguida que la especialidad es agregada puede ser modificada o eliminada.

<p>Resultado:</p> <p>Se insertan correctamente las especialidades.</p> <p>Se visualizan las especialidades correctamente.</p> <p>Se pueden modificar o eliminar una vez agregadas.</p>
<p>Evaluación de la prueba: Aceptada.</p>

Tabla 3.9 Prueba de aceptación para la Historia de Usuario Gestionar Material de Asignatura.

<p>Prueba de Aceptación: Gestionar Material de Asignatura</p>
<p>Historia de Usuario: Gestionar Material de Asignatura.</p>
<p>Nombre: Prueba para comprobar la gestión de los Material de Asignatura.</p>
<p>Descripción: Con esta prueba de aceptación se persigue validar la gestión de los materiales de asignatura.</p>
<p>Condiciones de ejecución: El profesor debe iniciar su sesión para acceder al módulo de gestión de los materiales y poder adicionar, eliminar y listar los materiales de asignatura.</p>
<p>Entrada / Pasos ejecución: El profesor hace clic en el botón “Agregar” de la tabla “Material de Asignatura” y en el formulario “Agregar Material de Asignatura” inserta los datos del material que desea agregar, luego presiona el botón “Agregar” y es insertado un nuevo material de asignatura. En seguida que el material es agregado puede ser eliminado.</p>
<p>Resultado:</p> <p>Se insertan correctamente los materiales de asignaturas.</p> <p>Se visualizan los materiales de las asignaturas correctamente.</p> <p>Se pueden eliminar una vez agregados.</p>
<p>Evaluación de la prueba: Aceptada.</p>

Tabla 3.10 Prueba de aceptación para la Historia de Usuario Gestionar Usuario.

Prueba de Aceptación: Gestionar Usuario
Historia de Usuario: 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 al módulo de gestión de usuarios y poder adicionar, buscar, eliminar y listar los usuarios.
Entrada / Pasos ejecución: El administrador hace clic en el botón “Agregar” de la tabla “Mostrar Usuarios” y en el formulario “Agregar Usuario” inserta los datos del usuario que desea agregar, luego presiona el botón “Agregar” y es insertado un nuevo usuario. En seguida que el usuario es agregado puede ser eliminado.
Resultado: Se insertan correctamente los usuarios. Se visualizan los usuarios correctamente. Se pueden eliminar una vez agregados.
Evaluación de la prueba: Aceptada.

Tabla 3.11 Prueba de aceptación para la Historia de Usuario Gestionar Solicitud.

Prueba de Aceptación: Gestionar Solicitud
Historia de Usuario: 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.
Condiciones de ejecución: El coordinador debe iniciar su sesión para acceder al módulo de atender solicitudes y poder revisarlas, aceptarlas o no.

<p>Entrada / Pasos ejecución: El coordinador selecciona una de las maestrías de la tabla “Maestrías con Solicitudes no atendidas” y presiona el botón “Mostrar Solicitudes”, luego selecciona una de las solicitudes que se mostrarán en la tabla “Solicitudes de la maestría” y presiona el botón “Atender Solicitud”. A continuación el coordinador podrá ver los datos del solicitante, descargar la planilla y aceptar o no la solicitud.</p>
<p>Resultado:</p> <p>Se aceptan o no correctamente las solicitudes.</p> <p>Se visualizan las solicitudes correctamente.</p> <p>Se almacenan correctamente.</p>
<p>Evaluación de la prueba: Aceptada.</p>

Tabla 3.12 Prueba de aceptación para la Historia de Usuario Gestionar Evaluación.

<p>Prueba de Aceptación: Gestionar Evaluación</p>
<p>Historia de Usuario: Gestionar Evaluación.</p>
<p>Nombre: Prueba para comprobar la gestión de las Evaluaciones.</p>
<p>Descripción: Con esta prueba de aceptación se persigue validar la gestión de las evaluaciones.</p>
<p>Condiciones de ejecución: El profesor debe iniciar su sesión para acceder al módulo de gestión de evaluaciones y poder adicionar, eliminar, modificar y listar las evaluaciones.</p>
<p>Entrada / Pasos ejecución: El profesor selecciona una de sus asignaturas, listadas en la tabla “Gestionar Evaluación” y hace clic en el botón “Mostrar Estudiantes”. A continuación se muestran todos los estudiantes que reciben la asignatura en la tabla “Estudiantes de la Asignatura”. Luego de seleccionar el estudiante que desea evaluar, podrá agregarle, modificarle y eliminarle la evaluación.</p>

Resultado:

Se insertan correctamente las evaluaciones.

Se visualizan las evaluaciones correctamente.

Se pueden modificar y eliminar una vez agregadas.

Evaluación de la prueba: Aceptada.