

Universidad de las Ciencias Informáticas
Facultad 8

Trabajo de Diploma para Optar por el Título de
Ingeniero en Ciencias Informáticas

Título: Sistema para el control de la información y evaluación de los profesores en los departamentos docentes de la facultad 8.

Autores:

Yaneisy Pérez Neira
Rolian Ruíz Rodríguez

Tutores:

Ing. Sergio Díaz Catalá
Ing. Joe Fernández Vanega

Ciudad de la Habana, Junio de 2009.

“Año del 50 Aniversario del Triunfo de la Revolución.”

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los _ días del mes de junio del año 2009.

Yaneisy Pérez Neira

Rolian Ruiz Rodríguez

Firma del autor

Firma del autor

Ing. Sergio Díaz Catalá

Firma del tutor

Ing. Joe Fernández Vanega

Firma del tutor

"Hay hombres que luchan un día y son buenos.

Hay otros que luchan un año y son mejores.

Hay quienes luchan muchos años y son muy buenos.

Pero hay los que luchan toda la vida: esos son los imprescindibles."

Bertolt Brecht.

sergiod@uci.cu

ypneira@estudiantes.uci.cu

rrrodriguez@estudiantes.uci.cu

Yaneisy

Mis agradecimientos a todas aquellas personas que de una forma u otra influyeron en la realización de este trabajo.

A mis compañeros de aula (grupo 8501 curso 2008-2009) por ser tan especiales y estar siempre disponibles cuando se necesitan.

A mis amigas de siempre y por siempre Marcia L. Rodríguez y Elizabeth Arencibia.

A mi compañero de tesis, Rolian Ruiz, que además de eso se ha convertido en un amigo especial.

Las amistades van cambiando y se renuevan al pasar del tiempo, pero cada una deja una huella que es lo que hace que se recuerden siempre. Un beso para esas amistades que no influyeron directamente pero que siempre estuvieron presentes y aportaron su granito de arena.

Agradecer a esta Revolución y al siempre comandante Fidel Castro la oportunidad de estudiar en esta Universidad.

Rolian

Mis agradecimientos a todas aquellas personas que de una forma u otra influyeron y me apoyaron en la realización de este trabajo.

A mis hermanos Rádriel y Emilio.

A mi familia por el apoyo que siempre me han dado.

A mis compañeros de aula y amistades.

A mis tutores.

A Michel , Lizy y Humberto, gracias por su ayuda que juntos formamos una comunidad.

A mi compañera de tesis, Yaneisy Pérez Neira que este tiempo juntos apoyándonos y ayudándonos mutuamente ha hecho que sea una persona muy especial para mí.

Agradecer a mis padres y a la Revolución la oportunidad de estudiar en esta Universidad y la formación que me han dado.

Yaneisy

Dedico esta tesis a mis padres, Omaidá (Mayi), Pedro (Pipe), Caridad (Mima), y Nelson (Nelsito Pérez) por todo el esfuerzo y abnegación que han puesto en mí. Por la confianza que me han dado y el ánimo de seguir siempre adelante.

A mis abuelos, (Tomasa, Ismael y Rey) que siempre han querido verme graduada.

A mis hermanos (Pedry, Nelsy y Noslen), por alegrarme la vida con su presencia.

A toda mi familia, por darme el apoyo que he necesitado.

A mis vecinos, por su apoyo y preocupación.

Rolian

Dedico esta tesis a mis padres, Zoraida Rodríguez y Emilio Ruiz que me han educado por el camino correcto.

A mis hermanos (Emilio y Rádriel).

A toda mi familia y amigos.

A Suyen Aragón que aunque no va a graduarse junto a nosotros, pero estará siempre presente como una más del grupo.

A Yonder Rodríguez un gran amigo que hoy ya no se encuentra entre nosotros del cual aprendí a luchar siempre por mis sueños.

En la Universidad de las Ciencias Informáticas los profesores son distribuidos en departamentos docentes en cada facultad. En estos departamentos los jefes de cada uno son los que llevan el control de la información de los profesores y medios básicos que pertenecen al mismo. En la facultad 8 no se cuenta con un software para llevar un control más eficiente de toda esta información y brindar facilidad a los jefes de departamento en el proceso de evaluar a los profesores.

Para dar solución a lo anteriormente planteado, en el presente trabajo se realizan las investigaciones necesarias en cuanto al desarrollo del software para lograr construir una aplicación que cumpla con los aspectos requeridos. Posteriormente se implementa un software, desarrollado con herramientas de software libre y utilizando la metodología XP, que tiene como objetivo almacenar la información de los medios básicos y profesores, obtener reportes y además brindar a los jefes de departamento la posibilidad de evaluar a los profesores según las actividades realizadas y del cumplimiento del plan de trabajo anual.

Palabras Claves

Departamentos docentes, profesores, medios básicos.

ÍNDICE DE CONTENIDOS

INDICE DE TABLAS III

ÍNDICE DE ILUSTRACIONES VI

INTRODUCCIÓN 1

1.1. Introducción 5

1.2. Sistemas de Gestión de la Información 5

1.3. Sistemas automatizados similares existentes 5

1.4. Tendencias y Tecnologías Actuales 8

1.4.1. Política de Migración a Software libre 8

1.4.2. ¿Por qué realizar una Aplicación Web? 8

1.4.3. Lenguajes de Programación 9

1.4.4. Lenguajes del lado del servidor 9

1.4.5. Lenguajes del lado del cliente 11

1.4.6. AJAX 13

1.5. Frameworks 13

1.5.1. ¿Por qué usar un Framework? 14

1.6. Arquitectura 18

1.7. ¿Qué son los Servicios Web? 19

1.7.1. Servidores Web 20

1.8. Sistemas Gestores De Base De Datos (SGBD) 22

1.9. Metodologías de Desarrollo De Software 24

1.9.1. Proceso de desarrollo de software 24

1.9.2. Metodologías 24

1.10. Herramientas de Desarrollo 30

1.11. Conclusiones 31

2.1. Introducción 32

2.2. Descripción de los procesos vinculados al campo de acción 32

2.2.1. Flujo actual del proceso 32

2.2.2. Objeto de automatización 33

2.2.3. Personas relacionadas con el sistema 33

2.3. Planificación 34

2.4. Exploración 34

2.5. Planificación de la Entrega 41

2.5.1. Estimación de esfuerzos por historias de usuario 42

2.5.2. Plan de entregas 43

2.6. Plan de iteraciones 43

2.6.3. Plan de duración de las iteraciones 44

2.7. Conclusiones 45

3.1. Introducción 46

3.2. Diseño 46

3.3. Tarjetas CRC 47

3.4.	Diseño de la Base de Datos.....	52
3.5.	Implementación	54
3.6.	Conclusiones	63
4.1.	Introducción	64
4.2.	Pruebas de Aceptación (PA).....	64
4.2.1.	Módulo Medios Básicos	65
4.2.2.	Módulo Profesor	67
4.2.3.	Módulo Plan de Trabajo.....	70
4.2.4.	Módulo Actividades y eventos.....	73
4.2.5.	Módulo Evaluaciones.....	76
4.2.6.	Módulo Reportes	77
4.3.	Conclusiones	77
	Conclusiones Generales.....	78
	Trabajos Citados.....	80
	Bibliografía.....	81
	Glosario de Términos.....	83

INDICE DE TABLAS

Tabla 1: Personas relacionadas con el sistema.34

Tabla 2: Representación de una historia de usuario.35

Tabla 3: HU Gestionar medios básicos.36

Tabla 4: HU Gestionar datos personales.36

Tabla 5: HU Gestionar actividades realizadas.....37

Tabla 6: HU Crear plan de trabajo.37

Tabla 7: HU Gestionar actividades del profesor.38

Tabla 8: HU Generar reportes.....38

Tabla 9: HU Asignar evaluación.....39

Tabla 10: HU Reportes del módulo Superación.39

Tabla 11: HU Gestionar Plan de Trabajo Individual del profesor.40

Tabla 12: HU Gestionar eventos realizados.40

Tabla 13: HU Gestionar usuarios41

Tabla 14: Estimación de esfuerzos por historias de usuario.....42

Tabla 15: Plan de entregas.43

Tabla 16: Plan de duración de las iteraciones.....44

Tabla 17: Cronograma de entrega.45

Tabla 18: Plantilla Tarjeta CRC.....47

Tabla 19: Tarjeta CRC de Departamento.....48

Tabla 20: Tarjeta CRC de Cuenta del Profesor.....48

Tabla 21: Tarjeta CRC de Usuario.49

Tabla 22: Tarjeta CRC de Medios Básicos.....49

Tabla 23: Tarjeta CRC de Actividades del profesor.....50

Tabla 24: Tarjeta CRC de Plan de trabajo.50

Tabla 25: Tarjeta CRC de Evaluaciones.51

Tabla 26: Tarjeta CRC de Eventos.51

Tabla 27: Tarjeta CRC de Reportes.....52

Tabla 28: TI Crear cuenta de usuario.....54

Tabla 29: TI Buscar cuenta de usuario.....55

Tabla 30: Modificar cuenta de usuario.	55
Tabla 31: Llenar datos personales.	56
Tabla 32: TI Ver y modificar datos personales.	56
Tabla 33: TI Crear medio básico.	57
Tabla 34: TI Buscar y listar medios básicos.	57
Tabla 35: TI Crear plan de trabajo	58
Tabla 36: TI Gestionar Plan de Trabajo Individual del profesor.	58
Tabla 37: TI Adicionar actividades realizadas	59
Tabla 38: TI Modificar actividades realizadas.....	59
Tabla 39: TI Adicionar y modificar eventos realizados	60
Tabla 40: TI Adjuntar constancia del evento.	60
Tabla 41: Gestionar actividades del profesor	61
Tabla 42: TI Reportes del módulo Superación	61
Tabla 43: Asignar evaluación.	62
Tabla 44: TI Generar reportes.....	62
Tabla 45: Plantilla Prueba de Aceptación.....	65
Tabla 46: PA Adicionar medio básico.....	65
Tabla 47: PA Modificar medio básico.	66
Tabla 48: PA Eliminar medio básico.....	66
Tabla 49: PA Mostrar medio básico.	67
Tabla 50: PA Modificar datos personales.	67
Tabla 51: PA Mostrar datos personales.	68
Tabla 52: PA Crear y modificar cuentas de usuarios.....	68
Tabla 53: PA Eliminar cuentas de usuarios.....	68
Tabla 54: PA Crear y modificar cuentas de los profesores.....	69
Tabla 55: PA Eliminar cuentas de profesores.	69
Tabla 56: PA Crear plan.....	70
Tabla 57: PA Modificar plan.	70
Tabla 58: PA Adicionar plan de trabajo del profesor.	71
Tabla 59: PA Adicionar plan de trabajo del profesor.	71

Tabla 60: PA Mostrar plan de trabajo individual del profesor.	72
Tabla 61: PA Eliminar el plan de trabajo del profesor.....	72
Tabla 62: PA Listar planes de trabajo.	73
Tabla 63: PA Adicionar, modificar y eliminar actividades.	73
Tabla 64: PA Adjuntar constancia.	74
Tabla 65: PA Adicionar y modificar actividades.....	74
Tabla 66: PA Adicionar, modificar y eliminar eventos.....	75
Tabla 67: PA Adjuntar documento.	75
Tabla 68: PA Evaluación mensual y trimestral del profesor.....	76
Tabla 69: PA Evaluación anual del profesor.....	76
Tabla 70: PA Generación de reportes.	77

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Flujo de Trabajo de symfony 16

Ilustración 2: Patrón MVC. 18

Ilustración 3: Metodología XP 27

Ilustración 4: Aptana Studio 30

Ilustración 5: EasyEclipse 30

Ilustración 6: Modelo de datos 53

INTRODUCCIÓN

En el mes de enero del año 2000 fue creado el Ministerio de la Informática y las Comunicaciones (MIC), con la misión fundamental de promover el uso masivo de las tecnologías de la información y las comunicaciones (TIC) en la economía nacional, la sociedad y ponerla a disposición de los ciudadanos. Como premisa fundamental de mantener el uso ordenado e intensivo de las TIC para satisfacer las necesidades de conocimiento e información en las personas, Cuba decide dar los primeros pasos en la informatización de la sociedad. Desde marzo del 2002 se introdujo la enseñanza de la computación en el Sistema Nacional de Educación. Su objetivo fundamental es elevar la calidad del aprendizaje y con ello garantizar un proceso de educación continuo, en el que se fomente la cultura general de la población sobre el uso de estas tecnologías. Conjuntamente con ello se suceden diferentes pasos primordiales que irían encaminados, en gran medida, a la materialización de una idea fundamentada en el desarrollo informático del país.

Disímiles son los ejemplos que se pueden citar entre ellos se engloba la creación de los Joven Club de Computación y Electrónica, llamados los núcleos iniciales de la informatización comunitaria, que han permitido el acceso gratuito a las TIC a personas de todas las edades, principalmente jóvenes y niños.

Se destaca sin duda alguna la creación de la Universidad de las Ciencias Informáticas (UCI), creada al calor de la Batalla de Ideas como un proyecto de la Revolución con el único objetivo de informatizar todos los sectores de la nación cubana y desarrollar la industria del Software para contribuir al desarrollo económico del país. Desde su inauguración en septiembre del año 2002, la UCI ha contado con una matrícula promedio anual de 2000 estudiantes, combinando el estudio con la producción y la investigación. Actualmente, cuenta con diez facultades en su sede principal a las cuales se suman otras tres facultades regionales situadas en La Habana, Granma y Ciego de Ávila. El centro universitario UCI en su sede principal posee actualmente 10 000 estudiantes y 3000 profesores, aproximadamente, que tienen el mérito de formar a los futuros profesionales del país. Muchos de estos profesores son graduados en la misma universidad y continúan la labor elemental de la enseñanza en el cumplimiento del servicio social.

Cada facultad de la UCI, identificada con un número del 1 al 10, tiene a los profesores distribuidos por departamentos docentes en dependencia de la asignatura que imparte. La Facultad 8, específicamente,

cuenta con cinco departamentos docentes (Humanidades, Sistemas Digitales, Ciencias Básicas, Programación e Ingeniería de Software – Práctica Profesional). Cada departamento tiene un jefe (máximo responsable en esta área), encargado de supervisar y controlar todos los datos, documentos, eventos, cursos y otras actividades, referente a cada uno de los profesores que pertenecen a su departamento y manejan dicha información con el objetivo de obtener resultados y evaluarlos trimestralmente, según su desempeño en Deficiente, Adecuado o Superior y anualmente, en Bien, Regular o Mal, para el expediente docente del profesor. Ellos mismos son los responsables de los medios básicos que pertenecen al departamento y de velar por su cuidado.

El plan de trabajo de cada profesor es creado individualmente, constituido por un grupo de tareas, acciones, actividades a realizar de tipo docente, extensión, productivo, científico, y de postgrado. Al elaborar este plan, el profesor se lo entrega al jefe de departamento, el cual tiene autoridad para revisarlo y modificarlo según la categoría docente (Instructor, Asistente, Auxiliar, Titular) o categoría científica (Máster o Doctor) del profesor y lo que se le exige para la misma por resolución ministerial. Velar por el cumplimiento de este plan es tarea fundamental del jefe de departamento, el cual en la actualidad lleva todos estos registros en documentos Word y Excel. La constancia de las actividades que realizan los profesores o eventos en los que participan se adjuntan al expediente del profesor para así tener referencia de su desempeño. Por otra parte, cada departamento tiene medios básicos que son utilizados por todos los profesores que pertenecen al mismo, de los que no se lleva un registro y control estricto, y deben ser chequeados por los jefes de departamentos.

Estos registros y datos guardados, utilizando esta vía, hacen que la información se encuentre distribuida, por lo que es engorroso obtener resultados de manera rápida y precisa. Además, puede haber pérdida de información y poca eficiencia en el momento que se utiliza la misma como apoyo al proceso de evaluación de los profesores.

Expuesta la situación problemática existente en los departamentos docentes de la Facultad 8, se delimita el siguiente **problema de investigación**:

¿Cómo lograr que los jefes de departamentos puedan controlar de una forma más eficiente toda la información referente a los profesores y medios básicos del departamento que le posibilite la toma de decisiones?

Debido a ello se definen como:

Objeto de estudio: Gestión de la información en los departamentos docentes.

Campo de acción: Gestión de la información de los profesores y medios básicos en los departamentos docentes de la facultad 8 de la Universidad de las Ciencias Informáticas.

Idea a defender: Si se implementa un software para la gestión de información en los departamentos, se automatizarían algunos procesos que se llevan a cabo en los mismos y serviría de apoyo a los jefes de departamentos para la toma de decisiones.

Objetivo general: Desarrollar un software que posibilite gestionar y almacenar de forma segura y eficiente toda la información de los profesores de la facultad 8 así como de los medios básicos pertenecientes a cada uno de los departamentos docentes.

Objetivos específicos:

1. Definir los objetivos y alcance del sistema.
2. Implementar el Sistema.
3. Validar el sistema propuesto.
4. Entregar un documento que recoja todo el proceso de desarrollo del software.

Tareas de investigación:

1. Investigar sobre la existencia de algún software con las mismas finalidades en nuestra universidad, en Cuba y en el mundo.
2. Entrevistar a los jefes de departamento de la Facultad 8.
3. Investigar sobre las tendencias en el desarrollo del software y tecnologías actuales, sus ventajas y desventajas.
4. Seleccionar las herramientas, lenguajes y metodología a utilizar.
5. Definir los procesos vinculados al campo de acción.

El presente trabajo está estructurado en 4 capítulos.

Capítulo 1: Contiene la fundamentación teórica del tema, abordando principalmente los lenguajes de programación y las tecnologías que se utilizan en el desarrollo de la aplicación. También se exploran soluciones existentes similares al campo de acción para tener una guía de las posibles automatizaciones que se pueden realizar.

Capítulo 2: Contiene los principales procesos que se realizan en los departamentos así como el flujo normal de los mismos. Aborda los aspectos funcionales para el desarrollo del sistema, se definen los procesos fundamentales por medio de las historias de usuarios creadas por el cliente y se realiza la planificación de entrega de los diferentes módulos que componen la aplicación.

Capítulo 3: Este capítulo contiene los aspectos relacionados con el diseño e implementación del sistema. Se presenta la técnica de tarjetas CRC, el modelo de datos y además las tareas de ingeniería de cada módulo del sistema para garantizar la entrega en la planificación establecida.

Capítulo 4: Este capítulo está dedicado a las pruebas que se le realizan al funcionamiento del software, las pruebas de aceptación del cliente. Las pruebas se realizan por módulos para la aceptación de cada uno de forma independiente.

1

FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

El siguiente capítulo aborda los elementos teóricos que soportan la presente investigación, en él se elabora un estudio sobre los sistemas similares existentes, así como de las principales tecnologías y tendencias que existen en la actualidad en el desarrollo de software. Además se justifica la selección de aquellas que formarán parte de la propuesta de solución, teniendo en cuenta las necesidades existentes y el entorno donde se enmarca el objeto de estudio.

1.2. Sistemas de Gestión de la Información.

De forma general la información es un conjunto de datos que se encuentran organizados y que tienen un significado sobre un determinado proceso o fenómeno. Además permite resolver problemas y tomar decisiones por lo que se ha convertido en un elemento importante en el desarrollo del mundo y de la sociedad. La Gestión de la información, es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración y conservación con el objetivo de obtener la información adecuada en el tiempo oportuno y con los resultados deseados. Los sistemas de gestión de la información surgen entonces con el objetivo de controlar los grandes volúmenes de información de una forma fácil, sencilla, confiable y además para obtener el mayor rendimiento de este recurso.

1.3. Sistemas automatizados similares existentes.

Actualmente cada día se hace necesario y frecuente la inserción de sistemas informáticos para controlar las actividades inherentes a los proceso de gestión de la información. Es imprescindible para las empresas u organizaciones mantener un estricto control sobre las actividades o cualidades que competen a cada uno de sus trabajadores lo que les garantiza un punto de partida para realizar sus procesos evaluativos. Es de sumo interés para las mismas, además, poseer un control eficiente de los medios básicos con los que cuenta. En el mundo globalizado en que se vive, la única manera de alcanzar el éxito es garantizando el cumplimiento de las actividades antes mencionadas por lo cual el volumen de los sistemas informáticos encargados de controlar las mismas se ha elevado considerablemente.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

A continuación se realiza el análisis de varios sistemas informáticos, a nivel internacional, nacional y de la universidad, que brindan un mejor punto de partida para el desarrollo de la aplicación.

En el plano internacional se encuentra el Sistema de Control de Bienes Web- UG, el mismo es un módulo que forma parte del Sistema Integral de Información Administrativa (SIIA) creado en la Universidad de Guanajuato (México). Este módulo ha sido desarrollado con el propósito de apoyar a las unidades académicas y administrativas de la Universidad en el control de los bienes materiales que tienen bajo su cuidado. Es un sistema administrativo que permite el registro oportuno, brindando información veraz, con el objetivo de garantizar el resguardo de los bienes, así como la identificación, registro y etiquetado de los bienes con eficacia, además de la custodia de los documentos que amparan la propiedad de los mismos. La aplicación permite al usuario, entre otras funcionalidades, realizar su propio registro de bienes, toda la información que el usuario captura queda disponible para el departamento de control de bienes, entre los datos que se guardan se encuentran la marca y modelo, la descripción y el número de serie. Una parte importante de la aplicación es el manejador de reportes, con esta herramienta es posible desplegar reportes con información del sistema y se realizan teniendo en cuenta distintos criterios.

Cuba por su parte cuenta con el sistema Suite Atena, el cual es un sistema Contable creado por la empresa DESOFT. Al sistema se integra un Módulo de Activos Fijos que permite entre otros aspectos controlar los activos fijos por entidad, actividad, centros de costos y áreas de responsabilidad. Controla además los movimientos de altas y bajas lo cual sin dudas es de suma importancia. El sistema también posibilita la generación de reportes así como la creación de usuarios y grupos con sus permisos con lo que se garantiza la seguridad.

En la Universidad de las Ciencias informáticas actualmente se trabaja con el Sistema de Gestión Integral ASSETS NS. Este es un sistema comercializado por la firma panameña D'MARCO S.A. y distribuido en Cuba por INFOMASTER, entidad informática perteneciente a la Empresa Nacional de Producción y Servicios a la Educación Superior del MES. Es un sistema flexible, amigable, con ayuda en línea que puede ser instalado en una microcomputadora o sobre varias, funcionando en ambiente multiusuario incluidas estaciones remotas. Asimismo, proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario. ASSETS NS es una aplicación cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000, utilizando adicionalmente Crystal Reports 7.0 para la generación de reportes de salidas. La aplicación tiene un gran

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

rendimiento debido a su filosofía cliente-servidor, que proporciona mayor rapidez y eficiencia en la operación de los diferentes procesos y en la obtención de resultados, además realiza una manipulación mínima de datos en el sitio del cliente, todos los procesos se ejecutan en el servidor viajando al usuario las respuestas a cada uno de los procesos, lo que proporciona por una parte, que la carga en la red no sea significativa, permitiendo tener mayor número de usuarios conectados y, por otra, disminuye los costos, dado que el hardware en el puesto del cliente no tiene grandes exigencias en cuanto a recursos. Entre los módulos que presenta la aplicación se encuentran Control de Inventarios y Activos Fijos donde el primero permite controlar las existencias, disponibilidad y reserva de los diferentes elementos en múltiples almacenes, realizar conteos físicos de los productos y hacer ajustes tanto de existencia física como de precios de costo de los artículos almacenados y el segundo permite controlar los medios básicos y definir la ubicación física de los mismos. Además, realizar revalorizaciones de los activos y emitir, al cierre del mes, el Comprobante de Operaciones por la depreciación de los activos.

En cuanto a la gestión de la información de los profesores por departamentos (actividades docentes, extradocentes, evaluación del profesor, entre otros) en la Universidad se han realizado varias tesis con este propósito, pero en la actualidad no se ha llegado a la materialización de un software, estas han quedado solo en análisis y diseño. A pesar de que existen semejanzas con el sistema a modelar porque tratan la gestión en departamentos docentes, la propuesta a desarrollar se enmarca en el funcionamiento interno de los departamentos en la facultad 8 donde existen actividades y procesos que son particulares del mismo.

Realizado el análisis de los sistemas automatizados se puede concluir que los mismos poseen un gran número de características que le permiten solucionar eficientemente un amplio espectro de problemas, sin embargo se detecta la no existencia de un sistema informático que agrupe la gestión de medios básicos y almacene la información de los profesores por departamento, a partir de lo cual se ratifica la necesidad del software a desarrollar. Al analizar estos sistemas se decide por parte del equipo de desarrollo tomar algunos aspectos positivos que ayuden a la confección de la propuesta de solución:

- Los procesos de registrar los medios incluyen todas las características de estos, que constituye una manera de normalizar los términos que serán utilizados y brindan la posibilidad de actualizar estos datos ya sea modificándolos o eliminándolos.

- Los procesos de reportes e inventarios permiten obtener información rápida y oportuna que facilita la toma de decisiones y el intercambio de información.

1.4. Tendencias y Tecnologías Actuales

1.4.1. Política de Migración a Software libre.

Se denomina software libre a todo aquel que permita a los usuarios ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. A menudo es confundido con el software gratuito, sin embargo no se trata de una cuestión de precio sino de libertad. Precisamente, las cuatro libertades que se definen son:

- La libertad de ejecutar el programa para cualquier propósito.
- La libertad de estudiar cómo trabaja el programa y adaptarlo a sus necesidades (El acceso al código fuente es una condición necesaria).
- La libertad de redistribuir copias para que pueda ayudar al vecino.
- La libertad de mejorar el programa y publicar sus mejoras y versiones modificadas en general para que se beneficie toda la comunidad (El acceso al código fuente es una condición necesaria). (1)

Las ventajas especialmente económicas que aportan las soluciones libres y las aportaciones de la comunidad de desarrollo han permitido un constante crecimiento del software libre hasta superar en ocasiones, como en el caso de los servidores web, al mercado propietario. Estas ventajas hacen que nuestro país siga una política de migración hacia el software libre y como parte de este proceso se decide para el desarrollo de la aplicación la utilización de herramientas y tecnologías pertenecientes al software libre.

1.4.2. ¿Por qué realizar una Aplicación Web?

Las aplicaciones Web forman un sistema íntegro de intercambio de datos, procesos e información entre el servidor (servidor Web) y los clientes (navegador). El usuario solicita una página Web existente en un servidor http mediante el navegador, el servidor envía entonces la información requerida al ordenador del usuario, el navegador interpreta los códigos HTML y presenta la información contenida en la página Web en la pantalla del ordenador del usuario. Las siguientes ventajas que ofrecen las aplicaciones Web avalan su utilización en la propuesta de solución:

- Los clientes y usuarios puedan tener acceso a ella en el momento que la requieren con el único requisito de tener instalado un navegador Web.
- Facilita la actualización y mantenimiento pues no es necesario distribuir e instalar software en las PC clientes.
- Posibilita que la información este centralizada.
- Evita dependencias de hardware, de software, así como del sistema operativo.

1.4.3. Lenguajes de Programación.

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. (2)

A continuación se realiza un estudio de los distintos tipos de lenguaje de programación Web atendiendo a que los mismos se dividen, a tono con la propia arquitectura cliente-servidor, en lenguajes del lado del cliente y lenguajes del lado del servidor.

1.4.4. Lenguajes del lado del servidor.

Los lenguajes del lado del servidor, son aquellos que son reconocidos, ejecutados e interpretados en el propio servidor Web, el cual envía una respuesta al cliente en un formato comprensible para él. (3)

Perl (Practical Extracting and Reporting Language)

Perl es gratuito, mucho más que eso, es software libre. Actualmente disponible para sistemas operativos de tipo Unix y otros como Windows. Constituye un lenguaje práctico para manipular información de archivos de texto y administración de sistemas, programación en red, y desarrollo de GUI (Interfaz Gráfica de Usuario). Es fácil de usar aunque es difícil de aprender. Además, es eficiente y completo, está enfocado hacia un desarrollador con conocimientos sobre el lenguaje y es ideal para desarrollo rápido de aplicaciones. Soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional. Una de las grandes ventajas del Perl es la amplia comunidad de usuarios donde se hace muy sencillo plantear y resolver las dudas que surjan en el desarrollo de una aplicación.

Python

Python es desarrollado como proyecto de software libre, manejado por la *Python Software Foundation*, es un lenguaje interpretado por lo que no es necesario realizar la compilación del código fuente para ejecutarlo, en esto radica su sencillez y velocidad. Es multiplataforma y está orientado a objetos. Python es considerado como un lenguaje exitoso debido a su facilidad de aprendizaje, su orientación a programadores promedio y la legibilidad de su código. Es además considerado como un lenguaje de muy alto nivel y de propósito general. La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero garantizando un mantenimiento mucho más sencillo que en otros lenguajes.

PHP

PHP es considerado como un puntal en el desarrollo de aplicaciones Web, es un lenguaje interpretado, sencillo, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Es además software libre, multiplataforma y rápido. Contiene varias bibliotecas para funciones matemáticas, de bases de datos e inclusive trabajo con redes. Es un lenguaje fácil de usar que cuenta, desde la versión 5.0, con un magnífico trabajo con el paradigma orientado a objetos. Se encuentra muy bien documentado, posee la capacidad de conexión y soporte con la mayoría de los gestores de bases de datos (MySQL, Oracle, PostgreSQL), así como con la mayoría de los servidores Web que se utilizan en la actualidad. Permite leer y manipular datos desde diversas fuentes como código XML (*Extensible Markup Language*), incluyendo datos que pueden ingresar los usuarios desde formularios HTML. Su diseño modular le permite expandir su potencial utilizando una enorme cantidad de librerías. Entre las desventajas adjudicadas a este lenguaje podemos encontrar:

- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- Cada desarrollador programa a su manera y por tanto es muy difícil mantener el código de terceras personas.

Sin embargo estas características negativas antes citadas pueden ser evitadas por el desarrollador, con la utilización de nuevas tecnologías como patrones y *frameworks*.

Fundamentación de la selección de PHP como lenguaje del lado del Servidor.

Cada lenguaje de programación tiene sus características propias que lo distingue, su selección varía teniendo en cuenta las características de la aplicación a desarrollar. Por tanto se selecciona PHP como lenguaje de programación del lado del servidor atendiendo a sus indudables ventajas sobre otros, entre las que se mencionan algunas:

- Actualmente es el lenguaje de software libre más utilizado en la Internet con una amplia documentación, posee una de las comunidades más grandes en Internet, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- Es multiplataforma y soporta POO (**P**rogramación **O**rientada a **O**bjetos).
- Posee mayor velocidad de ejecución con respecto a Perl y Python.
- Posee mayor velocidad de desarrollo que Perl y Python.
- Utiliza bajos recursos de hardware y software con respecto a Perl y Python.
- El equipo de desarrollo tiene un amplio dominio y experiencia en el desarrollo de aplicaciones Web utilizando PHP.

1.4.5. Lenguajes del lado del cliente.

Los lenguajes del lado del cliente son aquellos lenguajes que toman las acciones en el navegador del usuario, para el correcto desarrollo de la aplicación se destaca el uso de:

XHTML

XHTML (*EXtensible HyperText Markup Language*) es la nueva generación de HTML, que se suele definir como una versión XML validable de HTML o como algunos la llaman la versión limpia o estricta de HTML ya que cumplen las mismas funcionalidades, además de que todos los nombres de etiquetas y atributos van en minúsculas, todos los valores de los atributos entre comillas, todos los atributos deben tener un valor, todas las etiquetas deben estar cerradas, inclusive las etiquetas vacías, logrando que el código tenga mayor claridad.

Este lenguaje de marcado, simple, multiplataforma, está diseñado para definir la estructura y apariencia básica de textos y documentos con el fin de que éstos se visualicen correctamente al margen del contexto

y el navegador utilizado. Como ventajas que posee se puede destacar que es fácil de aprender, lo que permite que cualquier persona pueda utilizarlo, además se hace muy fácil actualizar los contenidos utilizando estilos en formato CSS (*Cascading Style Sheets*) en las páginas para una mayor facilidad en su modificación.

CSS

CSS (*Cascading Style Sheets*) en español Hojas de estilo en Cascada. Es una tecnología creada para controlar la forma de presentación de los documentos definidos con HTML y XHTML. Define el estilo visual de un sitio entero sin necesidad de hacerlo etiqueta por etiqueta, de manera que un cambio, por sencillo que este sea, a un estilo CSS, se reflejará automáticamente en todos los elementos que sean referidos a éste. De esta forma se reduce la complejidad de su mantenimiento, evita que los archivos sean demasiado pesados y permite visualizar el mismo documento en infinidad de dispositivos diferentes (ordenadores, PDAs¹, teléfonos móviles, entre otros).

JAVASCRIPT

Javascript es un lenguaje de programación interpretado por el navegador, de scripts, desarrollado para incrementar las funcionalidades del HTML y XHTML. Permite incluir macros en páginas Web, estas se ejecutan en el ordenador del visitante de las páginas y no en el servidor. Se utiliza para controlar las ventanas del navegador y el contenido que muestran, evitar depender del servidor Web para cálculos sencillos, capturar los eventos generados por el usuario y responder a ellos sin salir a Internet, y comprobar los datos que el usuario introduce en un formulario antes de ser enviados.

Fundamentación de los lenguajes del lado del cliente.

El lenguaje principal que se decide utilizar es XHTML, este se utilizará para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. Javascript será utilizado como apoyo a XHTML especialmente para la validación de datos. Las hojas de estilo en cascada (CSS) serán utilizadas para dar estilo a los documentos XHTML separando el estilo de la presentación, permitiendo controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Logrando con la integración de estos lenguajes, un trabajo sencillo, rápido y eficiente en el desarrollo de la propuesta de solución.

¹ PDAs: **P**ersonal **D**igital **A**ssistant (*Asistente Digital Personal*) es un computador de mano originalmente diseñado como agenda electrónica.

1.4.6. AJAX

Acrónimo de *Asynchronous JavaScript And XML* (Java Script y XML asíncronos). Ajax no es una tecnología sino que es la unión de varias tecnologías de la Web ya existentes que logran una sincronización sorprendente, estas son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM (*Document Object Model*), para la interacción y manipulación dinámica de la presentación.
- XML, XSLT² y JSON (*JavaScript Object Notation*/ Notación de Objetos de JavaScript), para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías. (4)

La unión de estas tecnologías logra crear aplicaciones interactivas que se ejecutan en el cliente (navegador), manteniendo una comunicación asíncrona con el servidor en segundo plano haciendo posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. Se han creado cientos de aplicaciones basadas en AJAX que en la mayoría de casos pueden sustituir completamente a otras técnicas como Flash y en el caso de las aplicaciones más avanzadas, pueden sustituir a complejas aplicaciones de escritorio.

1.5. Frameworks

Los *frameworks* son arquitecturas genéricas integradas por un extensible conjunto de componentes. En otras palabras un *framework* es un conjunto de librerías y componentes de probada capacidad, junto con una documentación y metodología de uso, que permite diseñar, construir e implantar aplicaciones de forma más uniforme, rápida, y con mayor calidad. Además simplifican el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes, proporcionan estructura al código fuente, forzando al programador a crear códigos más legibles y más fácil de mantener.

² **XSLT** o **Transformaciones XSL**: es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.

1.5.1 ¿Por qué usar un *Framework*?

Las ventajas y facilidades que nos ofrece un *framework* son numerosas:

- Reduce los tiempos de esfuerzo de desarrollo.
- Ofrece generadores de código y plantillas arquitectónicas desarrolladas con las últimas versiones de componentes y recogiendo todas las mejores prácticas de los mismos.
- Simplifica los procesos de mantenimiento a largo plazo.
- Simplifica la resolución de problemas en producción.
- La estandarización de un *framework* de desarrollo incrementa la calidad y reduce los costes en todo el ciclo de vida del software.
- Permite el diseño homogéneo de las soluciones.
- Propone una metodología de desarrollo.
- Proporciona los mecanismos de integración de aplicaciones.
- Agiliza la puesta en explotación y la operación de aplicaciones.

Como se ha podido apreciar son múltiples las ventajas que ofrece el desarrollo de aplicaciones mediante el uso de un *framework* por lo cual se incluye su utilización como parte de la propuesta de solución. A continuación se describirán dos de los *frameworks* para PHP más utilizados en la actualidad. Los mismos son *symfony* y *CakePHP*.

CakePHP

CakePHP es un *framework* de desarrollo de aplicaciones web escritas en PHP, fáciles de entender, orientado a objetos, ofrece una arquitectura extensible para el desarrollo, mantenimiento y despliegue de aplicaciones, facilita al usuario la interacción con la base de datos mediante el uso de *ActiveRecord*. Además hace uso del patrón **M**odelo **V**ista **C**ontrolador (MVC), disminuye los costos de desarrollo y ayuda a los desarrolladores escribir menos código.

Características de *CakePHP*:

- Compatible con PHP5.
- Formateo de URLs amigables.
- Implementa el patrón CRUD (crear, leer, actualizar y eliminar).
- Sistema de plantillas rápido y flexible.
- Validación integrada.
- Componentes para el manejo de la seguridad, sesiones y peticiones.
- *Helpers* para *AJAX*, *Javascript*, *HTML*, formularios y otros. (5)

Este *framework* tiene gran desventaja debido a que posee poca documentación.

symfony³

Es un *framework* completo diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web, separa la lógica de negocio, la lógica de Servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. *Symfony* le permite al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una aplicación (6). *Symfony* está desarrollado completamente con PHP 5, actualmente en su versión 1.2.5, bajo la licencia de software libre. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft.

³ Actualmente se encuentra en su versión 1.2.5. A partir de la versión 2.0 comenzara a llamarse *Symfony* con 'S' mayúscula.

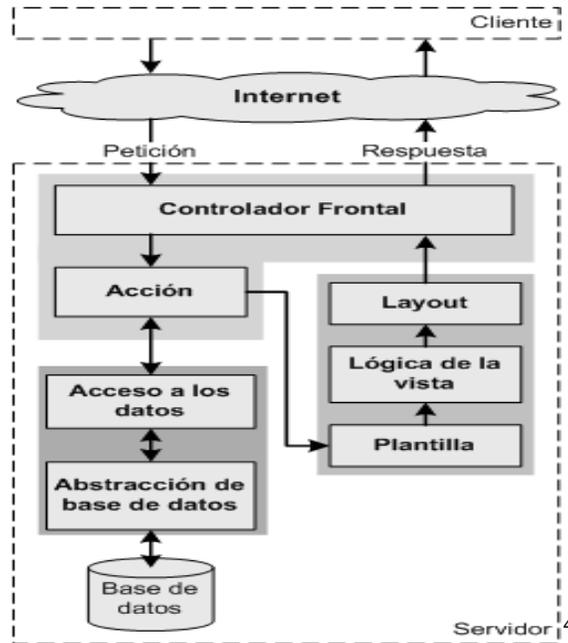


Ilustración 1: Flujo de Trabajo de symfony

Características de *symfony*:

- Implementa las reglas del negocio.
- Abstrae el acceso a datos.
- Es maduro, estable, profesional e infinitamente escalable.
- Simplifican el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes. Además proporciona estructura al código fuente, forzando al programador a crear códigos más legibles y más fácil de mantener.
- Seguro, incluye por defecto ataques contra XSS⁵, CSRF⁶.
- URLs amigables.

⁴ http://www.librosweb.es/symfony_1_0/capitulo2/el_patron_mvc.html

⁵ **XSS (Cross-site scripting)**: es un tipo de agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado.

⁶ **CSRF (Cross Site Request Forgery)**: es un tipo de agujero de seguridad basado en explotar la confianza que los sitios web tienen con sus usuarios.

- Implementa el patrón CRUD (crear, leer, actualizar y eliminar).
- *Helpers* para AJAX, Javascript, HTML, formularios y otros.
- Validación integrada.
- Componentes para el manejo de la seguridad, sesiones y peticiones.
- Toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- La estructura de archivos y directorios por defecto de *symfony*, permiten que cualquier programador con ciertos conocimientos de *symfony* sea capaz de continuar el desarrollo de cualquier proyecto *symfony* y en cuestión de minutos será capaz de profundizar en el código, solucionar errores y añadir nuevas características.
- Cuenta con una amplia documentación, tanto en español como en inglés, y una gran comunidad, de forma que cualquier problema que surja, o se encuentre es rápidamente corregido.

¿Por qué *symfony*?

En la búsqueda de un *framework* maduro, estable, bien documentado y con una gran comunidad que lo apoye, *symfony* sin duda alguna es una gran opción. Tanto *symfony* como *CakePHP* constituyen potentes *frameworks* que facilitan el desarrollo de aplicaciones Web. Aún cuando *CakePHP* es reconocido mundialmente por sus prestaciones, este cuenta con la gran desventaja de que la mayoría de la documentación sobre su uso se encuentra solo en idioma inglés. Por su parte *symfony* se hace muy fuerte en el aspecto antes mencionado contando a su favor con una gran documentación y una muy activa y creativa comunidad de usuarios que nos mantiene actualizados sobre los últimos cambios. Además en la UCI, se cuenta con facultades que poseen experiencia sobre el uso de dicho *framework* para el desarrollo de aplicaciones Web.

Se selecciona *symfony* como *framework* para PHP como parte de la propuesta de solución atendiendo además a la familiaridad del equipo de desarrollo en el trabajo con el mismo.

1.6. 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 grandes 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. El *framework symfony*, seleccionado anteriormente, implementa a su vez el patrón arquitectónico MVC, es por ello que adoptamos esta arquitectura para el desarrollo de la propuesta de solución.

Modelo Vista Controlador (MVC).

Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- **Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **Vista:** Presenta el modelo en un formato adecuado, como en una página Web que le permite al usuario interactuar con ella, usualmente un elemento de interfaz de usuario.
- **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

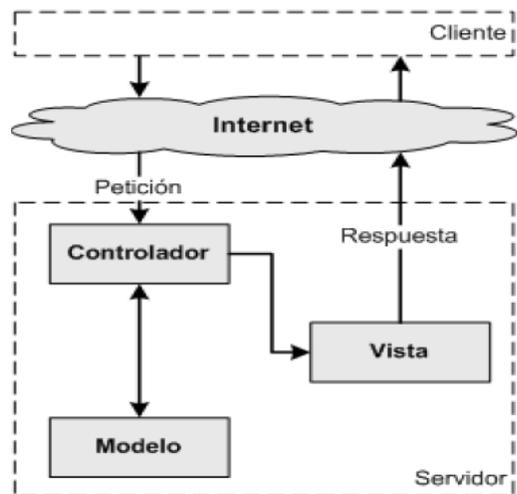


Ilustración 2: Patrón MVC.

⁷ http://www.librosweb.es/symfony_1_0/capitulo2/el_patron_mvc.html

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) logrando un mantenimiento más rápido y sencillo de las aplicaciones. Si se fuera a mostrar una misma aplicación en un navegador estándar, como en un navegador de un dispositivo móvil, sólo es necesario crear una vista nueva por cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (6)

Ventajas del Modelo Vista Controlador

- La separación del Modelo de la Vista, es decir, separa los datos de la representación visual de los mismos.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Permite el escalamiento de la aplicación en caso de ser requerido.

1.7. ¿Qué son los Servicios Web?

Los Servicios Web (*Web Service*) son un conjunto de aplicaciones o de tecnologías con capacidad para intercambiar datos entre sí, a través de protocolos y estándares (XML, SOAP, WSDL y UDDI) con el objetivo de ofrecer servicios en la Web. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios lo solicitan llamando a estos procedimientos a través de la Web. (7)

Un Servicio Web, es similar a un sitio Web pero no cuenta con un interfaz de usuario y da servicio a las aplicaciones y no a las personas. En lugar de obtener solicitudes desde el navegador y retornar páginas Web como respuesta, lo que hace es recibir solicitudes a través de un mensaje formateado en XML desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta también formateado en XML, permitiendo que las aplicaciones compartan información y que además, invoquen funciones de otras aplicaciones.

Ventajas de los servicios Web.

- La gran independencia entre la aplicación que usa el servicio Web y el propio servicio, de esta forma, los cambios a lo largo del tiempo en uno no afectan al otro.
- Aportan interoperabilidad y extensibilidad entre estas aplicaciones independientemente de sus propiedades o de las plataformas sobre las que se instalen y que al mismo tiempo sea posible su combinación para realizar operaciones complejas independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- El software como un Servicio, en lugar de un producto empaquetado. Los Servicios Web pueden ser brindados y pagados en línea y accedidos desde cualquier plataforma. Esto contribuye a sistemas más flexibles y estables.
- Mayor agilidad y flexibilidad debido a una mejor integración con los sistemas existentes.

1.7.1. Servidores Web

Un servidor web es un programa que se ejecuta continuamente en una computadora, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

Lighttpd

Lighttpd es un servidor Web seguro, rápido y muy flexible que ha sido optimizado para entornos de alto rendimiento. Posee una pequeña huella de memoria en comparación con otros servidores Web y logra una gestión eficaz de la carga de CPU (**Central Processing Unit /** Unidad Central de Procesamiento). Su avanzado conjunto de características (FastCGI, CGI, Auth, de salida-de compresión, reescritura de URL y otros) hacen la perfecta *Lighttpd web*-software para cada servidor en el que sufre los problemas de carga. Además, es *Open Source* bajo licencia BSD⁸.

⁸ La **licencia BSD** es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre.

Zope

Zope son las iniciales de "**Z Object Publishing Environment**", un entorno para publicar objetos. Es un servidor de aplicaciones Web de código escrito principalmente en el lenguaje de programación Python. La Licencia Pública de Zope ha sido certificada como código abierto además de compatible con la **General Public License** (GPL) de la **Free Software Foundation** (FSF). Todos los elementos de Zope son objetos, y estos se almacenan en una base de datos orientada a objetos ZOB. La potencia que proporciona este almacenamiento persistente de los objetos es enorme, sobre todo a la hora de modificar dinámicamente y buscar datos dentro de todos los objetos que forma el sitio web. Zope dispone de un producto conocido como *ZCatalog* que permite realizar búsquedas en todos los objetos del servidor.

Servidor Web Apache

Apache ha sido el servidor web más popular en internet, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Apache tiene muchas ventajas ejemplo de ello es que corre en una multitud de sistemas operativos lo que lo hace prácticamente universal, es una tecnología gratuita de código abierto, muy potente y altamente configurable por su diseño modular de forma que para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo y no es necesario volver a instalar el software. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache.
- **Módulos Multiproceso:** Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor. (8)

Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor y tiene una alta configurabilidad en la creación y gestión de logs.

¿Por qué Apache?

Después de ser analizadas las características fundamentales, y apreciar las diferencias que existen entre los servidores Web se decide la instalación de Apache, ya que se considera un servidor estable, potente, de probada solvencia y muy utilizado en la actualidad.

1.8. Sistemas Gestores De Base De Datos (SGBD).

Un SGBD es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Permite definir, construir y manipular una base de datos (BD), asegurando su integridad, confidencialidad y seguridad.

Algunas de las características deseables en un Sistema Gestor de base de datos son:

- **Control de la redundancia:** La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- **Restricción de los accesos no autorizados:** cada usuario ha de tener permisos de acceso y autorización.
- **Cumplimiento de las restricciones de integridad:** el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad. (9)

MySQL

MySQL es un gestor de base de datos desarrollado inicialmente por MySQL AB⁹ y actualmente pertenece a Sun Microsystems¹⁰, sencillo de usar e increíblemente rápido. Es uno de los motores de base de datos más usados en Internet. Es *Open Source* y usa la licencia GPL para aplicaciones no comerciales. Es una base de datos relacional, rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Presenta una alta escalabilidad ya que es posible manipular bases de datos enormes, está escrito en C y C++, probado con numerosos compiladores y dispone de muchas APIs (*Application Programming Interface*) para plataformas diferentes.

⁹ **MySQL AB** de las más grandes compañías de software libre del mundo.

¹⁰ **Sun Microsystems** es una empresa informática de Silicon Valley, fabricante de semiconductores y software.

Es multihilo y multiusuario, además permite la conexión entre diferentes máquinas con distintos sistemas operativos.

PostgreSQL

PostgreSQL es el servidor de bases de datos de código abierto, liberado bajo la licencia BSD más potente que existe. Soporta características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores (triggers) y procedimientos almacenados, lo que significa que podría implementar la lógica del negocio en el servidor, tiene lenguaje procedimental (PL), de alta disponibilidad y recuperación de fallas. Además, soporta gran cantidad de información y se puede utilizar sobre cualquier sistema operativo.

Ventajas:

- Mejor soporte que los proveedores comerciales.
- Ahorros considerables en costos de operación.
- Estabilidad y confiabilidad legendarias por la seguridad que aporta.
- Mayores prestaciones en caso de necesitar escalar la base de datos.
- Diseñado para ambientes de alto volumen de datos.
- También soporta el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o vídeo.
- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs (*Central Processing Unit*) y la cantidad de RAM (*Random Access Memory*) logrando un rendimiento óptimo.
- Dispone de gran cantidad de documentación en español.

Fundamentación de la selección del SGBD.

MySQL y PostgreSQL son 2 grandes rivales, es muy difícil decir a ciencia cierta cuál es el mejor. Tomar la decisión, de seleccionar PostgreSQL como la mejor alternativa, tuvo su mayor peso cuando a lo largo de la investigación salió a la luz que, Sun Microsystems, la compañía que adquirió MySQL, siguiendo las políticas del gobierno de los E.U.A (Estados Unidos de América) de continuar con el bloqueo económico y político a nuestro país, ha negado las actualizaciones de este potente gestor de base de datos al dominio

“cu”. Además, no se puede dejar de resaltar que en cuestiones de seguridad y rapidez en las consultas cuando la base de datos contiene un gran volumen de datos, PostgreSQL supera a MySQL.

1.9. Metodologías de Desarrollo De Software.

1.9.1. Proceso de desarrollo de software.

La calidad en el desarrollo y mantenimiento del software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales dependen de los sistemas informáticos para su buen funcionamiento. En los últimos años se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para la mejora de los procesos de software.

El **proceso de desarrollo de software** "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". (10)

Las piedras angulares del proceso de desarrollo del software son: el proyecto, las personas y el producto; siendo las características del cliente, el entorno de desarrollo y las condiciones del negocio, elementos que influyen en el proceso. Existe una estrecha relación entre personas, proyecto, producto y proceso. Estos términos son conocidos como las cuatro «P» en el desarrollo de software. El resultado final de un proyecto software es un producto, donde intervienen personas a través de un proceso de desarrollo de software que guía los esfuerzos de las personas implicadas en el proyecto.

1.9.2. Metodologías

Hoy en día, llevar a cabo el desarrollo de un buen software depende de un gran número de actividades y etapas donde elegir la mejor metodología para el equipo influye directamente en el futuro éxito del producto. El papel preponderante de las metodologías es sin duda esencial en un proyecto y en el paso inicial, que debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo.

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.

Teniendo en cuenta ambos enfoques damos paso al análisis de dos de las metodologías más usadas actualmente.

El Proceso Unificado de Desarrollo (RUP).

Es un proceso para el desarrollo de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como tres características esenciales está **dirigido por casos de uso**: que orientan al proyecto a la importancia para el usuario y lo que se quiere, está **centrado en la arquitectura**: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es **iterativo e incremental**: donde divide el proyecto en mini-proyectos donde los casos de uso y al arquitectura cumplen sus objetivos de manera depurada. RUP propone cuatro etapas para el desarrollo de un producto: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería pero con diferente énfasis; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros. El Proceso Unificado de Desarrollo tiene 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Flujos de trabajo:

- **Modelamiento del negocio**: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos**: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce *release* del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Es una de las metodologías más generales y más usadas de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto. Constituye además, una propuesta de proceso para el desarrollo de software orientado a objeto, utilizando UML (*Unified Model Language*), para describir todo el proceso basándose en componentes. Este lenguaje es estándar, con él se puede visualizar, especificar, construir y documentar los artefactos de un sistema.

XP (*Extreme Programming*).

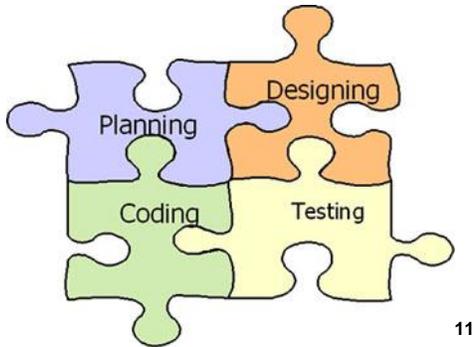


Ilustración 3: Metodología XP

Es la más destacada de los procesos ágiles de desarrollo de software formulada por Kent Beck. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los programadores que la practican consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. Es utilizada para proyectos de corto plazo, equipo pequeño y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Las características fundamentales son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera -

¹¹ http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html

el código es revisado y discutido mientras se escribe- es más importante que la posible pérdida de productividad inmediata.

- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

Ventajas

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

- Permite la retroalimentación.
- La presión esta a lo largo de todo el proyecto y no en una entrega final.

Desventajas

- Delimitar el alcance del proyecto con nuestro cliente.

Para mitigar esta desventaja se plantea definir un alcance a alto nivel basado en la experiencia.

Fundamentación de la selección de la metodología de desarrollo de software.

XP y RUP son dos grandes metodologías que después de analizar sus principales características y los aspectos preponderantes de cada una de ellas, se ha determinado la implantación de XP, una metodología ligera, con menos requerimientos de documentación y planificación para el desarrollo de la aplicación. Si se escoge RUP traería grandes dificultades como son:

Multitud de artefactos: El hecho de realizar varios artefacto y mantenerlos actualizados consume mucho tiempo.

El poco personal de desarrollo: Al ser solo dos personas a cargo del desarrollo de la aplicación, estas tomarían varios roles en cada etapa y sería muy complejo cumplir con las actividades de cada uno de ellos.

Requisitos cambiantes: Los cambios en un proceso de desarrollo son inevitables, al aparecer un nuevo requisito hace que se tenga que comenzar una nueva iteración para dar cumplimiento a su funcionalidad.

Planificación inexistente: La planificación que se realiza en las fases iniciales está sujeta a muchas variaciones en dependencia de los cambios que se experimenten en los requisitos. Por tanto se hace muy difícil planificar actividades específicas.

Los inconvenientes planteados pueden ser eliminados con la utilización de la metodología XP.

1.10. Herramientas de Desarrollo.

Aptana Studio.

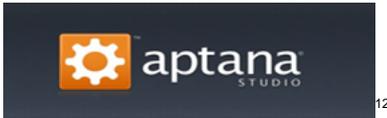


Ilustración 4: Aptana Studio

Aptana Studio es un IDE (*Integrated Development Environment*) basado en el conocido Eclipse, multiplataforma, se centra en el desarrollo de aplicaciones Web, con soporte a HTML, CSS, Javascript, Ajax, PHP, entre otros. Con esta herramienta se facilita el desarrollo integrado de Ajax con las tecnologías emergentes. Cuenta con una Edición *Open Source* para la comunidad y otra Profesional de pago. Permite, además, comprobar la compatibilidad de las funciones con los diferentes navegadores, la sincronización con carpetas locales y remotas, e incorporarle funciones mediante *plugins*.

EasyEclipse.



Ilustración 5: EasyEclipse

EasyEclipse es *Open Source*, multiplataforma, no es más complejo que lo necesario y se hace muy fácil de descargar e instalar. Dispone de varias distribuciones en las que se empaqueta el entorno de desarrollo Eclipse junto con una cuidada selección de "*plugins*" *Open Source* para obtener un IDE final excepcionalmente bueno para el desarrollo de aplicaciones en PHP, Python, Ruby y Java, con todos los *plugins* ya instalados y configurados de manera que funcionen correctamente en dependencia del entorno, para que el desarrollador final sólo tenga que preocuparse del código de su aplicación y no de afinar su IDE.

¹² http://afalcon.files.wordpress.com/2007/11/aptana_splash.jpg

¹³ http://farm4.static.flickr.com/3122/2666290071_699620ca01.jpg

Zend Studio.

Es un IDE destinado a desarrolladores profesionales. Es un IDE propietario y multiplataforma. Incluye todos los componentes necesarios durante el ciclo de vida de una aplicación en PHP. Incluye editor, análisis, depuración, optimizadores de código y herramientas de base de datos. *Zend Studio* nos permite agiliza el desarrollo web y permite simplificar proyectos complejos. Posee excelente completamiento de código, coloreado en la sintaxis del código, administración avanzada de proyectos, múltiples lenguajes, incorpora el *Framework* de Zend, *PHP Documentor*, manual de PHP. Integración con subversión, los navegadores, integración avanzada con FTP. Soporte para *Web Services*, PHP4, PHP5 y SQL.

Fundamentación de las herramientas de desarrollo escogida.

Teniendo en cuenta los IDEs antes caracterizados se sugiere la utilización de *Aptana Studio* para el trabajo con los lenguajes XHTML, Javascript y CSS y con la tecnología Ajax por el gran soporte e integración que tiene con este. Se hará uso de *EasyEclipse* para el trabajo con PHP puesto que ofrece tantas funcionalidades como *Zend Studio*, pero a su vez, *EasyEclipse* es software libre y es excepcionalmente bueno. Mientras que *Zend Studio* se descarta ya que requiere licencia de pago y no incluye editor visual HTML.

1.11. Conclusiones

Existen múltiples lenguajes, metodologías y tecnologías para el desarrollo de aplicaciones. Después de realizar un estudio de las más utilizadas se escogieron las que presentan mayores ventajas. Entre ellas se seleccionan los lenguajes de programación PHP, XHTML, JavaScript apoyándolos con CSS y Ajax, como gestor de base de datos PostgreSQL y servidor web Apache. Se elige el *framework symfony* para agilizar el proceso de desarrollo y los IDE's *Aptana* y *EasyEclipse*. Se decide realizar la aplicación sobre la base de la metodología ágil XP.

CARACTERÍSTICAS DEL SISTEMA

2.1. Introducción

No existe una metodología universal para hacer frente, con éxito, a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.). La metodología seleccionada para desarrollar el software fue XP motivo por el cual cada uno de los artefactos generados se encuentran condicionados al uso de la misma.

En el presente capítulo se describe el proceso actual que se lleva a cabo en los departamentos docentes y se abordan objetivos estratégicos que actualmente se necesitan y se requieren en los mismos. Se presentan de la primera fase de la metodología las historias de usuarios realizadas por el cliente, así como toda la planificación de entrega para la implementación de las mismas.

2.2. Descripción de los procesos vinculados al campo de acción

2.2.1. Flujo actual del proceso.

En los departamentos docentes se llevan a cabo varios procesos, donde la mayoría son controlados y supervisados por los jefes de los mismos.

Uno de los procesos más importantes es que todas las actividades o eventos realizados por los profesores son informados al jefe de departamento, quien se encarga de archivar en los expedientes o guardar en documentos Word y Excel, la constancia de su participación.

El plan de trabajo individual del profesor es otro proceso de gran valor ya que este tiene una gran influencia en la evaluación del mismo. Al ser creado el plan por el profesor, este lo hace llegar al jefe de departamento quien lo revisa y modifica en dependencia de la categoría docente o científica que tenga el profesor y posteriormente lo aprueban o no entre ambos. A continuación el jefe de departamento tiene que velar por el cumplimiento de las actividades de este plan y como se mencionó anteriormente en documentos Word y Excel son guardados estos datos.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

El profesor es evaluado por el jefe de departamento según su desempeño trimestralmente en Deficiente, Adecuado y Superior. En esta evaluación influyen las actividades realizadas por los profesores a lo largo del trimestre, por lo que tiene que consultar toda la información de cada uno para poder emitir el criterio de evaluación. Además se les evalúa anualmente en bien (B), regular (R) o mal (M), pero esta evaluación es para el expediente del profesor e influyen las evaluaciones trimestrales correspondientes y el cumplimiento del plan de trabajo individual.

Otro proceso que se lleva a cabo es el de llevar el control de la información de los medios básicos que pertenecen al departamento. En este caso el jefe de departamento tiene toda la responsabilidad en el cuidado y conservación de los mismos pero no lleva un registro actualizado y estricto de los datos de cada uno, solo realiza un reporte cuando es necesario para el inventario que se realiza a nivel de facultad.

2.2.2. Objeto de automatización.

Los jefes de departamentos en la facultad 8 tienen que manejar información muy importante y además en grandes cantidades. Gestionar todos estos datos para obtener algún resultado se hace difícil y tedioso ya que lo tienen que hacer manualmente aumentando así la posibilidad de cometer errores, de tener información repetida y también se desperdicia tiempo por el hecho de no tener la misma centralizada.

La realización de un sitio web para los departamentos docentes de la facultad 8, que recoja la mayor parte de la información que se maneja en los mismos y que se automaticen los procesos descritos anteriormente, es la propuesta que tiene este trabajo. En el mismo se almacenaría toda la información de cada uno de los profesores, brindando la posibilidad de crear el plan de trabajo y modificarlo, registrar las actividades realizadas por los mismos así como la constancia de su realización. También facilitaría la generación de reportes para apoyar en el proceso de asignación de las evaluaciones (mensual, trimestral, anual) y conjuntamente con todo esto registrar toda la información de los medios básicos del departamento detalladamente para así tener un control más eficaz y detallado de los mismos.

2.2.3. Personas relacionadas con el sistema

Se define como persona relacionada al sistema toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan en el mismo. También son las que se encuentran involucradas en dichos procesos, pues participan en ellos, pero no obtienen ningún resultado de valor.

Personas relacionadas con el sistema	Justificación
Administrador	Es la persona facultada para la gestión del sistema. Se encarga de crear y administrar las diferentes cuentas de usuarios que interactúan en la aplicación.
Jefe de Departamento	Es la persona que tiene la autoridad necesaria para controlar la información en el sistema sobre los usuarios (profesores) y medios básicos que se encuentren bajo su dirección así como de crear y administrar las diferentes cuentas de usuarios de los profesores de su departamento.
Profesor	Es la persona que tiene permisos para guardar y actualizar sus datos, la información referente a su plan de trabajo y a las actividades que realiza.

Tabla 1: Personas relacionadas con el sistema.

2.3. Planificación

Planificación es la primera fase de la metodología XP, en la misma quedan definidos los procesos que el cliente quiere automatizar y el tiempo en que quedará concluida la aplicación.

2.4. Exploración

En esta etapa el cliente crea las historias de usuarios que son de interés para la entrega del producto. Además el equipo de desarrollo se familiariza con las herramientas y tecnologías que se utilizarán en el proyecto.

Las historias de usuarios se utilizan para especificar las funcionales del software. Cada una de estas es suficientemente sencilla y comprensible para que el programador pueda implementarla en un corto período de trabajo. Se caracterizan por prioridad, riesgo y esfuerzo, además son la base de las pruebas

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

de aceptación, pues mediante estas se verifica si se cumplen las expectativas del cliente. Para definir las historias de usuario se emplea la siguiente plantilla.

Historia de Usuario	
Número: (Número de la historia de usuario incremental en el tiempo)	Nombre Historia de Usuario: (El nombre de la historia de usuario para identificarlas mejor entre los desarrolladores y el cliente)
Usuario: (sistema que protagoniza la historia)	
Prioridad en Negocio: (La importancia que tiene para el cliente) (Alta / Media / Baja)	Riesgo en Desarrollo: (Qué tan difícil es para el desarrollador) (Alto / Medio / Bajo)
Iteración Asignada: (La iteración a la que corresponde)	
Descripción: (Se especifican las operaciones del usuario y las respuestas que dará el sistema)	
Observaciones: (Algunas observaciones de interés, como glosario, información sobre usuarios, etc.)	

Tabla 2: Representación de una historia de usuario.

A continuación se presentan las historias de usuarios que fueron creadas por el cliente, teniendo estas los procesos fundamentales que se desean automatizar en los departamentos docentes de la facultad 8.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Gestionar medios básicos.
Usuario: Jefe de departamento docente.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración Asignada: 1	
Descripción: El Jefe de departamento debe tener la posibilidad de gestionar la información de los medios básicos que pertenecen al departamento. Debe almacenar todos los datos de los mismos. (PC, impresoras, escáneres, sillas, mesas), el rótulo, clasificación (dependiendo del medio), estado y observación.	
Observaciones: Gestionar incluye adicionar, modificar, eliminar y listar. Cada jefe de departamento gestiona solo los medios básicos que pertenecen a su local.	

Tabla 3: HU Gestionar medios básicos.

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Gestionar datos personales.
Usuario: Profesor, Jefe de departamento docente.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración Asignada: 1	
Descripción: El profesor o el Jefe de departamento docente deben tener la posibilidad de llenar sus datos personales y/o particulares así como modificarlos cuando sea necesario.	
Observaciones: Gestionar incluye adicionar, modificar y eliminar sus datos personales, como son: dirección particular, categoría docente, categoría científica, grupo que atiende, proyecto al que se encuentra vinculado, y en caso que sea un profesor interno además la dirección de la UCI y teléfono (s) donde localizarlo.	

Tabla 4: HU Gestionar datos personales.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Gestionar actividades realizadas.
Usuario: Profesor	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Iteración Asignada: 2	
Descripción: Brindar la posibilidad de que los profesores puedan gestionar las actividades que vayan realizando.	
Observaciones: Gestionar incluye adicionar, modificar y eliminar.	

Tabla 5: HU Gestionar actividades realizadas.

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Crear plan de trabajo.
Usuario: Profesor, Jefe de Departamento docente.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Iteración Asignada: 1	
Descripción: Brindar la posibilidad de que cada profesor y jefe de departamento docente puedan crear su plan de trabajo individual y guardar los cambios.	
Observaciones: El plan de trabajo a que se refiere es el anual, cada profesor y jefe de departamento tendrán uno solo por curso escolar (ej. 2008 - 2009).	

Tabla 6: HU Crear plan de trabajo.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Gestionar actividades del profesor.
Usuario: Jefe de departamento.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Iteración Asignada: 2	
Descripción: Debe tener el control total sobre las actividades de cada profesor de su departamento, para así poder verificar el cumplimiento de las mismas y modificarla o eliminarla en caso necesario.	
Observaciones: Las actividades son de tipo docente, extensión, productivas, científicas, formación y eventos.	

Tabla 7: HU Gestionar actividades del profesor.

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Generar reportes
Usuario: Jefe de departamento docente, Profesor.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Iteración Asignada: 3	
Descripción: El jefe de departamento docente debe tener la posibilidad de generar reportes concernientes a los datos, evaluaciones y actividades realizadas por los profesores además de los medios que están bajo su control y el profesor obtener el reporte de sus actividades cumplidas, así como de sus evaluaciones.	
Observaciones: Realizar los reportes por tipo de actividad, por profesor, por rango de fecha, por evaluación, por medios básicos.	

Tabla 8: HU Generar reportes.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Asignar evaluación
Usuario: Jefe de departamento	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración Asignada: 3	
Descripción: Debe tener la posibilidad de evaluar al profesor mensual y trimestral según su desempeño (Deficiente, Adecuado y Superior), y anualmente (Bien Regular o Mal).	
Observaciones: Para evaluar al profesor, el jefe de departamento debe consultar el plan de trabajo y los reportes de las actividades realizadas, así como las evaluaciones anteriores para la evaluación trimestral y anual.	

Tabla 9: HU Asignar evaluación.

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Reportes del módulo Superación.
Usuario: Jefe de departamento.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Iteración Asignada: 3	
Descripción: Obtener Información o reporte de cursos dados o impartidos.	
Observaciones: Esta funcionalidad está en dependencia del desarrollo de otra tesis, la que brindará servicio web a la presente.	

Tabla 10: HU Reportes del módulo Superación.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Historia de Usuario	
Número: 9	Nombre Historia de Usuario: Gestionar Plan de Trabajo Individual del profesor.
Usuario: Jefe de Departamento	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Iteración Asignada: 1	
Descripción: Debe brindar la posibilidad de adicionar o modificar las actividades en el plan de trabajo del profesor.	
Observaciones: El jefe de departamento modificará el plan de trabajo de cada profesor, cuando este lo considere necesario.	

Tabla 11: HU Gestionar Plan de Trabajo Individual del profesor.

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Gestionar eventos realizados.
Usuario: Profesor	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración Asignada: 2	
Descripción: Debe brindar la posibilidad de que el usuario adicione, modifique o elimine los eventos en los que ha participado y de adjuntar, en caso de premio, el documento que lo acredite o el documento de participación, así como su resultado, y el título del trabajo presentado.	
Observaciones:	

Tabla 12: HU Gestionar eventos realizados.

Historia de Usuario	
Número: 11	Nombre Historia de Usuario: Gestionar usuarios.
Usuario: Administrador, Jefe de departamento docente.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Iteración Asignada: 1	
Descripción: El Administrador debe tener la posibilidad de gestionar las cuentas que sean necesarias para acceder al sistema y el Jefe de departamento debe gestionar las cuentas de usuarios de los profesores de su departamento.	
Observaciones: Gestionar incluye crear, modificar y eliminar. El Administrador puede crear las cuentas de usuarios de los jefes de departamentos y profesores y los jefes de departamento solo pueden crear las cuentas de sus profesores.	

Tabla 13: HU Gestionar usuarios

2.5. Planificación de la Entrega

El cliente establece la prioridad de cada historia de usuario, y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Para la estimación se utiliza como medida el punto, el mismo equivale a una semana (6 días) ideal de programación. Generalmente las historias valen de 1 a 3 puntos.

2.5.1. Estimación de esfuerzos por historias de usuario

Historia de Usuario	Número	Puntos de Estimación
Gestionar medios básicos.	1	1/2
Gestionar datos personales.	2	1/2
Gestionar actividades realizadas.	3	1
Crear plan de trabajo.	4	1
Gestionar actividades del profesor.	5	1
Generar reportes	6	1/2
Asignar evaluación	7	1/2
Reportes del módulo Superación.	8	1
Gestionar Plan de Trabajo Individual del profesor.	9	1/2
Gestionar eventos realizados.	10	1
Gestionar usuarios	11	1/2

Tabla 14: Estimación de esfuerzos por historias de usuario.

El plan de entregas se realiza teniendo en cuenta las unidades funcionales que se quieren entregar y cada uno de estos módulos abarca un número de historias de usuarios a implementar para dar cumplimiento al funcionamiento del mismo.

2.5.2. Plan de entregas

Módulos	Historias de usuario que abarca
Medios Básicos	1
Profesor	2, 11
Plan de trabajo	4, 9
Actividades y eventos	3, 5, 10
Evaluaciones	7
Reportes	6, 8

Tabla 15: Plan de entregas.

2.6. Plan de iteraciones

Para el desarrollo del sistema en cuestión se han planificado 3 iteraciones teniendo en cuenta las características de las historias de usuario necesarias para su implementación, la distribución de las historias de usuario por cada iteración queda de la siguiente manera:

Iteración 1:

En esta iteración se les dará solución a las historias de usuarios correspondientes a los módulos **Medios Básicos, Profesor y Plan de trabajo**. El plan de trabajo es uno de los procesos de alta prioridad para el cliente. Se considera que algunas de las historias correspondientes a esta iteración no necesitan una semana completa de implementación, pues las mismas son sencillas y así se puede dedicar el tiempo restante a las demás que tienen un grado mayor de complejidad.

Iteración 2:

A esta iteración le corresponden las historias de mediana complejidad y que además tienen implícitas una parte de los procesos fundamentales que se realizan en los departamentos docentes. Serían las correspondientes al módulo **Actividades y eventos**.

Iteración 3:

Para esta iteración se implementarán las historias de usuarios que no se pueden realizar sin antes haber implementado las demás ya que se corresponden generalmente con la generación de reportes. Son las correspondientes a los módulos **Evaluaciones y Reportes**.

2.6.3. Plan de duración de las iteraciones

Iteraciones	Orden de las historias usuario a implementar	Duración total de las iteraciones en semanas
1	<ul style="list-style-type: none"> • Gestionar usuarios • Gestionar datos personales • Gestionar medios básicos • Crear plan de trabajo • Gestionar Plan de Trabajo Individual del profesor 	3
2	<ul style="list-style-type: none"> • Gestionar actividades realizadas • Gestionar eventos realizados • Gestionar actividades del profesor 	3
3	<ul style="list-style-type: none"> • Reportes del módulo Superación • Asignar evaluación • Generar reportes 	2

Tabla 16: Plan de duración de las iteraciones.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Como producto del plan de entregas y el plan de iteraciones se harán *releases* al sistema en las fechas que se muestran a continuación.

Módulos	Final 1ra iteración	Final 2da iteración	Final 3ra iteración
Medios Básicos	15-04-2009		
Profesor	15-04-2009		
Plan de trabajo	15-04-2009		
Actividades y eventos		6-05-2009	
Evaluaciones			18-05-2009
Reportes			18-05-2009

Tabla 17: Cronograma de entrega.

2.7. Conclusiones

La descripción de los procesos que actualmente ocurren en los departamentos docentes de la facultad 8 junto con las historias de usuarios creadas por el cliente es el objeto de automatización del presente trabajo. A partir de estas historias se realiza toda la planificación de entrega del software, la misma se realiza conjuntamente con el cliente que es el actor principal en la etapa de planificación de la metodología seleccionada. Se decide realizar el software en tres iteraciones de 2 y 3 semanas de trabajo cada una para garantizar la entrega al cliente en el tiempo establecido.

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3.1. Introducción

En el presente capítulo se abordan las fases diseño e implementación de la metodología XP. Uno de los artefactos fundamentales es la creación de las tarjetas CRC (Clase-Responsabilidades-Colaboración) las cuales permiten brindar un mayor enfoque orientado a objetos. Por otra parte se describen cada una de las tareas planificadas para llevar a cabo el desarrollo de cada una de las historias de usuario detectadas. Además se realiza el diseño de la base de datos, artefacto de suma importancia producto de que dará soporte a la futura aplicación.

3.2. Diseño

Symfony aplica para su diseño el conocido modelo-vista-controlador e incluso va un poco más allá al dividir cada una de estas tres capas en otras subcapas que garantizan un trabajo más efectivo al equipo de desarrollo. La capa de la vista puede dividirse en dos atendiendo a que la mayoría de las aplicaciones presentan un *layout* que es común para casi todas sus páginas mientras que solo cambian el interior de las mismas. Por otro lado el controlador recibe casi siempre un trabajo excesivo por lo cual es beneficioso crear un controlador frontal que es único para cada aplicación y las acciones que incluyen el código específico del controlador de cada página. En lo que respecta a la capa del modelo la misma se divide en la capa de acceso a los datos y en la de abstracción a la base de datos con lo cual se logra total independencia del gestor de base de dato. Dicha interfaz es conocida como ORM (***Object-Relational Mapping***). Mientras que PHP 5 y *symfony* son orientados a objetos las bases de datos son relacionales por lo cual se hace necesario el trabajo con una interfaz que traduzca la lógica relacional a la lógica de objetos. El *framework symfony* utiliza por defecto como ORM el desarrollado por el proyecto *propel*¹⁴, garantizando de esta manera que el acceso a la base de datos sea a través de objetos.

¹⁴ <http://propel.phpdb.org/>

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Relacional	Orientado a objetos
Tabla	Clase
Fila, registro	Objeto
Campo, columna	Propiedad

3.3. Tarjetas CRC

XP propone realizar diseños simples y sencillos, hacerlo todo lo menos complejo posible para lograr que sea entendible e implementable. Realizar una correcta especificación de los nombres de métodos y clases ayuda a comprender mejor lo diseñado y facilita las posteriores ampliaciones y la reutilización del código. Nunca se debe añadir funcionalidades extras al software aunque se piense que serán factibles en el futuro.

El uso de la técnica de las tarjetas CRC para diseñar con la máxima simplicidad posible permite al programador centrarse en el desarrollo orientado a objetos. Estas tarjetas representan objetos, la clase a la que pertenece el objeto se escribe en la parte superior de la tarjeta, en una columna a la izquierda se escriben las responsabilidades u objetivos que debe cumplir el objeto y a la derecha las clases que colaboran con cada responsabilidad (11) como se muestra en la siguiente plantilla:

Clase	
Responsabilidades	Colaboraciones

Tabla 18: Plantilla Tarjeta CRC.

Esta nueva técnica de diseño es adoptada como alternativa a los diagramas UML de las clases, pues en estas se plasman las responsabilidades que tienen cada objeto y las clases con las que tienen que interactuar para darles respuesta brindando así la información que se necesita a la hora de implementar.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Departamento	
Responsabilidades	Colaboraciones
Crear departamento.	Profesor Medios Básicos
Modificar departamento.	
Eliminar departamento.	

Tabla 19: Tarjeta CRC de Departamento.

Profesor	
Responsabilidades	Colaboraciones
Crear cuenta.	Usuario Departamento
Eliminar cuenta.	
Modificar cuenta.	
Listar cuentas de usuarios por departamento.	
Ver datos.	
Modificar datos.	
Generar Reporte.	

Tabla 20: Tarjeta CRC de Cuenta del Profesor.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Usuario	
Responsabilidades	Colaboraciones
Autenticarse.	Profesor.
Cerrar sesión.	

Tabla 21: Tarjeta CRC de Usuario.

Medios Básicos	
Responsabilidades	Colaboraciones
Crear medios básicos.	Profesor Usuario Departamento
Eliminar medios básicos.	
Modificar medios básicos.	
Buscar medios básicos.	
Mostrar datos de un medio básico.	
Listar medios básicos por departamento.	
Generar Reporte.	

Tabla 22: Tarjeta CRC de Medios Básicos.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Actividades del profesor	
Responsabilidades	Colaboraciones
Crear actividad.	Profesor Usuario Departamento
Modificar actividad.	
Eliminar actividad.	
Listar actividades realizadas por un profesor.	
Chequear actividades realizadas.	
Listar actividades a chequear.	
Listar actividades realizadas en el departamento.	
Generar reporte.	

Tabla 23: Tarjeta CRC de Actividades del profesor.

Plan de trabajo	
Responsabilidades	Colaboraciones
Crear plan de trabajo.	Profesor Usuario Departamento
Modificar plan de trabajo.	
Listar planes de trabajo por departamento.	
Imprimir plan de trabajo.	

Tabla 24: Tarjeta CRC de Plan de trabajo.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Evaluaciones	
Responsabilidades	Colaboraciones
Asignar evaluación a los profesores.	Profesor Usuario Departamento
Generar Reporte	

Tabla 25: Tarjeta CRC de Evaluaciones.

Eventos	
Responsabilidades	Colaboraciones
Crear Evento.	Profesor Usuario Departamento
Modificar Evento.	
Eliminar Evento.	
Listar Eventos realizadas por un profesor.	
Chequear eventos realizados	
Listar eventos a chequear.	
Listar eventos realizados en el departamento.	
Generar reporte.	

Tabla 26: Tarjeta CRC de Eventos.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Reportes	
Responsabilidades	Colaboraciones
Generar reportes.	Profesor Usuario Departamento Eventos Actividades del profesor Plan de trabajo Medios básicos.

Tabla 27: Tarjeta CRC de Reportes.

3.4. Diseño de la Base de Datos

El diseño de la base datos es algo esencial para el buen funcionamiento de cualquier software. Ayuda a comprender mejor la estructura del sistema y sus funcionalidades. Para apoyar la técnica de tarjetas CRC a continuación se presenta el diagrama del modelo de datos utilizado.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

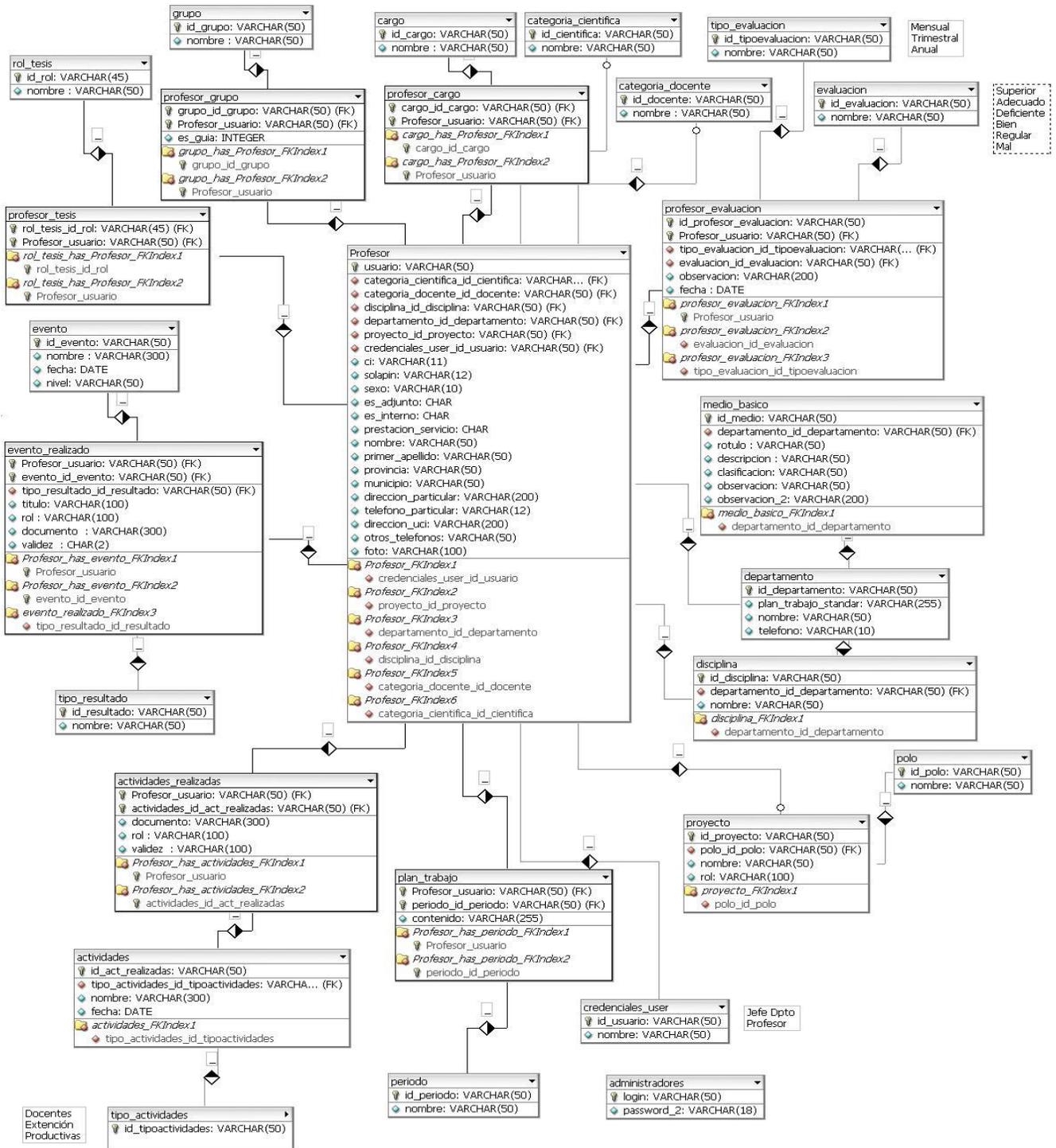


Ilustración 6: Modelo de datos

3.5. Implementación

El cliente es una parte más del equipo de desarrollo, su presencia es indispensable en las distintas fases de X.P. En el momento de codificar una historia de usuario su presencia es aún más necesaria. Los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que esta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada. Como ya se comentó anteriormente, la programación en parejas permite un código más eficiente y con gran calidad. La optimización del código siempre se debe dejar para el final.

De las historias de usuarios creadas por el cliente se generan tareas de ingeniería (TI) o tareas de programación como también se les conoce, algunas historias no se van a dividir en tareas pues se les dará solución conjuntamente como propone XP en sus principios, la programación en parejas para una mejor optimización del código.

El desarrollo del software se planificó en tres iteraciones de trabajo, en la primera de estas se implementan los módulos **Medios Básicos, Profesor y Plan de trabajo**. Las historias de usuarios a las que se les da solución en estos módulos se encuentran en Tabla #13: Plan de entrega.

HU #11

Tarea de Ingeniería.	
Número de la Tarea: 1	Número de la HU: 11
Nombre tarea: Crear cuenta de usuario.	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 27/03/09	Fecha Fin: 27/03/09
Programador Responsable: Rolian Ruiz Rodríguez	
Descripción: El sistema muestra la opción para que el administrador pueda crear una nueva cuenta de usuario.	

Tabla 28: TI Crear cuenta de usuario.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Tarea de Ingeniería.	
Número de la Tarea: 2	Número de la HU: 11
Nombre tarea: Buscar cuenta de usuario.	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 27/03/09	Fecha Fin: 28/03/09
Programador Responsable: Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción para que el administrador pueda buscar si ya la cuenta existe.	

Tabla 29: TI Buscar cuenta de usuario.

Tarea de Ingeniería.	
Número de la Tarea: 3	Número de la HU: 11
Nombre tarea: Modificar cuenta de usuario.	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 28/03/09	Fecha Fin: 29/03/09
Programador Responsable: Rolian Ruiz Rodríguez	
Descripción: El sistema muestra la opción para que el administrador pueda modificar los datos de la cuenta.	

Tabla 30: Modificar cuenta de usuario.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

HU #2

Tarea de Ingeniería.	
Número de la Tarea: 4	Número de la HU: 2
Nombre tarea: Llenar datos personales.	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 30/03/09	Fecha Fin: 30/03/09
Programador Responsable: Yaneisy Pérez Neira	
Descripción: El sistema muestra el formulario donde se pueden llenar los datos y da la opción de guardarlos.	

Tabla 31: Llenar datos personales.

Tarea de Ingeniería.	
Número de la Tarea: 5	Número de la HU: 2
Nombre tarea: Ver y modificar datos personales.	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 31/03/09	Fecha Fin: 31/03/09
Programador Responsable: Rolian Ruiz Rodríguez	
Descripción: El sistema brinda la opción de listar los datos personales dado algún criterio de búsqueda. Además si desea modificarlo se muestra el formulario correspondiente y da la opción de guardar los cambios.	

Tabla 32: TI Ver y modificar datos personales.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

HU #1

Tarea de Ingeniería.	
Número de la Tarea: 6	Número de la HU: 1
Nombre tarea: Crear medio básico.	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 1/04/09	Fecha Fin: 2/04/09
Programador Responsable: Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción para que el jefe de departamento pueda crear un medio básico.	

Tabla 33: TI Crear medio básico.

Tarea de Ingeniería.	
Número de la Tarea: 7	Número de la HU: 1
Nombre tarea: Buscar y listar medios básicos	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 3/04/09	Fecha Fin: 4/04/09
Programador Responsable: Rolian Ruiz Rodríguez	
Descripción: El sistema muestra la opción para que el jefe de departamento pueda buscar un medio básico determinado o listar los medios según algún criterio.	

Tabla 34: TI Buscar y listar medios básicos.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

HU # 4

Tarea de Ingeniería.	
Número de la Tarea: 8	Número de la HU: 4
Nombre tarea: HU Crear plan de trabajo	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 5/04/09	Fecha Fin: 11/04/09
Programador Responsable: Rolian Ruiz Rodríguez y Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción para que el profesor pueda crear y modificar su plan de trabajo anual.	

Tabla 35: TI Crear plan de trabajo

HU # 9

Tarea de Ingeniería.	
Número de la Tarea: 9	Número de la HU: 9
Nombre tarea: HU Gestionar Plan de Trabajo Individual del profesor	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 12/04/09	Fecha Fin: 15/04/09
Programador Responsable: Rolian Ruiz Rodríguez y Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción para que el jefe de departamento pueda buscar un determinado plan de trabajo y modificarlo si le es necesario.	

Tabla 36: TI Gestionar Plan de Trabajo Individual del profesor.

La **iteración 2** se dedica a al módulo **Actividades y eventos**. Esta iteración es de gran significado para el cliente ya que de este módulo se toma la mayor información que influye en la evaluación del profesor. Las historias de usuarios que abarca se pueden encontrar en la Tabla #13: Plan de entrega.

HU # 3

Tarea de Ingeniería.	
Número de la Tarea: 10	Número de la HU: 3
Nombre tarea: Adicionar actividades realizadas	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 16/04/09	Fecha Fin: 18/04/09
Programador Responsable: Rolian Ruiz Rodríguez	
Descripción: El sistema muestra la opción para que el profesor pueda adicionar las actividades realizadas.	

Tabla 37: TI Adicionar actividades realizadas

Tarea de Ingeniería.	
Número de la Tarea: 11	Número de la HU: 3
Nombre tarea: Modificar actividades realizadas	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 19/04/09	Fecha Fin: 22/04/09
Programador Responsable: Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción para que el profesor pueda modificar las actividades realizadas.	

Tabla 38: TI Modificar actividades realizadas

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

HU # 10

Tarea de Ingeniería.	
Número de la Tarea:12	Número de la HU: 10
Nombre tarea: Adicionar y modificar eventos realizados	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 23/04/09	Fecha Fin: 27/04/09
Programador Responsable: Rolian Ruiz Rodríguez	
Descripción: El sistema muestra la opción para que el profesor pueda adicionar o modificar los eventos en que ha participado.	

Tabla 39: TI Adicionar y modificar eventos realizados

Tarea de Ingeniería.	
Número de la Tarea:13	Número de la HU: 10
Nombre tarea: Adjuntar constancia del evento.	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 27/04/09	Fecha Fin: 29/04/09
Programador Responsable: Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción para que el profesor pueda adjuntar, en caso de que haya obtenido premio, el diploma que acredite el evento o la constancia de que participó en el mismo.	

Tabla 40: TI Adjuntar constancia del evento.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

HU # 5

Tarea de Ingeniería.	
Número de la Tarea:14	Número de la HU: 5
Nombre tarea: HU Gestionar actividades del profesor	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 30/04/09	Fecha Fin: 6/05/09
Programador Responsable: Rolian Ruiz Rodríguez y Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción de modificar las actividades, así como verificar su validez.	

Tabla 41: Gestionar actividades del profesor

En la iteración 3 se implementan los módulos **Evaluaciones y Reportes**. El primero de estos está dedicado a la evaluación por parte del jefe de departamento a los profesores. El segundo módulo se encarga de la creación de reportes necesarios para obtener información rápida y precisa. Las historias de usuarios que abarca se pueden encontrar en la Tabla #13: Plan de entrega.

HU# 8

Tarea de Ingeniería.	
Número de la Tarea:15	Número de la HU: 8
Nombre tarea: HU Reportes del módulo Superación	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 11/05/09	Fecha Fin: 17/05/09
Programador Responsable: Rolian Ruiz Rodríguez y Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción al jefe de departamento de obtener reportes del módulo de superación que serán brindados por <i>Web Service</i> .	

Tabla 42: TI Reportes del módulo Superación

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

HU #7

Tarea de Ingeniería.	
Número de la Tarea:16	Número de la HU: 7
Nombre tarea: HU Asignar evaluación	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 18/05/09	Fecha Fin: 21/05/09
Programador Responsable: Rolian Ruiz Rodríguez y Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción al jefe de departamento de asignar las diferentes evaluaciones (mensual, trimestral y anual) a los profesores.	

Tabla 43: Asignar evaluación.

HU # 6

Tarea de Ingeniería.	
Número de la Tarea:17	Número de la HU: 6
Nombre tarea: HU Generar reportes	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 22/04/09	Fecha Fin: 27/05/09
Programador Responsable: Rolian Ruiz Rodríguez y Yaneisy Pérez Neira	
Descripción: El sistema muestra la opción al jefe de departamento generar diferentes reportes por: Tipo de actividad. Profesor o usuario. Fecha. Evaluación. Medios básicos.	

Tabla 44: TI Generar reportes.

3.6. Conclusiones

Como parte de la fase de diseño de la metodología XP se emplea la técnica de las tarjetas CRC, con las mismas se logra que el equipo de desarrollo logre un mayor enfoque en la programación orientada a objetos garantizando así una mejor calidad del código en la fase de implementación. Para optimizar el tiempo de desarrollo de la aplicación algunas historias de usuarios fueron divididas en tareas de ingeniería a las cuales se le asignaron tiempos de implementación que fueron cumplidos cabalmente garantizando con esto el objetivo principal de su realización.

4

PRUEBA

4.1. Introducción

En el presente capítulo se muestran las pruebas de aceptación creadas por el cliente para verificar el buen funcionamiento de la aplicación. Las mismas fueron verificadas en cada entrega que se realiza del producto en la planificación establecida.

4.2. Pruebas de Aceptación (PA)

Uno de los pilares de la metodología XP es el proceso de pruebas. La metodología propone probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos no deseados a la hora de realizar modificaciones y refactorizaciones.

XP propone la realización de pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente. (12)

A la hora de codificar no se sigue la regla de XP que aconseja crear pruebas unitarias con entornos de desarrollo antes de programar. Las pruebas se obtienen de la descripción de requisitos plasmados en las historias de usuarios, y estas especifican las barreras que deben pasar las distintas funcionalidades del programa, procurando codificar teniendo en cuenta las pruebas que se deben vencer. Para realizar las pruebas de aceptación el cliente utiliza la siguiente plantilla.

Prueba de aceptación
HU: Nombre de la historia de usuario que va a comprobar su funcionamiento.
Nombre: Nombre del caso de prueba.
Descripción: Descripción del propósito de la prueba.
Condiciones de ejecución: Precondiciones para que la prueba se pueda realizar.
Entrada/Pasos de ejecución: Pasos para probar la funcionalidad.
Resultado esperado: Resultado que se desea de la prueba.
Evaluación de la prueba: Aceptada o Denegada.

Tabla 45: Plantilla Prueba de Aceptación.

A continuación se muestran las pruebas realizadas para verificar el funcionamiento de cada módulo en las entregas que se le hacen al cliente cumpliendo con lo establecido en el cronograma de entregas.

4.2.1. Módulo Medios Básicos

Prueba de aceptación
HU: Gestionar medios básicos.
Nombre: Adicionar medio básico.
Descripción: El jefe de departamento puede adicionar un medio básico a su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: Se intenta adicionar un nuevo medio básico al departamento docente.
Resultado esperado: Se adiciona el medio básico y se muestra un mensaje informativo.
Evaluación de la prueba: Satisfactoria.

Tabla 46: PA Adicionar medio básico.

Prueba de aceptación
HU: Gestionar medios básicos.
Nombre: Modificar medio básico.
Descripción: El jefe de departamento puede modificar los datos de un medio básico de su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir al menos un medio básico creado.
Entrada/Pasos de ejecución: Se intenta modificar los datos de un medio básico del departamento docente.
Resultado esperado: Se actualizan los datos del medio básico y se muestra un mensaje informativo.
Evaluación de la prueba: Satisfactoria.

Tabla 47: PA Modificar medio básico.

Prueba de aceptación
HU: Gestionar medios básicos.
Nombre: Eliminar medio básico.
Descripción: El jefe de departamento puede eliminar un medio básico de su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir al menos un medio básico creado.
Entrada/Pasos de ejecución: Se intenta eliminar un medio básico del departamento docente.
Resultado esperado: Se elimina de la lista de medios existentes y se muestra un mensaje informativo.
Evaluación de la prueba: Satisfactoria.

Tabla 48: PA Eliminar medio básico.

Prueba de aceptación
HU: Gestionar medios básicos.
Nombre: Mostrar medio básico.
Descripción: El jefe de departamento puede ver los datos de un medio básico determinado.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir al menos un medio básico creado.
Entrada/Pasos de ejecución: Se selecciona la opción de mostrar los datos.
Resultado esperado: Se muestran los datos.
Evaluación de la prueba: Satisfactoria.

Tabla 49: PA Mostrar medio básico.

4.2.2. Módulo Profesor

Prueba de aceptación
HU: Gestionar datos personales.
Nombre: Modificar datos personales
Descripción: El profesor puede modificar sus datos personales
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: Se selecciona la opción modificar los datos.
Resultado esperado: Se actualizan los datos y se muestra un mensaje informativo.
Evaluación de la prueba: Satisfactoria.

Tabla 50: PA Modificar datos personales.

Prueba de aceptación
HU: Gestionar datos personales.
Nombre: Mostrar datos personales
Descripción: El profesor puede ver sus datos personales.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: Se selecciona la opción mostrar los datos.
Resultado esperado: Se muestra un listado de los datos.
Evaluación de la prueba: Satisfactoria.

Tabla 51: PA Mostrar datos personales.

Prueba de aceptación
HU: Gestionar usuarios.
Nombre: Crear y modificar cuentas de usuarios.
Descripción: El administrador puede crear y modificar cuentas de usuarios.
Condición de ejecución:
Entrada/Pasos de ejecución: Crea o modifica cuentas de usuarios en la aplicación.
Resultado esperado: Se crean y se actualizan las cuentas de usuarios.
Evaluación de la prueba: Satisfactoria.

Tabla 52: PA Crear y modificar cuentas de usuarios.

Prueba de aceptación
HU: Gestionar usuarios.
Nombre: Eliminar cuentas de usuarios.
Descripción: El administrador puede eliminar cuentas de usuarios.
Condición de ejecución: Tiene que existir al menos una cuenta de usuario.
Entrada/Pasos de ejecución: Selecciona la opción de eliminar la cuenta de usuario.
Resultado esperado: Se eliminan la cuenta de usuario.
Evaluación de la prueba: Satisfactoria.

Tabla 53: PA Eliminar cuentas de usuarios.

Prueba de aceptación
HU: Gestionar usuarios.
Nombre: Crear y modificar cuentas de los profesores.
Descripción: El jefe de departamento puede crear o modificar cuentas de profesores de su departamento
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: Crea o modifica cuentas de profesores en la aplicación.
Resultado esperado: Se crean y se actualizan las cuentas de profesores.
Evaluación de la prueba: Satisfactoria.

Tabla 54: PA Crear y modificar cuentas de los profesores.

Prueba de aceptación
HU: Gestionar usuarios.
Nombre: Eliminar cuentas de profesores.
Descripción: El jefe de departamento puede eliminar cuentas de profesores.
Condición de ejecución: Tiene que existir al menos una cuenta de profesor creada.
Entrada/Pasos de ejecución: Selecciona la opción de eliminar la cuenta de profesor en la aplicación.
Resultado esperado: Se eliminan la cuenta de profesor.
Evaluación de la prueba: Satisfactoria.

Tabla 55: PA Eliminar cuentas de profesores.

4.2.3. Módulo Plan de Trabajo

Prueba de aceptación
HU: Crear Plan de Trabajo
Nombre: Crear plan.
Descripción: El profesor o jefe de departamento pueden crear su plan de trabajo.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: Se selecciona la opción de crear un plan de trabajo
Resultado esperado: Se brinda la posibilidad de confeccionar el plan, guardarlo y mostrar un mensaje de información posteriormente.
Evaluación de la prueba: Satisfactoria.

Tabla 56: PA Crear plan.

Prueba de aceptación
HU: Crear Plan de Trabajo
Nombre: Modificar plan.
Descripción: El profesor o jefe de departamento pueden modificar su plan de trabajo.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir al menos un plan de trabajo.
Entrada/Pasos de ejecución: Se selecciona la opción modificar un plan de trabajo
Resultado esperado: Se brinda la posibilidad de modificar el plan, guardarlo y mostrar un mensaje de información posteriormente.
Evaluación de la prueba: Satisfactoria.

Tabla 57: PA Modificar plan.

Prueba de aceptación
HU: Gestionar Plan de Trabajo Individual del profesor.
Nombre: Adicionar plan de trabajo del profesor.
Descripción: El jefe de departamento puede adicionar el plan de trabajo de los profesores de su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: Se selecciona la opción de adicionar un plan de trabajo.
Resultado esperado: Se actualiza el listado y se muestra un mensaje informativo.
Evaluación de la prueba: Satisfactoria.

Tabla 58: PA Adicionar plan de trabajo del profesor.

Prueba de aceptación
HU: Gestionar Plan de Trabajo Individual del profesor.
Nombre: Modificar las actividades del plan de trabajo del profesor.
Descripción: El jefe de departamento puede modificar las actividades del plan de trabajo de los profesores de su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir al menos un plan de trabajo creado.
Entrada/Pasos de ejecución: Se selecciona la opción de modificar un plan de trabajo.
Resultado esperado: Se actualiza el documento y se muestra un mensaje informativo.
Evaluación de la prueba: Satisfactoria.

Tabla 59: PA Adicionar plan de trabajo del profesor.

Prueba de aceptación
HU: Gestionar Plan de Trabajo Individual del profesor.
Nombre: Mostrar plan de trabajo del profesor.
Descripción: El jefe de departamento puede mostrar el plan de trabajo de los profesores de su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir al menos un plan de trabajo creado.
Entrada/Pasos de ejecución: Se selecciona la opción de mostrar un plan de trabajo.
Resultado esperado: Se muestra el plan de trabajo seleccionado.
Evaluación de la prueba: Satisfactoria.

Tabla 60: PA Mostrar plan de trabajo individual del profesor.

Prueba de aceptación
HU: Gestionar Plan de Trabajo Individual del profesor.
Nombre: Eliminar el plan de trabajo del profesor.
Descripción: El jefe de departamento puede eliminar los planes de trabajo que pertenecen a los profesores de su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir al menos un plan de trabajo creado.
Entrada/Pasos de ejecución: Se selecciona la opción eliminar un plan de trabajo.
Resultado esperado: Se actualiza el listado y se muestra un mensaje informativo.
Evaluación de la prueba: Satisfactoria.

Tabla 61: PA Eliminar el plan de trabajo del profesor.

Prueba de aceptación
HU: Gestionar Plan de Trabajo Individual del profesor.
Nombre: Listar planes de trabajo.
Descripción: El jefe de departamento puede ver todos los planes de trabajo de los profesores que pertenecen a su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir algún plan creado.
Entrada/Pasos de ejecución: El jefe selecciona la opción de mostrar todos los planes de trabajo.
Resultado esperado: Se muestra el listado de los planes de su departamento.
Evaluación de la prueba: Satisfactoria.

Tabla 62: PA Listar planes de trabajo.

4.2.4. Módulo Actividades y eventos

Prueba de aceptación
HU: Gestionar actividades realizadas.
Nombre: Adicionar, modificar y eliminar actividades.
Descripción: El profesor tiene la posibilidad de adicionar, modificar o eliminar las actividades que haya realizado.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: El profesor selecciona la opción de adicionar, modificar o eliminar las actividades realizadas.
Resultado esperado: Se adicionan, modifican o eliminan las actividades.
Evaluación de la prueba: Satisfactoria.

Tabla 63: PA Adicionar, modificar y eliminar actividades.

Prueba de aceptación
HU: Gestionar actividades realizadas.
Nombre: Adjuntar constancia.
Descripción: Si el profesor tiene algún documento que le acredite el cumplimiento de la actividad debe adjuntarlo a la misma.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: El profesor selecciona la opción adjuntar el documento a la actividad realizada.
Resultado esperado: Se guarda el documento.
Evaluación de la prueba: Satisfactoria.

Tabla 64: PA Adjuntar constancia.

Prueba de aceptación
HU: Gestionar actividades del profesor
Nombre: Adicionar y modificar actividades.
Descripción: El jefe de departamento adiciona o modifica las actividades realizadas por los profesores de su departamento.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: El jefe de departamento selecciona la opción de adicionar o modificar actividades.
Resultado esperado: Se adicionan o modifican las actividades y se muestra un mensaje informativo.
Evaluación de la prueba: Satisfactoria.

Tabla 65: PA Adicionar y modificar actividades.

Prueba de aceptación
HU: Gestionar eventos.
Nombre: Adicionar, modificar y eliminar eventos.
Descripción: El profesor adiciona, modifica o elimina los eventos en los que ha participado.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: El profesor selecciona la opción de adicionar, modificar o eliminar eventos.
Resultado esperado: Se adicionan, modifican o eliminan los eventos realizados.
Evaluación de la prueba: Satisfactoria.

Tabla 66: PA Adicionar, modificar y eliminar eventos.

Prueba de aceptación
HU: Gestionar eventos.
Nombre: Adjuntar documento.
Descripción: El profesor adjunta el documento que acredite la participación en el evento.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: El profesor selecciona la opción de adjuntar el documento al evento realizado.
Resultado esperado: Se adjunta el documento.
Evaluación de la prueba: Satisfactoria.

Tabla 67: PA Adjuntar documento.

4.2.5. Módulo Evaluaciones

Prueba de aceptación
HU: Asignar evaluación.
Nombre: Evaluación mensual y trimestral del profesor.
Descripción: El jefe de departamento les asigna una evaluación a sus profesores mensual y trimestralmente según su desempeño en Deficiente, Adecuado y Superior.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir algún profesor en el departamento.
Entrada/Pasos de ejecución: El jefe de departamento le asigna la evaluación a cada profesor.
Resultado esperado: El profesor tiene asignada la evaluación correspondiente al periodo que se trate.
Evaluación de la prueba: Satisfactoria.

Tabla 68: PA Evaluación mensual y trimestral del profesor.

Prueba de aceptación
HU: Asignar evaluación.
Nombre: Evaluación anual del profesor.
Descripción: El jefe de departamento les asigna una evaluación anual a sus profesores para incluirlo en el expediente docente del mismo en Bien, Regular o Mal.
Condición de ejecución: El usuario debe estar autenticado en el sistema. Debe existir algún profesor en el departamento.
Entrada/Pasos de ejecución: El jefe de departamento le asigna la evaluación a cada profesor.
Resultado esperado: El profesor tiene asignada la evaluación correspondiente.
Evaluación de la prueba: Satisfactoria.

Tabla 69: PA Evaluación anual del profesor.

4.2.6. Módulo Reportes

Prueba de aceptación
HU: Generar reportes
Nombre: Generación de reportes.
Descripción: El jefe de departamento genera reportes según los criterios seleccionados.
Condición de ejecución: El usuario debe estar autenticado en el sistema.
Entrada/Pasos de ejecución: El jefe de departamento elige los criterios por los cuales quiere generar los reportes y selecciona la opción de generarlos.
Resultado esperado: Se muestra el reporte.
Evaluación de la prueba: Satisfactoria.

Tabla 70: PA Generación de reportes.

4.3. Conclusiones

Con la realización de las pruebas de aceptación el cliente verifica el funcionamiento del software, probando individualmente cada módulo y asignándole la evaluación correspondiente. Todas las pruebas realizadas fueron aceptadas por el cliente cumpliendo entonces la aplicación con las historias de usuarios definidas inicialmente.

CONCLUSIONES GENERALES

A partir del análisis realizado en los departamentos docentes de la Facultad 8 de la Universidad de las Ciencias Informáticas se determinan una serie de procesos que se realizan manualmente en los mismos. En estos procesos están involucrados datos e informaciones de gran importancia con relación a los profesores y medios básicos de cada uno de los departamentos. Los jefes en estas áreas son los que se encargan de mantener bajo control todo este cúmulo de información, de la cual le es necesario obtener reportes y datos efectivos para emitir las evaluaciones correspondientes de cada profesor y mantener el cuidado y protección de los medios básicos. Para solucionar todas estas dificultades se desarrolló una aplicación que automatiza los procesos principales que tienen lugar en los departamentos docentes.

Para el desarrollo del sistema se efectuaron las siguientes actividades:

- Se hizo un análisis de los sistemas similares existente tanto nacional como internacionalmente de los cuales se obtienen ejemplos de cómo podrían solucionarse algunas funcionalidades que se requieren para la aplicación.
- Se llevó a cabo un estudio de las principales metodologías, lenguajes y herramientas que se consideraron factibles para el desarrollo del sistema.
- Se realizó todo el proceso de desarrollo del software siguiendo las fases de la metodología XP, lo cual queda plasmado en el presente documento.

Como resultado de la investigación se logra el desarrollo de un software en el que se da cumplimiento a las especificidades de los objetivos propuestos. Con el valor fundamental de simplificar la demora que produce el procesamiento manual de la información distribuida, de mejorar la gestión de las actividades que se realizan en los departamentos docentes de la facultad y además para contribuir a elevar la calidad del desarrollo del trabajo en los mismos.

RECOMENDACIONES

- Aplicar el software en los departamentos docentes de la Facultad 8 para obtener un mayor rendimiento del mismo.
- Extender las funcionalidades acorde a nuevos requisitos que surjan en los departamentos docentes.
- Agregar al módulo de reportes los datos correspondientes a la superación de los profesores.

TRABAJOS CITADOS

- [1] GNU Operating System. *La Definición de Software Libre*. [En línea] marzo de 2009.
<http://www.gnu.org/philosophy/free-sw.es.html>
- [2] **Gutierrez, Jorge A. Saavedra.** El Mundo Informático. *Software Libre*. [En línea] 2007.
<http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>
- [3] **Torre, Aníbal de la.** PHPNuke. *Lenguajes del lado servidor o cliente*. [En línea] 2007.
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html
- [4] **Amartino, Mariano.** AJAX un nuevo acercamiento a Aplicaciones Web. [En línea] 2005.
<http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>
- [5] **Paz, Carlos.** Linux Data. *CakePHP*. [En línea] 2009.
<http://www.linuxdata.com.ar/index.php?cakephp=1&seccion=2>
- [6] **Fabien Potencier, François Zaninotto.** "Symfony la guía definitiva". 2008.
- [7] W3C. *Guía Breve de Servicios Web*. [En línea] 01 de noviembre de 2008
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>
- [8] **Cuenca, Carlos Luis.** desarrolloweb. . *Arquitectura del servidor Apache*. [En línea] 2003.
<http://www.desarrolloweb.com/articulos/1112.php>
- [9] slideshare. *Sistemas Gestores de Bases de Datos*. [En línea] 2009.
<http://www.slideshare.net/alexmerono/sistemas-gestores-de-bases-de-datos>
- [10] **Mauricio, Ruben.** *Proceso de Desarrollo de Software RUP*. [En línea] 2007.
<http://pvgdata.com/Mauricio/index.php/archives/46>
- [11] **Fuentes, Isidro.** edib. *Tarjetas CRC*. [En línea] 28 de octubre de 2008.
http://informatica.escuelaedib.com/index.php?option=com_myblog&show=Tarjetas-CRC..html&Itemid=59
- [12] **J.J Gutierrez, M.J Escalona, M. Mejias, J.Torres.** *Pruebas del sistema en programación extrema*.
[En línea] http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf

BIBLIOGRAFÍA

Valdés, Damián Pérez. *Editores web que facilitan tu trabajo.* [En línea] 31 de enero de 2008.

<http://www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo/>

Arrúa, Cristhian. *Facultad de Ingenieria.* [En línea]

http://www.fing.edu.uy/tecnoinf/cursos/bd1/material/teo/Introduccion_BD.ppt

Castillo Oswaldo, Figueroa Daniel, Sevilla Hector. *Programación Extrema. Fases de la Programación Extrema.* [En línea] <http://programacionextrema.tripod.com/fases.htm>

Jacsonson. *"Applying UML in The Unified Process".* 1998.

Pérez, Daniel. *Lenguajes de Programación.* [En línea] 2009. <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>

García, Joaquín. *Ajax: Un nuevo acercamiento a las aplicaciones web.* [En línea]

<http://www.maestrosdelweb.com/editorial/ajax/>

Merelo Cuervos, Juan Julián. *Microsoft .Net . Servicios Web.* [En línea] 2002.

<http://geneura.ugr.es/~jmerelo/ws/>

LIGTTPD. [En línea] <http://www.lighttpd.net/>

Castillo, Alvaro del. *Zope: El servidor de aplicaciones libres.* [En línea] 2006.

<http://www.programacion.com/tutorial/zope/2/>

Sevilla, Lucio. *Una Introducción a APACHE.* [En línea] 2006.

http://linux.ciberaula.com/articulo/linux_apache_intro/ .

Murugarren, Joaquín Gracia. *Introducción a MySQL.* [En línea] 2005.

<http://www.webestilo.com/mysql/intro.phtml>

Pozo, Salvador. *MySQL con clase.* [En línea] marzo de 2005. <http://mysql.conclase.net/curso/>.

Teleformación. [En línea] http://teleformacion.uci.cu/file.php/102/Curso_2008-

[2009/Materiales_Basicos/Materiales_Basicos_CTP_1/Material_Clase_Teorico_Practica_1.doc](http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_CTP_1/Material_Clase_Teorico_Practica_1.doc)

- Alvarez, Miguel Angel.** Maestros del web. *Evaluando Zend Studio*. [En línea] 03 de noviembre de 2003.
<http://www.maestrosdelweb.com/editorial/zendstudio/>
- Gonzalez, Héctor.** desarrolloweb. *Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código*. [En línea] 04 de junio de 2003.
<http://www.desarrolloweb.com/articulos/1178.php>
- TORRES, LIC. Luis Gabriel Mondragón.** Slideshare. *CICLO DE VIDA PARA EL DISEÑO DE BASES DE DATOS*. [En línea] 2009. <http://www.slideshare.net/gabos/bdguia1>.
- Departamento de Control de Bienes.** [En línea] <http://www.siiia.ugto.mx/bienes/index.php>.
- EasyEclipse.** [En línea] 2007. <http://www.easyeclipse.org/site/about/index.html>.
- Torres, Raúl.** Lenguajes de Programación. [En línea] 2009. <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>.
- Sanchez, María A. Mendoza.** Informatizate. *Metodologías De Desarrollo De Software*. [En línea] 2005.
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
- Patricio Letelier, Ma. Carmen Penadés.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] <http://www.willydev.net/descargas/masyxp.pdf>.
- MSc., Hugo F. Arboleda Jiménez.** ACIS. *Modelos de ciclo de vida en desarrollo de software*. [En línea] <http://www.acis.org.co/index.php?id=551>.
- Alvarez, Miguel Angel.** desarrolloweb. *Un IDE para el desarrollo de aplicaciones web, enfocado en Ajax y la Web 2.0*. [En línea] 16 de 11 de 2007. <http://www.desarrolloweb.com/articulos/aptana-studio.html>.
- Barbone, Víctor A. González.** XP: Extreme Programming . [En línea] <http://ie.fing.edu.uy/~nacho/blandos/seminario/XProg1.html>.
- Pérez, Javier Eguíluz.** librosWeb.es. *Introducción a AJAX*. [En línea] <http://www.librosweb.es/ajax/capitulo1.html>.
- W3C.** *Guía Breve de Servicios Web*. [En línea] 2008.
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.

GLOSARIO DE TÉRMINOS

ActiveRecord: Es un patrón en el cual, el objeto contiene los datos que representan a un renglón (o registro) de la tabla o vista, además de encapsular la lógica necesaria para acceder a la base de datos.

BSD: Licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

CGI: *Common Gateway Interface* / Interface de Acceso Común o Interfaz Común de Puerta de Enlace. Es una importante tecnología de la World Wide Web que permite a un cliente (explorador web) solicitar datos de un programa ejecutado en un servidor web.

CCS: *Cascading Style Sheets* / Hojas de Estilo en Cascada. Tecnología desarrollada por el *World Wide Web Consortium* (W3C) con el fin de separar la estructura de la presentación.

CRUD: Es el acrónimo de *Create Read Update Delete* (Crear, Leer, Actualizar, Borrar) usado para referirse a las funciones básicas en Base de Datos o en la capa de persistencia de un sistema Software.

DESOFT: Sociedad mercantil de capital totalmente cubano denominada Sociedad para el Desarrollo de Productos Informáticos Especializados, S.A.

FastCGI: Es una alternativa al CGI estándar, cuya diferencia radica principalmente en el hecho de que el servidor crea un único proceso persistente por cada programa FastCGI en lugar de uno por cada solicitud del cliente.

GPL: *General Public License* / Licencia Pública General. Orientada principalmente a proteger la libre distribución, modificación y uso de software.

GUI: *Graphical User Interface* / Interfaz Gráfica de Usuario. Programa software que gestiona la interacción con el usuario de manera gráfica mediante el uso de íconos, menú, mouse, etc.

HTTP: *Hipertexto Transfer Protocol* / Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

HTML: *Hypertext Markup Language* / Lenguaje de marcado de hipertexto, usado para escribir documentos para servidores World Wide Web.

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

Logs: Son archivos en los que se recogen las visitas que tienen las páginas de un sitio web.

Microsoft: *Corporation Microcomputer Software*. Empresa multinacional estadounidense dedicada al sector de la informática.

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

PHP: *Hypertext Pre-processor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas.

Patrón: Es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

SQL: *Structured Query Language*. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas.

SOAP: *Simple Object Access Protocol* es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

UML: *Unified Modelling Language* / lenguaje de modelado gráfico que permite especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

Unix: Sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

UDDI: *Universal Discovery, Description and Integration*, es una especificación para un registro distribuido de información acerca de los servicios Web. Define la forma en la cual se publica y descubre información acerca de estos.

URL: *Uniform Resource Locator*/ localizador uniforme de recursos, es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

Windows: Es una familia de sistemas operativos desarrollados y comercializados por Microsoft. Existen versiones para hogares, empresas, servidores y dispositivos móviles.

WSDL: *Web Services Description Language*, un formato XML que se utiliza para describir servicios Web.

XML: *Extensible Markup Language* / lenguaje de marcado ampliable, es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C).