

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

**Título: Aplicación Web para la gestión de la
información de la Dirección de Relaciones
Internacionales del ISMMM**

Autor: José Luis Fernández Cuza

Tutor: Ing. Eloy Rafael Jiménez Iglesias

Consultante: Josefa Mestre.

DECLARACIÓN DE AUTORÍA

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

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

José Luis Fernández Cuza

Firma del Autor

Ing. Eloy Rafael Jiménez iglesias

Firma del Tutor

Agradecimientos

A mis padres, por su dedicación y apoyo incondicional, sin el cual me hubiera sido imposible seguir adelante.

A mi hermano, por darme la visión de quien ser en la vida, a mis tías Nela y Carmita, a mi primo Damisel. A mi socio Lázaro.

A mis profesores, los cuales me guiaron por el mundo del conocimiento, en especial a mi profe de Matemáticas, Yanet, que me aconsejó y me guió dándome razones para continuar.

A mis amigos, del barrio y del aula, a Angelita y su hija María, que recorrimos el camino desde el preescolar hasta el 12 e incluso trabajamos juntos, a Rene y su familia por el apoyo, a Usain y a Yosbani.

A Rafa, con sus locuras y sus despistes, Guerrero y sus historias, Roger, por su amistad, a Roco y a Juan Carlos, a Marlon y Eugenio, a todos por sus consejos y por mantenernos siempre unidos. A Deysi y sus canciones y por inventar la forma de escribir mi nombre, José, Indira y Yanet, las muchachitas del grupo.

A Yorvis, Héctor, Andy por su comprensión cuando necesitábamos trabajar en el lab.

A Exneider por los consejos de PHP, y a Néstor por las críticas en el DoTA.

Y por último y el más importante, a Yuris, por su amistad incondicional desde siempre, por estar ahí en los momentos buenos y malos, por aguantarme desde el politécnico hasta ahora y para siempre, por sus consejos y soporte, por ser simplemente MI HERMANO...

Pensamiento

*La virtud, como el arte, se consagra
constantemente a lo que es difícil de
hacer, y cuanto más dura es la tarea,
más brillante es el éxito.*

Aristóteles

Dedicatoria

A mis padres y mi hermano

A Ellis y su familia

A Nela y Damisel

A mis amigos

A mis profesores

A todos...

Resumen

La Dirección de Relaciones Internacionales es la encargada de manipular y controlar toda la información relacionada con los profesores que viajan al extranjero por razones académicas, así como también de los que vienen a recibir o impartir cursos de postgrado. También se encarga de controlar la información de los estudiantes extranjeros que se encuentran en nuestro Centro de estudio. La gestión de la información se ha convertido en un tema de amplia significación y debido al amplio volumen de información que se manipula, ha surgido la necesidad de crear un sistema automatizado que permita llevar el control de estos datos.

Con el desarrollo de esta investigación se propone la informatización de la gestión de la información correspondiente a los estudiantes extranjeros y profesores en la Dirección Relaciones Internacionales, a partir de la creación de una aplicación web que controle la información de manera ágil y segura.

Para la elaboración de la misma se realizó un estudio sobre las aplicaciones web y las herramientas para la construcción de las mismas. En este documento se recoge el estudio realizado y se presenta la metodología de Ingeniería de Software Programación Extrema, que se siguió para el desarrollo del software que se propone como solución de la problemática encontrada.

Abstract

The Direction of Internationals Relationship is the taken it upon to manipulate and control all the information related with the professors that they travel to the foreigner for academic reasons, as well as also of those that come to receive or impart courses of postgrad. It takes it upon to also control the information of the foreign students that they meet in our Center of study. The administration of the information has been converted in a topic of wide meaning and due to the wide volume of information that is manipulated; the necessity has surged of creating an automated system that allows carrying the control of these data.

With the development of this investigation proposes the digitalization of the management of the corresponding information to the foreign students and professors in Direction of Internationals Relationship, starting from the creation of a web application that controls the information of airy and sure manner.

For the elaboration of the same was carried out a study on the web applications and the tools for the construction of the same. In this document the carried out study is picked up and is introduced the methodology of Engineering of Software Extreme Programming that followed for the development of the software that proposes like solution of the opposing problem.



Índice

Introducción	1
Capítulo 1 Fundamentación Teórica.....	6
1.1 Estado del Arte.....	6
1.1.1 Informatización del proceso de gestión de los estudiantes extranjeros y profesores en la Dirección de Relaciones Internacionales del ISMMM.....	6
1.1.2 Flujo actual de los procesos.....	7
1.1.3 Análisis de la ejecución de los procesos.....	7
1.1.4 Antecedentes	8
1.1.5 Propuesta de Solución	9
1.2 Tendencias y tecnologías actuales	9
1.2.1 Aplicaciones Web	9
1.2.2 Servidores web	11
1.2.3 Sistemas de gestión de bases de datos	14
1.3 Lenguajes de programación	19
1.4 Metodologías de desarrollo	22
1.4.1 Scrum	22
1.4.2 Programación Extrema (Extreme Programming, XP)	28
1.4.3 Metodología propuesta para el desarrollo de la aplicación.	34
1.5 Patrones Arquitectónicos.....	35
1.5.1 Arquitectura en capas	36
1.5.2 Modelo Cliente – Servidor de 2 capas	37
1.5.3 Modelo – Vista – Controlador.....	38
1.6 Herramientas a utilizar en la propuesta de solución	41
1.6.1 XAMPP	41
1.6.2 Embarcadero ER/Studio	42



1.6.3	Yii Framework	42
1.6.4	NetBeans IDE	44
Capítulo 2 Planeación y Diseño.....		46
2.1	Funcionalidades Generales.....	46
2.1.1	Personal relacionado con el sistema	47
2.1.2	Lista de reserva	¡Error! Marcador no definido.
2.1.3	Características del sistema.....	49
2.1.4	Historias de usuario	50
2.2	Planificación de entregas	51
2.2.1	Estimación del esfuerzo por historias de usuario.....	52
2.2.2	Planificación de iteraciones	53
2.2.3	Plan de duración de las iteraciones	54
2.3	Diseño	55
2.3.1	Tarjetas CRC	56
Capítulo 3 Desarrollo y Pruebas.....		57
3.1	Modelo de datos.....	57
3.2	Interfaces de usuario.....	59
3.3	Desarrollo de las iteraciones	59
3.3.1	Tareas por historias de usuario.....	59
3.3.2	Tareas de ingeniería	64
3.4	Pruebas.....	64
3.4.1	Pruebas de aceptación	65
Capítulo 4 Estudio de Factibilidad		67
4.1	Evaluación Costo – Beneficio.....	67
4.2	Efectos Económicos	68
4.2.1	Efectos directos:	68



4.2.2	Efectos indirectos:.....	68
4.2.3	Efectos externos:	69
4.2.4	Intangibles	69
4.3	Ficha de costo de un producto informático.....	70
	Conclusiones Generales	74
	Recomendaciones.....	75
	Referencias Bibliográficas	76
	Bibliografía.....	78
	Glosario de Términos	80
	Anexo 1: Historias de usuarios.....	85
	Anexo 2: Tarjetas CRC.....	91
	Anexo 3: Tareas de ingeniería.....	95
	Anexo 4: Pruebas de Aceptación	103
	Anexo 5: Interfaces de usuario	108



Índice de ilustraciones y tablas

FIGURA 1.1 PILA DEL PRODUCTO	26
FIGURA 1.3 ELEMENTOS DE LA ARQUITECTURA EN DOS CAPAS.....	38
FIGURA 1.4 PATRÓN ARQUITECTURA MODELO VISTA CONTROLADOR	39
FIGURA 1.5 FUNCIONAMIENTO DE YII FRAMEWORK	43
TABLA 2.1: PERSONAL RELACIONADO CON EL SISTEMA.....	47
FIGURA 3.1 MODELO DE DATOS.	58
TABLA 2.6 HU NO. 1 AUTENTICAR USUARIO.....	85
TABLA 2.7 HU NO. 1 GESTIONAR USUARIO.....	85
TABLA 2.8 HU NO. 3 GESTIONAR ESTUDIANTE EXTRANJERO.....	86
TABLA 2.9 HU NO. 4 GESTIONAR ESTUDIANTE DE PREGRADO	86
TABLA 2.10 HU NO. 5 LLENAR REGISTRO DE VIAJE	86
TABLA 2.11 HU NO. 6 MOSTRAR GUÍA DEL EXPEDIENTE	87
TABLA 2.12 HU NO. 7 GESTIONAR PROFESOR DEL INSTITUTO.....	87
TABLA 2.13 HU NO. 8 GESTIONAR PASAPORTE.....	87
TABLA 2.14 HU NO. 9 MOSTRAR ALERTA DE VENCIMIENTO DEL PASAPORTE.	88
TABLA 2.15 HU NO. 10 GESTIONAR CONVOCATORIA.....	88
TABLA 2.16 HU NO. 11 MOSTRAR PERFIL DE UN USUARIO	89
TABLA 2.17 HU NO. 12 EXPORTAR PERFIL DE UN USUARIO.....	89
TABLA 2.18 HU NO. 13 EXPORTAR LISTADO DE LOS USUARIOS.	89
TABLA 2.19 HU NO. 14 EXPORTAR DATOS DE UN PROFESOR.....	90
TABLA 2.20 TARJETA CRC NO. 1	91
TABLA 2.21 TARJETA CRC NO. 2.....	91
TABLA 2.22 TARJETA CRC NO. 3.....	91
TABLA 2.23 TARJETA CRC NO. 4.....	92
TABLA 2.24 TARJETA CRC NO. 5.....	92
TABLA 2.25 TARJETA CRC NO. 6.....	92
TABLA 2.26 TARJETA CRC NO. 7	93
TABLA 2.27 TARJETA CRC NO. 8.....	93
TABLA 2.28 TARJETA CRC NO. 9.....	94
TABLA 3.10 TAREA DE PROGRAMACIÓN AUTENTICAR USUARIO.....	95
TABLA 3.11 TAREA DE PROGRAMACIÓN INSERTAR USUARIO	95
TABLA 3.12 TAREA DE PROGRAMACIÓN MOSTRAR USUARIO	95
TABLA 3.13 TAREA DE PROGRAMACIÓN MODIFICAR USUARIO	96



TABLA 3.14 TAREA DE PROGRAMACIÓN ELIMINAR USUARIO	96
TABLA 3.15 TAREA DE PROGRAMACIÓN INSERTAR ESTUDIANTE EXTRANJERO	96
TABLA 3.16 TAREA DE PROGRAMACIÓN MOSTRAR ESTUDIANTE EXTRANJERO	97
TABLA 3.17 TAREA DE PROGRAMACIÓN MODIFICAR ESTUDIANTE EXTRANJERO	97
TABLA 3.18 TAREA DE PROGRAMACIÓN ELIMINAR ESTUDIANTE EXTRANJERO	97
TABLA 3.19 TAREA DE PROGRAMACIÓN INSERTAR ESTUDIANTE DE PREGRADO	98
TABLA 3.20 TAREA DE PROGRAMACIÓN MOSTRAR ESTUDIANTE DE PREGRADO	98
TABLA 3.21 TAREA DE PROGRAMACIÓN MODIFICAR ESTUDIANTE DE PREGRADO.....	98
TABLA 3.22 TAREA DE PROGRAMACIÓN ELIMINAR ESTUDIANTE DE PREGRADO	99
TABLA 3.23 TAREA DE PROGRAMACIÓN LLENAR REGISTRO DE VIAJES DEL PROFESOR.	99
TABLA 3.24 TAREA DE PROGRAMACIÓN MOSTRAR GUÍA DEL EXPEDIENTE.	99
TABLA 3.25 TAREA DE PROGRAMACIÓN INSERTAR PROFESOR.	100
TABLA 3.26 TAREA DE PROGRAMACIÓN MOSTRAR PROFESOR.	100
TABLA 3.27 TAREA DE PROGRAMACIÓN MODIFICAR PROFESOR.....	100
TABLA 3.28 TAREA DE PROGRAMACIÓN ELIMINAR PROFESOR.	101
TABLA 3.29 TAREA DE PROGRAMACIÓN INSERTAR PASAPORTE.....	101
TABLA 3.30 TAREA DE PROGRAMACIÓN MOSTRAR PASAPORTE.	101
TABLA 3.31 TAREA DE PROGRAMACIÓN MODIFICAR PASAPORTE.	102
TABLA 3.32 TAREA DE PROGRAMACIÓN ELIMINAR PASAPORTE.	102
TABLA 3.33 TAREA DE PROGRAMACIÓN MOSTRAR ALERTA DE VENCIMIENTO DE PASAPORTE.....	102
TABLA 3.34 PA: PRUEBA PARA COMPROBAR LA ENTRADA DE UN USUARIO AL SISTEMA.	103
TABLA 3.35 PA: PRUEBA PARA COMPROBAR LA GESTIÓN DE USUARIOS.....	103
TABLA 3.36 PA: PRUEBA PARA COMPROBAR LA GESTIÓN ESTUDIANTES EXTRANJEROS	104
TABLA 3.37 PA: PRUEBA PARA COMPROBAR LA GESTIÓN DE ESTUDIANTES DE PREGRADO.....	105
TABLA 3.39 PA: PRUEBA PARA COMPROBAR LA ENTRADA DE DATOS DEL PROFESOR DEL INSTITUTO	106
TABLA 3.40 PA: PRUEBA PARA COMPROBAR LA ENTRADA DE DATOS DEL PASAPORTE	106
TABLA 3.41 PA: PRUEBA PARA COMPROBAR LA GESTIÓN DE ESTUDIANTES DE POSTGRADO.....	107
FIGURA 3.3 INTERFAZ DE USUARIO INSERTAR PROFESOR	108
FIGURA 3.4 INTERFAZ DE USUARIO ADMINISTRAR PROFESOR	109
FIGURA 3.5 INTERFAZ DE USUARIO ADMINISTRAR ESTUDIANTES EXTRANJEROS.....	109
FIGURA 3.6 INTERFAZ DE USUARIO INSERTAR ESTUDIANTE EXTRANJERO.	110
FIGURA 3.7 INTERFAZ DE USUARIO MODIFICAR ESTUDIANTE EXTRANJERO.	111
FIGURA 3.8 INTERFAZ DE USUARIO AUTENTICAR USUARIO. ERROR DE AUTENTICACIÓN.....	112
FIGURA 3.9 INTERFAZ DE USUARIO INSERTAR USUARIO.....	112



FIGURA 3.10 INTERFAZ DE USUARIO LISTAR PASAPORTES POR VENCERSE..... 113

FIGURA 3.11 INTERFAZ DE USUARIO LISTAR CARNET DE IDENTIDAD POR VENCERSE. 113



Introducción

En la era de la información, de la explosión de sus tecnologías, se vive la etapa en la que la humanidad ha alcanzado un desarrollo imprevisible; cada día son mayores las diferencias sociales, políticas y económicas. Se habla constantemente sobre la sociedad de la información, es visible el paso de las sociedades industriales a las posindustriales y del conocimiento, donde el factor esencial de progreso es el conocimiento. En este contexto, debe entenderse que las Tecnologías de Información y las Telecomunicaciones no son más que un medio para transmitir y gestionar datos, información y conocimiento, el conocimiento es factor fundamental para la creación de riquezas. (1) Estas son capaces de aportar soluciones eficientes y eficaces en cualquier esfera empresarial; permitiendo actuar con rapidez, obtener el máximo rendimiento del personal y tomar decisiones a la administración.

La información es un elemento fundamental para el desarrollo, con el decursar de los años, la gestión de la información ocupa, cada vez más, un espacio mayor en la economía de los países a escala mundial. Gestionar información es ir en busca de nuevos significados, análisis, aplicar el principio de que el todo, es más que la suma de las partes. Es producir un impacto en el ambiente de cualquier organización. Es un proceso que debe estar presente en cada uno de los pasos de la organización, es un proceso y a la vez un subproceso. Requiere acción, decisión y evaluación. En esta dirección han surgido los Sistemas de Gestión de Información, los cuales permiten la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. El Sistema de Gestión de Información aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones organizacional a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa.

Mediante la gestión se proporcionan los recursos de información necesarios para una buena toma de decisiones, se desarrollan nuevos conocimientos que posibilitan calidad



y eficiencia en los servicios y productos de las organizaciones. Cuba, con un proyecto de desarrollo que tiene como pilares la justicia social, la participación popular, la equidad y la solidaridad, ha diseñado e iniciado la aplicación de estrategias, donde las mismas permiten convertir los conocimientos y las tecnologías informáticas en instrumentos a disposición del avance y las profundas transformaciones revolucionarias.

La Dirección de Relaciones Internacionales es la encargada de manipular y controlar toda la información relacionada con los profesores que viajan al extranjero por razones académicas, así como también de los que vienen a recibir o impartir cursos de postgrado. También se encarga de controlar la información de los estudiantes extranjeros que se encuentran en nuestro Centro de estudio. Esta información es necesaria mantenerla organizada y accesible a las consultas. Debido al gran volumen de información que se manipula en la Dirección de Relaciones Internacionales y a la utilización de las Tecnologías de la Información y las Comunicaciones el Instituto Superior Minero Metalúrgico de Moa (ISMMM), como parte del sistema empresarial cubano, se ha trazado como estrategia la informatización de la mayoría de sus procesos.

La **situación problémica** que se presenta es que el proceso de gestión de la información se vuelve tedioso y lento. Pueden ocurrir pérdidas de información, además que las consultas deben realizarse manualmente, buscando en todos los documentos que se tienen. También se requiere de un mejor control de acceso a las informaciones posibilitando que los profesores puedan acceder a ciertos datos y llenarlos ellos mismos. Muchas de las informaciones están en una PC dentro de la Oficina de la Dirección de Relaciones Internacionales, por tanto si no se puede abrir por cualquier razón entonces no hay manera de acceder a las informaciones, esto es un inconveniente y crea demoras en los procesos de consultas de los datos. A partir de esta situación se identifica el **problema a resolver** ¿Cómo favorecer la eficiencia en la gestión de la información de los estudiantes extranjeros y profesores que se manipula en la Dirección de Relaciones Internacionales del ISMMM? En esta dirección surge el proyecto SGDRI, el cual tiene como **objeto de estudio**: las aplicaciones web que permitan la gestión de la información de los estudiantes extranjeros y profesores,



abarcando el **campo de acción** de la informatización de los procesos de gestión de la información en la Dirección de Relaciones Internacionales del ISMMM.

Para darle solución al problema se traza como **objetivo general** la creación de una aplicación web que permita la gestión de la información de los profesores y estudiantes extranjeros en la Dirección de Relaciones Internacionales en el Instituto Superior Minero Metalúrgico de Moa (ISMMM).

Para darle solución al objetivo general se trazan los siguientes **objetivos específicos**:

1. Establecer el Marco Teórico de la investigación.
2. Realizar un levantamiento sobre los procesos que se llevan a cabo en la dirección de relaciones internacionales del ISMMM.
3. Seguir los pasos de la metodología de desarrollo de software seleccionada.

Para darle cumplimiento a los objetivos específicos se definieron las siguientes **Tareas de investigación**:

1. Buscar los antecedentes de la investigación.
2. Seleccionar las tecnologías, herramientas y la metodología de desarrollo de software.
3. Realizar búsquedas bibliográficas.
4. Entrevistar a especialistas de la Dirección de Relaciones Internacionales.
5. Estudiar el proceso de control y manipulación de la información referente a los estudiantes extranjeros y profesores en la Dirección de Relaciones Internacionales del ISMMM con los especialistas.
6. Identificar los requerimientos del sistema.
7. Elaboración del diseño de la BD del sistema propuesto.
8. Elaboración de las interfaces de cada HU (Historia de Usuario) según la opinión del cliente.
9. Implementación de las funciones que deben tener cada HU según los requisitos que debe cumplir el software.
10. Realización de pruebas al sistema informático desarrollado.



11. Realizar el Manual de usuario.

De esta forma la **idea a defender** se plantea que si se elabora una aplicación web para la gestión de la información de los estudiantes extranjeros y profesores manejada en la Dirección de Relaciones Internacionales del ISMMM, se mejorará el acceso a la información, el proceso de obtención de informes, favorecerá la organización de la información así como también agilizar el proceso de trabajo.

Los **métodos empíricos** utilizados son:

1. Entrevista a expertos: para recopilar información necesaria para valorar la situación actual del problema, lo cual permitió identificar los principales requisitos del sistema.
2. Revisión y análisis de documentos: para la recopilación de la información las consultas bibliográficas.

Los **métodos teóricos** utilizados son:

1. Análisis y síntesis: este método se utiliza para desglosar el problema en partes o subproblemas, para de esta forma comprobar funcionamiento de los mismos, luego integrarlo todo para corroborar las relaciones entre estas y su integración con un todo llegando así a una mejor solución, también para arribar a conclusiones de la investigación.
2. Histórico – Lógico: se pudo de manifiesto durante el proceso de estudio de herramientas y metodologías de desarrollo de software a utilizar.



Estructura de la Tesis

El presente trabajo de diploma estará estructurado en 4 capítulos:

Capítulo 1: “Fundamentación Teórica”: Este capítulo incluye un estado del arte del tema tratado, además se analizarán los diferentes lenguajes, tecnologías y metodologías de desarrollo de software empleadas en el desarrollo de la aplicación propuesta.

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

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

Capítulo 4: “Estudio de Factibilidad”: En este se realiza un estudio para ver la factibilidad del producto por la metodología Coste-Beneficio. Además de un estudio de los esfuerzos requeridos para la realización del sistema propuesto.



Capítulo 1 Fundamentación Teórica

En el presente capítulo se presenta una descripción general del escenario donde radica el objeto de estudio, así como el campo de acción en que se trabaja. Se manifiesta una mejor visión sobre la herramienta a desarrollar. Este capítulo constituye la base teórica para la comprensión del trabajo que se desarrolla. Se abordarán conceptos asociados al análisis de las tecnologías y herramientas a utilizar, además de la metodología seleccionada para el desarrollo del software.

1.1 Estado del Arte

1.1.1 Informatización del proceso de gestión de los estudiantes extranjeros y profesores en la Dirección de Relaciones Internacionales del ISMMM.

El perfeccionamiento del control de la información se ha convertido en una tarea primordial en todas las instituciones, pues a través del análisis de los datos se pueden obtener valiosas conclusiones que pueden ayudar a mejorar el desempeño de la institución. Conocer con detalles los datos de los estudiantes extranjeros es de gran importancia pues así se puede entender cómo se están desarrollando sus conocimientos en sus respectivas carreras además de saber si se sienten a gusto con las clases y las actividades que se realizan. También es de vital relevancia que los profesores que viajen al extranjero tengan sus datos actualizados y que puedan acceder a ellos de manera rápida para modificarlos. Enmarcados en el perfeccionamiento de este proceso se ha despertado en el ISMMM la necesidad de informatizar el mismo.



1.1.2 Flujo actual de los procesos

En la Dirección de Relaciones Internacionales (DRI) del ISMMM se gestiona la información relacionada con los estudiantes extranjeros y profesores, además de las convocatorias a proyectos internacionales.

Las convocatorias llegan a la DRI y allí se guardan la organización, el tipo de proyecto, el monto, entre otros datos. Esta información se almacena en documentos digitales o en papel.

La Especialista para la atención de Becarios Extranjeros recibe y viabiliza los planteamientos de carácter académico, elabora el informe general de resultados docentes mensual, semestral y anual. Estos procesos se realizan manualmente por lo que resultan lentos y con un alto grado de complejidad.

También la DRI dirige y controla los datos de los profesores que viajen al extranjero y de profesores e investigadores extranjeros que llegan a nuestro centro.

1.1.3 Análisis de la ejecución de los procesos

Los factores o causas que afectan el proceso de gestión de la información de los estudiantes extranjeros y profesores han sido identificados mediante técnicas de observación directa y entrevistas a los especialistas de la DRI.

Información

Causas identificadas

- ✚ Aislamiento de la información.
- ✚ Dificultad en el manejo de información.
- ✚ Dificultad de obtención de información en el momento requerido.

Tiempo

Causas identificadas

- ✚ Retraso en la entrega de información.



- ✚ Retraso en el procesamiento de la información.
- ✚ Retraso en la elaboración de reportes.

Procedimientos

Causas identificadas

- ✚ Procesamiento mediante documentos en formato doc.
- ✚ Procesamiento de datos en forma impresa.

1.1.4 Antecedentes

En la intranet de la Universidad de Guantánamo, existe un enlace que permite visualizar e imprimir la relación de los profesores que se encuentran en el exterior. Esta información se muestra en el contenido de la misma página de la intranet, dentro de una tabla, con las opciones de imprimir y enviar por correo electrónico los datos de los profesores. (2)

DRI-UH: Sistema de Gestión de las Relaciones Internacionales de la Universidad de la Habana

Este es un sistema que permite la gestión de la información de la Dirección de Relaciones Internacionales de la Universidad de la Habana. La Universidad de la Habana (UH) también tiene incorporada a su intranet enlaces que direccionan a páginas que permiten visualizar información relacionada con la Dirección de Relaciones Internacionales de la UH. (3)

Aunque cuenta con muchas funcionalidades no es lo que se requiere en la Dirección de Relaciones Internacionales del ISMMM porque se necesita, además de controlar la información de los estudiantes extranjeros, que los profesores accedan al sistema por la red y llenen sus propios datos.

La Universidad “Oscar Lucero Moya” de Holguín también cuenta es su intranet con páginas que permiten obtener información sobre la Dirección de Relaciones Internacionales, su misión y sus procedimientos. (4)



Ninguno de estos métodos puede aplicarse a lo que el cliente requiere pues estos son solo páginas estáticas para visualizar información. No se adaptan a las necesidades requeridas por la Dirección de Relaciones Internacionales ya que no permite realizar consultas a los datos, ni los profesores pueden ellos mismos acceder a su información solamente para realizar algún cambio o actualización.

1.1.5 Propuesta de Solución

Después de analizada la problemática presentada se propone para darle solución el desarrollo de una aplicación Web que favorezca la eficiencia en la gestión de información en la Dirección de Relaciones Internacionales del Instituto Superior Minero Metalúrgico de Moa.

1.2 Tendencias y tecnologías actuales

1.2.1 Aplicaciones Web

En la ingeniería de software se denomina **aplicación web** a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (*HTML, JavaScript, Java, etc.*) en la que se confía la ejecución al navegador.

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa, un motor capaz de usar alguna tecnología web dinámica constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa.

El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario. (5)



1.2.1.1 Ventajas de una web

- **Compatibilidad multiplataforma:** Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo *Java*, *Flash*, *ASP* y *Ajax* permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- **Actualización:** Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo, pues no se hace necesario iniciar nuevas descargas y/o procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones).
- **Inmediatez de acceso:** Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y están listas para trabajar sin importar cuál es su configuración o su hardware.
- **Menos requerimientos de memoria:** Las aplicaciones basadas en Web tienen menos demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en Web usan en muchos casos la memoria de las computadoras donde ellas corren, dejando más espacio para correr múltiples aplicaciones sin incurrir en frustrantes deterioros en el rendimiento.
- **Menos Bugs:** Las aplicaciones basadas en Web son menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los *bugs* pueden ser corregidos tan pronto como son descubiertos.
- **Precio:** Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables y no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.



- **Múltiples usuarios concurrentes:** Las aplicaciones basadas en Web pueden ser utilizadas por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta.
- **Los datos son más seguros:** Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios escuchen mucho menos del tema. A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora. Las compañías que proveen aplicaciones basadas en Web van a brindar amplios servicios de resguardo de datos ya sea como una parte integral del servicio básico o como una opción paga. (6)

1.2.1.2 ¿Por qué utilizar una aplicación web?

Las aplicaciones Web son de un desarrollo poco costoso, sencillo y rápido. Presentan acceso ubicuo, sin necesidad de distribución e idealmente con pocos requerimientos técnicos. Con datos centralizados y fácil integración de datos múltiples fuentes. Por los aspectos antes expuestos se ha decidido usar la tecnología web para el desarrollo del sistema.

1.2.2 Servidores web

Un servidor Web es un programa que implementa el protocolo HTTP (*Hypertext Transfer Protocol*). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o paginas HTML (*Hypertext Markup Language*), textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproducciones de sonidos.

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza



una petición al servidor y este le responde con el contenido que el cliente solicita. Entre los servidores Web más utilizados se encuentran *Microsoft Internet Information Server*, *Microsoft Personal Web Server*, *Apache HTTP Server*, *Netscape Enterprise Server* y *Sun ONE Web Server*.

Apache es el servidor Web más difundido y utilizado en Internet debido a su alta estabilidad, seguridad, facilidad de expansión y por su costo ya que es software libre, lo que significa que se distribuye por medio de una licencia gratuita y de código abierto; además se encuentra disponible para una gran cantidad de sistemas operativos, tiene como fin crear un servidor HTTP (Web) robusto, de nivel comercial, con innumerables características y de distribución gratuita.

IIS (*Internet Information Server*), es el servidor web estándar del sistema operativo Windows NT/2000/XP, ofrece varios servicios de Internet: servidor Web (o HTTP), servidor FTP y servidor SMTP. (7)

Apache vs. ISS

Ventajas

ISS

- ✚ Fácil de usar.
- ✚ ASP preparado en la instalación por defecto.
- ✚ Soporte ODBC integrado.
- ✚ Configuración gráfica y en línea de comandos.

Apache

- ✚ Código fuente disponible.
- ✚ Existen versiones virtualmente para cualquier sistema operativo.
- ✚ Excelente integración con PHP y MySQL.
- ✚ Es el servidor web por excelencia en la Red.



Inconvenientes

ISS

- ✚ Multitud de nuevos fallos de seguridad.
- ✚ La mayoría de funcionalidad extra debe ser comprada separadamente.
- ✚ Solo funciona en Windows NT/2000/XP

Apache

- ✚ No existe aún configuración gráfica oficial.
- ✚ Curva larga de aprendizaje para sacarle el máximo partido.

1.2.2.1 ¿Por qué utilizar Apache?

- 1) Corre sobre una multitud de plataformas y Sistemas Operativos.
- 2) Ofrece tecnología libre y de código abierto, otorgándole una transparencia y dando la posibilidad de conocer que es lo que realmente estamos instalando.
- 3) Es un servidor Web altamente configurable y de diseño modular, capaz de ampliar su funcionalidad y calidad de servicios.
- 4) Trabaja en conjunto con gran cantidad de Lenguajes de Programación interpretados como PHP (*PHP Hypertext Pre-processor*), Perl, soporte con CGI (*Common Gateway Interface*), Java, JSP (*Java Server Pages*) y otros lenguajes de script, el complemento ideal para los sitios web dinámicos que vemos en la actualidad.
- 5) Es posible configurar y personalizar cada uno de los mensajes de error que se pueden producir por la utilización del servidor.



1.2.3 Sistemas de gestión de bases de datos

Los sistemas de gestión de base de datos (SGBD); en inglés: *Database Management System*, abreviado DBMS son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. Entre los SGBD disponibles en el mercado se encuentran: PostgreSQL, MySQL, Firebird, SQLite, Oracle, Microsoft SQL Server, entre otros.

PostgreSQL

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. (8)

Características

Alta concurrencia.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. (8)

Alta variedad de tipos nativos.

PostgreSQL provee nativamente soporte para:

1. Números de precisión arbitraria.
2. Texto de largo ilimitado.



3. Figuras geométricas (con una variedad de funciones asociadas).
4. Direcciones IP (IPv4 e IPv6).
5. Bloques de direcciones estilo CIDR.
6. Direcciones MAC.
7. Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura *GiST* de PostgreSQL. (8)

Otras características

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (*foreign keys*).
- Disparadores (*triggers*): Un disparador o *trigger* se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.
- Vistas.
- Integridad transaccional.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Soporte para transacciones distribuidas. Permite a PostgreSQL integrarse en un sistema distribuido formado por varios recursos (p.ej, una base de datos PostgreSQL, otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("*commit*") de la transacción global es el resultado del éxito de las transacciones locales. (8)

Desventajas

- ✚ Sin experiencia, configurar llega a ser un caos.
- ✚ Es fácil de vulnerar sin protección adecuada.
- ✚ El motor MyISAM es instalado por defecto y carece de capacidades de integridad relacional.



- ✚ InnoDB genera mucho footprint en memoria al indizar.
- ✚ El toolset empresarial tiene un costo adicional por suscripción anual.
- ✚ Realizar revisiones llegar a ser una labor manual y tediosa para el Administrador de la Base de Datos (DBA).

Microsoft SQL Server

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL. (9)

Características

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos. (9)

Desventajas

- MSSQL usa *Address Windowing Extensión (AWE)* para hacer el direccionamiento de 64-bit. Esto le impide usar la administración dinámica de memoria, y sólo le permite alojar un máximo de 64 GB de memoria compartida.
- MSSQL no maneja compresión de datos (excepto la versión 2008 *Enterprise Edition*, que sí lo hace), por lo que las bases de datos pueden llegar a ocupar mucho espacio en disco.
- MSSQL requiere de un sistema operativo Microsoft Windows, por lo que no puede instalarse, por ejemplo, en servidores Linux. (9)



Requisitos del hardware

SQL Server 2000 necesita aproximadamente de 128 MB RAM, de un procesador Pentium 166 MHz y de un espacio de disco de aproximadamente 270 MB para una completa instalación.

Costo

El *SQL Server 2000* está actualmente disponible bajo dos opciones de autorización:

- La licencia del procesador
- Licencia de acceso puesto cliente/ Servidor

La licencia del procesador requiere una sola licencia por cada CPU en el que se va a correr el SQL Server 2000 e incluye accesos ilimitados a clientes, esta licencia normalmente es más barata que la del acceso puesto Cliente/Servidor pues requiere una licencia para el servidor y las licencias para cada dispositivo del cliente.

MySQL

Es una base de datos relacional de libre distribución con potencia, usada principalmente como soporte de portales web que necesitan manejar datos dinámicamente, es una alternativa a los grandes servidores de bases de datos basados en Microsoft SQL Server y Oracle. MySQL se destaca especialmente por su velocidad al efectuar lecturas y búsquedas en la base de datos.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un



gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. (5)

MySQL cuenta con muchas **ventajas**, entre las que se encuentran las siguientes:

- Alto rendimiento
- Portabilidad
- MySQL software es Open Source.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet.
- El software MySQL usa la licencia GPL (*General Public License*). (10)

Requisitos de hardware

MySQL usa menos recursos del hardware. Para instalar MySQL, se necesita 32 RAM de Mb y casi cerca de 60Mb de espacio en disco duro. MySQL no requiere los recursos del CPU adicionales.



1.2.3.1 ¿Por qué MySQL?

MySQL es muy rápido, fiable y fácil de usar. Es un sistema multiplataforma de base de datos relacionales, lo que da velocidad y flexibilidad, cuenta con un sistema de privilegios contraseñas, muy seguro que permite la autenticación básica para el acceso al servidor. MySQL es un sistema de administración de base de datos. Opera en una arquitectura cliente/servidor. Es el sistema gestor de bases de datos “*open source*” más popular, o sea que puede ser bajado de Internet y usarlo sin tener que pagar, además que cualquiera puede estudiar su código y adecuarlo a las necesidades que requiera.

Luego de analizadas las características y facilidades del SGBD presentado, se decide usar el MySQL como SGBD, por las siguientes razones:

- ✚ Alto rendimiento y velocidad al realizar las operaciones.
- ✚ El PHP maneja más fácil al MySQL que al SQL Server, debido a la gran cantidad de funciones que tiene explícitas.
- ✚ El MySQL es multiplataforma.
- ✚ El MySQL no tiene precio en el mercado, se adquiere libremente.

1.3 Lenguajes de programación

En el dominio de la red, los lenguajes del lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP, entre otros.

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

ASP (*Active Server Pages*), desarrollado por Microsoft para funcionar junto a su servidor IIS, es un lenguaje poderoso sobre todo si se va a utilizar herramientas de Microsoft en conjunto como SQL Server



PHP vs ASP

Velocidad

Uno de los principales problemas en los portales de Internet que soportan una gran carga de usuarios es la velocidad de respuesta de su servidor, por lo que este debería ser un punto crítico al elegir un lenguaje u otro.

ASP, probablemente, nunca será tan rápido como PHP. ASP está construido en una arquitectura de objetos COM, y por lo tanto, cuando un programador de ASP usa VBScript realmente está corriendo un objeto COM. Si escribe un simple texto en la salida estándar, estaría llamando al objeto *Response* del método *Write*. Si accede a una base de datos, debe usar otro objeto, etc. Todo esto sobrecarga y hace más lento el sistema.

En cambio, al usar PHP todas las operaciones se realizan en el espacio de memoria de PHP. Esto significa que el código PHP ejecutará más rápido ya que no debe cargar dinámicamente ningún otro objeto en memoria en la ejecución de un programa.

Costes Ocultos

Una de las cosas que se suelen pasar por alto al usar un software determinado son los costes ocultos o derivados de la primera compra.

ASP cumple con esta especificación, es decir, un desarrollador posiblemente no pueda realizar todas las tareas que necesita con las opciones que ofrece ASP por defecto; si necesita encriptación comprará ASPEncrypt, si necesita gestión de correo comprará Qmail de ServerObject, por ejemplo, si necesita subir ficheros (file uploading) al servidor comprará SA-FileUP de Artisans. Todo esto y más están disponibles en PHP gratuitamente.

PHP vs ASP vs Perl con respecto a su ejecución en sistemas operativos.

PHP es un lenguaje multiplataforma. Un script PHP normal puede ejecutarse sin cambiar ni una sola línea de código en cualquier servidor que interprete PHP, es decir,



en servidores Windows, Linux etc. ASP, por contra, es un lenguaje propietario de Microsoft y solo puede ejecutarse en servidores Microsoft. Perl si puede ejecutarse bajo Windows y Unix, pero antes hay que realizar labores de adaptación de cada script al concreto sistema operativo. (11)

1.3.1.1 ¿Por qué usar PHP?

Es un lenguaje de programación usado generalmente en la creación de contenido para sitios o aplicaciones web. La versión 5 de PHP presenta un magnífico trabajo con el paradigma orientado a objeto que permite la reutilización de código entre otras facilidades.

Ventajas de trabajar con PHP comparado con otros lenguajes similares:

- ✚ Es un lenguaje multiplataforma.
- ✚ Rapidez de ejecución.
- ✚ Mantiene un bajo consumo de recursos de máquina.
- ✚ Gran seguridad, muy poca probabilidad de corromper los datos.
- ✚ Capacidad de conexión con la mayoría de los manejadores de bases de datos que se emplean en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✚ Posee una amplia documentación en internet, incluyendo una gran variedad de ejemplos y de ayudas.
- ✚ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✚ Permite las técnicas de programación orientada a objetos.
- ✚ Permite crear formularios para la web.
- ✚ No requiere definición de tipos de variables ni manejo detallado de bajo nivel.



1.4 Metodologías de desarrollo

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas informáticos. Las metodologías existentes en la actualidad se dividen en dos grandes grupos atendiendo a sus características: las metodologías tradicionales (RUP, MSF) y las metodologías ágiles (XP, SCRUM). Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y al mantener una buena relación con el cliente. (5)

Este último enfoque ha mostrado su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, no obstante, mantiene una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales.

1.4.1 Scrum

Scrum es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software. Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de software, puede ser utilizado en equipos de mantenimiento de software, o en una aproximación de gestión de programas. (13)

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

- ✚ Colaboración estrecha con el cliente.
- ✚ Predisposición y respuesta al cambio
- ✚ Prefiere el conocimiento tácito de las personas al explícito de los procesos



- ✚ Desarrollo incremental con entregas funcionales frecuentes
- ✚ Comunicación verbal directa entre los implicados en el proyecto
- ✚ Motivación y responsabilidad de los equipos por la auto-gestión, auto-organización y compromiso.
- ✚ Simplicidad. Supresión de artefactos innecesarios en la gestión del proyecto.

Roles

En Scrum se definen varios roles, estos están divididos en dos grupos: cerdos y gallinas. El nombre de los grupos está inspirado en la siguiente frase:¹

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

Las necesidades, deseos, ideas e influencias de los roles “gallina” se tienen en cuenta, pero no de forma que pueda afectar, distorsionar o entorpecer el proyecto Scrum. (13)

Roles "Cerdo"

Propietario del Producto

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

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

Equipo

¹ En un plato de huevos con tocino el cerdo está comprometido, la gallina sólo está involucrada.



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

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

Scrum Manager

Responsable del proceso Scrum.

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

Roles "Gallina"

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

Análisis de la frase "Rol gallina":

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

Usuarios

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

Stakeholders (Clientes, Proveedores, Inversores)



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

Managers

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

Sprint

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

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

Artefactos

Pila de producto (Product Backlog)

Listado con los requisitos del sistema

- ✚ Es responsabilidad del dueño del producto
 - ❖ Contenido
 - ❖ Priorización
 - ❖ Disponibilidad
- ✚ Nunca llega a ser una lista completa y definitiva



- + El empleado para planificar el proyecto es sólo una estimación inicial de requisitos
- + Es un documento dinámico que incorpora constantemente las necesidades del sistema
- + Se mantiene durante todo el ciclo de vida (hasta la retirada del sistema)

Pila del producto

Product Backlog		Estimación Inicial	Complejidad	Estim. ajustada	Trabajo pendiente			
					Sprint			
ID	Elemento				1	2	3	4
1	Nuevo formulario para peticiones de clientes	2	0.2	2,4	2,4	0	0	0
2	Configuración de respuestas automáticas	3	0.2	3,6	3,6	0	0	0
3	Envío automático de respuestas	1	0.2	1,2	1,2	0	0	0
4	Consulta para los clientes de peticiones enviadas	1	0.2	1,2	1,2	0	0	0
5	Modificación del cliente de sus peticiones enviadas	2	0.2	2,4	2,4	0	0	0
6	Acceso a peticiones sólo para clientes del portal jurídico	5	0.2	6	6	0	6	0
7	Consulta de peticiones por parte del staff	1	0.2	1,2	1,2	0	0	0
SPRINT 1		15		18	18	0	0	0
8	Inserción de comentarios y reasignación a peticiones (staff)	2	0.2	1,2	1,2	1,2	0	0
9	Consultas por clientes, fechas y temas	3	0,2	3,6	3,6	3,6	0	0
10	[Continúa]...							

Figura 1.1 Pila del Producto

Pila del Sprint (Sprint backlog)

Trabajo o tareas determinadas por el equipo para realizar en un sprint y lograr al final del mismo un incremento de la funcionalidad. Se recomienda que las tareas reflejadas tengan una duración comprendida entre las 4 y las 16 horas de trabajo. Las de mayor duración deben intentar descomponerse en sub-tareas de ese rango de tiempo. (13)

Gráfica de progreso

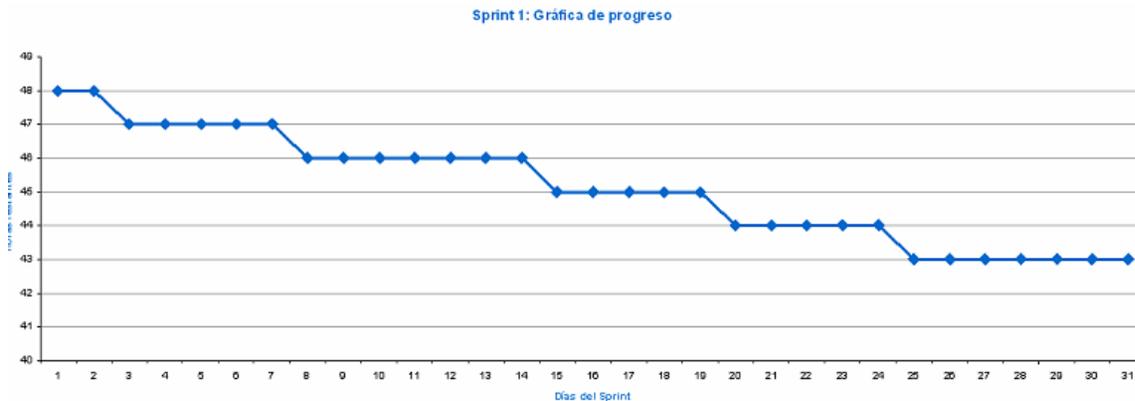


Figura 1.2 Gráfica de progreso

Burn down

La burn down chart es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, momento en el cual el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el Backlog). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos. (13)



1.4.2 Programación Extrema (Extreme Programming, XP)

XP es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, que describe la filosofía de XP en el "Manifiesto Ágil", sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. (11)

1.4.2.1 Valores que promueve XP

Cuatro valores que promueven la metodología XP:

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

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

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

Coraje: El coraje (valor) existe en el contexto de los otros 3 valores. (12)



1.4.2.2 El ciclo de vida ideal de XP

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

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

1.4.2.3 Prácticas en las que se fundamenta XP

Planificación incremental

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

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

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

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



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

Las responsabilidades del equipo de desarrollo son las siguientes:

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

Testing

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

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



Diseño simple

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

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

Propiedad colectiva del código

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

Integración continua.

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

Clientes en el equipo

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

La metodología XP intenta resolver este tipo de problemas integrando un representante del negocio dentro del equipo de desarrollo. Esta persona siempre está disponible para



resolver dudas y para decidir qué se hace en cada momento, en función de los intereses del negocio.

Entregas pequeñas

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

Semana de 40 horas

La programación extrema lleva un modo de trabajo en el que el equipo siempre está al 100%. Una semana de 40 horas en las que se dedica la mayor parte del tiempo a tareas que suponen un avance puede dar mucho de sí, y hace innecesario recurrir a sobreesfuerzos, excepto en casos extremos. Además, el esfuerzo continuado pronto lleva a un rendimiento menor y a un deterioro de la moral de todo el equipo. (12)

1.4.2.4 Fases de la Metodología XP

Fase I: Planificación

- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen *test* de aceptación para historias de usuarios (*user storie*) y permita hacer una estimación de tiempo de desarrollo del mismo.
- Se crea un plan de lanzamiento (*release planning*), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usa como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.



- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.
- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.

Fase II: Diseño

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

Fase III: Codificación

- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.
- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro, qué objetivos tiene que cumplir lo que se va a codificar.
- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores que integre código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos



de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.

- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.
- Se deja la optimización para el final.
- No se hacen horas extra de trabajo.

Fase IV: Pruebas

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las historias de usuarios (HU) elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que la correspondiente historia de usuario se ha completado. (12)

1.4.3 Metodología propuesta para el desarrollo de la aplicación.

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



independiente. Esto hace que el costo de su implementación sea mucho más accesible que el de otras metodologías.

Ventajas de XP

- ✚ Apropiado para entornos volátiles.
- ✚ Puede ser implementado en forma parcial (elegir sólo algunas de las prácticas) o en forma gradual.
- ✚ Puede adaptarse a las necesidades de cualquier equipo de desarrollo.
- ✚ Estar preparados para el cambio, significa reducir su coste.
- ✚ Permite definir en cada iteración cuáles son los objetivos de la siguiente.
- ✚ Exige que se establezca una comunicación más fluida con el cliente y que este tenga mayor participación en el proceso de desarrollo. La consecuencia de esto es que el cliente se involucre más en el desarrollo del producto.
- ✚ Permite la retroalimentación.
- ✚ Actualmente es la metodología ágil más extendida y documentada. Se realizan pruebas constantemente del sistema.

1.5 Patrones Arquitectónicos

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen una escala más grande.

Aunque un patrón arquitectónico comunica una imagen de un sistema, no es una arquitectura como tal. Un patrón arquitectónico es más un concepto que captura elementos esenciales de una arquitectura de software. Muchas arquitecturas diferentes



pueden implementar el mismo patrón y por lo tanto compartir las mismas características. Además, los patrones son a menudo definidos como una cosa "estrictamente descrita y comúnmente disponible". Por ejemplo, la arquitectura en capas es un estilo de llamamiento-y-regreso, cuando define uno un estilo general para interaccionar. Cuando esto es descrito estrictamente y comúnmente disponible, es un patrón.

Uno de los aspectos más importantes de los patrones arquitectónicos es que encarnan diferentes atributos de calidad. Por ejemplo, algunos patrones representan soluciones a problemas de rendimiento y otros pueden ser utilizados con éxito en sistemas de alta disponibilidad. A primeros de la fase de diseño, un arquitecto de software escoge qué patrones arquitectónicos mejor ofrecen las calidades deseadas para el sistema. (14)

1.5.1 Arquitectura en capas

Los sistemas o arquitecturas en capas constituyen uno de los estilos que parecen con mayor frecuencia mencionados como categorías mayores del catálogo, por el contrario, como una de las posibles imágenes de algún estilo envolvente.

Definen el estilo en capas como una organización jerárquica tal, que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones de la inmediatamente inferior.

La arquitectura por capas es un estilo de arquitectura en la que el objetivo primordial es la separación de la lógica de negocio de la lógica de diseño, un ejemplo básico es separar la capa de datos, de la capa de presentación al usuario.

La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de los niveles, simplemente es necesario conocer las API que existen entre niveles. El diseño de sistemas informáticos suele usar las arquitecturas multinivel o programación por capas. En dichas arquitecturas a cada nivel se le confía una misión



simple, lo que permite el diseño de arquitecturas escalables, (que pueden ampliarse con facilidad en caso de que las necesidades aumenten). (11)

1.5.2 Modelo Cliente – Servidor de 2 capas

Las aplicaciones cliente-servidor clásicas o de 2 capas, como su nombre lo indica, agrupan la lógica de presentación (interfaz) y la lógica de aplicación en la máquina cliente y acceden a fuentes de datos compartidos a través de una conexión de red que se encuentran en el servidor de datos.

La ventaja que presenta este tipo de aplicaciones es que los datos están centralizados. Esta centralización beneficia a la empresa pues es más fácil compartir los datos, se simplifica la generación de reportes y se proporciona consistencia en el acceso a los datos.

A continuación citamos las principales desventajas que presentan este tipo de aplicaciones:

- **Difíciles de mantener:** Esto viene dado por el hecho de que son difíciles de mantener las reglas de negocio de la lógica de aplicación ya que estas están programadas en cada cliente y esto implica que cualquier cambio tiene que ser redistribuido en todos los clientes.
- **Se compromete la confidencialidad:** Al tener programada la lógica de aplicación en el cliente este tiene a su disposición todas las reglas de negocio de la empresa.
- **Están estrechamente limitadas a una fuente de datos:** Los clientes casi siempre están configurados para acceder a una base de datos en particular por lo que mover los datos a una base de datos diferente se hace realmente complicado. (11)

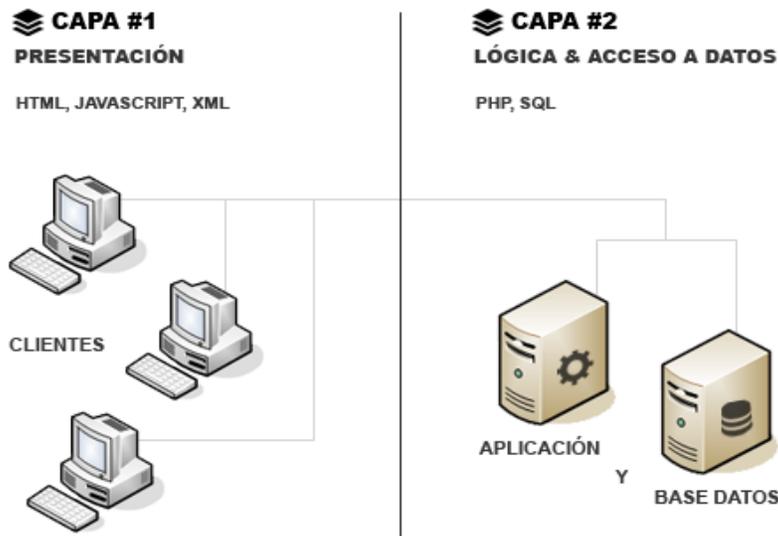


Figura 1.3 Elementos de la arquitectura en dos capas.

1.5.3 Modelo – Vista – Controlador

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Descripción

- + Modelo: Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.



- ✚ Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- ✚ Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

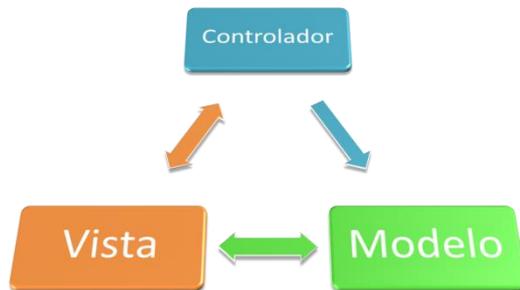


Figura 1.4 Patrón Arquitectura Modelo Vista Controlador

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (*handler*) o *callback*.



3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta relación entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista. Por ejemplo en el MVC usado por Apple en su framework Cocoa. Suele citarse como Modelo-Interface-Control, una variación del MVC más puro
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente. (15)

1.5.3.1 ¿Por qué MVC?

Proponemos como patrón arquitectónico a utilizar en nuestro proyecto el MVC porque este separa Modelo de la Vista, es decir, separa los datos de la representación visual de los mismos, crea independencia de funcionamiento, facilita el mantenimiento en caso de errores y permite el escalamiento de la aplicación en caso de ser requerido.



1.6 Herramientas a utilizar en la propuesta de solución

Luego del estudio realizado se arriba a la conclusión de que el sistema estará guiado por la metodología de desarrollo XP por sus facilidades, documentación y flexibilidad, además es la que mejor se adapta a las condiciones de desarrollo del sistema propuesto.

1.6.1 XAMPP

Es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl. El nombre proviene de X (para cualquier sistema operativo), A (Apache), M (MySQL), P (PHP), P (Perl). El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

XAMPP es regularmente actualizado para incorporar las últimas versiones de Apache/MySQL/PHP y Perl. También incluye otros módulos como OpenSSL, y PhpMyAdmin. (5) Para instalar XAMPP requiere solamente una pequeña fracción del tiempo necesario para descargar y configurar programas por separado.

Oficialmente, los diseñadores de XAMPP solo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica sin embargo, XAMPP es utilizado actualmente para servidor de sitios webs en WWW, y con algunas modificaciones es generalmente lo suficientemente seguro para serlo. Una herramienta especial es suministrada para proteger fácilmente las partes más importantes del paquete. (11)



1.6.2 Embarcadero ER/Studio

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejos.

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. ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes. (16)

1.6.3 Yii Framework

Yii es un framework para PHP orientado a objetos, libre y de alto rendimiento basado en componentes web para desarrollar aplicaciones de gran escala. Se pronuncia en español como se escribe y es un acrónimo para "Yes *It Is!*" (En español: ¡Si este es!). Permite una máxima reusabilidad en la programación web y puede acelerar significativamente el proceso de desarrollo. Yii es un framework genérico para programar Webs que puede ser utilizado para desarrollar virtualmente cualquier tipo de aplicaciones web. Ya que es liviano y está equipado con las soluciones más sofisticadas, está especialmente diseñado para trabajar con aplicaciones web de tráfico alto, como portales, foros, CMS, comercios electrónicos, etc.

Como la mayoría de los framework para PHP, Yii es un framework que sigue el patrón de diseño MVC. Yii sobresale sobre los otros frameworks PHP por su eficiencia y su rica librería de funcionalidades así como también su clara documentación, puesto que es el resultado de la experiencia que tienen sus autores en el desarrollo de aplicaciones

web ricas en funcionalidad y la investigación y reflejo de los frameworks y aplicaciones más populares para programar Webs. (17)

Para crear aplicaciones utilizando Yii, debe seguir un simple proceso compuesto de 3 pasos:

- ✚ Crear la base de datos.
- ✚ Generar el código PHP de la base, utilizando Yii.
- ✚ Modificar el código para adecuarlo a sus necesidades.

Ejemplo de funcionamiento del framework Yii:

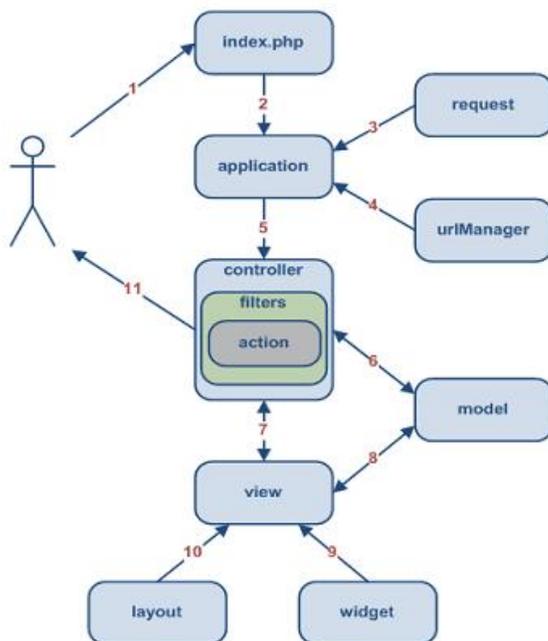


Figura 1.5 Funcionamiento de Yii Framework

1. Un usuario hace una petición en la siguiente URL: <http://www.example.com/index.php?r=post/show&id=1> Y el servidor web maneja la petición ejecutando el *script* index.php.
2. El script crea una instancia de la aplicación y la ejecuta.
3. La aplicación obtiene la información detallada de la petición de los usuarios desde un componente de la aplicación llamado request (petición).



4. La aplicación determina la petición del controlador y actúa con la ayuda de un componente de la aplicación llamado *urlManager* (manejador de url's).
 5. Para este ejemplo, el controlador es *Post* que hace referencia a la clase *PostController*; y la acción es *show* (mostrar) el cuál su significado es determinado por el controlador.
 6. La aplicación crea una instancia de la petición del controlador para más adelante manejar la petición del usuario. El controlador determina que la acción *show* (mostrar) se refiere a un método llamado *actionShow* en la clase del controlador.
 7. Entonces crea y ejecuta filtros (por ej.: controles de acceso) asociados a esta acción. La acción es ejecutada si los filtros lo permiten.
 8. La acción lee un modelo *Post* el cuál si ID es 1 en la base de datos.
 9. La acción asocia una vista llamada *show* (mostrar) con el modelo *Post*.
 10. La vista lee y muestra los atributos del modelo *Post*.
 11. La vista ejecuta algunos artilugios.
 12. Los resultados de la vista son embebidos en un diseño (*layout*)
 13. La acción completa la prestación de la vista y muestra el resultado al usuario.
- (18)

1.6.4 NetBeans IDE

El IDE (Entorno de Desarrollo Integrado) NetBeans es un entorno de desarrollo integrado - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. NetBeans permite crear aplicaciones Web con PHP 5, un potente *debugger* integrado y además viene con soporte para crear aplicaciones HTML5.

El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.



Conclusiones del Capítulo

En este capítulo se ha hecho referencia a los procesos referentes a la gestión de los profesores que cumplen misiones en el extranjero, se detallaron las condiciones y problemas actuales que rodean el objeto de estudio, y por último se han fundamentado las herramientas que se utilizarán para el diseño y desarrollo del producto.

En la implementación del sistema, se utilizó el entorno de programación, **NetBeans 7.3** por las facilidades que brinda para el trabajo con el lenguaje de programación **PHP**, el cual fue seleccionado para el desarrollo de la aplicación por las características ventajosas que brinda en comparación con otros lenguajes de programación. Como servidor Web: **Apache 5.0**, por ser multiplataforma y garantizar que la aplicación a su vez lo sea.

Como Sistema Gestor de Base de Datos se determina el uso de **MySQL 5.0**, por su rapidez cuando se trabaja con una base de datos pequeña o mediana, además de estar totalmente integrado con los entornos de desarrollo seleccionados y el lenguaje de programación escogido y la aplicación se sustentará en una Arquitectura Modelo – Vista – Controlador utilizando el framework de desarrollo **Yii**.



Capítulo 2 Planeación y Diseño

En este capítulo se introduce la fase de planeación y diseño, donde se detallan las necesidades del cliente, se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU). Además se presentará un plan de iteraciones para cada una de las historias aplicando una estimación de esfuerzo de las mismas. Se presentan las llamadas tarjetas CRC y sus principales clases.

Introducción

A continuación se realiza la introducción y desarrollo de las Fases de Planeación y Diseño de la metodología de desarrollo de software XP, la cual se empleó para la realización de este trabajo. En esta fase tiene lugar las reuniones con el cliente, en las cuales se definen las principales funcionalidades del sistema, los requisitos funcionales, se elaboran las historias de usuario, donde se ofrece una breve descripción de cada funcionalidad de la aplicación.

2.1 Funcionalidades Generales

La gestión de la información referente a los estudiantes extranjeros y profesores que salen al extranjero o los que vienen a cursar postgrados en la Dirección de Relaciones Internacionales se realiza de forma manual, guardando las informaciones en planillas en papel o documentos digitales aislados, lo cual hace que el proceso se vuelva lento y engorroso para el especialista encargado de realizarlo, así como que las consultas no sean eficientes, en algunos casos trae como consecuencia pérdida de información y deterioro de la misma. De la descripción anterior se identifican como funcionales generales del sistema guardar los datos de los profesores y estudiante extranjeros así como realizar búsquedas por requisitos.



2.1.1 Personal relacionado con el sistema

Se encargará de la centralización, gestión y emisión de la información manejada en la Dirección de Relaciones Internacionales.

Tabla 2.1: Personal relacionado con el sistema.

Personas relacionadas con el sistema	Justificación
Director	Es el encargado de controlar y actualizar la información de los estudiantes y del personal del Centro en el extranjero.
Secretaria Ejecutiva	Es la encargada de recopilar la información y realizar las planillas.
Desarrollador	Es la persona responsable de llevar a cabo la implementación del sistema.

2.1.2 Funcionalidades del sistema

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

De acuerdo a lo antes expuesto el sistema debe ser capaz de:

1. Autenticar Usuarios.
2. Gestionar usuarios del sistema.
 - a) Insertar usuario al sistema.
 - b) Mostrar usuario del sistema.
 - c) Modificar usuario del sistema.
 - d) Eliminar usuario del sistema.
3. Gestionar estudiante extranjero.
 - a) Insertar estudiante extranjero.
 - b) Mostrar estudiante extranjero.



- c) Modificar estudiante extranjero.
- d) Eliminar estudiante extranjero.
- 4. Gestionar estudiante de pregrado.
 - a) Insertar estudiante de pregrado.
 - b) Mostrar estudiante de pregrado.
 - c) Modificar estudiante de pregrado.
 - d) Eliminar estudiante de pregrado.
- 5. Llenar registro de viajes.
- 6. Mostrar guía del expediente.
- 7. Gestionar profesor del instituto.
 - a) Insertar profesor del instituto.
 - b) Mostrar profesor del instituto.
 - c) Modificar profesor del instituto.
 - d) Eliminar profesor del instituto.
- 8. Gestionar pasaporte.
 - a) Insertar pasaporte.
 - b) Mostrar pasaporte.
 - c) Modificar pasaporte.
 - d) Eliminar pasaporte.
- 9. Mostrar alerta de vencimiento del pasaporte.
- 10. Gestionar convocatoria.
 - a) Insertar convocatoria.
 - b) Mostrar convocatoria.
 - c) Modificar convocatoria.
 - d) Eliminar convocatoria.
- 11. Mostrar perfil de un usuario.
- 12. Exportar el perfil de un usuario.
- 13. Exportar el listado de los usuarios.
- 14. Exportar datos de un profesor.
- 15. Exportar el listado de los profesores.
- 16. Mostrar lista de pasaportes que se vencen.



17. Exportar lista de pasaportes que se vencen.
18. Exportar el listado de los pasaportes.
19. Mostrar un listado de los currículos vitae.
20. Descargar el currículum de un profesor.
21. Mostrar un listado de los registros de viaje.
22. Exportar el listado de los registros de viaje.
23. Exportar el listado de convocatorias.
24. Gestionar beca de postgrado.
 - a) Insertar beca de postgrado.
 - b) Mostrar beca de postgrado.
 - c) Modificar datos de una beca.
 - d) Eliminar beca de postgrado.
25. Exportar un listado de las becas de postgrado.
26. Exportar listado de estudiantes de pregrado.
27. Exportar listado de las solicitudes de visa.
28. Gestionar estudiante de postgrado.
 - a) Insertar estudiante de postgrado.
 - b) Mostrar estudiante de postgrado.
 - c) Modificar estudiante de postgrado.
 - d) Eliminar estudiante de postgrado.
29. Exportar listado de estudiantes de postgrado.

2.1.3 Características del sistema

1. Requerimientos de software.
 - a. Navegador que soporte las características del lenguaje HTML5 y CSS3.
 - b. Servidor web Apache 2.x.
 - c. Servidores de bases de datos MySQL 5.x.
2. Requerimientos de Apariencia e interfaz externa.
 - a. Diseño gráfico no debe ser complejo.
 - b. Utilizar colores que proporcionen una interfaz amigable al usuario.
3. Requerimientos de seguridad



- a. Contar con un control de acceso por roles.
 - b. Asegurar que las funcionalidades solo sean accedidas por las personas con el rol y los permisos establecidos.
4. Requerimientos de usabilidad
- a. Mostrar en cada página un título con el nombre de la sección correspondiente.
 - b. Resaltar en todos los formularios los campos obligatorios con un asterisco o con un texto en rojo.
 - c. Agregar etiquetas a los campos de entrada de datos con una breve descripción del formato que admiten y un ejemplo.
 - d. Verificar la información de todos los campos de entrada de datos, señalar aquellos con datos erróneos y mostrar mensajes que ayuden a corregirlos.

2.1.4 Historias de usuario

Las historias de usuarios (HU), son la técnica utilizada en XP para detallar los requisitos del software. Son el resultado directo del intercambio entre los usuarios y desarrolladores a través de reuniones donde las conocidas *tormenta de ideas* (*brain storm*) arrojan no solo los requerimientos, sino también las posibles soluciones; representan una forma rápida de administrar las necesidades de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para gestionarlos, debido a que un requerimiento de software es descrito de forma concreta y sencilla utilizando el lenguaje común del usuario. Las HU permiten responder ágilmente a los requerimientos cambiantes y aunque se redactan desde las perspectivas de los clientes, también los desarrolladores pueden brindar ayuda en la identificación de las mismas. [[Anexo 1: Historias de usuarios](#)]



Tabla 2.2 HU No. 7 Gestionar profesor del instituto

Historia de usuario	
Número: 7	Usuario: admin
Nombre: Gestionar profesor del instituto.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: José Luis Fernández Cuza	
Descripción: el usuario debe entrar los datos del profesor que desea insertar.	
Observaciones: Confirmado por el cliente.	

2.2 Planificación de entregas

En esta fase se establece la prioridad de cada HU, y a continuación, se realiza una estimación del esfuerzo necesario de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de dos a tres meses.

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo, trabajan el tiempo planeado sin ningún tipo de interrupción, este punto de estimación que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor entre 1 y 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de los puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.



2.2.1 Estimación del esfuerzo por historias de usuario

Para el buen desarrollo de la aplicación se realizó una estimación para cada una de las historias de usuario identificadas, y se obtienen los resultados que se muestran a continuación:

Tabla 2.3 Estimación de esfuerzos por historias de usuario.

Historias de usuarios	Puntos de estimación
Autenticar usuario	0.1
Gestionar usuarios.	1
Gestionar estudiante extranjero	3
Gestionar estudiante de pregrado	1
Llenar registro de viajes	0.1
Mostrar guía del expediente	0.1
Gestionar profesor del instituto	1
Gestionar pasaporte	1
Mostrar alerta de vencimiento del pasaporte	0.2
Gestionar convocatoria	1
Mostrar perfil del usuario	0.1
Exportar el perfil del usuario	0.1
Exportar el listado de los usuarios	0.1
Exportar datos de un profesor	0.1
Exportar el listado de los profesores	0.1
Mostrar lista de pasaportes que se vencen	0.1
Exportar lista de pasaportes que se vencen	0.1
Exportar listado de los pasaportes	0.1
Mostrar un listado de los currículos vitae	0.1
Descargar el currículum de un profesor	0.1
Mostrar un listado de los registros de viaje	0.1
Exportar el listado de los registros de viaje	0.1
Exportar el listado de convocatorias	0.1
Gestionar beca de postgrado	1



Exportar un listado de las becas de postgrado	0.1
Exportar listado de estudiantes de pregrado	0.1
Exportar listado de las solicitudes de visa	0.1
Gestionar estudiante de postgrado	1
Exportar listado de estudiantes de postgrado	0.1

2.2.2 Planificación de iteraciones

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este plan se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU se incluirán en un *release*, lo cual debe ser consistente con el contenido de una o dos iteraciones.

En relación con lo antes tratado se decide realizar el sistema en 4 iteraciones, la cual se explica de forma detallada a continuación:

Primera iteración:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU 1, 2, 5, 6, 7, 8 y 9 las cuales hacen alusión a la autenticación de usuarios y a la inserción de la información de los usuarios y de los profesores del instituto y sus pasaportes. Además se tendrá la primera versión de prueba, que incorporan todas las funcionalidades antes vistas, la cual se presentará al cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

Segunda Iteración:



Esta iteración tiene como finalidad desarrollar las historias de usuario 4, 10, 24, 25, 26 y 27. Las mismas brindan las funcionalidades para gestionar la información de los estudiantes de pregrado, las convocatorias a proyectos internacionales y de becas de postgrado así como exportar los listados con esa información.

Tercera Iteración:

En la tercera iteración se llevaran a cabo las historias de usuario 11 – 23. Con estas se tendrán las funcionalidades para mostrar y exportar a formato PDF los diferentes listados de usuarios, profesores, registros de viaje, convocatorias y currículos vitae.

Cuarta Iteración:

En la cuarta y última iteración se dará cumplimiento a las historias de usuario 3, 28, y 29, con las cuales se obtendrán las funcionalidades para gestionar la información de los estudiantes extranjeros, los estudiantes de postgrado y exportar sus informaciones en formato PDF.

2.2.3 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto guiado por la metodología de desarrollo de software XP, se crea el plan de duración de cada una de las iteraciones que se llevaran a cabo durante el desarrollo del mismo. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas la HU en cada una de las mismas.

Tabla 2.4 Plan de duración de las iteraciones.

Iteración	Historia de usuario	Duración total
1	Autenticar usuarios	20 días
	Gestionar usuarios	
	Mostrar guía del expediente	
	Gestionar profesor del instituto	
	Gestionar pasaporte	
	Mostrar alerta de vencimiento de pasaporte	



	Llenar registro de viaje	
2	Gestionar estudiante de pregrado	18 días
	Gestionar convocatoria	
	Gestionar beca de postgrado	
	Exportar un listado de las becas de postgrado	
	Exportar listado de estudiantes de pregrado	
	Exportar listado de las solicitudes de visa	
3	Mostrar perfil del usuario	13 días
	Exportar perfil de un usuario	
	Exportar el listado de los usuarios	
	Exportar datos de un profesor	
	Exportar el listado de los profesores	
	Mostrar lista de pasaportes que se vencen	
	Exportar lista de pasaportes que se vencen	
	Exportar listado de los pasaportes	
	Mostrar listado de los currículos vitae	
	Descargar el currículum de un profesor	
	Mostrar un listado de los registros de viaje	
	Exportar el listado de los registros de viaje	
	Exportar el listado de las convocatorias	
4	Gestionar estudiante extranjero	21 días
	Gestionar estudiante de postgrado	
	Exportar listado de estudiantes de postgrado	

2.3 Diseño

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como tarjetas CRC, las cuales se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño.



2.3.1 Tarjetas CRC

Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las 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. Para mejor comprensión de las mismas, se determina agruparlas por HU.

A continuación se presenta la tarjeta CRC de la clase Usuario [[Ver Anexo 2: Tarjetas CRC](#)]

Tabla 2.5 Tarjeta CRC No. 7

Nombre de la Clase: Gestionar Profesor	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del profesor del instituto	Insertar_profesor
Muestra los datos del profesor	Mostrar_profesor
Modifica los datos del profesor	Modificar_profesor
Eliminar el profesor de la base de datos	Eliminar_profesor

Conclusiones del Capítulo

Como resultado de este capítulo se abordó la fase de planeación y diseño donde se delinearon las historias de usuarios (HU) con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas. Se presentaron además las principales clases que se utilizarán en el desarrollo de la aplicación a través de las llamadas tarjetas CRC culminando así esta fase y se determina que el equipo de trabajo está listo para pasar a la siguiente etapa de desarrollo.



Capítulo 3 Desarrollo y Pruebas

En este capítulo se siguen las fases de Desarrollo y Pruebas conformadas en la Metodología XP. Se muestra un modelo de datos correspondiente a la aplicación concluyente, así como las interfaces de usuarios diseñadas para la naciente aplicación. Se realiza a través del desarrollo de las iteraciones un desglose de las tareas a realizar para cada una de las HU expuestas en el capítulo anterior. Por último se describen las pruebas realizadas, mostrando las diferentes respuestas al aplicarse cada una de las funcionalidades.

Introducción

En este capítulo se adentra en las últimas fases de la metodología XP Desarrollo y Pruebas de la aplicación. En las cuales se encuentran como encargado el programador en la parte de desarrollo y el cliente en la realización de las pruebas de aceptación.

3.1 Modelo de datos

Después de analizados los requisitos que debe cumplir el sistema y los datos con los que opera, se realizó el modelo de datos, el cual cuenta con las tablas **acceso** y **usuario**, las que almacenarán la información de los usuarios de la aplicación. Entre las tablas más relevantes del modelo se encuentran **profesor** y **estudiante_extranjero**, que son las encargadas de llevar los datos de los profesores que han cumplido o cumplen misión en el exterior, así como de los estudiantes extranjeros que se encuentran cursando estudios en nuestro centro. También se encuentran las tablas **pregrado** y **postgrado** las cuales almacenas el concepto por el cual los estudiantes cursan estudios en el instituto y las tablas **ci** y **pasaporte** que guardan la información de los documentos de profesores y estudiantes.

A continuación mostramos el modelo de datos utilizado en el sistema.

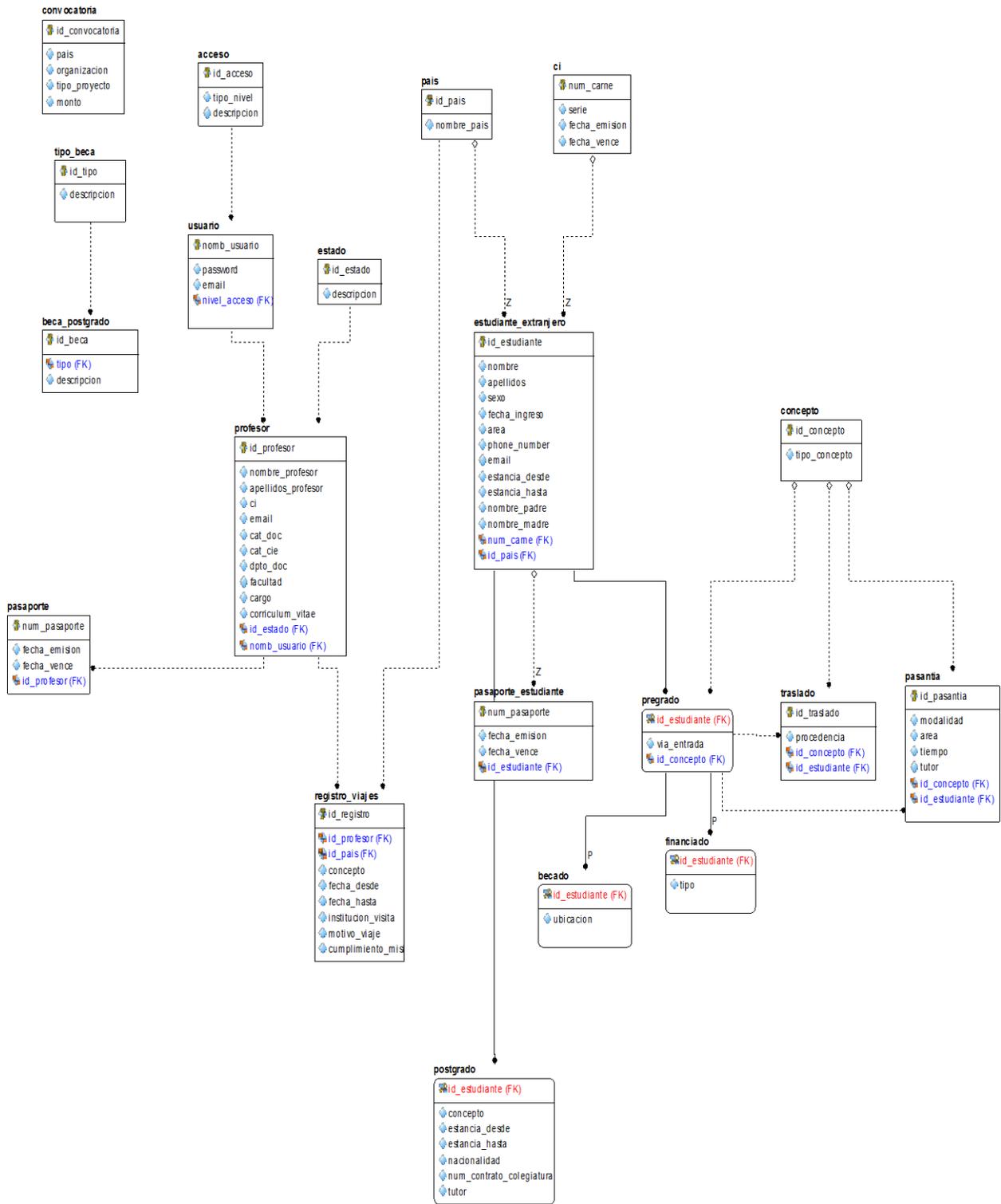


Figura 3.1 Modelo de datos.



3.2 Interfaces de usuario

En el capítulo solo está representada la interfaz de usuario correspondiente a autenticar usuario del sistema, las demás están reflejadas en el [[Anexo 5: Interfaces de usuario](#)]

Inicio » Entrar

Entrar al Sistema Bienvenido

Por favor llene el formulario con sus credenciales:

Usuario *

Password *

Recordarme

Figura 3.2 Interfaz de usuario Autenticar usuario.

3.3 Desarrollo de las iteraciones

En el capítulo anterior se detallaron todas las historias de usuarios de cada una de las iteraciones en la que se va a desarrollar el sistema, y para la realización de cada una de las HU se realizará una revisión al plan de iteraciones, para ver si pueden ocurrir cambios.

3.3.1 Tareas por historias de usuario

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



que pueden ser escritas en lenguaje técnico. Estas se representan mediante las tarjetas de tareas.

Distribución de tareas por historias de usuario

Tabla 3.3 Distribución de tareas por historias de usuario.

Historia de usuario	Tareas
Autenticar usuario	Autenticar usuario
Gestionar usuario	Insertar usuario Mostrar usuario Modificar usuario Eliminar usuario
Gestionar estudiante extranjero	Insertar estudiante extranjero Mostrar estudiante extranjero Modificar estudiante extranjero Eliminar estudiante extranjero
Gestionar estudiante de pregrado	Insertar estudiante de pregrado Mostrar estudiante de pregrado Modificar estudiante de pregrado Eliminar estudiante de pregrado
Llenar registro de viajes	Llenar registro de viajes
Mostrar guía del expediente	Mostrar guía del expediente
Gestionar profesor del instituto	Insertar profesor del instituto Mostrar profesor del instituto Modificar profesor del instituto Eliminar profesor del instituto
Gestionar pasaporte	Insertar pasaporte Mostrar pasaporte Modificar pasaporte Eliminar pasaporte
Mostrar alerta de vencimiento del pasaporte	Mostrar alerta de vencimiento del pasaporte
Gestionar convocatoria	Insertar convocatoria Mostrar convocatoria



	Modificar convocatoria Eliminar convocatoria
Mostrar perfil de un usuario	Mostrar perfil de un usuario
Exportar el perfil de un usuario	Exportar el perfil de un usuario
Exportar el listado de los usuarios	Exportar el listado de los usuarios
Exportar datos de un profesor	Exportar datos de un profesor
Exportar el listado de los profesores	Exportar el listado de los profesores
Mostrar lista de pasaportes que se vencen	Mostrar lista de pasaportes que se vencen
Exportar lista de pasaportes que se vencen	Exportar lista de pasaportes que se vencen
Exportar el listado de los pasaportes	Exportar el listado de los pasaportes
Mostrar un listado de los currículos vitae	Mostrar un listado de los currículos vitae
Descargar el currículum del profesor	Descargar el currículum del profesor
Mostrar lista de los registros de viaje	Mostrar lista de los registros de viaje
Exportar lista de los registros de viaje	Exportar lista de los registros de viaje
Exportar lista de convocatorias	Exportar lista de convocatorias
Gestionar beca de postgrado	Insertar beca de postgrado Mostrar beca de postgrado Modificar beca de postgrado Eliminar beca de postgrado
Exportar un listado de las becas de postgrado	Exportar un listado de las becas de postgrado
Exportar listado de estudiantes de pregrado	Exportar listado de estudiantes de pregrado
Exportar listado de las solicitudes de visa	Exportar listado de las solicitudes de visa
Gestionar estudiante de postgrado	Insertar estudiante de postgrado Mostrar estudiante de postgrado Modificar estudiante de postgrado Eliminar estudiante de postgrado
Exportar listado de estudiantes de postgrado	Exportar listado de estudiantes de postgrado



Historias de usuario abordadas en la Iteración No. 1

Tabla 3.4 Historias de usuario abordadas en la primera iteración.

Historia de usuario	Tiempo de estimación (días)
Autenticar usuarios	1
Gestionar usuarios	5
Mostrar guía del expediente	1
Gestionar profesor del instituto	5
Gestionar pasaporte	5
Mostrar alerta de vencimiento del pasaporte	2
Llenar registro de viaje.	1

Historias de usuario abordadas en la Iteración No. 2

Tabla 3.5 Historias de usuario abordadas en la segunda iteración.

Historia de usuario	Tiempo de estimación (días)
Gestionar estudiante de pregrado	5
Gestionar convocatoria	5
Gestionar beca de postgrado	5
Exportar un listado de becas de postgrado	1
Exportar listado de estudiantes de pregrado	1
Exportar listado de solicitudes de visa	1



Historias de usuario abordadas en la Iteración No. 3

Tabla 3.6 Historias de usuario abordadas en la tercera iteración

Historia de usuario	Tiempo de estimación (días)
Mostrar perfil de un usuario	1
Exportar el perfil de un usuario	1
Exportar el listado de los usuarios	1
Exportar datos de un profesor	1
Exportar el listado de los profesores	1
Mostrar lista de pasaportes que se vencen	1
Exportar lista de pasaportes que se vencen	1
Exportar el listado de los pasaportes	1
Mostrar un listado de los currículos vitae	1
Descargar el currículum de un profesor	1
Mostrar un listado de los registros de viaje	1
Exportar el listado de los registros de viaje	1
Exportar el listado de convocatorias	1

Historias de usuario abordadas en la Iteración No. 4

Tabla 3.7 Historias de usuario abordadas en la cuarta iteración.

Historia de usuario	Tiempo de estimación (días)
Gestionar estudiante extranjero	15
Gestionar estudiante de postgrado	5
Exportar listado de estudiantes de postgrado	1



3.3.2 Tareas de ingeniería

En XP generalmente cada historia se divide en tareas de ingeniería o tareas de programación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Estas se crean para obtener una mejor planificación y cumplir con las funcionalidades básicas que luego conformarán las funcionalidades generales de cada historia de usuario [[Anexo 3: Tareas de ingeniería](#)].

Tabla 3.8 Tarea de programación Autenticar usuario.

Tarea de Programación	
No. Tarea: 16	No. Historia: 7
Nombre de la tarea: insertar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 10/02/2014	Fecha fin: 11/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite la entrada de los usuarios al sistema.	

3.4 Pruebas

En la metodología XP las pruebas juegan un papel fundamental, pues esta permite la comprobación continua del código. El desarrollo constante de las pruebas da lugar a que se desarrolle un software con mayor calidad dando una mayor seguridad de lo que se está haciendo.

En la metodología escogida hay dos tipos de prueba; las unitarias o desarrollo dirigido por pruebas, 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.



3.4.1 Pruebas de aceptación

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en otras metodologías de desarrollo, 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 la HU 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. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los arquetipos anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve a cabo el proceso mediante las pruebas de aceptación.

La planilla utilizada para plasmar el contenido de las pruebas de aceptación se muestra a continuación. [[Anexo 4: Pruebas de Aceptación](#)].

Tabla 3.9 Prueba de aceptación HU # 7 Gestionar profesores.

Prueba de aceptación
Historia de usuario # 7: Gestionar profesores
Nombre: Prueba para comprobar la entrada de datos del profesor del instituto.
Descripción: Validación de entrada de los datos del profesor.
Condiciones de ejecución: El usuario debe introducir los datos de un profesor.
Entrada/Pasos de ejecución: El usuario escribe los datos del profesor.
Resultado esperado: <ul style="list-style-type: none"> ➤ Si se insertan los datos correctamente se creará un nuevo registro en la tabla profesor Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"> ➤ Se dejen campos obligatorios vacíos.
Evaluación de la prueba: Aceptada.



Conclusiones del Capítulo

Con el desarrollo de este capítulo se llevó a cabo la fase de desarrollo y diseño donde se presenta el modelo de datos de la aplicación a obtener, logrando una visión detallada de sus atributos y las relaciones entre sus clases, además, las principales clases mediante el empleo de las tarjetas CRC. Se realiza el desarrollo de las iteraciones a partir de la distribución de tareas de ingeniería por HU, y se les practica las pruebas de aceptación a las funcionalidades de mayor importancia.



Capítulo 4 Estudio de Factibilidad

En la actualidad para realizar de manera satisfactoria el desarrollo de cualquier proyecto se hace necesario el estudio de factibilidad para tener en cuenta una valoración de los costos a incurrir determinando si será factible o no desarrollar el mismo.

En general los productos informáticos no están exentos de posibles riesgos en la concepción del proyecto, por lo que es válido minimizar de forma razonable recursos humanos, materiales y financieros, de ahí que es de vital importancia estimar la relación costo – beneficio, así como el esfuerzo, capital humano y el tiempo de desarrollo que se demanda en la ejecución de los mismos.

En este capítulo se expone el estudio de factibilidad del proyecto, centrado en estimaciones de esfuerzo humano, tiempo de desarrollo para su ejecución y costo. Se presentarán los efectos económicos, los beneficios y costos intangibles.

4.1 Evaluación Costo – Beneficio

Para el estudio de la factibilidad de este proyecto fue utilizada 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: (5)

Costo: que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados.

Eficiencia: 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). (6)

Esta es una de las partes más importantes en la elaboración de cualquier proyecto ya que haciendo un correcto estudio de factibilidad se puede ahorrar meses e incluso años de trabajo, hasta evitar poner en duda la reputación profesional si se realiza un sistema mal planificado desde una etapa temprana.



4.2 Efectos Económicos

- ✚ Efectos directos
- ✚ Efectos indirectos
- ✚ Efectos externos
- ✚ Intangibles

4.2.1 Efectos directos:

Positivos:

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

Negativos:

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

4.2.2 Efectos indirectos:

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



4.2.3 Efectos externos:

- ✚ Se obtendrá un producto disponible que le facilitará gran parte del trabajo a los usuarios encargados de la gestión de la información referente a los estudiantes extranjeros y profesores en el ISMMM.

4.2.4 Intangibles

En la estimación económica siempre hay elementos como perjuicio o beneficio, pero al momento de darle valor en unidades monetarias esto resulta difícil o prácticamente imposible. (6)

A fin de medir con precisión los efectos, deberán considerarse dos situaciones: la situación sin proyecto y la situación con proyecto.

Situación sin proyecto (Solución manual):

Son complejas las actividades de llenar los datos del profesor que va a viajar, del estudiante extranjero que está becado en residencia estudiantil o de las notas de este, ya que no se cuenta con la infraestructura informática para que el profesor acceda a sus datos y los llene él mismo o que se tengan registrados todos los resultados del estudiante en sus carreras, trayendo consigo no tener una vía rápida de operación y análisis de la información.

Situación con proyecto (Solución automatizada)

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

Costos:

- ✚ Resistencia al cambio

**Beneficios:**

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

4.3 Ficha de costo de un producto informático.

Para determinar el costo económico del proyecto se utilizará el Procedimiento para elaborar una ficha de costo de un producto informático. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional. (19)

Costos en Moneda Libremente Convertible:

✚ Costos Directos

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

Total: \$ 64.94 CUC

✚ Costos indirectos

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

Total: \$0.00.



+ Gastos de distribución y venta.

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

Total: \$0.00.

Total General: \$ 64.94 CUC.

Costos en Moneda Nacional:

+ Costos Directos.

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

+ Costos indirectos

1. Know How: \$ 0.0.
Total: \$ 145.99.

Total General: \$ 145.99

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

Compara los beneficios y los costos de un proyecto particular y si los primeros exceden a los segundos entrega un elemento de juicio inicial, que indica su aceptabilidad. Mientras que el análisis costo-efectividad sigue la misma lógica, compara los costos con las potencialidades de alcanzar más eficientemente los objetivos no expresables en moneda; si no en productos. Para esta técnica es imprescindible definir una variable



directa que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado por los especialistas para realizar las actividades de gestión de la información y la variable sería, la complejidad de las pruebas que se desarrollan durante este proceso.

Valores de la variable (Solución manual)

1. Gestionar información de los profesores. (10 min)
2. Gestionar información de los estudiantes extranjeros. (11 min)
3. Buscar información de los pasaportes que se vencen. (8 min)
4. Crear reportes. (15 min)

Valores de la variable (Solución con el sistema)

1. Gestionar información de los profesores. (2 min)
2. Gestionar información de los estudiantes extranjeros. (2 min)
3. Buscar información de los pasaportes que se vencen. (0.4 min)
4. Crear Reportes. (1 min).

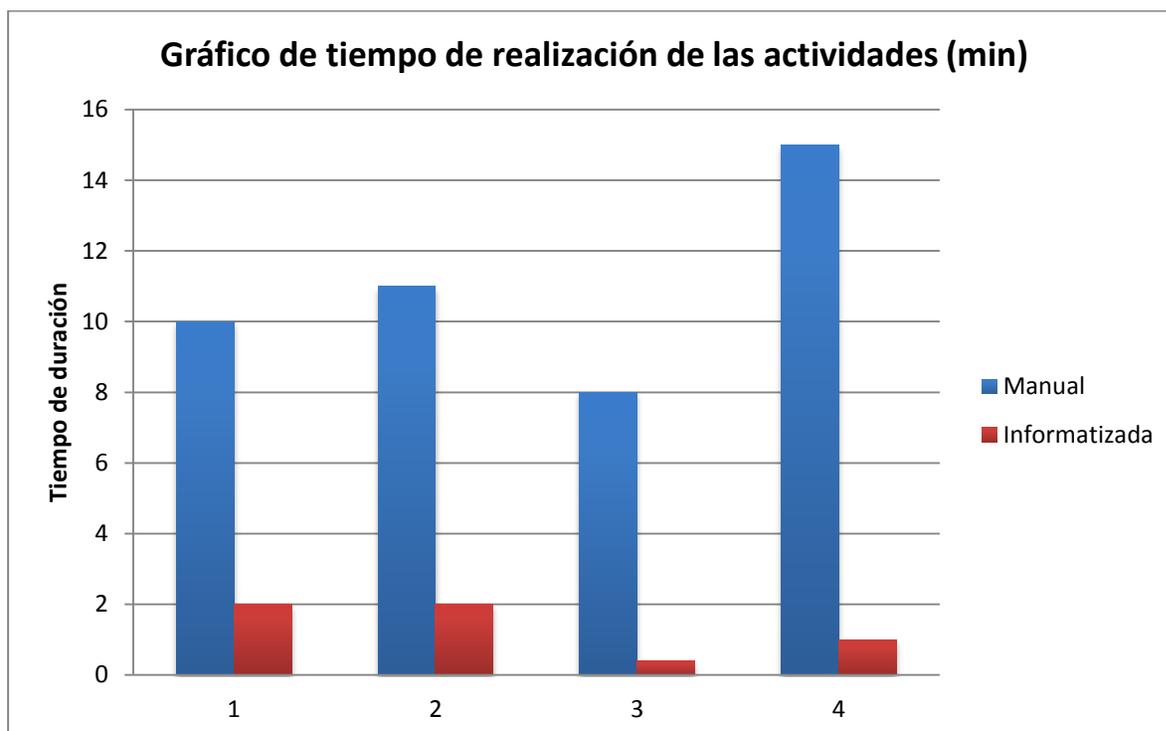




FIGURA 4.1 GRÁFICA DE LA SOLUCIÓN SIN EL PRODUCTO Y SOLUCIÓN CON EL PRODUCTO.

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

Conclusiones del Capítulo

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



Conclusiones Generales

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

- ✚ Se elaboró el marco teórico metodológico que fundamenta la investigación, esto permitió que quedara identificada la situación problemática existente y las bases para comenzar con el diseño e implementación de la aplicación.
- ✚ Se realizó un levantamiento sobre los procesos que se llevan a cabo en la dirección de relaciones internacionales del ISMMM.
- ✚ Durante el desarrollo de la metodología XP se generaron 58 requisitos funcionales, 31 historias de usuario, 10 tarjetas CRC, 11 pruebas de aceptación y 58 tareas de ingeniería.
- ✚ Se efectuó un estudio de la factibilidad técnica, económica, y operativa del software en cuestión; arrojando resultados satisfactorios para el proyecto.

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



Recomendaciones

El autor de la presente investigación ha considerado pertinente ofrecer las siguientes recomendaciones:

- ✚ Realizar la implantación del software en el ISMMM.
- ✚ Ejecutar la aplicación preferentemente en los navegadores Mozilla Firefox ya que traduce los estilos de diseño con mayor claridad.
- ✚ Realizar un estudio más profundo de este sistema en vista a perfeccionarlo en versiones futuras.
- ✚ Realizar encuestas de satisfacción a los usuarios finales que permitan realizarle nuevas versiones a la Aplicación Web teniendo en cuenta las recomendaciones hechas por ellos.



Referencias Bibliográficas

1. **Robles, R. García.** El nuevo paradigma de la gestión del conocimiento y su aplicación en el ámbito educativo. [Online] [Cited: 06 02, 2014.] <http://tecnologiedu.us.es/edutec/paginas/125.htm..>
2. Intranet Universidad de Guantánamo. [Online] <http://intranet.cug.co.cu>.
3. **Universidad de la Habana.** matcom. *Facultad de Matemática y Computación.* [Online] Casa de Software. [Cited: 05 28, 2014.] <http://www.matcom.uh.cu>.
4. Intranet Universidad de Holguín. [Online] <http://intranet.uho.edu.cu>.
5. **Lecusay, Pablo.** *Sistema informático para favorecer el proceso de selección de personal en el instituto superior minero metalúrgico de moa.* Moa : ISMMM, 2012.
6. **Silva, Ing. Exneider proenza.** *Sistema de Costos de Calidad para la Empresa .* 2010.
7. **Ferrer, J.** [Online] 02 25, 2003. [Cited: 02 15, 2014.] <http://libresoft.es/downloads/ferrer-20030312.pdf>.
8. **www.postgresql.org/support/professional_support.** [Online] [Cited: 02 18, 2014.] http://www.postgresql.org/support/professional_support.
9. **www.microsoft.com/spain/sql/default.msp.** [Online] [Cited: 02 18, 2014.] <http://www.microsoft.com/spain/sql/default.msp>.
10. MySQL: Ventajas. [Online] [Cited: 06 8, 2014.] <http://sistemaspyt.blogspot.com/2008/09/ventajas.html>.
11. **Meriño, Pricido Fidel Rusiaux.** *Herramienta Informática para la gestión de información sobre Incidentes y Accidentes Ambientales en la Empresa Che Guevara.* Moa : ISMMM, 2013.



12. **Jeffries, R.** What is Extreme Programming? [Online] [Cited: 02 16, 2014.]
<http://www.xpprogramming.com/xpmag/whatisxp.htm>.
13. **Bañeres, Juan Palacio.** Gestión ágil de proyectos: Scrum. *Navegapolis.net*.
[Online] <http://www.navegapolis.net>.
14. **Pressman, Roger S.** *Ingeniería de Softwar, un enfoque práctico*. s.l. : McGraw-Hill, 2011.
15. **Gabriel Duarte.** Preguntale al Gabo. [Online] [Cited: 04 14, 2014.]
<http://gabrielduarte77.wordpress.com/category/yii-framework>.
16. EmbarcaderoER/Studio. [Online] [Cited: 02 25, 2014.]
http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=115375597.
17. **Chile, Fundación GNU.** Taller introducción Framework Yii. [Online]
<http://www.fundaciongnuchile.cl>.
18. **Duarte, Gabriel.** Guía básica de yiiframework. [Online] 10 2011. [Cited: 02 20, 2014.] <http://leninmhs.files.wordpress.com/2011/10/guia-basica-de-yii-framework.pdf>.
19. Costo-Efectividad. [Online] [Cited: 06 01, 2014.]
www.crid.or.cr/digitalizacion/pdf/spa/doc9321-02.pdf.
20. **www.oracle.com.** [Online] <http://www.oracle.com>.
21. *Presentación-XP.pdf*. **Fernández Escribano, Gerardo.** 2002.
22. Yii-Bootstrap - Bringing together the Yii PHP framework and Twitter's Bootstrap.htm.
[Online] <http://www.yiiframework.com>.
23. **Pérez, Agustín Rodríguez.** *Sistema para La Gestión del Presupuesto*. ISMM : s.n., 2012.



Bibliografía

1. [BECK, 1999]. Kent Beck. *Extreme Programming Explained: Embrace Change*, Pearson Education 1999. Addison-Wesley, 2000.
2. [BEIGHLEY; MORRISON, 2005]. Beighley, Lynn, Morrison, Michael. *Head First PHP & MySQL*. O'Reilly, 2005.
3. [CANOS; LETELIER; PENADES, 2005]. H. Canós, José, Letelier, Patricio, Penadés, M. Carmen. *Metodologías Ágiles en el desarrollo de software*. Universidad Politécnica de Valencia, 2005.
4. Calvo Burke, Aurelio. *Sistema de Gestión de Información sobre Derrames de Hidrocarburos en el puerto de Moa*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", Moa, julio 2013.
5. [DUARTE, 2011]. Gabriel Duarte. *Guía básica de Yii Framework*. España, 2011.
6. [FERNANDEZ, 2002]. Gerardo Fernández Escribano. *Introducción a Extreme Programming*. Ingeniería de Software II. 9-12-2002.
7. [FREEMAN, 2005]. Elisabeth Freeman. *A Brain-Friendly Guide to HTML & CSS*. *Head First Html with Css and Xhtml*. O'Reilly, 2005.
8. GDP. *Manual de PHP*. Grupo de Documentación de PHP, 2002.
9. [HERNANDEZ, 2006]. Edelsys Hernández Meléndrez. *Como escribir una tesis*. Escuela Nacional de Salud Pública, 2006.
10. Hernández Luque, Eylin. *Aplicación Web para la gestión de información de instrumentos meteorológicos del Instituto de Meteorología*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Universidad de las Ciencias Informáticas, Ciudad de la Habana, mayo 2007.
11. [LETELIER; PENADES, 2005]. Letelier, Patricio; Penadés, M^a Carmen. *Metodologías ágiles para el desarrollo de software: Extreme Programming (XP)*. Universidad Politécnica de Valencia, 2005.
12. *Manual de SQL*. WebExperto.COM, 2001.
13. [MARTEENS, 1999]. Ian Marteens. *La cara oculta de C++ Builder*. Madrid, 1999.
14. Mesa Martínez, Jorge I. *Modelos de formularios con Yii*. <http://blog.jorgeivanmeza.com/category/desarrollo-de-software> (20/02/14).
15. Peñaloza B, Leisewitz T, Bastías G, Zárata V, Depaux R, Villarroel L, et al. *Metodología para la evaluación de la relación costo-efectividad en centros de atención primaria de Chile*. *Revista Panam. Salud Pública*. 2010.
16. Rodríguez Pérez, Agustín. *SIGEP: Sistema para La Gestión del Presupuesto en el ISMMM*. Trabajo de Diploma para optar por el título de Ingeniero



- Informático, Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, junio 2012.
17. Salazar Pupo, Ricardo. Sistema de gestión de capacitación ECRIN. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, julio 2013.
 18. Schenone Marcelo Hernán *Diseño de una Metodología Ágil de Desarrollo de Software*. Tesis de Grado en Ingeniería en Informática. Universidad de Buenos Aires, argentino, abril 2004.
 19. Suárez Falcón, Yuniel. Sistema de gestión de información para la estimación y mitigación de riesgos de desastres en la Provincia Holguín. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, julio 2013.
 20. [WELLING; THOMSON, 2004]. Welling Luke, Thomson, Laura. Desarrollo Web con PHP y MySQL. Anaya Multimedia, 2004.
 21. Yii Framework. [online]. www.yiiframework.com (15/04/14).



Glosario de Términos

Actualización: Cambios que se aplican a un programa, para corregir errores, agregarle funcionalidades, etc.

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

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

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

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

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

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

Beca: Subvención para realizar estudios o investigaciones.



Becado: Persona que disfruta de una beca.

Becario Extranjero: Refiérase a aquel estudiante de otro país que cursa estudios de pregrado en el Instituto Superior Minero Metalúrgico de Moa y se encuentran becados en las instalaciones del centro.

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

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

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

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

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

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

Firebird: Sistema de administración de base de datos relacional (o RDBMS) (Lenguaje consultas: SQL) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++.

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

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



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

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

IDE: Integrated Development Environment / Entorno de Desarrollo Integrado. Entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

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

Interfaz: Conjunto de representaciones de operaciones públicas.

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

Múltiples iteraciones contribuyen a crear un producto completamente integrado.

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

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

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



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

Metodología Ágil: Constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

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

Metodologías tradicionales: Metodologías basadas en procesos.

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

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

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

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

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

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



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

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

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

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

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

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

Servidor Web: Un programa que corre sobre el servidor que escucha las peticiones HTTP (Hyper Text Markup Language) que le llegan y las satisface. Dependiendo del tipo de petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición

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

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

Usuario: Persona que interactúa con el sistema



Anexo 1: Historias de usuarios

Tabla 2.6 HU No. 1 Autenticar usuario

Historia de usuario	
Número: 1	Usuario: Cliente
Nombre: Autenticar usuarios.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: José Luis Fernández Cuza	
Descripción: Los usuarios del sistema ingresan sus datos para entrar al sistema (nombre de usuario y contraseña). El sistema verifica que los datos estén correctos, en caso de que no sean correctos la aplicación muestra un mensaje de error.	
Observaciones: Confirmado por el cliente.	

Tabla 2.7 HU No. 1 Gestionar Usuario

Historia de usuario	
Número: 2	Usuario: Administrador
Nombre: Gestionar usuarios sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: José Luis Fernández Cuza	
Descripción: el usuario debe entrar los datos del requeridos después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	


Tabla 2.8 HU No. 3 Gestionar estudiante extranjero

Historia de usuario	
Número: 3	Usuario: Administrador
Nombre: Gestionar estudiante extranjero.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Cuarta
Programador responsable: José Luis Fernández Cuza	
Descripción: el usuario debe entrar los datos del requeridos después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.9 HU No. 4 Gestionar estudiante de pregrado

Historia de usuario	
Número: 4	Usuario: Administrador
Nombre: Gestionar estudiante de pregrado.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: José Luis Fernández Cuza	
Descripción: el usuario debe entrar los datos del requeridos después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.10 HU No. 5 Llenar registro de viaje

Historia de usuario	
Número: 5	Usuario: Administrador
Nombre: Llenar registro de viaje.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: José Luis Fernández Cuza	
Descripción: el usuario debe entrar los datos del requeridos después podrá realizar las	



operaciones correspondientes.
Observaciones: Confirmado por el cliente.

Tabla 2.11 HU No. 6 Mostrar guía del expediente

Historia de usuario	
Número: 6	Usuario: profesor
Nombre: Mostar guía del expediente	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: José Luis Fernández Cuza	
Descripción: el sistema muestra un listado con la guía del expediente del profesor luego el usuario puede descargarlos.	
Observaciones: Confirmado por el cliente.	

Tabla 2.12 HU No. 7 Gestionar profesor del instituto.

Historia de usuario	
Número: 7	Usuario: Administrador
Nombre: Gestionar profesor del instituto.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: José Luis Fernández Cuza	
Descripción: el usuario debe entrar los datos del requeridos después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.13 HU No. 8 Gestionar pasaporte.

Historia de usuario	
Número: 8	Usuario: Administrador
Nombre: Gestionar pasaporte.	



Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: José Luis Fernández Cuza	
Descripción: el usuario debe entrar los datos del requeridos después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.14 HU No. 9 Mostrar alerta de vencimiento del pasaporte.

Historia de usuario	
Número: 9	Usuario: profesor
Nombre: Mostrar alerta de vencimiento del pasaporte.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: José Luis Fernández Cuza	
Descripción: el sistema muestra un mensaje de advertencia cuando la fecha de vencimiento del pasaporte está por cumplirse.	
Observaciones: Confirmado por el cliente.	

Tabla 2.15 HU No. 10 Gestionar convocatoria.

Historia de usuario	
Número: 10	Usuario: Administrador
Nombre: Gestionar convocatoria.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: José Luis Fernández Cuza	
Descripción: el usuario introduce los datos requeridos, luego podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	



Tabla 2.16 HU No. 11 Mostrar perfil de un usuario

Historia de usuario	
Número: 11	Usuario: TODOS
Nombre: Mostrar perfil de un usuario.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: Tercera
Programador responsable: José Luis Fernández Cuza	
Descripción: el sistema muestra los datos del usuario y este puede hacer las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.17 HU No. 12 Exportar perfil de un usuario.

Historia de usuario	
Número: 12	Usuario: TODOS
Nombre: Exportar perfil de un usuario.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: Tercera
Programador responsable: José Luis Fernández Cuza	
Descripción: el sistema muestra los datos del usuario y permite que este lo descargue en formato pdf.	
Observaciones: Confirmado por el cliente.	

Tabla 2.18 HU No. 13 Exportar listado de los usuarios.

Historia de usuario	
Número: 13	Usuario: Administrador
Nombre: Exportar listado de los usuarios.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: Tercera
Programador responsable: José Luis Fernández Cuza	
Descripción: el sistema muestra los datos de todos los usuarios y permite ser descargado en	



formato pdf.
Observaciones: Confirmado por el cliente.

Tabla 2.19 HU No. 14 Exportar datos de un profesor

Historia de usuario	
Número: 14	Usuario: TODOS
Nombre: Exportar datos un profesor.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: Tercera
Programador responsable: José Luis Fernández Cuza	
Descripción: el sistema muestra los datos del profesor y permite ser descargado en formato pdf.	
Observaciones: Confirmado por el cliente.	



Anexo 2: Tarjetas CRC

Tabla 2.20 Tarjeta CRC No. 1

Nombre de la Clase: Autenticar Usuario	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Autentica al usuario en el sistema.	login

Tabla 2.21 Tarjeta CRC No. 2

Nombre de la Clase: Gestionar Usuario	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del usuario	Insertar_usuario
Muestra los datos del usuario	Mostrar_usuario
Modifica los datos del usuario	Modificar_usuario
Eliminar el usuario de la base de datos	Eliminar_usuario
Lista y exporta los datos de un usuario	Listar_usuario

Tabla 2.22 Tarjeta CRC No. 3

Nombre de la Clase: Gestionar Estudiante Extranjero	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del estudiante	Insertar_estudiante
Muestra los datos del estudiante	Mostrar_estudiante
Modifica los datos del estudiante	Modificar_estudiante
Eliminar el estudiante de la base de datos	Eliminar_estudiante
Lista y exportar los datos del estudiante	Listar_estudiante



Tabla 2.23 Tarjeta CRC No. 4

Nombre de la Clase: Gestionar Estudiante de pregrado	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del estudiante de pregrado	Insertar_profesor_ext
Muestra los datos del estudiante de pregrado	Mostrar_profesor_ext
Modifica los datos del estudiante de pregrado	Modificar_profesor_ext
Eliminar el estudiante de la base de datos	Eliminar_profesor_ext
Lista y exporta los datos del estudiante de pregrado	Listar_profesor

Tabla 2.24 Tarjeta CRC No. 5

Nombre de la Clase: Gestionar Pasaporte	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del pasaporte	Insertar_pasaporte
Muestra los datos del pasaporte	Mostrar_pasaporte
Modifica los datos del pasaporte	Modificar_pasaporte
Eliminar el pasaporte de la base de datos	Eliminar_pasaporte
Lista y exporta los pasaportes	Listar_pasaporte
Lista, exporta y muestra alerta de vencimiento	Alerta_vencimiento

Tabla 2.25 Tarjeta CRC No. 6

Nombre de la Clase: Gestionar Convocatoria	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos de la convocatoria	Insertar_convocatoria



Muestra los datos de la convocatoria	Mostrar_convocatoria
Modifica los datos de la convocatoria	Modificar_convocatoria
Eliminar la convocatoria de la base de datos	Eliminar_convocatoria
Lista y exportar la convocatoria	Listar_convocatoria

Tabla 2.26 Tarjeta CRC No. 7

Nombre de la Clase: Gestionar Profesor	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del profesor del instituto	Insertar_profesor
Muestra los datos del profesor	Mostrar_profesor
Modifica los datos del profesor	Modificar_profesor
Eliminar el profesor de la base de datos	Eliminar_profesor
Lista y exporta datos de los profesores	Listar_profesor

Tabla 2.27 Tarjeta CRC No. 8

Nombre de la Clase: Gestionar Beca de postgrado	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos de la beca	Insertar_beca
Muestra los datos de la beca	Mostrar_beca
Modifica los datos de la beca	Modificar_beca
Eliminar la beca de la base de datos	Eliminar_beca
Lista y exporta datos de becas de postgrado	Listar_beca



Tabla 2.28 Tarjeta CRC No. 9

Nombre de la Clase: Gestionar estudiante de postgrado	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del estudiante de postgrado	Insertar_estudiante_postgrado
Muestra los datos del estudiante de postgrado	Mostrar_estudiante_postgrado
Modifica los datos del estudiante de postgrado	Modificar_estudiante_postgrado
Eliminar el estudiante de postgrado de la base de datos	Eliminar_estudiante_postgrado
Lista y exporta los estudiantes de postgrado	Listar_estudiante_postgrado



Anexo 3: Tareas de ingeniería

Tabla 3.10 Tarea de Programación autenticar usuario

Tarea de Programación	
No. Tarea: 1	No. Historia: 1
Nombre de la tarea: Autenticar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 22/01/2014	Fecha fin: 22/01/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite autenticar los usuarios al sistema.	

Tabla 3.11 Tarea de Programación Insertar usuario

Tarea de Programación	
No. Tarea: 2	No. Historia: 2
Nombre de la tarea: Insertar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 23/01/2014	Fecha fin: 23/01/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite insertar los usuarios que van a interactuar con el sistema.	

Tabla 3.12 Tarea de Programación Mostrar usuario

Tarea de Programación	
No. Tarea: 3	No. Historia: 2
Nombre de la tarea: Mostrar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 24/01/2014	Fecha fin: 24/01/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite mostrar los datos de los usuarios del sistema.	


Tabla 3.13 Tarea de Programación Modificar usuario

Tarea de Programación	
No. Tarea: 4	No. Historia: 2
Nombre de la tarea: Modificar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 25/01/2014	Fecha fin: 26/01/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite modificar los datos de los usuarios del sistema.	

Tabla 3.14 Tarea de Programación Eliminar usuario

Tarea de Programación	
No. Tarea: 5	No. Historia: 2
Nombre de la tarea: Eliminar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 27/01/2014	Fecha fin: 27/01/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite eliminar el usuario seleccionado de la base de datos.	

Tabla 3.15 Tarea de Programación insertar estudiante extranjero

Tarea de Programación	
No. Tarea: 6	No. Historia: 3
Nombre de la tarea: insertar estudiante extranjero	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 20/05/2014	Fecha fin: 20/05/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite insertar el estudiante en la base de datos.	



Tabla 3.16 Tarea de Programación mostrar estudiante extranjero

Tarea de Programación	
No. Tarea: 7	No. Historia: 3
Nombre de la tarea: mostrar estudiante extranjero	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 21/05/2014	Fecha fin: 21/05/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite mostrar los datos del estudiante en la base de datos.	

Tabla 3.17 Tarea de Programación modificar estudiante extranjero

Tarea de Programación	
No. Tarea: 8	No. Historia: 3
Nombre de la tarea: modificar estudiante extranjero	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 22/05/2014	Fecha fin: 23/05/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite modificar el estudiante en la base de datos.	

Tabla 3.18 Tarea de Programación eliminar estudiante extranjero

Tarea de Programación	
No. Tarea: 9	No. Historia: 3
Nombre de la tarea: eliminar estudiante extranjero	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 24/05/2014	Fecha fin: 24/05/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite eliminar el estudiante de la base de datos.	

**Tabla 3.19 Tarea de Programación insertar estudiante de pregrado**

Tarea de Programación	
No. Tarea: 10	No. Historia: 4
Nombre de la tarea: insertar estudiante de pregrado	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 20/04/2014	Fecha fin: 20/04/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite insertar el estudiante en la base de datos.	

Tabla 3.20 Tarea de Programación mostrar estudiante de pregrado

Tarea de Programación	
No. Tarea: 11	No. Historia: 4
Nombre de la tarea: mostrar estudiante de pregrado	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 21/04/2014	Fecha fin: 21/04/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite mostrar el estudiante en la base de datos.	

Tabla 3.21 Tarea de Programación modificar estudiante de pregrado

Tarea de Programación	
No. Tarea: 12	No. Historia: 4
Nombre de la tarea: modificar estudiante de pregrado	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 22/04/2014	Fecha fin: 23/04/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite modificar el estudiante en la base de datos.	



Tabla 3.22 Tarea de Programación eliminar estudiante de pregrado

Tarea de Programación	
No. Tarea: 13	No. Historia: 4
Nombre de la tarea: eliminar estudiante de pregrado	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 24/04/2014	Fecha fin: 24/04/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite eliminar el estudiante en la base de datos.	

Tabla 3.23 Tarea de Programación llenar registro de viajes del profesor.

Tarea de Programación	
No. Tarea: 14	No. Historia: 5
Nombre de la tarea: llenar registro de viaje	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 02/02/2014	Fecha fin: 02/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite llenar el registro de viaje del profesor.	

Tabla 3.24 Tarea de Programación mostrar guía del expediente.

Tarea de Programación	
No. Tarea: 15	No. Historia: 6
Nombre de la tarea: mostrar guía del expediente	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 03/02/2014	Fecha fin: 03/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite mostrar la guía del expediente.	



Tabla 3.25 Tarea de Programación insertar profesor.

Tarea de Programación	
No. Tarea: 16	No. Historia: 7
Nombre de la tarea: insertar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 04/02/2014	Fecha fin: 04/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite insertar un profesor del instituto.	

Tabla 3.26 Tarea de Programación Mostrar profesor.

Tarea de Programación	
No. Tarea: 17	No. Historia: 7
Nombre de la tarea: mostrar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 05/02/2014	Fecha fin: 05/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite mostrar un profesor del instituto.	

Tabla 3.27 Tarea de Programación modificar profesor.

Tarea de Programación	
No. Tarea: 18	No. Historia: 7
Nombre de la tarea: modificar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 06/02/2014	Fecha fin: 07/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite modificar un profesor del instituto.	



Tabla 3.28 Tarea de Programación eliminar profesor.

Tarea de Programación	
No. Tarea: 19	No. Historia: 7
Nombre de la tarea: eliminar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 09/02/2014	Fecha fin: 09/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite eliminar un profesor del instituto.	

Tabla 3.29 Tarea de Programación insertar pasaporte.

Tarea de Programación	
No. Tarea: 20	No. Historia: 8
Nombre de la tarea: insertar pasaporte.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 10/02/2014	Fecha fin: 10/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite insertar un pasaporte.	

Tabla 3.30 Tarea de Programación mostrar pasaporte.

Tarea de Programación	
No. Tarea: 21	No. Historia: 8
Nombre de la tarea: mostrar pasaporte.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 11/02/2014	Fecha fin: 11/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite mostrar un pasaporte.	



Tabla 3.31 Tarea de Programación modificar pasaporte.

Tarea de Programación	
No. Tarea: 22	No. Historia: 8
Nombre de la tarea: modificar pasaporte.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 11/02/2014	Fecha fin: 12/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite modificar un pasaporte.	

Tabla 3.32 Tarea de Programación eliminar pasaporte.

Tarea de Programación	
No. Tarea: 23	No. Historia: 8
Nombre de la tarea: eliminar pasaporte.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 13/02/2014	Fecha fin: 13/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite eliminar un pasaporte.	

Tabla 3.33 Tarea de Programación mostrar alerta de vencimiento de pasaporte.

Tarea de Programación	
No. Tarea: 24	No. Historia: 9
Nombre de la tarea: mostrar alerta de vencimiento del pasaporte.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 14/02/2014	Fecha fin: 15/02/2014
Programador responsable: José Luis Fernández Cuza	
Descripción: Permite mostrar un alerta de vencimiento del pasaporte.	



Anexo 4: Pruebas de Aceptación

Tabla 3.34 PA: Prueba para comprobar la entrada de un usuario al sistema.

Prueba de aceptación
Historia de usuario # 1: Autenticar usuarios
Nombre: Prueba para comprobar la entrada de un usuario al sistema.
Descripción: Validación de entrada de los datos del usuario.
Condiciones de ejecución: El usuario debe introducir su nombre de usuario y contraseña.
Entrada/Pasos de ejecución: El usuario escribe su nombre de usuario y contraseña y luego pulsa el botón "Entrar".
<p>Resultado esperado:</p> <ul style="list-style-type: none"> ➤ Si el usuario tiene acceso para entrar a la aplicación e inserta sus datos correctamente entrara sin problemas al sistema. <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none"> ➤ Se inserten los datos de un usuario no valido para el sistema o datos incorrectos. ➤ Se dé clic en el botón Entrar sin insertar nada en los campos de texto.
Evaluación de la prueba: Aceptada.

Tabla 3.35 PA: Prueba para comprobar la gestión de usuarios.

Prueba de aceptación
Historia de usuario # 2: Gestionar usuarios
Nombre: Prueba para comprobar la gestión de usuarios.
Descripción: Validación de gestión de usuarios.
Condiciones de ejecución: El usuario debe entrar a la aplicación y tener el rol de Administrador para poder crear, modificar, listar y eliminar usuarios.
Entrada/Pasos de ejecución: El administrador escribe los datos del usuario que se piden para crear el usuario luego presiona el botón Crear. Después que el usuario esté insertado se podrá eliminar, modificar y acceder a sus datos en el listado de usuarios.
Resultado esperado:



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

Se emite un mensaje de error en caso de que:

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

Evaluación de la prueba: Aceptada.

Tabla 3.36 PA: Prueba para comprobar la gestión estudiantes extranjeros.

Prueba de aceptación
Historia de usuario # 3: Gestionar estudiante extranjero
Nombre: Prueba para comprobar la gestión de estudiantes extranjeros.
Descripción: Validación de la gestión de estudiantes extranjeros.
Condiciones de ejecución: El usuario debe estar autenticado con rol de Administrador para introducir los datos de un estudiante extranjero, modificarlo, acceder al listado y eliminarlo.
Entrada/Pasos de ejecución: El Administrador escribe los datos que se piden del estudiante y pulsa el botón Insertar. Después que el estudiante esté insertado se podrá eliminar, modificar y acceder a ella en el listado de estudiantes.
Resultado esperado: <ul style="list-style-type: none"> ➤ Si se introducen los datos correctamente se insertará el estudiante en la base de datos. ➤ Si se desea eliminar el estudiante se muestra un mensaje de confirmación.
Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"> ➤ Se dejen campos obligatorios vacíos. ➤ Se inserten letras en campos que deben ser numéricos. ➤ Se inserte una dirección de correo inválida. ➤ Las fechas de inicio sean mayores que las finales. ➤ La longitud de la serie del carnet sea menor que lo especificado. ➤ Se inserte un número de carnet o pasaporte que ya exista.
Evaluación de la prueba: Aceptada.



Tabla 3.37 PA: Prueba para comprobar la gestión de estudiantes de pregrado

Prueba de aceptación
Historia de usuario # 4: Gestionar estudiante de pregrado
Nombre: Prueba para comprobar la gestión estudiante de pregrado.
Descripción: Validación de entrada de los datos del estudiante de pregrado.
Condiciones de ejecución: El usuario debe estar autenticado con rol de Administrador para introducir los datos de un estudiante de pregrado, modificarlo y acceder al listado.
Entrada/Pasos de ejecución: El Administrador accede a la interfaz de gestionar estudiantes de pregrado, donde puede modificar y acceder a la información que se muestra.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> ➤ Si se introducen los datos correctamente se modificará el estudiante. ➤ Si se desea eliminar el estudiante se muestra un mensaje de confirmación. <p>Se emite un mensaje de error en caso de que:</p> <ul style="list-style-type: none"> ➤ Se dejen campos obligatorios vacíos.
Evaluación de la prueba: Aceptada.

Tabla 3.38 PA: Prueba para comprobar el registro de viajes

Prueba de aceptación
Historia de usuario # 5: Llenar registro de viaje.
Nombre: Prueba para comprobar el registro de viajes.
Descripción: Validación de del registro de viajes.
Condiciones de ejecución: El usuario debe estar autenticado con rol de Administrador, acceder a la vista <i>Visualizar Profesor</i> y pinchar en el enlace <i>Registrar Viaje</i> . Luego debe introducir los datos necesarios. Una vez insertado podrá modificarlo, eliminarlo o acceder a la información en el listado.
Entrada/Pasos de ejecución: El Administrador selecciona a un profesor y hace clic en <i>Registrar Viaje</i> , luego escribe los datos que se piden para registrar el viaje.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> ➤ Si se insertan los datos correctamente se registrará el viaje.



- Si se desea eliminar se muestra un mensaje de confirmación.

Se emite un mensaje de error en caso de que:

- Se dejen campos obligatorios vacíos.
- La fecha de inicio es mayor que la final.

Evaluación de la prueba: Aceptada.

Tabla 3.39 PA: Prueba para comprobar la entrada de datos del profesor del instituto

Prueba de aceptación
Historia de usuario # 7: Gestionar profesores
Nombre: Prueba para validar la gestión de los profesores del instituto.
Descripción: Validación de la gestión de los profesores del instituto.
Condiciones de ejecución: El usuario debe estar autenticado con rol de Administrador para introducir los datos de un profesor modificarlo o eliminarlo así como acceder a su información.
Entrada/Pasos de ejecución: El Administrador escribe todos los datos del profesor y luego presiona el botón <i>Crear</i> . Después que los datos estén insertados es que se puede eliminar, modificar y acceder a los datos del profesor.
Resultado esperado: <ul style="list-style-type: none"> ➤ Si se insertan los datos correctamente se creará un nuevo registro en la tabla profesor. ➤ Si se desea eliminar se muestra un mensaje de confirmación.
Se emite un mensaje de error en caso de que: <ul style="list-style-type: none"> ➤ Se dejen campos obligatorios vacíos. ➤ Se inserte una dirección de correo inválida.
Evaluación de la prueba: Aceptada.

Tabla 3.40 PA: Prueba para comprobar la entrada de datos del pasaporte

Prueba de aceptación
Historia de usuario # 8: Gestionar pasaporte
Nombre: Prueba para comprobar la validación de datos del pasaporte
Descripción: Validación de entrada de los datos del pasaporte.



Condiciones de ejecución: El usuario debe estar autenticado como Administrador, luego introduce los datos correspondientes al pasaporte.

Entrada/Pasos de ejecución: El Administrador escribe los datos que se piden sobre el pasaporte.

Resultado esperado:

- Si se insertan los datos correctamente se inserta el pasaporte.
- Si se desea eliminar se muestra un mensaje de confirmación

Se emite un mensaje de error en caso de que:

- Se dejen campos obligatorios vacíos.
- Que la fecha de vencimiento sea menor que la de emisión.

Evaluación de la prueba: Aceptada.

Tabla 3.41 PA: Prueba para comprobar la gestión de estudiantes de postgrado.

Prueba de aceptación

Historia de usuario # 28: Gestionar estudiante de postgrado.

Nombre: Prueba para comprobar la entrada de datos del estudiante de postgrado.

Descripción: Validación de entrada de los datos del estudiante de postgrado.

Condiciones de ejecución: El usuario introduce los datos correspondientes.

Entrada/Pasos de ejecución: El usuario debe estar autenticado como administrador para acceder a la interfaz de Gestionar estudiantes de postgrado, luego escribe los datos del estudiante de postgrado, una vez insertados podrá eliminar, modificar o acceder a los datos en el listado.

Resultado esperado:

- Si se insertan los datos correctamente se inserta el estudiante de postgrado.
- Si se desea eliminar se muestra un mensaje de confirmación

Se emite un mensaje de error en caso de que:

- Se dejen campos obligatorios vacíos.
- La fecha de inicio sea mayor que la final.

Evaluación de la prueba: Aceptada.

Anexo 5: Interfaces de usuario

Profesores *Insertar Profesor*

Campos con * son obligatorios.

Usuario *	<input type="text" value="admin"/>
Nombre *	<input type="text" value="nombre profesor"/>
Apellidos *	<input type="text" value="apellidos profesor"/>
CI *	<input type="text" value="carnet de identidad"/>
Email	<input type="text" value="correo electronico"/>
Categoría Docente	<input type="text" value="Seleccione Categoría"/>
Categoría Científica	<input type="text" value="Seleccione Categoría"/>
Dpto Docente	<input type="text" value="Seleccione Departamento"/>
Facultad	<input type="text" value="Seleccione Facultad"/>
Cargo	<input type="text" value="cargo"/>
Curriculum Vitae	<input type="text"/> <input type="button" value="Examinar..."/>

Operaciones

- Listar
- Administrar

Figura 3.3 Interfaz de usuario Insertar profesor

Profesores *Administrar información de los profesores*

Opcionalmente puede entrar un operador de comparación (<, <=, >, >=, <> o =) al inicio de cada consulta para especificar como debe realizarse la comparación.

Avanzado

Viendo 1-3 de 3 resultados.

Nombre	Apellidos	CI	Email	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Roilandy	Torres Fonseca	90125425654	rtorres@info.ismm.edu.cu	  
Yuris	Acosta Zayas	88101725264	yaacosta@info.ismm.edu.cu	  
Rafael	Rodriguez Rignack	90125425654	rwilliams@info.ismm.edu.cu	  

Operaciones

- Listar
- Crear
- Administrar
- Descargar PDF

Figura 3.4 Interfaz de usuario administrar profesor

Estudiantes *Administrar estudiantes*

Opcionalmente puede entrar un operador de comparación (<, <=, >, >=, <> o =) al inicio de cada consulta para especificar como debe realizarse la comparación.

Avanzado

Viendo 1-4 de 4 resultados.

Nombre	Apellidos	Correo	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Nina	Dobrev Tayler	nina.dobrev@gmail.com	  
Luisao	Mtongo Mtongo	lmtongo@ismm.edu.cu	  
Luis	Almenteros	lusi@hotmail.com	  
Roger	Pressman	rpressman@gmail.com	  

Operaciones

- Listar
- Insertar
- Descargar PDF

Figura 3.5 Interfaz de usuario Administrar estudiantes extranjeros.

Estudiantes [Insertar estudiantes](#)

Nombre <input type="text" value="nombre"/>	Apellidos <input type="text" value="apellidos"/>
Sexo <input type="text" value="Femenino"/>	Fecha Ingreso <input type="text" value="fecha_ingreso"/>
Telefono <input type="text" value="# de telefono"/>	Area <input type="text" value="area a la que pertenece"/>
Pais <input type="text" value="Afganistán"/>	Correo <input type="text" value="correo electronico"/>
Estancia Desde <input type="text" value="estancia desde"/>	Nombre del Padre <input type="text" value="nombre del padre"/>
Estancia Hasta <input type="text" value="estancia hasta"/>	Nombre de la Madre <input type="text" value="nombre de la madre"/>
Num Carne <input type="text" value="# del carne"/>	# Serie <input type="text" value="# de serie"/>
Fecha Emision <input type="text" value="fecha de emision"/>	Fecha Vence <input type="text" value="fecha de vencimiento"/>
Num Pasaporte <input type="text" value="# del pasaporte"/>	Concepto <input type="text" value="--Seleccione--"/>
Emission <input type="text" value="fecha de emision"/>	Fecha Vence <input type="text" value="fecha de emision"/>

Operaciones

- [Listar](#)
- [Administrar](#)

Figura 3.6 Interfaz de usuario insertar estudiante extranjero.

Estudiantes *Modificar estudiantes*

Nombre	Apellidos
<input type="text" value="Nina"/>	<input type="text" value="Dobrev Tayler"/>
Sexo	Fecha Ingreso
<input type="text" value="Femenino"/>	<input type="text" value="10-02-2010"/>
Telefono	Area
<input type="text" value="45127894265"/>	<input type="text" value="Cultura"/>
Pais	Correo
<input type="text" value="Bulgaria"/>	<input type="text" value="nina.dobrev@gmail.com"/>
Estancia Desde	Nombre del Padre
<input type="text" value="19-06-2014"/>	<input type="text" value="Antof Dobrev"/>
Estancia Hasta	Nombre de la Madre
<input type="text" value="24-06-2016"/>	<input type="text" value="Katerina Dobreva"/>
Num Carne	# Serie
<input type="text" value="89010912425"/>	<input type="text" value="L12457869"/>
Fecha Emision	Fecha Vence
<input type="text" value="20-06-2013"/>	<input type="text" value="18-06-2020"/>
Num Pasaporte	Concepto
<input type="text" value="571668422554"/>	<input type="text" value="Pregrado"/>
Emision	Fecha Vence
<input type="text" value="26-06-2014"/>	<input type="text" value="24-06-2015"/>

Operaciones

- Listar
- Insertar
- Ver
- Administrar

Figura 3.7 Interfaz de usuario Modificar estudiante extranjero.



Entrar al Sistema Bienvenido

Por favor llene el formulario con sus credenciales: ✕

Usuario *

Password * Usuario o contraseña incorrectos.

Recordarme

Figura 3.8 Interfaz de usuario Autenticar usuario. Error de autenticación.

Estudiantes Insertar estudiantes

Por favor corrija los siguientes errores de ingreso:

- Num Carne no puede estar vacío.
- Telefono debe ser un número.
- Debe seleccionar un concepto
- Correo no es una dirección de email válida.
- La fecha de inicio no puede ser mayor a la final

Operaciones

- [Listar](#)
- [Administrar](#)

Nombre Apellidos

Sexo Fecha Ingreso

Telefono Area
Telefono debe ser un número.

Pais Correo
Correo no es una dirección de email válida.

Figura 3.9 Interfaz de usuario insertar usuario



Pasaportes *Pasaportes que estan en riesgo de vencimiento*

Estudiantes que deben renovar el pasaporte ×

# Pasaporte	Nombre	Apellidos	Email	Fecha Expira
456789123004	Luisao	Mtongo Mtongo	lmtongo@ismm.edu.cu	04/06/2015
712366459961	Luis	Almenteros	lusi@hotmail.com	26/07/2014
424511345676	Roger	Pressman	rpressman@gmail.com	09/06/2016

Operaciones

- Inicio
- Descargar PDF

Figura 3.10 Interfaz de usuario Listar pasaportes por vencerse.

Carnet de Identidad *Carnet de Identidad que estan en riesgo de vencimiento*

Estudiantes que deben renovar el carnet de identidad ×

# Carnet	Nombre	Apellidos	Email	Fecha Expira
95032514256	Luisao	Mtongo Mtongo	lmtongo@ismm.edu.cu	09/04/2018
75425688455	Luis	Almenteros	lusi@hotmail.com	26/06/2014

Operaciones

- Inicio
- Descargar PDF

Figura 3.11 Interfaz de usuario Listar carnet de identidad por vencerse.