

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

Título: Sistema de gestión de información para las actividades del Colectivo de Año en el Departamento de Informática del ISMMM.

Autor (es): Yuris Angel Acosta Zayas

Tutor (es): Ing. Exneyder Proenza Silva
Lic. Yelins de la Cruz Navarro

Moa, 2014
“Año 56 de la Revolución”

Tabla de contenido

Introducción.....	1
Capítulo 1. Fundamentación Teórica.....	6
1.1 Introducción.....	6
1.2 El Colectivo de Año y su Trabajo Metodológico.....	6
1.2.1 Principales funciones.....	6
1.3 Niveles de dirección.....	7
1.4 Funcionamiento del eslabón de base como sistema en la labor metodológica.	7
1.5 Tendencias y tecnologías actuales	8
1.5.1 Aplicaciones Web.....	8
1.5.2 Ventajas de una web	9
1.5.3 Servidores web.....	10
1.5.4 Sistemas de gestión de bases de datos.....	11
1.5.5 Herramientas CASE	16
1.5.6 Lenguajes de programación.....	16
1.6 Metodologías de desarrollo	18
1.6.1 Metodología propuesta para el desarrollo de la aplicación.....	18
1.6.2 Programación Extrema (Extreme Programming, XP)	19
1.6.3 Valores que promueve XP.....	19
1.6.4 El ciclo de vida ideal de XP	20
1.6.5 Planificación incremental	20
1.6.6 Testing	21
1.6.7 Fases de la Metodología XP	24
1.6.8 Scrum.....	27

1.7	Patrones Arquitectónicos	34
1.7.1	Arquitectura	35
1.8	Herramientas a utilizar en la propuesta de solución.....	37
1.8.1	XAMPP	37
1.8.2	Macromedia Dreamweaver 8.0.....	38
1.8.3	Embarcadero ER/Studio	39
Capítulo 2 Planeación y Diseño.....		41
2.1	Funcionalidades Generales	41
2.1.1	<i>Personal relacionado con el sistema</i>	41
2.1.2	<i>Funcionalidades del sistema</i>	42
2.1.3	<i>Características del sistema</i>	45
2.1.4	<i>Historias de usuarios</i>	45
2.2	Planificación de entregas	46
2.2.1	<i>Estimación del esfuerzo por historias de usuario</i>	47
2.2.2	<i>Planificación de iteraciones</i>	48
2.2.3	<i>Plan de duración de las iteraciones</i>	49
2.3	Diseño	50
2.3.1	<i>Tarjetas CRC</i>	50
Capítulo 3 Desarrollo y Pruebas.....		52
3.1	Modelo de datos	52
3.2	Interfaces de usuario	53
3.3	Desarrollo de las iteraciones	53
3.3.1	Tareas por historias de usuario	53
3.3.2	Tareas de ingeniería.....	57
3.4	Pruebas.....	58

3.4.1	Pruebas de aceptación	58
Capítulo 4	Estudio de Factibilidad.....	61
4.1	Evaluación Costo – Beneficio.....	61
4.2	Efectos Económicos	62
4.2.1	Efectos directos:	62
4.2.2	Efectos indirectos:.....	62
4.2.3	Efectos externos:	63
4.2.4	Intangibles	63
4.3	Ficha de costo de un producto informático.	64
Conclusiones	Generales	68
Recomendaciones	69
Referencia bibliográfica.....		70
Bibliografía		71
Anexo 1:	Historias de usuarios	I
Anexo 2:	Tarjetas CRC	X
Anexo 3:	Tareas de ingeniería	XV
Anexo 4:	Pruebas de Aceptación	XXXII
Anexo 5:	Interfaces de usuario	XXXV

Índice de tablas

Tabla 2.1: Personal relacionado con el sistema	42
Tabla 2.2 HU No. 1: Gestión usuarios.....	46
Tabla 2.3 Estimación de esfuerzo por HU	47
Tabla 2.4 Plan de duración de las iteraciones.	49
Tabla 2.5 Tarjeta CRC No. 1 Usuario	50
Tabla 3.3 Distribución de tareas por historias de usuario	54
Tabla 3.4 Historias de usuario abordadas en la primera iteración.	55
Tabla 3.5 Historias de usuario abordadas en la segunda iteración.....	56
Tabla 3.6 Historias de usuario abordadas en la tercera iteración.....	56
Tabla 3.7 Historias de usuario abordadas en la cuarta iteración	57
Tabla 3.8 Tarea de programación autenticar usuario	58
Tabla 3.9 Tarea de aceptación hu #2 gestionar profesor.....	59
Tabla 2.6 HU No. 1 Autenticar usuario	I
Tabla 2.7 HU No. 2 gestionar profesores.....	I
Tabla 2.8 HU No.3 gestionar estudiantes	I
Tabla 2.9 HU No. 4 gestionar actividades	II
Tabla 2.10 HU No. 5 gestionar tutor	II
Tabla 2.11 HU No. 6 gestionar profesor guía	III
Tabla 2.12 HU No. 7 gestionar asignatura.....	III
Tabla 2.13 HU No. 8 gestionar administrador	IV
Tabla 2.14 HU No. 9 gestionar evaluación	IV
Tabla 2.15 HU No. 10 exportar reporte de evaluaciones por semestre	IV
Tabla 2.16 HU No. 11 exportar reporte de evaluaciones por año	V
Tabla 2.17 HU No. 12 exportar reporte de estudiantes	V
Tabla 2.18 HU No. 13 gestionar documentos	VI
Tabla 2.19 HU No. 14 buscar estudiantes	VI
Tabla 2.20 HU No.15 buscar coordinador de año	VI
Tabla 2.21 HU No.16 buscar tutor.....	VII
Tabla 2.22 HU No.17 buscar profesor guía	VII

Tabla 2.23 HU No.18 buscar jefe de departamento	VIII
Tabla 2.24 HU No.19 buscar jefe de carrera	VIII
Tabla 2.25 HU No.20 modificar contraseña.....	IX
Tabla 2.26 Tarjeta CRC No. 1	X
Tabla 2.27 Tarjeta CRC No. 2	X
Tabla 2.28 Tarjeta CRC No. 3	X
Tabla 2.29 Tarjeta CRC No. 4	XI
Tabla 2.30 Tarjeta CRC No. 5	XI
Tabla 2.31 Tarjeta CRC No. 6	XI
Tabla 2.32 Tarjeta CRC No. 7	XII
Tabla 2.33 Tarjeta CRC No. 8	XII
Tabla 2.34 Tarjeta CRC No. 9	XII
Tabla 2.35 Tarjeta CRC No. 10	XIII
Tabla 2.36 Tarjeta CRC No. 11	XIII
Tabla 2.37 Tarjeta CRC No. 12	XIV
Tabla 3.10 Tarea de Programación autenticar profesor	XV
Tabla 3.11 Tarea de Programación Insertar profesor	XV
Tabla 3.12 Tarea de Programación Mostrar profesor	XV
Tabla 3.13 Tarea de Programación Modificar profesor	XV
Tabla 3.14 Tarea de Programación Eliminar profesor	XVI
Tabla 3.15 Tarea de Programación insertar estudiante	XVI
Tabla 3.16 Tarea de Programación mostrar estudiante	XVI
Tabla 3.17 Tarea de Programación modificar estudiante	XVII
Tabla 3.18 Tarea de Programación eliminar estudiante	XVII
Tabla 3.19 Tarea de Programación insertar actividades	XVII
Tabla 3.20 Tarea de Programación mostrar actividades.....	XVIII
Tabla 3.21 Tarea de Programación modificar actividades	XVIII
Tabla 3.22 Tarea de Programación eliminar actividades	XVIII
Tabla 3.23 Tarea de Programación insertar profesor guía.	XIX
Tabla 3.24 Tarea de Programación mostrar profesor guía.	XIX
Tabla 3.25 Tarea de Programación modificar profesor guía.	XIX

Tabla 3.26 Tarea de Programación eliminar profesor guía.	XX
Tabla 3.27 Tarea de Programación insertar asignatura.	XX
Tabla 3.28 Tarea de Programación mostrar asignatura.	XX
Tabla 3.29 Tarea de Programación modificar asignatura.	XX
Tabla 3.30 Tarea de Programación eliminar asignatura.	XXI
Tabla 3.31 Tarea de Programación salvar curso.	XXI
Tabla 3.32 Tarea de Programación insertar administrador.	XXI
Tabla 3.33 Tarea de Programación mostrar administrador.	XXII
Tabla 3.34 Tarea de Programación modificar administrador.	XXII
Tabla 3.35 Tarea de Programación eliminar administrador.	XXII
Tabla 3.36 Tarea de Programación insertar semestre.	XXIII
Tabla 3.37 Tarea de Programación mostrar semestre.	XXIII
Tabla 3.38 Tarea de Programación modificar semestre.	XXIII
Tabla 3.39 Tarea de Programación eliminar semestre.	XXIV
Tabla 3.40 Tarea de Programación insertar evaluación.	XXIV
Tabla 3.41 Tarea de Programación mostrar evaluación.	XXIV
Tabla 3.42 Tarea de Programación modificar evaluación.	XXIV
Tabla 3.43 Tarea de Programación eliminar evaluaciones.	XXV
Tabla 3.44 Tarea de Programación Exportar listado de evaluaciones por semestre.	XXV
Tabla 3.45 Tarea de Programación Exportar listado de evaluaciones por año. ...	XXV
Tabla 3.46 Tarea de Programación Exportar listado de estudiantes por año. ...	XXVI
Tabla 3.47 Tarea de Programación Exportar un listado de estudiantes en general.	XXVI
Tabla 3.48 Tarea de Programación Exportar listado de estudiantes de pregrado.	XXVI
Tabla 3.49 Tarea de Programación Exportar listado los profesores.	XXVII
Tabla 3.50 Tarea de Programación Insertar documentos.	XXVII
Tabla 3.51 Tarea de Programación mostrar documentos.	XXVII
Tabla 3.52 Tarea de Programación modificar documentos.	XXVIII
Tabla 3.53 Tarea de Programación eliminar documentos.	XXVIII

Tabla 3.54 Tarea de Programación buscar estudiante	XXVIII
Tabla 3.55 Tarea de Programación buscar coordinador de año	XXIX
Tabla 3.56 Tarea de Programación buscar tutor	XXIX
Tabla 3.57 Tarea de Programación buscar profesor guía	XXIX
Tabla 3.58 Tarea de Programación buscar jefe de departamento	XXX
Tabla 3.59 Tarea de Programación buscar jefe de carrera	XXX
Tabla 3.60 Tarea de Programación modificar contraseña	XXX
Tabla 6.61 PA: Prueba para comprobar la entrada de un usuario al sistema. ..	XXXII
Tabla 3.62 PA: Prueba para comprobar la entrada de datos del usuario	XXXII
Tabla 3.63 PA: Prueba para comprobar la entrada de datos del estudiante.	XXXIII
Tabla 3.64 PA: Prueba para comprobar la entrada de datos de las actividades	XXXIII
Tabla 3.65 PA: Prueba para comprobar la entrada de datos de la asignatura	XXXIV

Índice de imágenes

Figura 1.1 Patrón arquitectónico tres capas	36
Figura 3.1 Modelo de datos	52
Figura 3.2 Interfaz de usuario Autenticar usuario	53
Figura 4.1 Gráfica de la solución sin el producto y solución con el producto.	66
Figura 3.3 Interfaz de usuario Insertar profesor	XXXV
Figura 3.4 Interfaz de usuario administrar profesor	XXXV
Figura 3.5 Interfaz de usuario gestionar actividades	XXXVI
Figura 3.6 Interfaz de usuario administrar actividades	XXXVI
Figura 3.7 Interfaz de usuario gestionar estudiantes	XXXVI
Figura 3.8 Interfaz de usuario administrar estudiantes	XXXVII

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.

Yuris Angel Acosta Zayas

Firma del Autor

Ing. Exneyder Proenza Silva

Firma del Tutor

Lic. Yelins de la Cruz Navarro

Firma del Tutor

Pensamiento

“La ciencia es la expresión de una necesidad inherente al ser humano y, en todo caso, está ligada a la función superior de su naturaleza inteligente: la capacidad de crear.”

René Gerónimo Favaleoro

Dedicatoria

A mis queridos padres, Juan Manuel Cedeño Sánchez y Delaida Zayas Céspedes por su confianza, cariño, dedicación, y la fuerza necesaria para culminar con mi carrera.

A la memoria de mi abuela Berja Luisa Céspedes por ser el pilar y soporte de todos mis momentos.

A la memoria de mi padre Víctor M. Acosta por su empeño, confianza y amor.

A mis hermanos, por ser las personas más especiales y únicas en la faz de la tierra.

A Daimara Chapman Fonseca por brindarme esta felicidad que aún me llena y a las amistades que nos rodearon todo este tiempo.

A mi familia completa por brindarme su comprensión y su afecto, sin dejar de mencionar de forma especial a mis tías Dolores Zayas y Reyna Cedeño, y las atenciones de mis sobrinos y demás familiares.

A mis amigos y hermanos del alma, José Luis Fernández Cuza, Reydel Carralero Belett y Juan Ramón La ' O por siempre estar ahí, dándome mucho fuerza y apoyo.

A mis tutores Exneider Proenza y Yelins de la Cruz por su ayuda en todo el proceso.

Gracias

Agradecimiento

A mis queridos padres Juan Manuel Cedeño Sánchez y Delaida Zayas Céspedes por su confianza, cariño, dedicación, y la fuerza necesaria para culminar con mi carrera.

A mis hermanos Yusel, Yuliana y Yusmani por estar pendientes a mi crecimiento como profesional.

A Daimara, a su familia, a sus amistades que son mías y a Nieves por ser una persona especial en mi corazón.

A mis compañeros de estudio y a todos aquellos que aguantaron mis altas y bajas.

A mis tutores Exneider y Yelins por brindarme una mano guía para enseñarme el camino a transitar.

Al Colectivo del Departamento de Ingeniería Informática en el ISMM.

A todos los que de una forma u otra hicieron posible que esto sucediera.

A todos ustedes les brindo mi entero agradecimiento.

Resumen

El colectivo de año se encarga de llevar a cabo el trabajo metodológico en este nivel organizativo. Agrupa a los profesores que imparten las asignaturas del año, a los profesores guías de cada grupo, tutores y a los representantes de las organizaciones estudiantiles. Este tiene como propósito lograr el cumplimiento con calidad de los objetivos del año, propiciando la integración de los aspectos educativos e instructivos con un enfoque interdisciplinario.

En el Departamento de Ingeniería Informática de la facultad Geología – Minas del ISMMM; existe la necesidad de desarrollar una herramienta informática que favorezca el trabajo del Colectivo de Año; elaborando una base de datos para almacenar la información y realizar búsquedas con mayor velocidad y exactitud.

Por esta razón se necesita gestionar un gran volumen de información, tenerlo actualizado y que esté disponible para consultarlo cuando sea necesario. Este proceso se realiza de forma manual y es guardado en documentos impresos. Por tanto se requiere de una forma más organizada de almacenar los datos para evitar su deterioro o pérdida; aunque algunas de estas informaciones se guardan en archivos digitales no cumplen con la seguridad requerida.

Para la elaboración de la misma se realizó un estudio bibliográfico sobre las aplicaciones Web, las reglas correspondientes al colectivo de año y se presenta la metodología de desarrollo de software que se siguió.

Abstract

The group this year is responsible for carrying out the methodological work in this organizational level. It brings together teachers who teach the subjects of the year, teachers guides each group, guardians and representatives of student organizations. This is intended to ensure compliance with quality objectives for the year, promoting the integration of educational and instructional issues with an interdisciplinary approach.

In the Department of Computer Engineering Faculty of Geology - ISMMM Mines; there is a need to develop a software tool that encourages the work of the Collective of the Year; developing a database to store information and perform searches with greater speed and accuracy.

For this reason you need to manage a large volume of information, to have it updated and available for reference when needed. This process is performed in hard format which is stored in printed documents. Therefore requires a more organized way of storing data to prevent deterioration or loss; although some of this information are made into digital files they do not meet safety requirements.

For the preparation of the same bibliographic study of Web applications was made, the group of the year for the rules and the methodology that was followed for software development is proposed as a solution to the problem found is presented.

Introducción

La implementación de las ciencias informáticas en el mundo tiene un considerable empleo en numerables áreas de desempeño social, manifestándose como parte de las nuevas tecnologías de mayor uso en la actualidad. El hombre en su constante desarrollo ha experimentado prósperos resultados en el uso de los sistemas informatizados; éstos controlan, gestionan, procesan y brindan información; representando una significativa importancia en el alcance del desarrollo científico-tecnológico y económico a nivel global.

En el mundo de hoy no se conciben entidades que no empleen las Tecnologías de la Información y las Comunicaciones (TIC). Estas son capaces de aportar soluciones eficientes en cualquier esfera empresarial; permitiendo actuar con rapidez, obtener el máximo rendimiento del personal y tomar decisiones a la administración. 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.

Una buena gestión de la información permite tomar decisiones acertadas en el momento preciso, si contamos con aplicaciones desarrolladas en computadoras que ofrezcan un acceso rápido y fácil a través del equipo humano, explotando los conocimientos y la tecnología informática actual.

Cuba, siendo aún un país subdesarrollado, ha logrado potencializar la formación de profesionales a través de la Educación Superior, ganando en prestigio y ejemplo ante el mundo. Las universidades en el mundo y en la isla funcionan de manera estructurada, garantizando la formación integral y la superación continua de profesionales en las ciencias técnicas, económicas y sociales; el desarrollo de investigaciones científicas, así como en acciones de innovación tecnológica y gestión del conocimiento. El Instituto Superior Minero Metalúrgico de Moa (ISMMM) de la provincia Holguín no está ajeno a ello.

SIGA

La organización del proceso docente educativo en este centro de altos estudios, implica de manera directa la labor que realiza para este fin el Colectivo de Año de cada carrera. Éste es el encargado de llevar a cabo el trabajo metodológico. Agrupa a los profesores guías de cada grupo y a los que desarrollan las asignaturas del año, a los tutores y a los representantes de las organizaciones estudiantiles. Todos en conjunto realizan una ardua labor educativa que se consolida durante los cinco años de la especialidad; ello se deriva del control sistemático de los profesores hacia los estudiantes; los cuales son evaluados en la marcha del proceso docente educativo y del cumplimiento de los proyectos educativos.

Se ha conocido sobre la existencia de una herramienta informática desarrollada por el Ministerio de Educación Superior (MES) con el fin de facilitar el trabajo de la Secretaría General en varias universidades del país: el Sistema de Gestión de la Nueva Universidad (SIGENU). El cual solo funciona como un sistema de gestión académica, basada en la información de un egresado desde que se matricula hasta que se gradúa o causa baja definitiva. Entre sus funcionalidades se encuentra la inscripción de un estudiante, asignaturas a cursar, registro de evaluaciones, control de bajas, emisión de reportes oficiales; entre otros. Sin embargo, su acceso es limitado; solo las secretarías docentes tienen acceso a ella, no así los profesores de los diferentes colectivos de año de cada carrera ni los directivos de sus respectivos departamentos.

El Departamento de la carrera de Ingeniería Informática del Instituto en estos momentos no cuenta con un sistema de gestión de la información que facilite la labor metodológica realizada en cada año referente a las actividades que en este se realizan, así como para la gestión de la información docente de sus estudiantes. Por lo que su trabajo carece de eficiencia y en ocasiones es engorroso, debido a lo lento que son los pasos para darle finalidad a sus procesos que a su vez provoca la pérdida de documentos en formato duro y por ello extravío de información. Por tal motivo dicho departamento se vio en la necesidad de informatizar el control y gestión de estas actividades a través de una herramienta.

Yuris Angel Acosta Zayas

SIGA

Por lo planteado anteriormente y conociendo la necesidad del Departamento de Ingeniería Informática, de desarrollar una herramienta informática que favorezca esta labor, se plantea como **problema científico** en esta investigación: ¿Cómo favorecer el proceso de gestión de la información del trabajo del Colectivo de Año del Departamento de Ingeniería Informática en el ISMMM?

Para darle solución a este problema, el **objeto de estudio** es: Aplicación web que facilite la gestión de la información.

Este objeto de estudio nos delimita como **campo de acción**: Informatización del proceso de gestión de la información del trabajo del Colectivo de Año del Departamento de Ingeniería Informática en el ISMMM.

Como **objetivo general**: Desarrollar una herramienta informática que favorezca el trabajo del Colectivo de Año del Departamento de Ingeniería Informática en el ISMMM.

Por tanto entendemos como **idea a defender** que: El desarrollo de una aplicación web favorecerá el trabajo del Colectivo de Año del Departamento de Ingeniería Informática en el ISMMM.

Objetivos específicos:

- Realizar un estudio de las tecnologías y herramientas a utilizar para el diseño e implementación de la aplicación.
- Implementar una aplicación que permita llevar a cabo el proceso de gestión de información de los estudiantes, para los profesores del departamento de Ingeniería Informática en el ISMMM.

Para darle cumplimiento al objetivo general, se propusieron las siguientes **Tareas**:

SIGA

- Análisis de los procesos actuales de gestión de información de los estudiantes, para los profesores del Departamento de Ingeniería Informática en el ISMMM.
- Análisis bibliográfico de las literaturas científicas relacionadas con los estudios de colectivo de año.
- Elaboración del marco teórico metodológico que sustenta a la investigación.
- Selección de la metodología de desarrollo de software, lenguajes de programación y herramientas que se utilizarán.
- Análisis, diseño e implementación de la aplicación informática para la gestión de información.
- Realización de las pruebas al software desarrollado.
- Realización del estudio de factibilidad.
- Documentación del software.

En la confección de este sistema se han empleado métodos **empíricos** y **teóricos** con finalidades a resolver necesidades para la terminación de la aplicación.

Los **métodos empíricos** empleados para la recopilación de información son los siguientes:

Observación no participante: Permite observar cómo funciona el proceso de gestión y seguimiento del Trabajo Metodológico en el Departamento de Ingeniería Informática en el ISMMM.

Entrevista a expertos: Permite recopilar información necesaria para valorar la situación actual del problema, lo cual permitió realizar un análisis y determinar los principales requisitos del sistema.

Los **métodos teóricos** proporcionan calidad en la investigación y en esta los utilizados son:

Histórico-Lógico: Para conocer el fenómeno que se estudia en sus antecedentes y tendencias actuales, lo cual permite establecer las bases teóricas que sustentan la investigación. También para reflejar de forma lógica la esencia, necesidad y el

comportamiento del proceso del trabajo metodológico de dicho departamento y el desarrollo del software.

Análisis y Síntesis: Permitió resumir los aspectos más importantes de la bibliografía consultada vinculada con el problema tratado y otras investigaciones relacionadas con el objeto de estudio; y se realizó la descomposición de cada uno de los requisitos del sistema para un mejor análisis.

Hipotético – Deductivo: Se empleó en la elaboración de la idea a defender.

Estructura de la tesis

El presente trabajo consta de introducción, 4 capítulos, conclusiones, recomendaciones, bibliografía, glosario de términos y anexos.

Capítulo 1 “Fundamentación Teórica”: Se analizan aspectos relacionados con el Trabajo Metodológico, conceptualización y características, se realiza un estudio acerca de los diferentes sistemas existentes vinculados al campo de acción, además se presenta la metodología y las herramientas a utilizar en el desarrollo del sistema propuesto.

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

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

Capítulo 4 “Estudio de Factibilidad y Sostenibilidad”: Se hace referencia a los beneficios tangibles e intangibles y se analizan los costos de desarrollo de la aplicación contra los beneficio para definir si es factible o no la aplicación y se valora la sostenibilidad del producto.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En este capítulo se abordarán conceptos asociados al trabajo de los Colectivos de Año que son necesarios para un entendimiento y darle solución a la situación problemática anteriormente planteada. Se realiza un estudio sobre los diferentes procesos en la gestión docente, además de su estructura y funciones asociadas; algunos sistemas automatizados existentes, así como las definiciones básicas relacionadas con las aplicaciones Web, las tecnologías para su desarrollo, herramientas y lenguaje de programación a utilizar, más la metodología implementada en el desarrollo de la aplicación.

1.2 El Colectivo de Año y su Trabajo Metodológico.

El Colectivo de Año es el encargado de llevar a cabo el Trabajo Metodológico en este nivel organizativo. Agrupa a los profesores que imparten las asignaturas del año, a los profesores guías de cada grupo, a los tutores y a los representantes de las organizaciones estudiantiles. Este colectivo se propone lograr el cumplimiento de los objetivos del año con calidad, propiciando la integración de los aspectos educativos e instructivos con un enfoque interdisciplinario. La conducción de este colectivo metodológico corresponde al Jefe de Colectivo de Año. (9)

1.2.1 Principales funciones

- ✓ La concreción de la estrategia educativa de la carrera en ese año, propiciando la integración de las clases, el trabajo científico estudiantil y las prácticas laborales con las diferentes tareas de impacto social, deportivas, y culturales, entre otras, que cumplen los estudiantes; en correspondencia con los objetivos educativos e instructivos de ese año. Esta estrategia se plasmará en el proyecto educativo para cada uno de los grupos que conforman el año.
- ✓ La participación en el proceso de diagnóstico integral y evaluación de los integrantes de la brigada o grupo estudiantil.

- ✓ La conducción y el control sistemático de la marcha del proceso docente educativo y del cumplimiento de los proyectos educativos de los grupos que conforman el año, desarrollando acciones para eliminar las deficiencias detectadas y proponiendo las medidas que permitan el mejoramiento continuo de la calidad de dicho proceso. (9)

Conociendo el funcionamiento del Colectivo de Año de año podemos valorar las tendencias y tecnologías actuales necesarias para la creación de un sistema enfocado en las necesidades planteadas.

1.3 Niveles de dirección

- ✓ Departamento docente
- ✓ Facultad, filial o unidad docente
- ✓ Sede universitaria
- ✓ Centro de educación superior

Estos niveles son los encargados de dirigir esta labor en los colectivos de carrera, de año, de disciplina y de asignatura, según corresponda. Trazarán estrategias pedagógicas comunes, en correspondencia con su papel y lugar en la estructura del centro de educación superior, dirigidas al perfeccionamiento del proceso docente educativo y divulgarán las mejores experiencias de los colectivos metodológicos en los claustros para su generalización. (9)

1.4 Funcionamiento del eslabón de base como sistema en la labor metodológica.

La estabilidad del Colectivo de Año constituye el elemento esencial para lograr que el eslabón de base funcione como sistema, lo que supone, en primer lugar, el adecuado cumplimiento de las funciones de cada uno de los jefes y coordinadores mencionados anteriormente. Para ello, en las diferentes reuniones de trabajo previstas, así como en las diferentes actividades metodológicas planificadas a ese nivel; deben participar, de conjunto con el jefe de colectivo de año, todos los

profesores que imparten docencia ese curso, los profesores guía, los tutores y los jefes de brigada de la FEU.

Dentro de sus principales funciones se tiene que:

- ❖ El modo en que se concreta la estrategia educativa de la carrera, propiciando la integración de las clases, el trabajo científico estudiantil y las prácticas laborales con las diferentes tareas de impacto social, deportivas, y culturales, entre otras, que cumplen los estudiantes; en correspondencia con los objetivos educativos e instructivos de ese año.
- ❖ El cumplimiento de lo previsto en los proyectos educativos de cada uno de los grupos que conforman el periodo pertinente, donde deben estar plasmados adecuadamente, entre otros aspectos, las diferentes actividades que permiten concretar la estrategia educativa de la carrera.
- ❖ Los resultados docentes que se vienen alcanzando en las diferentes evaluaciones previstas en cada una de las asignaturas que se desarrollan en ese año.
- ❖ La evaluación del progreso de los estudiantes en cada una de las asignaturas, básicamente a partir de los resultados docentes alcanzados por cada uno de ellos, enfatizando en los que presentan mayores dificultades. (9)

1.5 Tendencias y tecnologías actuales

1.5.1 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, envía peticiones a la capa intermedia que ofrece servicios, valiéndose de consultas y actualizaciones a la base de datos que a su vez proporciona una interfaz de usuario. (2)

1.5.2 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 o interferir con sus hábitos 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.

SJCA

- 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. (10)

1.5.3 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. (3)

Apache

Es el servidor Web más difundido y utilizado en Internet debido a su alta

SIGA

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.

Características de Apache

- 1) Fundamentalmente 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.

1.5.4 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

SIGA

SGBD disponibles en el mercado se encuentran: PostgreSQL, MySQL, Firebird, SQLite, Oracle, Microsoft SQL Server, entre otros. (1)

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.

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

- ✓ Alto rendimiento
- ✓ Bajo coste
- ✓ Facilidad de configuración y aprendizaje
- ✓ Portabilidad

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.

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.

Características

1- 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.

2- 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.

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.

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.

Características

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.

SJCA

- 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.

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.

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. (5)

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

SIGCA

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.

1.5.5 Herramientas CASE

Embarcadero ERStudio 8.0: Es una herramienta profesional para diseñar bases de datos, facilitándote herramientas para la creación de diagramas de relación, modelado de datos y gestión de estructuras (Es compatible con ambos SGBD, MySQL y PostgreSQL), fue la seleccionada para el diseño de la base de datos de SIGEDO.

MYSQL Workbench 5.1 OSS: Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. Se diseña para hacer más fácil de entender el estado actual de los datos. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia.
(11)

1.5.6 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.
(10)

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

SJCA

(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

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.

1.6 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. (2)

1.6.1 Metodología propuesta para el desarrollo de la aplicación.

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

Una posible mejora es incluir en los procesos de desarrollo más actividades, artefactos y restricciones, basando en los puntos débiles detectados, sin embargo,

el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuáles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas.

Este 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. (2)

1.6.2 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. (7)

1.6.3 Valores que promueve XP

Cuatro valores que promueven la metodología XP:

SJCA

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. (7)

1.6.4 El ciclo de vida ideal de XP

Fases:

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

1.6.5 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.(7)

El objetivo de XP es generar versiones de la aplicación tan pequeñas como sea

SJCA

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.

1.6.6 Testing

SIGCA

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.(7)

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. (7)

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. (7)

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. (7)

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. (7)

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. (7)

Sirve de presentación del trabajo y orienta al lector sobre los aspectos fundamentales de este. Se explica con absoluta claridad y de forma corrida. Debe contener los siguientes aspectos (no necesariamente tienen que aparecer todos los puntos):

SIGA

- Descripción de la organización donde se realiza la práctica profesional. (Caracterización de la organización, objeto social, misión, etc.)
- Situación problémica y problema a resolver
- Actualidad y necesidad del trabajo.
- Antecedentes (deben indicarse las referencias bibliográficas utilizadas como fuente de información).
- Aportes prácticos esperados del trabajo.
- Objeto de estudio.
- Objetivos del trabajo (objetivo general y objetivos específicos)
- Tareas principales desarrolladas para cumplir los objetivos.
- Estudio de herramientas, lenguajes y tecnologías a utilizar.
 - Opciones existentes en la actualidad.
 - Selección para el desarrollo del trabajo y justificación de la misma.
- Efecto económico: exposición de los beneficios del Proyecto elaborado, los cuales pueden ser:
 - Beneficios de carácter técnico.
 - Beneficios intangibles (económicos, sociales o políticos).
 - Beneficios económicos tangibles. (Esto último es obligatorio para todos los trabajos, debiéndose especificar los beneficios o aportes económicos del trabajo)

1.6.7 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

SJCA

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. (7)

1.6.8 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. (8)

Características

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el Scrum Master, que mantiene los procesos y trabaja de forma similar al director de proyecto, el Product Owner, que representa a los stakeholders (interesados externos o internos), y el Team que incluye a los desarrolladores.

Durante cada sprint, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie

SJCCA

puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Scrum permite la creación de equipos auto organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirement churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software.

Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

Roles

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

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

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

Roles "Cerdo"

SJCA

ProductOwner representa la voz del cliente. Se asegura de que el equipo Scrum trabaja de forma adecuada desde la perspectiva del negocio. El ProductOwner escribe historias de usuario, las prioriza, y las coloca en el ProductBacklog.

ScrumMaster (o Facilitador)

El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

ScrumTeam o Equipo

El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (diseñador, desarrollador, etc).

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?

Yuris Angel Acosta Zayas

SIJCA

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.

Reuniones en Scrum

DailyScrum

Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama "daily stand up". El scrum tiene unas guías específicas:

1. La reunión comienza puntualmente a su hora. A menudo hay castigos - acordados por el equipo- para quien llegue tarde (por ejemplo: dinero, flexiones, llevar colgando una gallina de plástico del cuello, etc)
2. Todos son bienvenidos, pero sólo los "cerdos" pueden hablar.
3. La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
4. Todos los asistentes deben mantenerse de pie (esto ayuda a mantener la reunión corta)
5. La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.

Durante la reunión, cada miembro del equipo contesta a tres preguntas:

1. ¿Qué has hecho desde ayer?
2. ¿Qué es lo que estás planeando hacer hoy?

3. ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? (Es el papel del ScrumMaster recordar estos impedimentos).

Scrum de Scrum

Cada día normalmente después del “DailyScrum”

- Estas reuniones permiten a los grupos de equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración.
- Asiste una persona asignada por cada equipo.

La agenda será la misma que la del DailyScrum, añadiendo además las siguientes cuatro preguntas:

1. ¿Qué ha hecho tu equipo desde nuestra última reunión?
2. ¿Qué hará tu equipo antes que nos volvamos a reunir?
3. ¿Hay algo que demora o estorba a tu equipo?
4. ¿Estás a punto de poner algo en el camino del otro equipo?

Reunión de Planificación del Sprint (Sprint Planning Meeting)

Al inicio del ciclo Sprint (cada 15 o 30 días), una “Reunión de Planificación del Sprint” se lleva a cabo.

- Seleccionar qué trabajo se hará
- Preparar, con el equipo completo, el Sprint Backlog que detalla el tiempo que tomará hacer el trabajo.
- Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint
- Ocho horas como límite

Al final del ciclo Sprint, dos reuniones se llevarán a cabo: la “Reunión de Revisión del Sprint” y la “Retrospectiva del Sprint”

Reunión de Revisión del Sprint (Sprint Review Meeting)

1. Revisar el trabajo que fue completado y no completado
2. Presentar el trabajo completado a los interesados (alias “demo”)
3. El trabajo incompleto no puede ser demostrado
4. Cuatro horas como límite

Retrospectiva del Sprint (Sprint Retrospective)

Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

Sprint

El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Es recomendado que la duración de los sprints sea constante y definida por el equipo con base en su propia experiencia. Se puede comenzar con una duración de sprint en particular (2 o 3 semanas) e ir ajustándolo con base en el ritmo del equipo, aunque sin relajarlo demasiado. Al final de cada sprint, el equipo deberá presentar los avances logrados, y deberán entregar un producto con características de utilizable por el cliente. Asimismo, se recomienda no cambiar los objetivos del sprint o sprint backlog a menos que la falta de estos cambios amenace al éxito del proyecto. La constancia permite la concentración y mejora la productividad del equipo de trabajo.

Documentos

Product backlog

El product backlog es un documento de alto nivel para todo el proyecto. Contiene descripciones genéricas de todos los requerimientos, funcionalidades deseables, etc. priorizadas según su retorno sobre la inversión (ROI). Es el qué va a ser construido. Es abierto y cualquiera puede modificarlo. Contiene estimaciones

SJCA

grosso modo, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido. Esta estimación ayuda al product owner a ajustar la línea temporal y, de manera limitada, la prioridad de las diferentes tareas. Por ejemplo, si dos características tienen el mismo valor de negocio la que requiera menos tiempo de desarrollo tendrá probablemente más prioridad, debido a que su ROI será más alto.

Sprint backlog

El sprint backlog es un documento detallado donde se describe el cómo el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas con ninguna tarea de duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser dividida en otras menores. Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

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.

Scrum aplicado al desarrollo de software

Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren

rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software.

Jeff Sutherland aplicó el modelo Scrum al desarrollo de software en 1993 en Easel Corporation (Empresa que en los macro-juegos de compras y fusiones se integraría en VMARK, luego en Informix y finalmente en Ascential Software Corporation). En 1995 lo presentó junto con Ken Schwaber como proceso formal, también para gestión del desarrollo de software en OOPSLA 95. Más tarde, en 2001 serían dos de los promulgadores del Manifiesto ágil. En el desarrollo de software scrum está considerado como modelo ágil por la Agile Alliance.

La ficha adjunta incluye una descripción sinóptica del proceso y sus elementos que son:

- Roles: Propietario del producto, Gestor o Manager del Scrum, Equipo e Interesados.
- Componentes del proceso: Pila del producto (ProductBacklog), Pila del sprint (Sprint Backlog), Incremento.
- Reuniones: Planificación del sprint, Revisión diaria, Revisión del sprint.
- Sprint

1.7 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 escalera más grande.

SIGA

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. (2)

1.7.1 Arquitectura

La Arquitectura es el esqueleto o base de una aplicación. Representa la organización fundamental de un sistema. Desde los pequeños programas hasta los sistemas más grande poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". En la Web es muy común la utilización de la arquitectura "3-capas", "n-capas", "MVC", entre otras.

Arquitectura en tres capas

Arquitectura en capas: es donde se define como organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo que quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son consistentes de ningún detalle o interfaz de las superiores.

SJCA

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



Figura 1.1 Patrón arquitectónico tres capas

La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio solo 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 niveles; simplemente es necesario conocer la API que existe entre niveles. En el diseño de sistemas informáticos actual se suele usar las arquitecturas multi-nivel 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). (2)

Capas o niveles

Capa de presentación: es la que ve el usuario (hay quien la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

SIGA

Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

Capa de datos: es donde residen los datos y es la encargada de acceder ellos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador (no es lo típico). Si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

1.8 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.

1.8.1 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

SJCCA

programa esta 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. 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. (1)

1.8.2 Macromedia Dreamweaver 8.0

Dreamweaver es una aplicación en forma de suite (basada en la forma de estudio de Adobe Flash) que está destinada a la construcción, diseño y edición de sitios y aplicaciones Web basados en estándares. Creado inicialmente por Macromedia (actualmente producido por Adobe Systems) es el programa más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Expression Web y tiene soporte tanto para edición de imágenes como para animación a través de su integración MALVERDE'S Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer y no validaba como HTML estándar. Esto se ha ido corrigiendo en las versiones recientes.

SJCA

La gran ventaja de este editor sobre otros es su gran poder de ampliación y personalización del mismo, puesto que en este programa, sus rutinas (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en JavaScript-C, lo que le ofrece una gran flexibilidad en estas materias. Esto hace que los archivos del programa no sean instrucciones de C++ sino rutinas de JavaScript que hace que sea un programa muy fluido, que todo ello hace, que programadores y editores web hagan extensiones para su programa y lo ponga a su gusto.

Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para pre visualizar las páginas web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio web completo. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código. (2)

1.8.3 Embarcadero ER/Studio

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

Conclusiones de Capítulo

En este capítulo se trataron temas esenciales para el desarrollo de la aplicación sin los cuales hubiera sido imposible la materialización del objeto de estudio y el campo de acción. Además se realizó un estudio para la selección de las herramientas, metodología y lenguajes a utilizar dada la gama de ventajas que presentan.

SIGA

En la implementación del sistema, se utilizó el entorno de programación, **Dreamweaver** por las facilidades que brinda para el trabajo de diseño Web, además este entorno de desarrollo soporta 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.

Capítulo 2 Planeación y Diseño

Introducción

En el presente capítulo se introduce la fase de planeación y diseño, en la cual se detallan las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU) y se describen las necesidades del cliente. A su vez se presentará un plan de iteraciones para cada una de las HU aplicando la estimación de esfuerzo a estas. Se presentan las tarjetas CRC y sus principales clases.

También 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 tienen lugar las reuniones con el cliente, en las cuales se definen las principales funcionalidades del sistema, los requisitos funcionales y 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 y profesores de la carrera Ingeniería Informática y el Trabajo Metodológico que se realiza desde la base del Departamento de Informática del centro se realiza de forma manual, guardando las informaciones en planillas de papel o documentos digitales aislados, lo cual hace que el proceso se vuelva lento y en ocasiones engorroso para los trabajadores de este departamento, así como que las consultas no sean eficientes, en algunos casos trae como consecuencia la pérdida de información y el deterioro de la misma. Tomando como referencia lo antes expuesto se identifican como funcionalidades generales del sistema guardar todos los datos de los profesores y estudiante 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 el Departamento de Informática.

Yuris Angel Acosta Zayas

Tabla 2.1: Personal relacionado con el sistema

Personas relacionadas con la Justificación sistema	
Jefe de Departamento	Es el encargado de actualizar todas las actividades existentes en el sistema.
Jefe de Carrera	Es el encargado de actualizar todas las actividades existentes en el sistema.
Jefe de Colectivo de año	Es el encargado de actualizar y organizar las diferentes actividades metodológicas a mostrar en el sistema.
Coordinador de año	Es el encargado de organizar, insertar nuevos estudiantes al sistema.
Profesor guía	Es el encargado de realizar e insertar las plantillas semestrales de los estudiantes en el sistema.
Tutor	Es el encargado de recopilar información referente a estudiantes para introducirla en el sistema.
Administrador	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.

1. Autenticar profesor.
2. Gestionar profesores.
 - a) Insertar profesor al sistema.

- b) Mostrar profesor del sistema.
 - c) Modificar profesor del sistema.
 - d) Eliminar profesor del sistema.
3. Gestionar estudiantes.
- a) Insertar estudiantes.
 - b) Mostrar estudiantes.
 - c) Modificar estudiantes.
 - d) Eliminar estudiantes.
4. Gestionar actividades del sistema.
- a) Insertar actividades al sistema.
 - b) Mostrar actividades del sistema.
 - c) Modificar actividades del sistema.
 - d) Eliminar actividades del sistema.
5. Gestionar coordinadores del sistema.
- a) Insertar coordinadores al sistema.
 - b) Mostrar coordinadores del sistema.
 - c) Modificar coordinadores del sistema.
 - d) Eliminar coordinadores del sistema.
6. Gestionar tutores del sistema.
- a) Insertar tutores al sistema.
 - b) Mostrar tutores del sistema.
 - c) Modificar tutores del sistema.
 - d) Eliminar tutores del sistema.
7. Gestionar profesor guía del sistema.
- a) Insertar profesor guía al sistema.
 - b) Mostrar profesor guía del sistema.
 - c) Modificar profesor guía del sistema.
 - d) Eliminar profesor guía del sistema.
8. Gestionar asignatura al sistema.
- a) Insertar asignatura al sistema.
 - b) Mostrar asignatura del sistema.

- c) Modificar asignatura del sistema.
- d) Eliminar asignatura del sistema.
- 9. Salvar fin de curso.
- 10. Gestionar administrador al sistema.
 - a) Insertar administrador al sistema.
 - b) Mostrar administrador del sistema.
 - c) Modificar administrador del sistema.
 - d) Eliminar administrador del sistema.
- 11. Gestionar semestre al sistema.
 - a) Insertar semestre al sistema.
 - b) Mostrar semestre del sistema.
 - c) Modificar semestre del sistema.
 - d) Eliminar semestre del sistema.
- 12. Gestionar evaluación al sistema.
 - a) Insertar evaluación al sistema.
 - b) Mostrar evaluación del sistema.
 - c) Modificar evaluación del sistema.
 - d) Eliminar evaluación del sistema.
- 13. Exportar reporte de evaluaciones por semestre.
- 14. Exportar reporte de evaluaciones por año.
- 15. Exportar reporte de estudiantes por año.
- 16. Exportar reporte de estudiantes en general.
- 17. Exportar reporte de profesores.
- 18. Gestionar documentos al sistema.
 - a) Insertar documentos al sistema.
 - b) Mostrar documentos del sistema.
 - c) Modificar documentos del sistema.
 - d) Eliminar documentos del sistema.
- 19. Buscar Estudiantes en el sistema.
- 20. Buscar Coordinador de año en el sistema.
- 21. Buscar Tutor en el sistema.

- 22. Buscar Profesor guía en el sistema.
- 23. Buscar Jefe de departamento en el sistema.
- 24. Buscar Jefe de carrera en el sistema.
- 25. Modificar contraseña

2.1.3 Características del sistema

1. Requerimientos de software.
 - a. Navegador que soporte las características del lenguaje HTML5, JS3 y CSS3.
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.
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 usuarios

Las historias de usuarios (HU), es 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.

Tabla 2.2 HU No. 1: Autenticar usuarios.

Tabla 2.2 HU No. 1: Gestión usuarios.

Historia de usuario	
Número: 1	Usuario: Cliente
Nombre: Autenticar usuarios.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Los usuarios del sistema ingresan sus datos para entrar al sistema	
Observaciones: Confirmado por el cliente.	

2.2 Planificación de entregas

En esta fase se establece la prioridad de cada HU y 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 y según sus necesidades. Las entregas no deben exceder los dos o tres meses.

Las estimaciones asociadas a la implementación de las HU 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 HU generalmente tienen un valor de 0 a 1 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 que el sistema funcionara mediante a lo deseado, se realizó una estimación para cada una de las historias de usuario identificadas, y mediante estos resultados se obtienen los resultados presentes:

Tabla 2.3 Estimación de esfuerzo por HU

Historias de usuarios	Puntos de estimación
Autenticar profesor.	1
Gestionar profesores.	1
Gestionar estudiante	1
Gestionar actividades del sistema.	1
Gestionar coordinadores del sistema.	1
Gestionar tutores del sistema.	1
Gestionar profesor guía del sistema.	1
Gestionar asignatura al sistema.	1
Salvar fin de curso.	0.2
Gestionar administrador al sistema.	1
Gestionar semestre al sistema.	1
Gestionar evaluación al sistema.	1
Exportar reporte de evaluaciones por semestre.	0.1
Exportar reporte de evaluaciones por año.	0.1
Exportar reporte de estudiantes por año.	0.1
Exportar reporte de estudiantes en general.	0.1
Exportar reporte de profesores.	0.1
Gestionar documentos al sistema.	1
Buscar Estudiantes en el sistema.	0.1
Buscar Coordinador de año en el sistema.	0.1
Buscar Tutor en el sistema.	0.1
Buscar Profesor guía en el sistema.	0.1
Buscar Jefe de departamento en el sistema.	0.1
Buscar Jefe de carrera en el sistema.	0.1
Modificar contraseña.	0.1

2.2.2 Planificación de iteraciones

Partiendo de las HU anteriormente 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, tomando como referencia 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 1 iteración, la cual se explica de forma detallada a continuación:

Primera iteración:

El objetivo es darle cumplimiento a las HU que se consideran de mayor importancia para el desarrollo de la aplicación. Al finalizar esta iteración se contará con todas las funcionalidades descritas en las HU 1, 2, 5, 6, 7 y 10 las cuales hacen mención a la autenticación de usuarios y a la inserción de la información de los profesores según los roles que juegan en el sistema.

Segunda iteración:

Esta iteración tiene como finalidad desarrollar las historias de usuario 3, 4, 8, 11, 12 y 18. Las mismas brindan las funcionalidades para gestionar la información de estudiantes, asignaturas, actividades, semestres, evaluaciones y documentos que toman curso en la carrera. De manera que el cliente pueda interactuar con las ventajas que les brindará la aplicación para su uso.

Tercera iteración:

En esta iteración se llevarán a cabo las historias de usuario 13 – 17. Con estas se tendrán las funcionalidades para llevar a cabo la exportación de los reportes realizados a evaluaciones por semestre y por año, a su vez se les realizará

reportes a los estudiantes por año y en general, y finalizando con los reportes realizados a los profesores de la carrera.

Cuarta Iteración

En la presente iteración se desarrollan las historias de usuarios 19 – 25, 9. Estas brindan la información referente a la búsqueda de usuarios en el sistema.

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 llevarán 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
Iteración 1	Autenticar profesor.	3 semana
	Gestionar profesores	
	Gestionar coordinadores del sistema.	
	Gestionar tutores del sistema.	
	Gestionar profesor guía del sistema.	
	Gestionar administrador al sistema.	
Iteración 2	Gestionar actividades del sistema.	3 semanas
	Gestionar asignatura al sistema.	
	Gestionar semestre al sistema.	
	Gestionar evaluación al sistema.	
	Gestionar documentos al sistema.	
Iteración 3	Exportar reporte de evaluaciones por	2 semanas
	Exportar reporte de evaluaciones por año.	
	Exportar reporte de estudiantes por año.	
	Exportar reporte de estudiantes en general.	
	Exportar reporte de profesores.	

	Modificar contraseña	
Iteración 4	Buscar Estudiantes en el sistema.	1 semana
	Buscar Coordinador de año en el sistema.	
	Buscar Tutor en el sistema.	
	Buscar Profesor guía en el sistema.	
	Buscar Jefe de departamento en el sistema.	
	Buscar Jefe de carrera en el sistema.	
	Salvar fin de curso.	

2.3 Diseño

Se muestra en este epígrafe la realización de las tarjetas de clases, colaboración y conocidas 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

Tabla 2.5 Tarjeta CRC No. 1 Usuario

Nombre de la Clase: Gestionar Profesor	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores

Inserta datos del profesor	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

En este capítulo se logró abordar la fase de planeación y diseño donde se delinearon las historias de usuarios (HU) con la participación del cliente, se llevó a cabo 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 el siguiente capítulo se continúa con las fases de Desarrollo y Pruebas expuestas en la Metodología XP. Se muestra el modelo de datos correspondiente a la aplicación, así como las interfaces de usuarios diseñada. Se realiza a través del desarrollo y avance de las iteraciones un desglose de las tareas a realizar para cada una de las HU expuestas anteriormente. Se describen las pruebas realizadas, dando como resultado las respuestas al aplicarse cada una de las funcionalidades.

Introducción

En el siguiente capítulo se profundiza en las últimas fases de la Metodología XP Desarrollo y Pruebas de la aplicación. Donde el encargado es el programador en la parte de desarrollo y el cliente en la realización de las pruebas de aceptación.

3.1 Modelo de datos

Modelo de datos utilizado en el sistema.

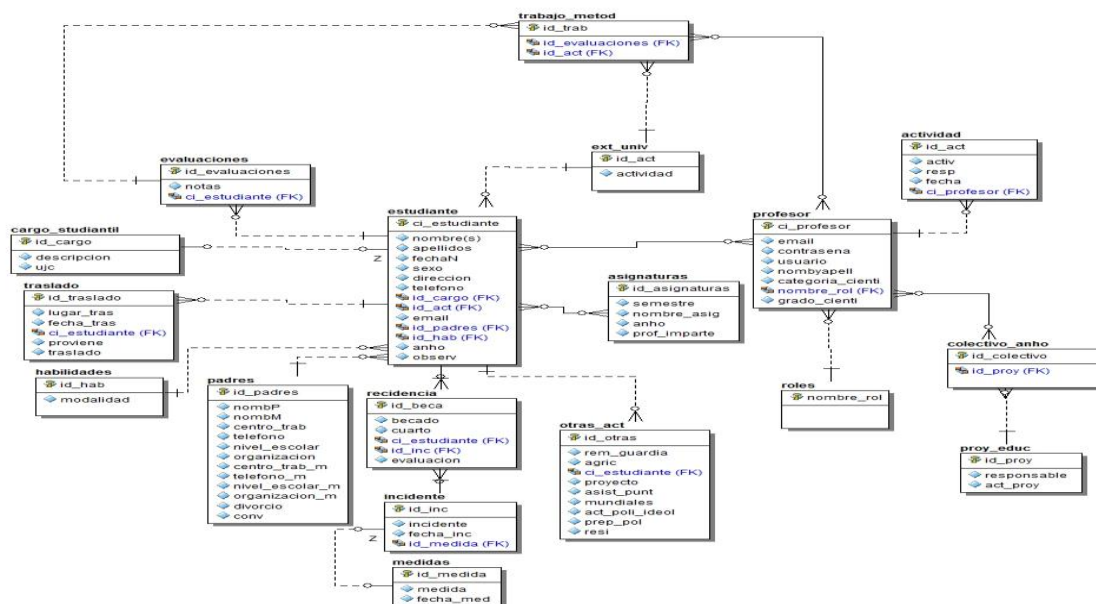


Figura 3.1 Modelo de datos

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 #: Interfaces de usuario](#)]

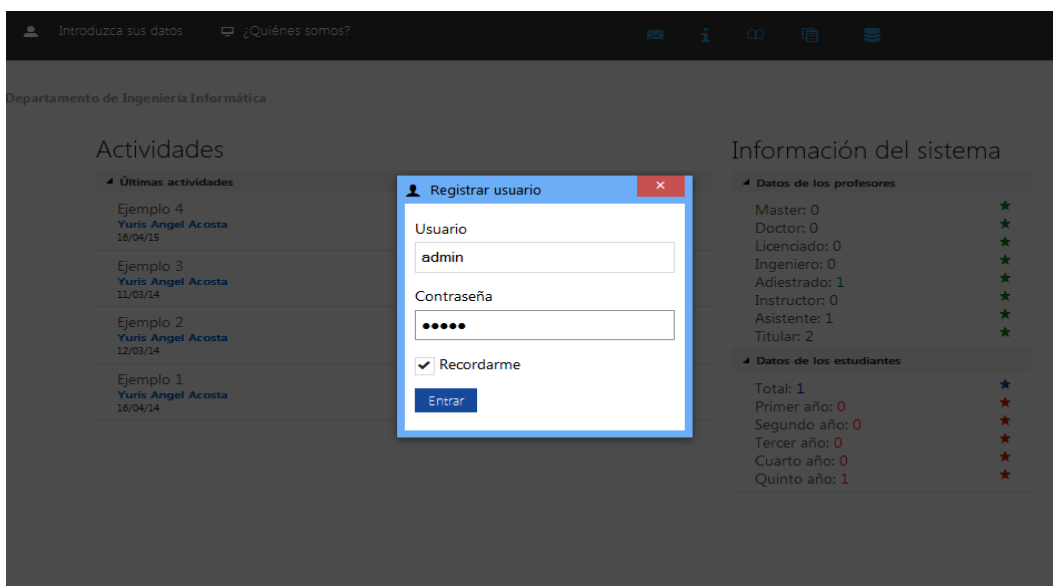


Figura 3.2 Interfaz de usuario Autenticar usuario.

3.3 Desarrollo de las iteraciones

En el anterior capítulo se detallaron todas las HU 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 conformado, para detectar si pueden existir cambios en las mismas.

3.3.1 Tareas por historias de usuario

Dentro del contenido de presente 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. El cliente no tiene por qué entender necesariamente las tareas, 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 profesor.
Gestionar profesor	Insertar profesor Mostrar profesor Modificar profesor Eliminar profesor
Gestionar estudiante	Insertar estudiante Mostrar estudiante Modificar estudiante Eliminar estudiante
Gestionar actividad	Insertar actividades Mostrar actividades Modificar actividades Eliminar actividades
Gestionar coordinador	Insertar Coordinadores Listar Coordinadores Modificar Coordinadores Eliminar Coordinadores
Gestionar tutor	Insertar tutores Listar tutores Modificar tutores Eliminar tutores
Gestionar profesor guía	Insertar profesores guía Listar profesores guía Modificar profesores guía Eliminar profesores guía
Gestionar asignatura	Insertar asignatura Listar asignatura Modificar asignatura Eliminar asignatura
Gestionar administrador	Insertar administrador al sistema. Mostrar administrador del sistema. Modificar administrador del sistema. Eliminar administrador del sistema.
Salvar fin de Curso	Salvar fin de Curso
Gestionar semestre	Insertar Semestres Mostrar Semestres Modificar Semestres Eliminar Semestres

Gestionar semestre	Insertar semestre al sistema. Mostrar semestre del sistema. Modificar semestre del sistema. Eliminar semestre del sistema.
Gestionar evaluaciones	Insertar Evaluaciones Mostrar Evaluaciones Modificar Evaluaciones Eliminar Evaluaciones
Exportar reporte de evaluaciones por semestre	Exportar reporte de evaluaciones por semestre
Exportar reporte de evaluaciones por año	Exportar reporte de evaluaciones por año
Exportar reporte de estudiantes por año.	Exportar reporte de estudiantes por año.
Exportar reporte de estudiantes en general.	Exportar reporte de estudiantes en general.
Exportar reporte de profesores.	Exportar reporte de profesores.
Gestionar documentos al sistema	Insertar Documentos Listar Documentos Eliminar Documentos Modificar documentos Eliminar Informaciones
Buscar Estudiantes en el sistema.	Buscar Estudiantes en el sistema.
Buscar Coordinador de año en el sistema.	Buscar Coordinador de año en el sistema.
Buscar Tutor en el sistema.	Buscar Tutor en el sistema.
Buscar Profesor guía en el sistema.	Buscar Profesor guía en el sistema.
Buscar Jefe de departamento en el sistema.	Buscar Jefe de departamento en el sistema.
Buscar Jefe de carrera en el sistema.	Buscar Jefe de carrera en el sistema.
Modificar contraseña.	Modificar contraseña.

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 profesor	1
Gestionar profesor	4

Gestionar coordinadores	4
Gestionar tutores	4
Gestionar profesor guía	4
Gestionar administrador	4

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 estudiantes	3
Gestionar actividades	3
Gestionar asignatura	2
Gestionar semestre	2
Gestionar evaluación	3
Gestionar documentos	3

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)
Exportar reporte de evaluaciones por semestre.	1
Exportar reporte de evaluaciones por año.	1

Exportar reporte de estudiantes por año.	1
Exportar reporte de estudiantes en general.	1
Exportar reporte de profesores.	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)
Buscar Estudiantes en el sistema.	1
Buscar Coordinador de año en el sistema.	1
Buscar Tutor en el sistema.	1
Buscar Profesor guía en el sistema.	1
Buscar Jefe de departamento en el sistema.	1
Buscar Jefe de carrera en el sistema.	1
Modificar contraseña	1
Salvar fin de curso.	1

3.3.2 Tareas de ingeniería

En XP generalmente cada HU se divide en tareas de ingeniería o tareas de programación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Estas se crean para obtener una mejor planificación y cumplir con las funcionalidades básicas que

luego conformaran las funcionalidades generales de cada historia de usuario [\[Anexo #3: Tareas de ingeniería\]](#).

Tabla 3.8 Tarea de programación autenticar usuario

Tarea de Programación	
No. Tarea: 2	No. Historia: 2
Nombre de la tarea: insertar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 22/03/2014	Fecha fin: 23/03/2014
Programador responsable: Yuris Angel Acosta Zayas	
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 Tarea de aceptación hu #2 gestionar profesor

Prueba de aceptación
Historia de usuario # 2: Gestionar profesores
Nombre: Prueba para comprobar la entrada de datos del profesor del departamento.
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.

- Las contraseñas no coincidan.

Evaluación de la prueba: Aceptada.

Conclusiones

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 los momentos actuales es muy importante tener en cuenta que a la hora de desarrollar un proyecto el mismo cuenta con eficacia y eficiencia a la hora de su implementación, y por supuesto se hace preciso efectuar y evaluar la factibilidad antes de su elaboración. 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 y factibilidad del proyecto, centrado en estimaciones de esfuerzo humano, tiempo de desarrollo para su ejecución y costo. Se estiman los beneficios tangibles e intangibles que representan para el sistema propuesto, un análisis de costos y beneficios.

4.1 Evaluación Costo – Beneficio





Para el estudio de la factibilidad de este proyecto fue utilizada la metodología Costo Efectividad (Beneficio), esta metodología, plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores (1):

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

- **La efectividad**, que se entiende como la capacidad del proyecto para satisfacerla necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo). El desarrollo de un producto informático, siempre tiene un costo.






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

4.2 Efectos Económicos


-  Efectos directos
-  Efectos indirectos
-  Efectos externos
-  Intangibles

4.2.1 Efectos directos:

Positivos:

-  Disminución de la acumulación de materiales impresos relacionados con los estudiantes extranjeros y profesores.
-  Los usuarios pueden revisar sus datos en todo momento.
-  Seguridad al guardar la información.
-  Ahorro de tiempo en la búsqueda de información de un profesor o estudiante.
-  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. (5)

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.

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: \$ 64.93 CUC.

Costos en Moneda Nacional:

Costos Directos.

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

Costos indirectos

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

Total General: \$ 145.99

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

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

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

Valores de la variable (Solución manual)

1. Gestionar información de los profesores. (10 min)
2. Gestionar información de los estudiantes. (15 min)
3. Buscar información de las actividades que se realizan. (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. (2 min)
3. Buscar información de las actividades que se realizan. (0.4 min)
4. Crear Reportes. (1 min).



Figura 4.1 Gráfica de la solución sin el producto y solución con el producto.

SIGA

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

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología, la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes, el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tómese como costo el tiempo empleado por el Webmaster para mantener el sitio actualizado en todo momento.

Este capítulo realizó el estudio de factibilidad mediante La Metodología Costo Efectividad (Beneficio), analizó los efectos económicos, los beneficios y costos intangibles, además se calculó el costo de ejecución del proyecto mediante la ficha de costo arrojando como resultado \$64.93 CUC y \$145.99 MN demostrándose la factibilidad del proyecto.

Conclusiones Generales

Después de analizados los resultados del producto final, es evidente el cumplimiento de los objetivos planteados al iniciar la investigación, se obtuvieron los siguientes resultados:

- El estudio de la literatura relacionada con el tema permitió examinar y establecer comparaciones entre las herramientas existentes relacionadas con el campo de acción.
- El análisis y estudio del proceso del Trabajo Metodológico brindó una fácil comprensión para el desarrollo del sistema.
- El estudio adecuado de las herramientas y tecnologías utilizadas agilizó la implementación del software.
- El proceso de desarrollo del software estuvo guiado por las fases de la metodología XP, en las que quedaron plasmadas las tarjetas CRC, se desarrollaron las tareas correspondientes para dar solución a las historias de usuario y se realizaron las pruebas de aceptación, demostrando que el cliente está de acuerdo con el producto final.
- Se desarrolló una herramienta Web flexible y orientada al usuario que permite el seguimiento de los procesos del Trabajo Metodológico.

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:

- ❖ Extender el uso de la Aplicación Web a todas las carreras del ISMMM siempre que se cumpla el mismo proceso con que se realiza el Trabajos Metodológico.

Referencia bibliográfica

1. **Lecusay, Pablo.** *Sistema informático para favorecer el proceso de selección de personal en el instituto superior minero metalúrgico de moa.* Moa : ISMMM, 2012.
2. **Silva, Ing. Exneider proenza.** *Sistema de Costos de Calidad para la Empresa .* 2010.
3. **Ferrer, J.** [Online] 02 25, 2003. [Cited: 02 15, 2014.] <http://libresoft.es/downloads/ferrer-20030312.pdf>.
4. **www.postgresql.org/support/professional_support.** [Online] [Cited: 02 18, 2014.] http://www.postgresql.org/support/professional_support.
5. **www.microsoft.com/spain/sql/default.msp.** [Online] [Cited: 02 18, 2014.] <http://www.microsoft.com/spain/sql/default.msp>.
6. **www.oracle.com.** [Online] <http://www.oracle.com>.
7. **Jeffries, R.** What is Extreme Programming? [Online] [Cited: 02 16, 2014.] <http://www.xpprogramming.com/xpmag/whatisxp.htm>.
8. **Bañeres, Juan Palacio.** Gestión ágil de proyectos: Scrum. *Navegapolis.net.* [Online] <http://www.navegapolis.net>.
9. **MES.** El funcionamiento del Eslabón Base del proceso de formación, con prioridad en la labor educativa y político ideológica. 2012.
10. **López Labañino, Carlos Mario.** "Sistema de Gestión Documental Informatizado para el Departamento de Informática". Moa, Holguín: ISMMM 2012.
11. **ER/STUDIO.** Herramienta para el modelaje de datos Embarcadero. [En línea] Consultado 2013 – 3 – 25. Disponible en: <http://bureaudeprensa.com>.

Bibliografía

1. **FERRER, J.** Metodologías Ágiles. [En línea]. [Citado: 2013-02-25].
<http://libresoft.es/downloads/ferrer-20030312.pdf>
2. **MARTÍNEZ CORDERO, Yadiris.** “Sistema de gestión docente para la carrera de Informática en el ISMMM”. Moa-Holguín : ISMMM, 2012.
3. **JEFFRIES, R.** What is Extreme Programming? [Online] [Cited: 02 16, 2014.]
http://www.postgresql.org/support/professional_support.
4. **LÓPEZ LABAÑINO, Carlos Mario.** “Sistema de Gestión Documental Informatizado para el Departamento de Informática”. Moa, Holguín: ISMMM 2012
5. *Funcionamiento del eslabón de base. Versión para los CES. 8.10.12*
6. *Metodología XP.* [en línea], 2008. [Consultado: 2013-02-2]. Disponible en:
http://2008/04/características-de-la-metodologia-xp_25.html
7. **DATE, C. J.** “Introducción a los Sistemas de Bases de Datos”, Félix Varela, 2003.
8. **EGUÍLUZ PÉREZ, J.** “Introducción a CSS”, www.librosweb.es.
9. **EGUÍLUZ PÉREZ, J.** “Introducción a JavaScript”, www.librosweb.es.
10. CAVSI. 2004. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] [Consultado 2013-3-26]. Disponible en:
<http://www.cavsi.com/preguntasrespuestas/que-es-un-Sistema-gestor-de-bases-de-datos-o-sgbd/>.
11. Eaprende.com Aprende.com. [En línea]. Gestor de Base de Datos: MySQL, PostgreSQL, SQLite. - 2001. [Consultado 2013-3-10.] -
<http://www.eaprende.com/gestor-de-basededatos-mysql-postgresql-sqlite.html>
12. Ciberaula.com. 2005. Ciberaula.com. Una Introducción a APACHE. [En línea] [Consultado 2013-3-28]. Disponible en: <http://linux.ciberaula.com>
13. Blanco Criado, A. 2008. XAMPP. [En línea]. [Consultado 2013-3-28].
Disponible en: <http://www.adictosaltrabajo.com/tutoriales>
14. ER/STUDIO. Herramienta para el modelaje de datos Embarcadero. [En

línea] Consultado 2013-3-25]. Disponible en: <http://bureaudeprensa.com>

15. DATE, C. J. *“Introducción a los Sistemas de Bases de Datos”*, Félix Varela, 2003.
16. HERNÁN RUIZ, M. *“Programación Web Avanzada Soluciones Rápidas y Efectivas Para Desarrolladores de Sitios”*, Félix Varela, 2006.
17. *El funcionamiento del Eslabón Base del proceso de formación, con prioridad en la labor educativa y político ideológica*. MES, 2012.

Anexo 1: Historias de usuarios

Tabla 2.6 HU No. 1 Autenticar usuario

Historia de usuario	
Número: 1	Usuario: Cliente
Nombre: Autenticar profesor.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: Yuris Angel Acosta Zayas	
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. 2 gestionar profesores

Historia de usuario	
Número: 1	Usuario: Administrador
Nombre: Gestionar profesores.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.8 HU No.3 gestionar estudiantes

Historia de usuario	
----------------------------	--

Número: 2	Usuario: Administrador
Nombre: Gestionar estudiantes.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.9 HU No. 4 gestionar actividades

Historia de usuario	
Número: 4	Usuario: Administrador
Nombre: Gestionar actividades del sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.10 HU No. 5 gestionar tutor

Historia de usuario	
Número: 6	Usuario: Administrador
Nombre: Gestionar tutores del sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Primera

Programador responsable: Yuris Angel Acosta Zayas

Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.

Observaciones: Confirmado por el cliente.

Tabla 2.11 HU No. 6 gestionar profesor guía

Historia de usuario

Número: 7

Usuario: Administrador

Nombre: Gestionar profesor guía del sistema.

Prioridad en el negocio: Alta

Riesgo en desarrollo: media

Puntos estimados: 1

Iteración asignada: Primera

Programador responsable: Yuris Angel Acosta Zayas

Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.

Observaciones: Confirmado por el cliente.

Tabla 2.12 HU No. 7 gestionar asignatura

Historia de usuario

Número: 8

Usuario: Administrador

Nombre: Gestionar asignatura al sistema.

Prioridad en el negocio: Alta

Riesgo en desarrollo: media

Puntos estimados: 1

Iteración asignada: Segunda

Programador responsable: Yuris Angel Acosta Zayas

Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.

Observaciones: Confirmado por el cliente.

Tabla 2.13 HU No. 8 gestionar administrador

Historia de usuario	
Número: 10	Usuario: Administrador
Nombre: Gestionar administrador al sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Primera
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.14 HU No. 9 gestionar evaluación

Historia de usuario	
Número: 12	Usuario: Administrador
Nombre: Gestionar evaluación.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Segunda
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.	
Observaciones: Confirmado por el cliente.	

Tabla 2.15 HU No. 10 exportar reporte de evaluaciones por semestre

Historia de usuario	
Número: 13	Usuario: Administrador
Nombre: Exportar reporte de evaluaciones por semestre.	

Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Tercera
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El sistema muestra las evaluaciones y permite que se descargue en formato pdf.	
Observaciones: Confirmado por el cliente.	

Tabla 2.16 HU No. 11 exportar reporte de evaluaciones por año

Historia de usuario	
Número: 14	Usuario: Administrador
Nombre: Exportar reporte de evaluaciones por año.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Tercera
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El sistema muestra las evaluaciones por año y permite que se descargue en formato pdf.	
Observaciones: Confirmado por el cliente.	

Tabla 2.17 HU No. 12 exportar reporte de estudiantes

Historia de usuario	
Número: 16	Usuario: Administrador
Nombre: Exportar reporte de estudiantes en general.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Tercera
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El sistema muestra los datos de los estudiantes del curso y permite que se	

descargue en formato pdf.

Observaciones: Confirmado por el cliente.

Tabla 2.18 HU No. 13 gestionar documentos

Historia de usuario

Número: 18

Usuario: Administrador

Nombre: Gestionar documentos al sistema.

Prioridad en el negocio: Alta

Riesgo en desarrollo: media

Puntos estimados: 1

Iteración asignada: Segunda

Programador responsable: Yuris Angel Acosta Zayas

Descripción: El usuario debe entrar los datos requeridos y después podrá realizar las operaciones correspondientes.

Observaciones: Confirmado por el cliente.

Tabla 2.19 HU No. 14 buscar estudiantes

Historia de usuario

Número: 19

Usuario: TODOS

Nombre: Buscar Estudiantes en el sistema.

Prioridad en el negocio: Alta

Riesgo en desarrollo: media

Puntos estimados: 1

Iteración asignada: Cuarta

Programador responsable: Yuris Angel Acosta Zayas

Descripción: El usuario debe entrar los datos requeridos y después podrá realizar la operación correspondiente.

Observaciones: Confirmado por el cliente.

Tabla 2.20 HU No.15 buscar coordinador de año

Historia de usuario	
Número: 20	Usuario: TODOS
Nombre: Buscar Coordinador de año en el sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Cuarta
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar la operación correspondiente.	
Observaciones: Confirmado por el cliente.	

Tabla 2.21 HU No.16 buscar tutor

Historia de usuario	
Número: 21	Usuario: TODOS
Nombre: Buscar Tutor en el sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Cuarta
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar la operación correspondiente.	
Observaciones: Confirmado por el cliente.	

Tabla 2.22 HU No.17 buscar profesor guía

Historia de usuario	
Número: 22	Usuario: TODOS
Nombre: Buscar Profesor guía en el sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media

Puntos estimados: 1	Iteración asignada: Cuarta
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar la operación correspondiente.	
Observaciones: Confirmado por el cliente.	

Tabla 2.23 HU No.18 buscar jefe de departamento

Historia de usuario	
Número: 23	Usuario: TODOS
Nombre: Buscar Jefe de departamento en el sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Cuarta
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar la operación correspondiente.	
Observaciones: Confirmado por el cliente.	

Tabla 2.24 HU No.19 buscar jefe de carrera

Historia de usuario	
Número: 24	Usuario: TODOS
Nombre: Buscar Jefe de carrera en el sistema.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: media
Puntos estimados: 1	Iteración asignada: Cuarta
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: El usuario debe entrar los datos requeridos y después podrá realizar la operación correspondiente.	

Observaciones: Confirmado por el cliente.

Tabla 2.25 HU No.20 modificar contraseña

Historia de usuario

Número: 25

Usuario: Administrador

Nombre: Modificar contraseña.

Prioridad en el negocio: Alta

Riesgo en desarrollo: media

Puntos estimados: 1

Iteración asignada: Cuarta

Programador responsable: Yuris Angel Acosta Zayas

Descripción: El usuario debe entrar los datos requeridos y después podrá realizar la operación correspondiente.

Observaciones: Confirmado por el cliente.

Anexo 2: Tarjetas CRC

Tabla 2.26 Tarjeta CRC No. 1

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

Tabla 2.27 Tarjeta CRC No. 2

Nombre de la Clase: Gestionar Profesor	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del profesor	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 los datos de un profesor	Listar_profesor

Tabla 2.28 Tarjeta CRC No. 3

Nombre de la Clase: Gestionar Estudiante	
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 2.29 Tarjeta CRC No. 4

Nombre de la Clase: Gestionar Actividades del sistema	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos de las actividades	Insertar_actividad
Muestra los datos de las actividades	Mostrar_actividad
Modifica los datos de las actividades	Modificar_actividad
Eliminar las actividades de la base de datos	Eliminar_actividad
Lista y exporta los datos de las actividades	Listar_actividad

Tabla 2.30 Tarjeta CRC No. 5

Nombre de la Clase: Gestionar Coordinadores	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del coordinador	Insertar_coordinador
Muestra los datos del coordinador	Mostrar_coordinador
Modifica los datos del coordinador	Modificar_coordinador
Eliminar los datos del coordinador	Eliminar_coordinador
Lista y exporta los datos del coordinador	Listar_coordinador

Tabla 2.31 Tarjeta CRC No. 6

Nombre de la Clase: Gestionar Tutores	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores

Inserta datos de los tutores	Insertar_tutores
Muestra los datos de los tutores	Mostrar_tutores
Modifica los datos de los tutores	Modificar_tutores
Eliminar tutores de la base de datos	Eliminar_tutores
Lista y exportar los tutores	Listar_tutores

Tabla 2.32 Tarjeta CRC No. 7

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

Tabla 2.33 Tarjeta CRC No. 8

Nombre de la Clase: Gestionar Asignaturas	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos de la asignatura	Insertar_asignatura
Muestra los datos de la asignatura	Mostrar_asignatura
Modifica los datos de la asignatura	Modificar_asignatura
Eliminar la asignatura de la base de datos	Eliminar_asignatura
Lista y exporta datos de asignatura	Listar_asignatura

Tabla 2.34 Tarjeta CRC No. 9

Nombre de la Clase: Gestionar Administrador	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del administrador	Insertar_administrador
Muestra los datos del administrador	Mostrar_administrador
Modifica los datos del administrador	Modificar_administrador
Eliminar los administrador de la base de datos	Eliminar_administrador
Lista y exporta los administrador	Listar_administrador

Tabla 2.35 Tarjeta CRC No. 10

Nombre de la Clase: Gestionar Semestre	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del semestre	Insertar_semestre
Muestra los datos del semestre	Mostrar_semestre
Modifica los datos del semestre	Modificar_semestre
Eliminar los semestre de la base de datos	Eliminar_semestre
Lista y exporta los semestre	Listar_semestre

Tabla 2.36 Tarjeta CRC No. 11

Nombre de la Clase: Gestionar Evaluación	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del evaluación	Insertar_evaluación
Muestra los datos del evaluación	Mostrar_evaluación
Modifica los datos del evaluación	Modificar_evaluación

Eliminar los evaluación de la base de datos	Eliminar_evaluación
Lista y exporta los evaluación	Listar_evaluación

Tabla 2.37 Tarjeta CRC No. 12

Nombre de la Clase: Gestionar Documentos	
Tipo de clase:	
Lógica del negocio	
Responsabilidades	Colaboradores
Inserta datos del documentos	Insertar_documentos
Muestra los datos del documentos	Mostrar_documentos
Modifica los datos del documentos	Modificar_documentos
Eliminar los documentos de la base de datos	Eliminar_documentos
Lista y exporta los documentos	Listar_documentos

Anexo 3: Tareas de ingeniería

Tabla 3.10 Tarea de Programación autenticar profesor

Tarea de Programación	
No. Tarea: 1	No. Historia: 1
Nombre de la tarea: Autenticar usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 22/02/2014	Fecha fin: 22/02/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite autenticar los profesor al sistema.	

Tabla 3.11 Tarea de Programación Insertar profesor

Tarea de Programación	
No. Tarea: 2	No. Historia: 2
Nombre de la tarea: Insertar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 23/02/2014	Fecha fin: 23/02/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar los profesor que van a interactuar con el sistema.	

Tabla 3.12 Tarea de Programación Mostrar profesor

Tarea de Programación	
No. Tarea: 3	No. Historia: 2
Nombre de la tarea: Mostrar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 24/02/2014	Fecha fin: 24/02/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite mostrar los datos de los profesor del sistema.	

Tabla 3.13 Tarea de Programación Modificar profesor

Tarea de Programación	
No. Tarea: 4	No. Historia: 2
Nombre de la tarea: Modificar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 25/02/2014	Fecha fin: 26/02/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar los datos de los profesor del sistema.	

Tabla 3.14 Tarea de Programación Eliminar profesor

Tarea de Programación	
No. Tarea: 5	No. Historia: 2
Nombre de la tarea: Eliminar profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 27/02/2014	Fecha fin: 27/02/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite eliminar el profesor seleccionado de la base de datos.	

Tabla 3.15 Tarea de Programación insertar estudiante

Tarea de Programación	
No. Tarea: 6	No. Historia: 3
Nombre de la tarea: insertar estudiante	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 15/03/2014	Fecha fin: 15/03/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar el estudiante en la base de datos.	

Tabla 3.16 Tarea de Programación mostrar estudiante

Tarea de Programación	
No. Tarea: 7	No. Historia: 3

Nombre de la tarea: mostrar estudiante	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 16/03/2014	Fecha fin: 16/03/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar el estudiante en la base de datos.	

Tabla 3.17 Tarea de Programación modificar estudiante

Tarea de Programación	
No. Tarea: 8	No. Historia: 3
Nombre de la tarea: modificar estudiante	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 17/03/2014	Fecha fin: 17/03/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar el estudiante en la base de datos.	

Tabla 3.18 Tarea de Programación eliminar estudiante

Tarea de Programación	
No. Tarea: 9	No. Historia: 3
Nombre de la tarea: eliminar estudiante	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 18/03/2014	Fecha fin: 18/03/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite eliminar el estudiante de la base de datos.	

Tabla 3.19 Tarea de Programación insertar actividades

Tarea de Programación	
No. Tarea: 10	No. Historia: 4
Nombre de la tarea: insertar actividad	

Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 20/03/2014	Fecha fin: 20/03/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar actividades en la base de datos.	

Tabla 3.20 Tarea de Programación mostrar actividades

Tarea de Programación	
No. Tarea: 11	No. Historia: 4
Nombre de la tarea: mostrar actividades	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 21/03/2014	Fecha fin: 21/03/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite mostrar actividades en la base de datos.	

Tabla 3.21 Tarea de Programación modificar actividades

Tarea de Programación	
No. Tarea: 12	No. Historia: 4
Nombre de la tarea: modificar actividades	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 22/03/2014	Fecha fin: 22/03/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar actividades en la base de datos.	

Tabla 3.22 Tarea de Programación eliminar actividades

Tarea de Programación	
No. Tarea: 13	No. Historia: 4
Nombre de la tarea: eliminar actividades	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 24/03/2014	Fecha fin: 24/03/2014

Programador responsable: Yuris Angel Acosta Zayas
Descripción: Permite eliminar actividades en la base de datos.

Tabla 3.23 Tarea de Programación insertar profesor guía.

Tarea de Programación	
No. Tarea: 14	No. Historia: 7
Nombre de la tarea: insertar profesor guía	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 07/04/2014	Fecha fin: 07/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar un profesor guía en la base de datos	

Tabla 3.24 Tarea de Programación mostrar profesor guía.

Tarea de Programación	
No. Tarea: 15	No. Historia: 7
Nombre de la tarea: mostrar profesor guía.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 08/04/2014	Fecha fin: 08/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite mostrar un profesor guía en la base de datos.	

Tabla 3.25 Tarea de Programación modificar profesor guía.

Tarea de Programación	
No. Tarea: 16	No. Historia: 7
Nombre de la tarea: modificar profesor guía.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 09/04/2014	Fecha fin: 10/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar un profesor guía en la base de datos.	

Tabla 3.26 Tarea de Programación eliminar profesor guía.

Tarea de Programación	
No. Tarea: 17	No. Historia: 7
Nombre de la tarea: insertar convocatoria.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 11/04/2014	Fecha fin: 11/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite eliminar un profesor guía en la base de datos.	

Tabla 3.27 Tarea de Programación insertar asignatura.

Tarea de Programación	
No. Tarea: 18	No. Historia: 8
Nombre de la tarea: insertar asignatura.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 14/04/2014	Fecha fin: 14/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar una asignatura en la base de datos.	

Tabla 3.28 Tarea de Programación mostrar asignatura.

Tarea de Programación	
No. Tarea: 19	No. Historia: 8
Nombre de la tarea: mostrar asignatura.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 16/04/2014	Fecha fin: 16/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite mostrar una asignatura en la base de datos.	

Tabla 3.29 Tarea de Programación modificar asignatura.

Tarea de Programación

No. Tarea: 20	No. Historia: 8
Nombre de la tarea: modificar asignatura.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 17/04/2014	Fecha fin: 17/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar una asignatura en la base de datos.	

Tabla 3.30 Tarea de Programación eliminar asignatura.

Tarea de Programación	
No. Tarea: 21	No. Historia: 8
Nombre de la tarea: eliminar una asignatura	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 17/04/2014	Fecha fin: 17/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite eliminar una asignatura en la base de datos	

Tabla 3.31 Tarea de Programación salvar curso.

Tarea de Programación	
No. Tarea: 22	No. Historia: 9
Nombre de la tarea: salvar fin de curso.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 18/04/2014	Fecha fin: 18/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite realizar una salva a cada curso.	

Tabla 3.32 Tarea de Programación insertar administrador.

Tarea de Programación	
No. Tarea: 23	No. Historia: 10
Nombre de la tarea: insertar administrador.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 21/04/2014	Fecha fin: 21/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar un administrador en la base de datos	

Tabla 3.33 Tarea de Programación mostrar administrador.

Tarea de Programación	
No. Tarea: 24	No. Historia: 10
Nombre de la tarea: mostrar administrador.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 22/04/2014	Fecha fin: 22/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite mostrar un administrador en la base de datos	

Tabla 3.34 Tarea de Programación modificar administrador.

Tarea de Programación	
No. Tarea: 25	No. Historia: 10
Nombre de la tarea: insertar administrador.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 23/04/2014	Fecha fin: 23/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar un administrador en la base de datos	

Tabla 3.35 Tarea de Programación eliminar administrador.

Tarea de Programación	
No. Tarea: 26	No. Historia: 10
Nombre de la tarea: insertar administrador.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 24/04/2014	Fecha fin: 24/04/2014

Programador responsable: Yuris Angel Acosta Zayas
Descripción: Permite eliminar un administrador en la base de datos

Tabla 3.36 Tarea de Programación insertar semestre.

Tarea de Programación	
No. Tarea: 27	No. Historia: 11
Nombre de la tarea: insertar semestre.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 25/04/2014	Fecha fin: 25/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar un semestre en la base de datos.	

Tabla 3.37 Tarea de Programación mostrar semestre.

Tarea de Programación	
No. Tarea: 28	No. Historia: 11
Nombre de la tarea: mostrar semestre.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 28/04/2014	Fecha fin: 28/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite mostrar un semestre en la base de datos.	

Tabla 3.38 Tarea de Programación modificar semestre.

Tarea de Programación	
No. Tarea: 29	No. Historia: 11
Nombre de la tarea: modificar semestre.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 29/04/2014	Fecha fin: 29/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar un semestre en la base de datos.	

Tabla 3.39 Tarea de Programación eliminar semestre.

Tarea de Programación	
No. Tarea: 30	No. Historia: 11
Nombre de la tarea: eliminar semestre.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 30/04/2014	Fecha fin: 30/04/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite eliminar un semestre en la base de datos.	

Tabla 3.40 Tarea de Programación insertar evaluación.

Tarea de Programación	
No. Tarea: 31	No. Historia: 12
Nombre de la tarea: insertar evaluaciones.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 05/05/2014	Fecha fin: 05/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar evaluaciones de estudiantes en la base de datos	

Tabla 3.41 Tarea de Programación mostrar evaluación.

Tarea de Programación	
No. Tarea: 32	No. Historia: 12
Nombre de la tarea: mostrar evaluaciones.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 06/05/2014	Fecha fin: 06/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite mostrar evaluaciones de estudiantes en la base de datos	

Tabla 3.42 Tarea de Programación modificar evaluación.

Tarea de Programación

No. Tarea: 33	No. Historia: 12
Nombre de la tarea: modificar e valuaciones	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 07/05/2014	Fecha fin: 07/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar evaluaciones de estudiantes en la base de datos.	

Tabla 3.43 Tarea de Programación eliminar evaluaciones.

Tarea de Programación	
No. Tarea: 34	No. Historia: 12
Nombre de la tarea: eliminar evaluaciones.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 08/05/2014	Fecha fin: 08/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite eliminar evaluaciones de estudiantes en la base de datos.	

Tabla 3.44 Tarea de Programación Exportar listado de evaluaciones por semestre.

Tarea de Programación	
No. Tarea: 35	No. Historia: 13
Nombre de la tarea: exportar listado de evaluaciones por semestre.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 09/05/2014	Fecha fin: 10/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite exportar un listado de las evaluaciones de estudiantes por semestre en formato pdf.	

Tabla 3.45 Tarea de Programación Exportar listado de evaluaciones por año.

Tarea de Programación	
No. Tarea: 36	No. Historia: 14

Nombre de la tarea: exportar listado de evaluaciones por año.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 09/05/2014	Fecha fin: 10/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite exportar un listado de las evaluaciones de estudiantes por año en formato pdf.	

Tabla 3.46 Tarea de Programación Exportar listado de estudiantes por año.

Tarea de Programación	
No. Tarea: 37	No. Historia: 15
Nombre de la tarea: exportar listado de estudiantes por año.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 11/05/2014	Fecha fin: 12/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite exportar un listado de los estudiantes por año en formato pdf.	

Tabla 3.47 Tarea de Programación Exportar un listado de estudiantes en general.

Tarea de Programación	
No. Tarea: 38	No. Historia: 16
Nombre de la tarea: Exportar un listado de los estudiantes de la carrera.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 13/05/2014	Fecha fin: 14/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite Exportar un listado de los estudiantes existentes en la carrera en formato pdf.	

Tabla 3.48 Tarea de Programación Exportar listado de estudiantes de pregrado.

Tarea de Programación	
No. Tarea: 39	No. Historia: 26

Nombre de la tarea: Exportar listado de estudiantes de pregrado.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 15/05/2014	Fecha fin: 16/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite Exportar listado de estudiantes de pregrado	

Tabla 3.49 Tarea de Programación Exportar listado los profesores.

Tarea de Programación	
No. Tarea: 40	No. Historia: 17
Nombre de la tarea: Exportar listado de los profesores.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 17/05/2014	Fecha fin: 18/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite Exportar listado de los profesores existentes en el departamento en formato pdf.	

Tabla 3.50 Tarea de Programación Insertar documentos.

Tarea de Programación	
No. Tarea: 41	No. Historia: 18
Nombre de la tarea: insertar documentos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 19/05/2014	Fecha fin: 19/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite insertar documentos para la bibliografía del departamento en la base de datos.	

Tabla 3.51 Tarea de Programación mostrar documentos

Tarea de Programación	
No. Tarea: 42	No. Historia: 18

Nombre de la tarea: mostrar documentos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 20/05/2014	Fecha fin: 20/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite mostrar documentos para la bibliografía del departamento en la base de datos.	

Tabla 3.52 Tarea de Programación modificar documentos

Tarea de Programación	
No. Tarea:43	No. Historia: 18
Nombre de la tarea: modificar documentos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 21/05/2014	Fecha fin: 22/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite modificar documentos para la bibliografía del departamento en la base de datos.	

Tabla 3.53 Tarea de Programación eliminar documentos

Tarea de Programación	
No. Tarea: 44	No. Historia: 18
Nombre de la tarea: eliminar documentos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 23/05/2014	Fecha fin: 23/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite eliminar documentos de la bibliografía del departamento en la base de datos.	

Tabla 3.54 Tarea de Programación buscar estudiante

Tarea de Programación

No. Tarea: 45	No. Historia: 19
Nombre de la tarea: buscar estudiantes en el sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 26/05/2014	Fecha fin: 26/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite buscar estudiantes en la base de datos del sistema.	

Tabla 3.55 Tarea de Programación buscar coordinador de año

Tarea de Programación	
No. Tarea: 46	No. Historia: 20
Nombre de la tarea: buscar coordinador de año en el sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 27/05/2014	Fecha fin: 27/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite buscar coordinador de año en la base de datos del sistema.	

Tabla 3.56 Tarea de Programación buscar tutor

Tarea de Programación	
No. Tarea: 47	No. Historia: 21
Nombre de la tarea: buscar tutor en el sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 28/05/2014	Fecha fin: 28/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite buscar tutor en la base de datos del sistema.	

Tabla 3.57 Tarea de Programación buscar profesor guía

Tarea de Programación	
No. Tarea: 48	No. Historia: 22
Nombre de la tarea: buscar profesor guía en el sistema	

Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 29/05/2014	Fecha fin: 29/05/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite buscar profesor guía en la base de datos del sistema.	

Tabla 3.58 Tarea de Programación buscar jefe de departamento

Tarea de Programación	
No. Tarea: 49	No. Historia: 23
Nombre de la tarea: buscar jefe de departamento en el sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 01/06/2014	Fecha fin: 01/06/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite buscar jefe de departamento en la base de datos del sistema.	

Tabla 3.59 Tarea de Programación buscar jefe de carrera

Tarea de Programación	
No. Tarea: 50	No. Historia: 24
Nombre de la tarea: buscar jefe de carrera en el sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 02/06/2014	Fecha fin: 02/06/2014
Programador responsable: Yuris Angel Acosta Zayas	
Descripción: Permite buscar estudiantes en la base de datos del sistema.	

Tabla 3.60 Tarea de Programación modificar contraseña

Tarea de Programación	
No. Tarea: 51	No. Historia: 25
Nombre de la tarea: modificar contraseña	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 03/06/2014	Fecha fin: 03/06/2014

Programador responsable: Yuris Angel Acosta Zayas
Descripción: Permite modificar la contraseña de los usuarios del sistema en la base de datos.

Anexo 4: Pruebas de Aceptación

Tabla 6.61 PA: Prueba para comprobar la entrada de un usuario al sistema.

Prueba de aceptación
Historia de usuario # 1: Autenticar profesor
Nombre: Prueba para comprobar la entrada de un profesor al sistema.
Descripción: Validación de entrada de los datos del profesor.
Condiciones de ejecución: El profesor debe introducir su nombre y contraseña.
Entrada/Pasos de ejecución: El profesor escribe sus datos y pulsa “Entrar”.
<p>Resultado esperado:</p> <ul style="list-style-type: none">➤ Si el profesor 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 profesor 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 3.62 PA: Prueba para comprobar la entrada de datos del usuario

Prueba de aceptación
Historia de usuario # 2: Gestionar profesor
Nombre: Prueba para comprobar la entrada de datos del profesor.
Descripción: Validación de entrada de los datos del profesor.
Condiciones de ejecución: El profesor debe introducir los datos.
Entrada/Pasos de ejecución: El profesor escribe los datos.
<p>Resultado esperado:</p> <ul style="list-style-type: none">➤ Si se introducen los datos correctamente se creará el nuevo profesor. <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 ya creado. ➤ Se inserte en el campo Email un correo no válido. ➤ Se dejen campos obligatorios vacíos.
Evaluación de la prueba: Aceptada.

Tabla 3.63 PA: Prueba para comprobar la entrada de datos del estudiante.

Prueba de aceptación
Historia de usuario # 3: Gestionar estudiante
Nombre: Prueba para comprobar la entrada de datos del estudiante.
Descripción: Validación de entrada de los datos del estudiante.
Condiciones de ejecución: El profesor debe introducir los datos de un estudiante.
Entrada/Pasos de ejecución: El profesor escribe los datos del estudiante.
<p>Resultado esperado:</p> <ul style="list-style-type: none"> ➤ Si se introducen los datos correctamente se insertará el estudiante. <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 3.64 PA: Prueba para comprobar la entrada de datos de las actividades

Prueba de aceptación
Historia de usuario # 8: Gestionar actividad
Nombre: Prueba para comprobar la entrada de datos de las actividades
Descripción: Validación de entrada de los datos de las actividades.
Condiciones de ejecución: El usuario introduce los datos correspondientes.
Entrada/Pasos de ejecución: El usuario escribe los datos de las actividades.
Resultado esperado:

<p>➤ Si se insertan los datos correctamente se inserta la actividad.</p> <p>Se emite un mensaje de error en caso de que:</p> <p>➤ Se dejen campos obligatorios vacíos.</p> <p>➤ Que la fecha de cumplimiento sea menor que la actual.</p>
Evaluación de la prueba: Aceptada.

Tabla 3.65 PA: Prueba para comprobar la entrada de datos de la asignatura

Prueba de aceptación
Historia de usuario # 28: Gestionar asignatura.
Nombre: Prueba para comprobar la entrada de datos de la asignatura.
Descripción: Validación de entrada de los datos de la asignatura.
Condiciones de ejecución: El profesor introduce los datos correspondientes.
Entrada/Pasos de ejecución: El profesor escribe los datos de la asignatura.
<p>Resultado esperado:</p> <p>➤ Si se insertan los datos correctamente se inserta la asignatura.</p> <p>Se emite un mensaje de error en caso de que:</p> <p>➤ Se dejen campos obligatorios vacíos.</p>
Evaluación de la prueba: Aceptada.

Anexo 5: Interfaces de usuario

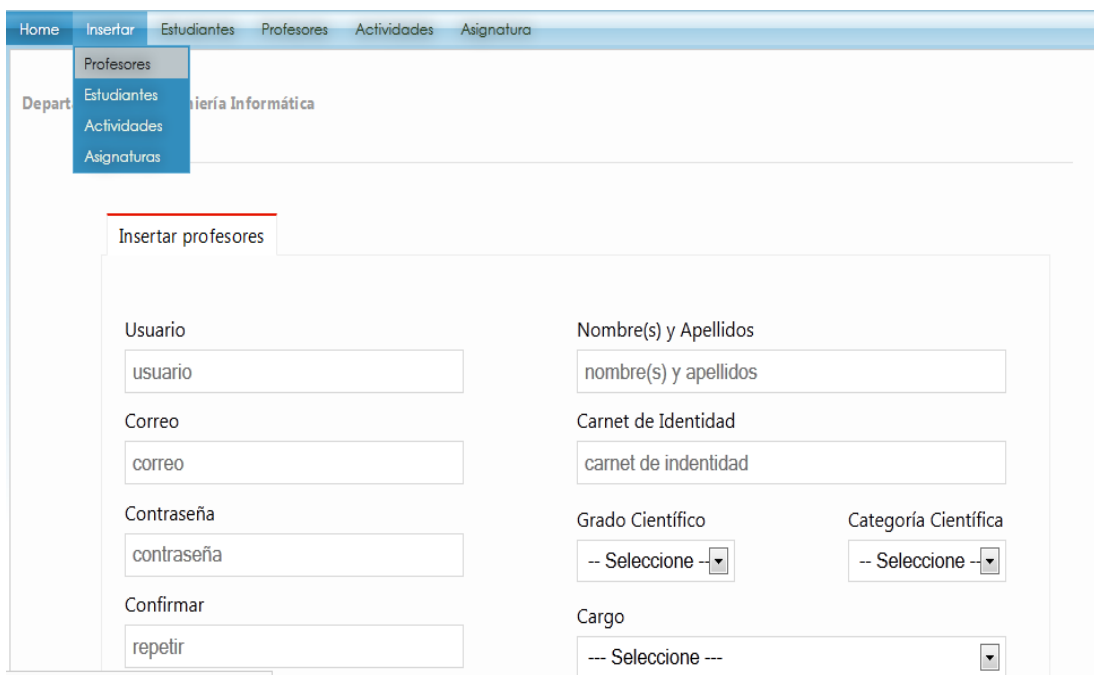
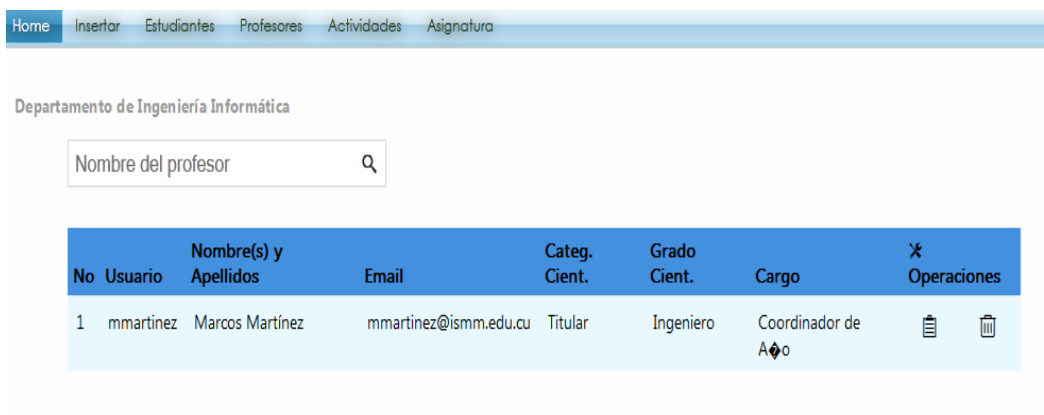


Figura 3.3 Interfaz de usuario Insertar profesor





No	Usuario	Nombre(s) y Apellidos	Email	Categ. Cient.	Grado Cient.	Cargo	Operaciones
1	mmartinez	Marcos Martínez	mmartinez@ismm.edu.cu	Titular	Ingeniero	Coordinador de A&D	 

Figura 3.4 Interfaz de usuario administrar profesor

Home Insertar Estudiantes Profesores Actividades Asignatura

Insertar actividades

Actividad

nueva actividad

Actividad
Llene este campo por favor

Fecha

+ Insertar

Figura 3.5 Interfaz de usuario gestionar actividades

Home Insertar Estudiantes Profesores Actividades Asignatura

Departamento de Ingeniería Informática

No	Actividad	Responsable	Año	XOperaciones
1	prueba 1	Ivan Feria Salmon	20/05/14	

Figura 3.6 Interfaz de usuario administrar actividades

Home Insertar Estudiantes Profesores Actividades Asignatura

+ Insertar

Llene todos los campos

Estudiante Padres Convivencia

Nombre(s)

nombres

Apellidos

Dirección particular

dirección

Carnet de Identidad

CI

Correo electrónico

correo

Teléfono

Cargo que ocupa

--- Seleccione ---

Proviene de:

ORDEN 18

Habilidades

habilidades

Centro de traslado

Seleccione el centro

Fecha de traslado

31/12/12

Figura 3.7 Interfaz de usuario gestionar estudiantes




Home Insertar Estudiantes Profesores Actividades Asignatura									
Departamento de Ingeniería Informática									
Nombre del estudiante 									
No	Año	CI	Sexo	Nombre(s) y Apellidos	Email	F/Nacimiento	Dirección	✕Operaciones	
1	1ro	33645457858	M	Yuris Angel Acosta Zayas	yaacosta@info.ismm.edu.cu	17/10/88	Guatemala #36 Mayarí Holguín		

Figura 3.8 Interfaz de usuario administrar estudiantes