



Universidad de las Ciencias Informáticas

Facultad 2

Sistema de gestión de la planificación de las actividades
de los trabajadores docentes en la Facultad 2 de la
Universidad de las Ciencias Informáticas.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS

AUTOR:

Jorge Ernesto Bauzá Becerra

TUTOR:

Ing. Danisleydi Noroña Blanco.

La Habana, 2013

“Año 55 de la Revolución”

Declaración de autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

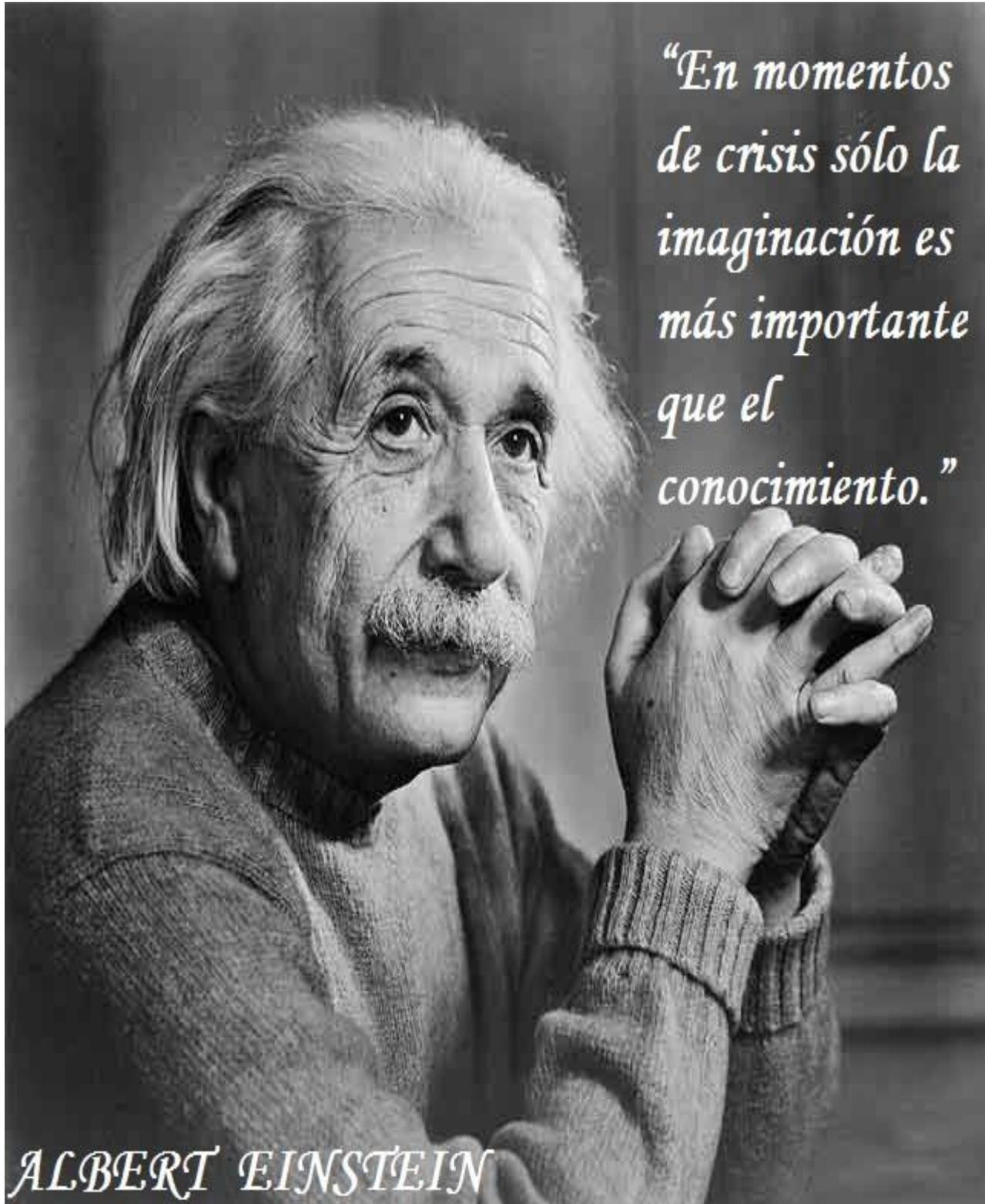
Para que así conste firmo la presente a los ____ días del mes de junio del año 2013.

Jorge Ernesto Bauzá Becerra

Autor

Ing. Danisleydi Noroña Blanco

Tutor



*“En momentos
de crisis sólo la
imaginación es
más importante
que el
conocimiento.”*

ALBERT EINSTEIN

Dedicatoria

Dedico este trabajo a mis padres María Luisa Becerra Roque y Jorge Bauzá Sosa por ser un gran ejemplo de respeto, responsabilidad y amor, por educarme sin descanso, por su preocupación constante, por darme todo a cambio de nada, gracias por existir.

A mi familia, al piquete de mis amigos por esos momentos inolvidables y a todos los que me quieren.

Agradecimientos

A mis padres por ser guía y luz de mi vida, por su ejemplo y dedicación.

A Francis por estar siempre presente en los buenos y malos momentos que tuve como estudiante universitario y por ayudarme a alcanzar mi meta, el ser ingeniero.

A mis amigos del barrio Carlos, Yasel, Lázaro (Pini), Tony, Ronni, Fernando, Pedro y Ronal que vienen conmigo desde el pre.

A la gente del 2108, 2305 y 2504 por compartir conmigo cada día que he transcurrido en esta universidad haciéndola más placentera y divertida.

A Roberto, Olivio y Grabiél por ayudarme en aquellos momentos en que los errores de Symfony me volvían loco.

A los tribunales que me han atendido, gracias por sus críticas constructivas y consejos que me han ayudado a prepararme para este momento tan importante.

A mi tutora por su dedicación y empeño.

A la UCI por darme la oportunidad de compartir con todas estas personas los mejores años de mi vida.

Resumen

La Universidad de las Ciencias Informática (UCI) cuenta con un gran número de trabajadores docentes, los cuales tienen como objetivo principal formar nuevos ingenieros informáticos capacitados para ejercer su profesión en cualquier lugar del país. Todo esto es posible gracias a la dirección docente de las distintas facultades las cuales se esfuerzan al máximo para que las actividades docente que realizan lleguen a todos sus estudiantes.

El presente trabajo de diploma se centra en la necesidad de desarrollar un sistema informático que permita gestionar la planificación de las actividades de los trabajadores docentes en la Facultad 2. Para dar cumplimiento al objetivo de la investigación se realizó la fundamentación teórica donde se analizaron los conceptos fundamentales relacionados con el tema para profundizar los conocimientos a la hora de informatizar los procesos a gestionar. Además, se definieron las tecnologías, herramientas y lenguajes utilizados en el desarrollo del sistema, decidiéndose el uso de lenguaje PHP, el framework Symfony y como gestor de base de datos MySQL. Se efectuó un análisis de aplicaciones similares existentes en el ámbito nacional e internacional. El proceso de desarrollo de software estuvo guiado por la metodología XP.

Palabras clave: Plan de actividades, proceso docente, trabajadores docentes.

Índice

| | |
|--|----|
| Introducción..... | 1 |
| Capítulo 1: Fundamentación Teórica..... | 4 |
| 1.1 Introducción..... | 4 |
| 1.2 Conceptos fundamentales..... | 4 |
| 1.3 Estudio del estado del arte..... | 8 |
| 1.4 Metodología de desarrollo de software..... | 10 |
| 1.5 Lenguajes de programación..... | 12 |
| 1.6 El Lenguaje de Modelado..... | 14 |
| 1.7 Herramientas de software y patrón de arquitectura utilizados..... | 15 |
| 1.8 Conclusiones Parciales..... | 20 |
| Capítulo 2: Descripción de la solución..... | 21 |
| 2.1 Introducción..... | 21 |
| 2.2 Objeto de estudio..... | 21 |
| 2.2.1 Descripción de los procesos vinculados al campo de acción..... | 21 |
| 2.2.2 Descripción del flujo actual del proceso..... | 21 |
| 2.2.3 Objetos a informatizar..... | 22 |
| 2.3 Propuesta de solución..... | 22 |
| 2.3.1 Personas relacionadas con la aplicación..... | 22 |
| 2.3.2 Lista de reserva del producto..... | 23 |
| 2.3.3 Fase de exploración..... | 29 |
| 2.3.4 Planificación..... | 35 |
| 2.4 Conclusiones Parciales..... | 37 |
| Capítulo 3: Diseño e implementación del sistema..... | 38 |
| 3.1 Introducción..... | 38 |
| 3.2 Arquitectura del sistema..... | 38 |
| 3.3 Diseño..... | 38 |
| 3.3.1 Tarjetas CRC..... | 40 |
| 3.4 Implementación..... | 43 |
| 3.4.1 Tareas de ingeniería..... | 44 |
| 3.4.2 Diseño de la Base de Datos del sistema..... | 45 |
| 3.4.3 Diagrama de despliegue..... | 46 |

| | |
|---|----|
| 3.5 Conclusiones Parciales..... | 47 |
| Capítulo 4: Validación del Sistema..... | 48 |
| 4.1 Introducción..... | 48 |
| 4.2 Tipos de pruebas..... | 48 |
| 4.3 Conclusiones parciales | 63 |
| Conclusiones | 64 |
| Recomendaciones | 65 |
| Trabajos citados | 66 |
| Bibliografía | 68 |

Introducción

La educación universitaria logra una superación individual física, emocional e intelectual del individuo. Los estudiantes universitarios desarrollan diferentes habilidades y destrezas que le permiten explotar al máximo todos los conocimientos adquiridos durante su formación académica, debido a esto es preciso lograr que la formación del estudiante sea de excelencia de manera que se pueda potenciar el capital humano de los profesionales como una importante vía para el desarrollo económico y social.

En el caso particular de Cuba, todas sus universidades están inmersas en un nuevo proceso educativo con el objetivo de lograr que los graduados de nivel superior, posean la suficiente preparación general integral que les permita desarrollar un alto rigor científico y eficiencia en el cumplimiento de las tareas a ellos encomendadas.

En el 2002 surge la Universidad de las Ciencias Informáticas (UCI) como parte de la Batalla de Ideas llevada a cabo en el país. Su objetivo es formar profesionales eficientes y comprometidos, capaces de llevar el desarrollo a cada rincón del país de la mejor manera posible. Esta universidad, por sus características particulares, establece nuevas formas de control y evaluación del proceso docente.

La UCI cuenta con un claustro de profesores encargados de llevar a cabo fundamentalmente las actividades docentes. Este claustro tiene la responsabilidad de planificar, coordinar, decidir e informar sobre los aspectos docentes del Centro, siendo su función primordial la de impartir la docencia con calidad a todo el alumnado.

Para llevar a cabo el proceso de planificación de las actividades docentes y otras actividades en la Facultad 2 se envía a los profesores, por parte de la dirección de la misma, un plan de trabajo mensual. Este plan es utilizado por los trabajadores docentes para elaborar su plan individual el cual debe ser revisado por su jefe inmediato superior. Sin embargo, debido al dinamismo de la UCI es natural que surjan actividades que no han sido contempladas inicialmente. Por otro lado cada trabajador docente cuenta con un plan anual que es elaborado por su jefe inmediato superior y posteriormente revisado y aprobado por ambos.

Además de estos procesos de planificación los trabajadores docentes se ven envueltos en otras actividades como son los controles a clases y la evaluación trimestral de su desempeño. En la primera, además de una evaluación cuantitativa, reciben un informe donde se plasma la valoración de cada indicador evaluado, los principales logros, las deficiencias y las recomendaciones. En la segunda se llena un modelo de evaluación a partir del trabajo realizado en cada trimestre.

Todos estos procesos se realizan de forma manual y descentralizada lo que trae como inconvenientes:

- Afectaciones en la participación de los trabajadores en las actividades de última hora ya sea por falta de información, por la no llegada en tiempo de la información o por coincidencias con otras actividades, así como inconformidad por parte de los mismos con las planificaciones realizadas.
- Necesidad de emplear mucho tiempo de la jornada laboral de los trabajadores docentes para la elaboración de su plan de trabajo individual.
- Falta de disponibilidad de documentación y/o informaciones como la evaluación trimestral y el resultado de los controles a clases, necesarias para la consulta sistemática por parte de los trabajadores docentes.

Todas estas dificultades entorpecen la organización, planificación y control de las actividades docentes conllevando a que no se alcancen los resultados esperados por el claustro de profesores de la facultad. Es evidente también que la ausencia de un espacio, donde se puedan consultar los documentos generados por las distintas actividades docentes traiga como consecuencia que coincidan actividades en horario y fecha, se entorpezca el proceso de evaluación o que su confección requiera más tiempo del necesario.

A partir de todo lo anteriormente planteado se define la siguiente interrogante como **problema a resolver** ¿Cómo gestionar el proceso de planificación de las actividades y la disponibilidad de la información para el colectivo docente en la Facultad 2 de la UCI?

Se define como **objeto de estudio**: el proceso de gestión de la planificación de los colectivos docentes.

Para dar solución al problema planteado se define el siguiente **objetivo general**: Desarrollar un sistema informático que centralice la gestión de la planificación de las actividades que realizan los trabajadores docentes de la Facultad 2 de la UCI.

Delimitando como **campo de acción** el proceso de gestión de la planificación mensual y anual del colectivo docente de la Facultad 2 de la UCI.

De este objetivo se derivan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación identificando las principales tendencias, limitaciones y ventajas de los sistemas informáticos de gestión de planificación docente así como las actividades llevadas a cabo por los trabajadores docentes de la facultad 2 de la UCI.
- Diseñar un sistema de gestión que logre centralizar la planificación de las actividades de los trabajadores docentes.
- Implementar las funcionalidades requeridas para el sistema de gestión, cumpliendo con estándares de codificación y calidad definidos en la UCI.
- Validar las funcionalidades del sistema de gestión desarrollado.

Para dar cumplimiento a los objetivos se definen las siguientes **tareas de investigación**:

- Caracterización de las tendencias existentes para el desarrollo de los procesos de gestión de información.
- Establecimiento de los fundamentos que deben sostener el proceso de gestión de las actividades de los trabajadores docentes en la Facultad 2 de la UCI para su informatización.
- Selección de las tecnologías, herramientas, estándares, patrones y metodologías necesarias para el desarrollo del sistema de gestión.
- Estructuración de los requerimientos necesarios para el desarrollo del sistema de gestión.
- Diseño del sistema de gestión de la planificación de las actividades de los trabajadores docentes en la Facultad 2 de la UCI.

- Implementación del sistema de gestión de la planificación de las actividades de los trabajadores docentes en la Facultad 2 de la UCI
- Aplicación de las pruebas de funcionalidad al sistema de gestión desarrollado para su validación.

Se hará uso de los siguientes métodos como técnicas para la recopilación y búsqueda de información.

Métodos teóricos

- Analítico-Sintético: Permite analizar la bibliografía encontrada permitiendo extraer de ella los elementos más importantes que sirvan para la implementación del sistema en cuestión.
- Método Inductivo – Deductivo: Permite analizar y seleccionar las herramientas más adecuadas.
- Modelación: Permite realizar los modelos correspondientes al ciclo de vida del desarrollo del Sistema de Gestión propuesto.

Se propone como **idea a defender** que el desarrollo de un sistema que gestione el proceso de planificación de las actividades que realizan los trabajadores docentes de la Facultad 2 de la UCI permitirá centralizar la información que se genera así como mejorar la organización y planificación del trabajo de una manera rápida y eficiente.

Posibles resultados:

- Sistema de Gestión para la informatización y centralización de las actividades de los trabajadores docentes en la Facultad 2 de la UCI.
- Informe detallado de la base teórico-práctica sobre la que se sustenta la solución propuesta.

Este informe de investigación está estructurado de la siguiente manera: **Introducción, 4 Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografías Consultadas, Anexos.**

Capítulo 1 Fundamentación Teórica: Se hace un análisis de las tendencias y tecnologías actuales sobre las que se apoyará la propuesta de solución. Se fundamentan las herramientas y lenguaje a emplearse para el análisis y modelado de un sistema acorde con los objetivos.

Capítulo 2 Descripción de la solución: Se presenta una propuesta del sistema a desarrollar, se muestran los requerimientos de la propuesta de solución mediante las historias de usuario, estas a su vez proporcionan los detalles sobre la estimación del riesgo y una planificación del tiempo de implementación de dicha historia de usuario.

Capítulo 3 Diseño e implementación del sistema: Se muestra el diseño del sistema y se describen los artefactos relacionados con la implementación. Se definen las tarjetas CRC (Contenido, Responsabilidad, Colaboración) y se detallan las iteraciones para el desarrollo del sistema, exponiéndose las tareas generadas en cada una de ellas por cada historia de usuario.

Capítulo 4 Validación del Sistema: Se describen las pruebas realizadas a la aplicación. Para ello se diseñan los casos de pruebas de aceptación a los que será sometida la aplicación en cada una de las iteraciones.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se muestra el resultado de las investigaciones realizadas, teniendo como base el objeto de estudio. Se hace un análisis de las tendencias y tecnologías actuales sobre las que se apoyará la propuesta de solución. Se fundamentan las herramientas y lenguajes a emplearse para el análisis y modelado de un sistema acorde con los objetivos.

1.2 Conceptos fundamentales

Colectivo docente

El origen del uso del término colectivo docente se ubica en España, a finales de la década de los 80's del siglo pasado. Surge con la reforma educativa que se generó en ese país (1). Este cambio llevó a los docentes a emigrar de un entorno en el que la individualidad era el sentido del trabajo diario a otro donde se sabe parte de un grupo de colegas que comparten responsabilidades, tareas, éxitos y problemáticas diversas. En este contexto se realiza un modelo pedagógico en el que el alumno se convierte en el centro de la intervención docente (2).

Méndez (3) recoge una primera aproximación sobre el concepto de colectivo docente desde una perspectiva amplia, al afirmar que es “el conjunto de profesionales de una institución educativa con responsabilidades docentes, sin importar el estatus que tengan en la institución, este primer acercamiento aporta una panorámica orientada hacia el trabajo áulico de los profesores y la noción de pertenecía a un grupo de profesionales de la educación con un fin en común, la enseñanza”.

Varios especialistas describen al colectivo docente en tres valoraciones principales con relación a los diferentes lazos definidos por el entorno contextual y temporal en el que se ubican:

- Tareas pedagógicas
- Identidad profesional
- Formación profesional

Wass (4), Le Gal (5) y Gather Thurler (6) afirman que “el colectivo docente desarrolla como compromiso fundamental todo tipo de tareas pedagógicas entre las que sobresalen la orientación y la educación integral de los alumnos, así como las insertas en el cotidiano laboral que van desde diseñar planes de clase, estrategias de intervención, materiales y recursos didácticos hasta un reconocimiento profundo de todas las dimensiones de los alumnos que atienden.”

Existen otro grupo de especialistas orientados al estudio de los significados del que hacer docente en base a procesos de investigación cualitativa como la investigación-acción, ellos han profundizado sobre el tema; algunos de sus hallazgos identifican al colectivo docente como un medio favorecedor para contribuir en la construcción de la identidad profesional, consideran que el compartir un espacio de trabajo laboral favorece la construcción de significados entre ellos sobre su práctica profesional (7).

Una tercera perspectiva está orientada hacia la formación profesional. El colectivo docente es visto como un espacio en el que el maestro aprende y se forma, de esta forma se generan procesos que favorecen el desarrollo profesional (8) en un ambiente de aprendizaje caracterizado por el trabajo colaborativo y la reflexión de la práctica docente (9).

Planificación

La Planificación es la primera función de la administración, y consiste en determinar las metas u objetivos a cumplir. La planificación incluye seleccionar misiones y objetivos como las acciones para alcanzarlos; requiere tomar decisiones; es decir, seleccionar entre diversos cursos de acción futuros. Así la planificación provee un enfoque racional para lograr objetivos preseleccionados (10).

Los esfuerzos que se realizan a fin de cumplir objetivos y hacer realidad diversos propósitos se enmarcan dentro de una planificación. Este proceso exige respetar una serie de pasos que se fijan en un primer momento, para lo cual aquellos que elaboran una planificación emplean diferentes herramientas y expresiones. La planificación supone trabajar en una misma línea desde el comienzo de un proyecto, ya que se requieren múltiples acciones cuando se organiza cada uno de los proyectos. Su primer paso, dicen los expertos, es trazar el plan que luego será concretado. En otras palabras, la planificación es un método que permite ejecutar planes de forma directa, los cuales serán realizados y supervisados en función del planeamiento (11).

La planificación cumple dos propósitos principales en las organizaciones: el protector que consiste en minimizar el riesgo, reduciendo la incertidumbre que rodea al mundo de los negocios y definiendo las consecuencias de una acción administrativa determinada; y el afirmativo que consiste en elevar el nivel de éxito organizacional. Un propósito adicional de la planificación consiste en coordinar los esfuerzos y los recursos dentro de las organizaciones.

Para que la planificación sea efectiva, es preciso tener en consideración los siguientes principios:

- **Precisión:** Los planes no deben hacerse con afirmaciones genéricas, sino con la máxima precisión posible, porque están destinados a regir acciones concretas.
- **Flexibilidad:** Dentro de la precisión, todo plan debe dejar margen para los posibles cambios que puedan surgir en razón de lo imprevisible o de circunstancias que hayan variado desde el origen.
- **Unidad:** Los planes deben ser de naturaleza tal, que pueda afirmarse la existencia de un solo plan para cada función, los cuales estarán integrados y coordinados de modo que constituyan un solo plan general (12).

Las etapas de la planificación varían su descripción, de acuerdo a varios autores; sin embargo, analizando un poco sus planteamientos, se pueden establecer de forma general, las siguientes:

- **Elaboración del plan o planeamiento:** en esta etapa se hace un estudio de la situación o del objeto a planificar; se diseña el modelo normativo que contiene el futuro propuesto para el objeto, las correspondientes estrategias a aplicar y los medios necesarios y el modelo operativo con los respectivos programas que permitan lograr los objetivos formulados en el modelo normativo.

- **Preparación de las condiciones para la ejecución del plan:** en esta etapa se toman una serie de medidas que pueden ser: políticas, sociales, legales, institucionales, financieras, organizativas, relativas a los recursos materiales, con respecto al personal; que garanticen el éxito en la aplicación del plan.
- **Ejecución del plan:** en esta etapa se pone en ejecución el plan, siguiendo lo establecido en el modelo operativo y en particular, en sus proyectos y programas. Además se controla la ejecución y los resultados parciales que se van obteniendo, para introducir los correctivos necesarios.
- **Evaluación de los resultados:** se evalúan los resultados obtenidos, la corrección de la estrategia y la pertinencia de los medios (12).

Planificación Docente

La docencia podemos considerarla como una actividad internacional cuyo desarrollo exige una planificación que defina el currículo de propuestas susceptibles a llevar a cabo. Esta actividad de puntualización corresponde esencialmente al profesor que es el último responsable de la acción educativa ante los estudiantes. La habilidad para planificar se considera un aspecto fundamental dentro de las competencias que definen el rol profesional de los profesores. Es el aspecto central de la fase pre activa, es decir, del período anterior a la instrucción directa con el alumno. Por otra parte, a lo largo de los años de formación inicial, los profesores reciben un entrenamiento específico en tareas de planificación y programación del currículo (13).

La administración docente exige normalmente, entre los requisitos de acceso al ejercicio de la profesión educativa, una programación de la actividad a desempeñar. Del mismo modo, cada año, al principio del curso escolar, en todos los centros de enseñanza, los profesores abordan diversas tareas relacionadas con el diseño curricular, los proyectos curriculares del centro y las diferentes programaciones que intervienen en el proceso de planificación.

Por lo tanto la planificación docente es, un proceso secuencial a través del cual se establecen una serie de pasos que conducen la enseñanza a una meta final. Una planificación eficaz requiere poner en marcha una serie de habilidades y conocimientos, que no siempre resultan conscientes para el que planifica (13).

Gestión

El término gestión hace referencia a la acción y a la consecuencia de administrar o gestionar algo. Por lo que ella es la encargada de llevar a cabo el servicio que permite el desarrollo de las operaciones comerciales. La gestión se compone por las actividades de acción, control, decisión e información, dichas actividades forman parte de un ciclo de permanente retroalimentación, esto permite crecer y alcanzar los objetivos propuesto por la persona o entidad. (14)

Tomando el concepto de gestión como un ciclo, se puede afirmar que la gestión es más bien un proceso, al definirlo así es posible analizarlo y descomponerlo en subprocesos siendo más fácil su estudio e investigación. Este proceso aún mantiene las cuatro componentes básicas de la gestión como son: planificación, organización, dirección y control. Para que este proceso funcione, todas las personas involucradas en él deben presentar una comunicación constante, con tal de lograr la meta propuesta. En aras de facilitar la comunicación, el proceso de gestión se divide estructuralmente en grupos pequeños los cuales interactúan habitualmente entre ellos debido a su cercanía física, por funciones, o por intereses comunes, ya sean formales o informales.

Proceso de Gestión Docente

Es importante señalar que el Proceso Docente Educativo es más que la mera actividad de impartir la docencia, mucho más que la relación estudiante-profesor. Hoy en día dado que los sistemas educacionales están permanentemente sometidos a transformaciones estructurales se ha introducido el concepto de gestión, proveniente del mundo empresarial, como una forma de responder a dichas transformaciones, ya sea a nivel macro o micro. A nivel macro podemos hablar de gestión docente, como aquella disciplina encargada de la gestión global de los sistemas educacionales y de la creación de políticas educativas tomando como referente el marco curricular nacional, mientras que a nivel micro podemos situar a esta gestión como aquella disciplina que enfatiza principalmente el trabajo en equipo y, en consecuencia, el aprendizaje organizacional, tendientes a la generación de procesos que permitan diagnosticar, planificar, implementar y evaluar el quehacer pedagógico de la institución educativa. La gestión docente, por tanto, puede ser entendida como la capacidad de dirigir una organización educativa tendiendo como principal foco la generación de aprendizaje, e involucrando a las diversas dimensiones que allí se dan cita. (15)

Varios autores han tratado el término de Gestión Docente, la que se caracteriza fundamentalmente por enfocar de manera amplia las posibilidades reales de una institución, en el sentido de resolver situaciones o el de alcanzar un propósito en cuestión. Se afirma que esta gestión constituye la acción principal de la administración y es un eslabón intermedio entre la planificación y los objetivos concretos que se pretenden alcanzar. He aquí algunas definiciones al respecto: (16)

- Son los trámites que se realizan para la resolución de los asuntos o proyectos educativos.
- Proceso mediante el cual se dirige, conduce, orienta y administra una institución educativa.
- Proceso orientado al fortalecimiento de los proyectos educativos de las instituciones, que ayuda a mantener la autonomía institucional, en el marco de las políticas públicas, y que enriquece los procesos pedagógicos con el fin de responder a las necesidades educativas locales o regionales.
- Es la disposición y organización de los recursos de un individuo o grupo para obtener los resultados esperados.
- Es el arte de anticipar participativamente el cambio con el propósito de crear permanentemente estrategias que permitan garantizar el futuro deseado o una forma de alinear los esfuerzos y recursos para alcanzar un fin determinado.
- Es el conjunto de procesos, de toma de decisiones y ejecución de acciones que permiten llevar a cabo las prácticas pedagógicas, su ejecución y evaluación.
- Es la gestión del entorno interno de la entidad docente hacia el logro de sus objetivos.
- Es el proceso de las acciones, transacciones y decisiones que la organización o escuela lleva a cabo para alcanzar los objetivos propuestos.
- Conjunto de acciones de movilización de recursos orientadas a la consecución de objetivos.

Pozner, respecto al concepto de gestión docente, planteó que ésta puede ser entendida como “el conjunto de acciones, articuladas entre sí, que emprende el equipo directivo en una institución educativa, para promover y posibilitar la consecución de la intencionalidad pedagógica en y con la comunidad educativa”. (15) Esta autora plantea que, aparte de la ejecución de reglamentaciones, la gestión docente debe preocuparse además de la calidad y cantidad de los aprendizajes que se produzcan en la institución educativa.

La gestión docente consiste en:

- Presentar un perfil integral, coherente y unificado de decisiones.
- Definir los objetivos institucionales, las propuestas de acción y las prioridades en la administración de recursos.
- Definir acciones para extraer ventajas a futuro; se consideran tanto las oportunidades y amenazas del medio en el que está inserta, como los logros y problemas de la misma organización.
- Comprometer a todos los actores institucionales.
- Definir el tipo de servicio educativo que se ofrece.

El desarrollo del proceso es responsabilidad de la directiva de la institución académica (pero no es la que realiza todas las tareas), debe: (17)

- Planificar
- Controlar
- Definir los objetivos
- Tomar decisiones para solucionar problemas
- La comunicación
- Capacitación del personal
- La influencia del poder.

En síntesis, la Gestión Docente se conceptualiza como el "conjunto de operaciones y actividades encaminadas a la adquisición y manejo de los recursos educativos". Este concepto interrelaciona las situaciones físicas y humanas de las organizaciones.

La gestión docente supone, en consecuencia, una nueva forma de comprender y conducir la organización escolar de tal manera que la labor cotidiana de la enseñanza llegue a ser un proceso práctico generador de decisiones y comunicaciones específicas que tienden al mejoramiento de las prácticas educativas, a la exploración y explotación de posibilidades, y a la innovación permanente como proceso sistemático.

1.3 Estudio del estado del arte.

En el mundo actual los sistemas de gestión de información permiten aprovechar al máximo los recursos de información en función de la mejora continua y de la toma de decisiones. Entre ellos los sistemas de gestión de la planificación docente facilitan la manipulación de la información relacionada con el proceso de planificación de los centro de estudios. El desarrollo de un nuevo sistema de gestión de este tipo necesita de un análisis minucioso de otros sistemas que posean características o funcionalidades similares al que se desea implementar.

A continuación se presenta un resumen de los elementos más significativos de algunos sistemas actuales de gestión de la planificación e información docente encontrados por el autor de la presente investigación en Cuba y en el mundo, con el objetivo de identificar aquellos que puedan ser utilizados como posible solución a la problemática planteada:

Panorama Internacional

En el ámbito internacional con respecto al proceso de gestión de la planificación docente, han sido creados en diversas universidades numerosos sistemas informáticos con el objetivo de facilitar la planificación de las actividades docentes y la disponibilidad de información sobre estas actividades, entre los sistemas encontrados están:

Sistema de Gestión de la Actividad Docente en la Universidad de León, España. Sistema basado en una aplicación desktop mediante la cual se gestiona la información sobre la carga docente en la universidad, algunos ejemplos de sus funciones son: controlar las altas y bajas del profesorado y de las asignaturas que imparten, gestionar el plan docente existente, conocer si un profesor está disponible, listar las asignaturas existentes por áreas y brindar información a los profesores sobre todo lo que ocurre en función de la docencia.

La Universidad de Alicante en España tiene un portal Web en el cual se aprecian varios servicios importantes relacionados con la vida estudiantil. Además se hace uso del mismo portal para gestionar información docente como publicación de noticias, planificación de horarios y actividades docentes, o información sobre departamentos docentes.

Es importante resaltar que estas universidades tuvieron que pagar una licencia, ya que estos sistemas se basan en software propietario.

Panorama Nacional

En el panorama nacional se encontraron diferentes sistemas enfocados a la informatización de procesos relacionados con la planificación docente. Entre los sistemas encontrados es importante resaltar que mucho de ellos utilizan tecnología web. A continuación se mencionan las soluciones más destacadas:

Sistema de Gestión de la planificación docente en la Facultad de Ingeniería Eléctrica de la Universidad de Oriente, permite gestionar la planificación docente en el Departamento de Informática de dicha facultad, con el objetivo de garantizar la ejecución de la misma de una manera más rápida y eficiente.

Sistema de Gestión para el control de las Actividades Científicas de los Docentes (SGCAD) en la Facultad de Ciencias Técnicas de la Universidad de Las Tunas, su objetivo es gestionar los trabajos de los docentes, los premios obtenidos por los trabajos en los distintos eventos así como las actividades científicas a realizarse a lo largo del curso. Permite realizar búsquedas por diferentes criterios y guardar resúmenes y/o trabajos a presentar.

Sistema de Gestión de las Actividades Metodológicas de la Facultad Regional “Mártires de Artemisa”, Universidad de las Ciencias Informáticas. Este tiene como objetivo, la planificación de las actividades realizadas por cada jefe de departamento, asignatura, colectivo de año y disciplina que se encuentran involucrados en dicho proceso, para de esa forma contribuir al fortalecimiento del proceso docente – educativo de la facultad. La información brindada por el sistema le permite a los directivos y usuarios acceder y consultar los diferentes datos relacionados con el plan de las actividades metodológicas.

Sistema de gestión de información de la facultad 8. Módulo para la gestión de la información docente. Sistema informático que ayuda en la gestión de la información docente que se manipula en la Facultad 4, anteriormente Facultad 8, de la UCI. El sistema tiene el propósito de digitalizar la información que se genera en la facultad incluyendo los reportes, contribuye en la simplificación del trabajo y la demora que produce el procesamiento manual de la información. Permite mejorar la gestión de las actividades que se realizan en la facultad.

Sistema automatizado para la gestión de información de los cursos optativos de la Facultad 2. Mejora la manipulación y administración de la información que se genera, mediante este sistema se lleva un control de la participación de los estudiantes en los cursos optativos que tienen relación con la facultad, los cuales en cierto sentido son de obligatorio cumplimiento.

Herramienta para la gestión de proyectos “Redmine” de la UCI. Dicho sistema es una aplicación web, de trabajo colaborativo, orientada a la coordinación de proyectos, principalmente de desarrollo de software, sirviendo como herramienta principal de comunicación entre los distintos componentes del proyecto. Esta es usada para mostrar y gestionar las tareas asignadas a cada estudiante y profesor. Hay que resaltar que esta herramienta en la UCI está configurada para la gestión de los proyectos que presenta la universidad, por lo cual no es utilizada en el ámbito de la gestión docente de las facultades.

Luego de un análisis de los sistemas de gestión de información y planificación docente encontrados se decide que no es conveniente optar por ninguna de esas soluciones. Los sistemas internacionales analizados se basan en software propietario y sería necesario pagar las licencias correspondientes para poder utilizarlos. En el caso de las soluciones nacionales las que tienen como objetivo el proceso docente no abarcan todos los aspectos a los que se pretende dar solución en esta investigación y las otras solo se refieren a la información particular de los estudiantes no respondiendo así a las exigencias del presente trabajo.

1.4 Metodología de desarrollo de software.

Las metodologías son un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayudan a los desarrolladores a realizar nuevos software. La metodología indica cómo hay que obtener los distintos productos parciales y finales. Las metodologías de desarrollo de software se pueden clasificar en dos tipos: pesadas y ágiles. (18)

Las metodologías pesadas están guiadas por una fuerte planificación durante todo el proceso de desarrollo. El proceso de desarrollo de software llevado a cabo a partir de este tipo de metodología es mucho más controlado, con cierta resistencia a posibles cambios, además de poseer numerosas políticas o normas. Existe un contrato prefijado y el cliente interactúa con el equipo de desarrollo mediante reuniones o entrevistas, los grupos de trabajos son grandes y posiblemente distribuidos por roles que posibilitan un mejor trabajo. La más conocida y usada de estas metodologías es el Proceso Racional Unificado (RUP).

Sin embargo este enfoque no resulta ser el más adecuado para proyectos donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo manteniendo una alta calidad. Como solución a esta dificultad surgen las metodologías ágiles.

La metodología ágil combina una filosofía y un conjunto de directrices de desarrollo, busca la satisfacción del cliente y la entrega temprana de software incremental, equipos de proyecto pequeños y con alta motivación; métodos informales; un

mínimo de productos de trabajo de la ingeniería del software; y una simplicidad general del desarrollo. Las directrices de desarrollo resaltan la entrega sobre el análisis y el diseño (aunque estas actividades se no se descartan), y la comunicación activa y continua entre los desarrolladores y los clientes. (19) Entre estas metodologías se destacan Programación Extrema (XP) y SCRUM.

Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Está especialmente indicada para proyectos con un rápido cambio de requisitos.

Algunas de sus principales características son: el desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de treinta días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. Otra característica importante es el intercambio constante con el cliente, donde destaca la reunión diaria del equipo de desarrollo para coordinación e integración.

Scrum es un complemento a las metodologías ágiles para el control, seguimiento y corrección de errores, cuyo objetivo primario es que toda la organización esté alineada en un mismo sentido ante este escenario. (20)

XP

XP está guiada por una rápida programación y se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y en la reutilización de código. También se enfoca en la realización de pruebas a los principales procesos con el objetivo de tratar de obtener los posibles errores futuros, esto conocido como pruebas unitarias, así como en la simplicidad de las soluciones implementadas y coraje para enfrentar los cambios. Impone un alto nivel de disciplina entre los programadores, lo cual permite mantener un mínimo nivel de documentación que a su vez se traduce en una gran velocidad de desarrollo. XP se define como especialmente adecuada para proyectos de corto plazo con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. Está formada por cuatro partes fundamentales que encierran sus características fundamentales, las cuales son: historia de usuarios, roles, procesos y prácticas.

Las historias de usuario son tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Estas tarjetas también presentan un análogo en formato digital facilitando el trabajo al equipo desarrollo.

XP define los siguientes roles: (21)

- Programador
- Cliente
- Encargado de pruebas
- Encargado de seguimiento
- Entrenador
- Consultor

- Gestor

El ciclo de desarrollo consiste en los siguientes pasos (21):

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

Luego de un análisis realizado se decidió que la metodología XP es la más adecuada a utilizar en el desarrollo de la aplicación, pues ella es empleada para proyectos de corto plazo, el equipo de trabajo cuenta con pocos integrantes por lo que la dimensión del proyecto es pequeña. Además posibilita tener un proceso de desarrollo motivado ya que no permite excesos de trabajos, propone una realimentación continua entre el cliente y el equipo de desarrollo, se consigue una mayor integración entre los miembros mediante la comunicación. Esta metodología está probada en disímiles proyectos y tiene suficiente soporte y documentación accesible en la red.

1.5 Lenguajes de programación.

PHP versión 5.3.10

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Además, es un lenguaje de programación de estilo clásico con variables, sentencias condicionales, bucles, funciones, etc.

PHP es un lenguaje multiplataforma. Presenta gran capacidad de conexión con la mayoría de los manejadores de base de datos que se utiliza en la actualidad y destaca su conectividad con MySQL. Posee gran capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones). Puede leer y manipular datos desde diversas fuentes. Permite las técnicas de programación orientada a objetos, así como crear los formularios para la web. Se presenta como una alternativa de fácil acceso, debido a que es libre. La mayoría de los sitios web publicados en internet se basan en este lenguaje de programación. (22)

Se escogió PHP como lenguaje de programación del lado del servidor para llevar a cabo el desarrollo de este sistema, principalmente por las ventajas y características que el presenta: (23)

- PHP es un lenguaje de programación web que es muy fácil de aprender, además de poseer una amplia documentación en su página oficial (Sitio Oficial), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.

- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

JavaScript versión 1.8

JavaScript es un lenguaje de programación que permite añadir interactividad a las páginas. Es ideal para agregar ciertas funciones rápidas a una página web. Es un lenguaje muy potente de alto nivel. Es cierto que no se puede hacer nada directamente al nivel de la máquina, pero es capaz de trabajar con muchas propiedades de los exploradores web, páginas web y, a veces, con el propio sistema donde se ejecuta el explorador. No necesita una fase de compilación como Java o C y el explorador no ha de cargar ninguna máquina virtual para ejecutar el código. Solo hay que crear el código y cargarlo. JavaScript también utiliza una arquitectura orientada a objetos parecida a la de Java o C++. También se trabaja con propiedades como las funciones del constructor o la estratificación en jerarquías. Con lo que hay más opciones para utilizar el código. (24)

Tipos de JavaScript: (24)

- JavaScript cliente. Se desarrolla y aplica en el navegador del cliente.
- JavaScript de servidor. Se desarrolla y ejecuta en el servidor, lo mismo que cualquier Interfaz de Entrada Común (CGI), pero con una sencillez que estos últimos desconocen.

JavaScript permite:

- Máxima interactividad entre el usuario y la página.
- Verificación de los datos introducidos por el usuario, antes de enviar el formulario al servidor.
- Ejecución de carritos de compra en el navegador.
- Ejecución de pequeñas cantidades de información al igual que en una base de datos.
- Manejo de applets y plug-ins dentro de múltiples marcos de HTML.
- Pre-procesado de información antes de enviarla al servidor.

Se selecciona este lenguaje por su sencillez y su compatibilidad con la mayoría de los navegadores. Además tiene la ventaja que mediante él se puede realizar gran parte de las funciones del cliente, lo que reduce la carga del servidor. JavaScript también es un lenguaje muy potente de alto nivel, es dinámico y responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página, puede cambiar totalmente el aspecto de la misma. Permite realizar cálculos en base a variables cuyo valor es determinado por el usuario.

JQUERY

jQuery es una biblioteca de java script, que puede ser usada en proyectos libres y privativos, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. Logra simplificar la manera de interactuar con los documentos HTML. Permite manejar eventos, desarrollar animaciones, y agregar interacción con la tecnología Ajax a páginas web. Entre las ventajas de utilizar esta librería se encuentran reducción y reutilización de código, código más conciso y de fácil lectura, desarrollo de componentes dinámicos que enriquecen las aplicaciones y fácil manejo de eventos y animaciones y. (25) Ésta permite:

- Acceder a las partes de una página. Ofrece un mecanismo selector robusto y eficaz para recuperar exactamente la parte del documento que se vaya a inspeccionar o manipular.
- Modificar la apariencia de una página. Proporciona el mismo soporte de estándares en todos los navegadores.
- Modificar el contenido de una página. Puede modificar el contenido de un documento en sí mismo con sólo pulsar unas teclas
- Responder a la interacción de un usuario con una página. Ofrece una forma elegante para interceptar una amplia variedad de eventos. Al mismo tiempo, su Interfaz de Programación de Aplicaciones (API) de gestión de eventos elimina inconsistencias de navegación que a menudo plagan los desarrolladores web.
- Agregar animación a una página. Proporciona información visual al usuario al ofrecer una serie de efectos como fundidos y cortinillas, así como un conjunto de herramientas para la elaboración de otros nuevos.
- Recuperar información de un servidor sin actualizar la página.
- Simplifica las tareas comunes de JavaScript. Ofrece mejoras en las construcciones básicas de JavaScript, como iteración y la manipulación de matrices. (25)

HTML

HTML es un lenguaje de marcas hipertextuales que está diseñado para estructurar textos para generar páginas web. El lenguaje HTML es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo. Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, vídeo, etc.

Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML. (26)

1.6 El Lenguaje de Modelado

Un modelo es una representación, en cierto medio, de algo en el mismo u otro medio. El modelo capta los aspectos importantes de lo que estamos modelando, desde cierto punto de vista, y simplifica u omite el resto. La ingeniería, la arquitectura y muchos otros campos creativos usan modelos. (27)

UML versión 2.1

El Lenguaje Unificado de Modelado (UML) permite modelar sistemas de información, y su objetivo es lograr modelos que, además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los clientes o usuarios de aquello que se modela. UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. No es un lenguaje de programación. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. (27).Es común utilizarlo para desarrollar diagramas en proyectos realizados con XP. (28)

UML es un método formal de modelado que aporta las siguientes ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa.

Las principales funciones de UML son:

- Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

1.7 Herramientas de software y patrón de arquitectura utilizados.

Symfony versión 2.1.0.

Symfony es un marco de trabajo que ofrece herramientas y clases que reducen el tiempo de desarrollo de una aplicación web compleja. Separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Es compatible con la mayoría de los gestores de bases de datos.

Es un completo framework php diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (29)

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.

- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Características de Symfony: (29)

- Escalable. Es infinitamente escalable si se disponen de los recursos necesarios.
- Probado. Probado con éxito durante años en varias aplicaciones gigantescas (Yahoo! Answers, Dailymotion, delicious) y en otros miles de sitios pequeños y medianos.
- Soporte. Sigue una política de tipo LTS (Long Term Support), por la que las versiones estables se mantienen durante 3 años sin cambios pero con una continua corrección de errores.
- Licencia. Se publica bajo licencia MIT, con la se puede desarrollar aplicaciones web comerciales, gratuitas y/o de software libre.
- Seguro. Permite controlar hasta el último acceso a la información e incluye por defecto protección contra ataques XSS y CSRF.
- Código. Desde su primera versión Symfony ha sido creado sólo para PHP 5, para obtener el máximo rendimiento de PHP y aprovechar todas sus características.
- Compromiso. Los creadores de Symfony no viven del framework, sino de las aplicaciones que desarrollan con él, por lo que les interesa tanto aspectos como el rendimiento, la buena documentación y el soporte.
- Documentado. Ya ha publicado cinco libros gratuitos de calidad y siempre actualizados. Además, toda la documentación está traducida al español.
- Calidad. Su código fuente incluye más de 9.000 pruebas unitarias y funcionales.
- Internacionalización. Está traducido a más de 40 idiomas e incluye todas las herramientas necesarias para traducir fácilmente las aplicaciones.

Arquitectura Modelo Vista Controlador

La Arquitectura Modelo Vista Controlador (MVC) es un patrón del diseño web formado por tres niveles: Modelo, Vista, Controlador. El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. La vista transforma el modelo en una página web que permite al usuario interactuar con ella. El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Es utilizado por Symfony para separar las distintas partes que forman una aplicación web.

La separación de capas como presentación, lógica de negocio, acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y fácilmente sustentables, lo que constituye un ahorro de tiempo en desarrollo en posteriores proyectos.

Al existir la separación de vistas, controladores y modelos es más sencillo realizar labores de mejora como:

- Agregar nuevas vistas.
- Agregar nuevas formas de recolectar las órdenes del usuario.
- Modificar los objetos de negocios bien sea para mejorar el performance o para migrar a otra tecnología.
- Las labores de mantenimiento también se simplifican y se reduce el tiempo necesario para ellas.

- Las vistas también son susceptibles de modificación sin necesidad de provocar que todo el sistema se paralice.

Herramienta Case para el modelado de UML.

Visual Paradigm versión 8.0

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. (30)

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos.

Se caracteriza por: (30)

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Capacidades de ingeniería directa e inversa.
- Licencia: gratuita y comercial.
- Fácil de instalar y actualizar.
- Generación de bases de datos.
- Transformación de diagramas de Entidad-Relación en tablas de base de datos.

Sistema gestor de base de datos

Los Sistemas Gestores de Base de Datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Estos permiten manejar con facilidad grandes volúmenes de información en muy poco tiempo y ofrecen independencia y seguridad en el tratamiento de información.

El propósito general de los sistemas de gestión de Base de Datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. Dentro de los SGBD más utilizados se encuentra MySQL, PostgreSQL y Oracle.

Oracle

Oracle es un sistema de base de datos relacional extremadamente potente y flexible. Esta potencia y flexibilidad, sin embargo, implican también una cierta complejidad. Para poder diseñar aplicaciones útiles basadas en Oracle es necesario entender como manipula los datos almacenados en el sistema. PL/SQL es una herramienta de gran importancia diseñada para la manipulación de datos, tanto internamente dentro de Oracle como externamente, en las propias aplicaciones. PL/SQL está disponible en diversos entornos, cada uno de los cuales tiene diferentes ventajas. Es una aplicación propietaria y sus precios son muy altos en el mercado. (31)

PostgreSQL

Es un SGBD objeto-relacional que se caracteriza por ser un potente motor de bases de datos, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos. Está considerado uno de los SGBD de código abierto más avanzado del mundo.

Entre sus principales características se pueden citar las siguientes: (32)

- Se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, Mac OS, Windows, entre otros.
- Documentación bien organizada, pública y libre.
- Cuenta con múltiples comunidades de desarrolladores muy activas.
- Altamente adaptable a las necesidades del cliente.
- Soporta distintos tipos de datos: además del soporte para los de tipo base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, entre otros.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, entre otras.

MySQL versión 5.5.20

Es un sistema desarrollado bajo la filosofía de código abierto, y su código fuente está disponible. Ideal en aplicaciones web donde hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante adelantar monitoreo sobre el desempeño para detectar y corregir errores tanto de SQL como de programación.

Este gestor de bases de datos es uno de los gestores más utilizado en el mundo del software libre, ya que ofrece una gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen muchas librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de tener una fácil instalación y configuración.

A continuación se presentan algunas de las ventajas de su utilización: (33)

- Acceso a las bases de datos: acceso de forma simultánea por varios usuarios y/o aplicaciones.
- Seguridad en forma de permisos y privilegios: determinados usuarios tendrán permisos para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos.
- Rapidez: se caracteriza por ser veloz, multihilo, multiusuario y robusto.
- Escalabilidad: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros.

- **Conectividad:** permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es usual que servidores Linux o Unix, usando MySQL, sirvan datos para ordenadores con sistema operativo Windows, Linux, Solaris, entre otros.

MySQL está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes. Es multihilo, con lo que puede beneficiarse de sistemas multiprocesador. Permite manejar multitud de tipos para columnas. Permite manejar registros de longitud fija o variable.

¿Por qué se utilizará MySQL? Para seleccionar el Sistema Gestor de Bases de Datos más indicado para usar en esta propuesta de solución se tuvo en cuenta que no fueran software propietario por los cuales hubiera que pagar una licencia para su utilización, lo cual limita el marco de elección entre MySQL y PostgreSQL. Por las características específicas del sistema a implementar se determinó que resultaría óptimo el uso de MySQL para la gestión de datos del sistema, debido a la poca complejidad de los procesos a realizar en la base de datos y la necesidad de rapidez al ejecutar las operaciones sobre los datos. Otro aspecto importante que se tuvo en cuenta es la facilidad de uso e instalación. MySQL viene incorporado en los paquetes más usados para hosting de sitios web profesionales, entiéndase WAMP, XAMPP y AppServ, lo que lo hace ideal para usuarios con poca experiencia en la instalación y configuración de Sistemas Gestores de Bases de Datos.

Netbeans versión 7.2.1

Tras la decisión de utilizar PHP como lenguaje de programación, se selecciona como IDE de desarrollo Netbeans. NetBeans es un entorno de desarrollo para que los programadores puedan escribir, compilar, depurar y ejecutar programas, escrito para Java pero puede servir para cualquier otro lenguaje de programación. Es un producto libre y gratuito sin restricciones de uso, fácil de instalar, utilizar y se ejecuta en muchos sistemas operativos como Windows y Linux. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida. Expande el soporte de los lenguajes dinámicos con apoyo para PHP5.3, así como el esquema de Symfony, y acelera el desarrollo de aplicaciones web con PHP.

Servidor Web Apache versión 2.2.21

El servidor Apache es el servicio que se encarga de resolver las peticiones de página de los clientes utilizando el protocolo de internet HTTP. Es un software que se estructura por módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Apache se adapta a una gran variedad de entornos a través de su diseño modular. (34)

Los módulos de Apache se pueden clasificar en tres categorías:

- **Módulos Base.** Módulo con las funcionalidades básicas de 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 las peticiones.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Características de Apache: (34)

- Flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos.
- Multiplataforma.

- Tecnología gratuita de código abierto.
- Servidor altamente configurable de diseño modular.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP.

Además de las características antes mencionadas, se selecciona Apache como servidor web porque es uno de los servidores web más utilizados, además es una tecnología gratuita de código fuente abierto que ofrece instalaciones sencillas para sitios pequeños como es el caso y si se requiere se puede expandir. Además, permite la creación de sitios web dinámicos utilizando PHP que es el lenguaje de programación seleccionado.

1.8 Conclusiones Parciales

En el presente capítulo se realizó un análisis de las características, estructura y funcionamiento de los sistemas de gestión de planificación docente, determinando que ninguno de ellos resuelve la problemática planteada en el presente trabajo. También se abordaron las tecnologías que permitirán el desarrollo del sistema a implementar teniendo en cuenta las tendencias actuales. Se definió que la construcción de la base de datos se realizará con MySQL, como lenguajes de programación se utilizarán JavaScript y PHP. Se trabajará haciendo uso del framework Symfony2 con el IDE de desarrollo Netbeans. La metodología de desarrollo seleccionada es XP con UML como lenguaje de modelado y como herramienta CASE el Visual Paradigm.

Capítulo 2: Descripción de la solución

2.1 Introducción

En este capítulo se describe la propuesta de solución del sistema, se definen los requerimientos funcionales y no funcionales. Además se detallan dos de las fases de la metodología seleccionada, las cuales son Exploración y Planificación, donde se confeccionan las Historias de Usuarios (HU) para cada iteración, definiendo los tiempos de entrega de cada una.

2.2 Objeto de estudio

Para la organización del proceso docente de la UCI se desarrollan un conjunto de actividades con vistas a realizar una efectiva planificación, ésta envuelve a toda una jerarquía integrada por jefes de departamentos, jefes de asignatura, vicedecanos de formación, profesores, entre otros. Entre las actividades más importantes de este proceso de planificación están el desarrollo del plan de trabajo, la gestión de los resultados obtenidos en los controles a clases y la evaluación trimestral de desempeño. La deficiencia de este proceso radica, entre otros aspectos, en que no prevé lo que pueda acarrear el surgimiento de una u otra actividad no planificada. Además su desarrollo de forma manual y descentralizada trae como inconvenientes afectaciones en la participación de los trabajadores en las actividades, falta de disponibilidad de documentación y/o informaciones y empleo de mucho tiempo de la jornada laboral para el desarrollo de los distintos planes de trabajo.

2.2.1 Descripción de los procesos vinculados al campo de acción.

Actualmente en la Facultad 2 el proceso de planificación mensual y anual de los trabajadores docentes se ve afectado por su ejecución manual y descentralizada, la cual trae un sin número de afectaciones para aquellos que se ven envueltos en ella. Además los profesores no cuentan con un medio para acceder a la documentación necesaria u obtener información sobre el proceso.

2.2.2 Descripción del flujo actual del proceso.

El flujo actual del proceso de planificación del proceso docente en la Facultad 2 se comporta la siguiente manera:

El vicedecano de formación envía el plan de trabajo de la facultad a los jefes de los departamentos docentes. Los jefes de departamento utilizan este plan de trabajo y elaboran el plan de trabajo mensual de su departamento el que es utilizado por los profesores para elaborar su plan de trabajo individual. Este plan de trabajo realizado por cada profesor es revisado por su jefe de departamento. Además, cada trabajador cuenta con un plan de trabajo anual que es elaborado por su jefe de departamento y luego revisado por el profesor y aprobado por ambos.

Por otro lado a todos los profesores se le realizan controles a clases durante el curso, en estos controles, además de la evaluación cualitativa, reciben un informe donde se plasma la valoración de cada indicador evaluado, los principales logros, las deficiencias y las recomendaciones realizadas por la o las personas que realizan el control. Otro proceso es la evaluación trimestral del profesor en la que se llena un modelo de evaluación a partir del trabajo realizado en cada trimestre. En este proceso se asignan evaluaciones de deficiente, adecuado o superior.

2.2.3 Objetos a informatizar.

Para llevar a cabo la gestión de la planificación de las actividades de los trabajadores docentes en la Facultad 2 existen varios procesos que deben ser informatizados, pues el manejo de estos datos de forma manual o vía correo resulta engorroso y muchas veces complicado. Esto trae consigo una serie de problemas que atentan contra la organización y divulgación de los mismos. Serán objetos de informatización los procesos vinculados a la realización de los diferentes planes de trabajo, evaluación de los controles a clases y evaluación trimestral.

2.3 Propuesta de solución.

Después del análisis de los procesos expuestos anteriormente se ha llegado a la conclusión que existen varios flujos de actividades que necesitan ser informatizados de manera que se contrarreste la descentralización que sufren debido a su creación manual.

Como propuesta de solución, fundamentada en esta investigación, se plantea la creación de un sistema web desarrollado mediante el framework Symfony, que permita concentrar toda la información referente a la planificación de las actividades docentes de la Facultad 2. El mismo debe permitir un manejo adecuado y una gestión eficiente de la información referente a estos procesos. Incluyendo funcionalidades como son las publicaciones de artículos interesantes para la comunidad de profesores, la gestión de los diferentes planes de trabajo, las evaluaciones trimestrales de los profesores y los resultados de los controles a clases. Se desarrollan otras como el envío de notificaciones vía correo a los diferentes profesores que pertenecen al sistema. En el sistema se habilitan cinco tipos de usuarios con roles diferentes los cuales van a ser los encargados de formalizar toda la información referente al sistema, limitar el acceso, dar soporte, mantenimiento e interactuar con este.

2.3.1 Personas relacionadas con la aplicación

Son aquellas personas que interactúan con la aplicación y obtienen un resultado del valor de uno o varios procesos que se ejecutan en la misma. Los roles que pueden desempeñar los usuarios en la aplicación se muestran a continuación así como una breve descripción.

| Roles que pueden desempeñar los usuarios | Justificación |
|--|--|
| Administrador | Además de gestionar la información del sistema tiene entre sus obligaciones darle soporte y mantenimiento a la aplicación. |
| Profesor | Es el encargado de gestionar el plan de trabajo individual. |
| Jefe de departamento | Gestiona el plan de trabajo mensual de su departamento al igual que el plan de trabajo anual de los profesores, sus evaluaciones trimestrales y los resultados obtenidos por los mismos en los controles a clases. |
| Jefe de asignatura | Gestiona evaluación de control a clases. |

Vicedecano de formación

Gestiona el plan de trabajo mensual de la facultad. Además puede observar toda la información existente en el sistema sobre el proceso de planificación de las actividades docente de los profesores.

Tabla 2.1. Personas relacionadas con el sistema.

2.3.2 Lista de reserva del producto

Los requisitos funcionales son las características o funcionalidades que el sistema debe cumplir. Estos, en la metodología XP, son abarcados en la Lista de Reserva del Producto (LRP) donde aparecen ordenados según la prioridad de implementación. Los requisitos no funcionales también son abarcados en LRP.

Requisitos funcionales

RF1 Autenticar usuario:

El sistema debe identificar los usuarios, y permitir el acceso de los mismos una vez identificados.

- Usuario
- Contraseña

RF2 Insertar usuario:

El sistema debe permitir añadir un nuevo usuario, el cual ocupará determinado rol.

- Nombre
- Usuario
- CI
- Solapín
- Categoría docente
- Cargo
- Correo
- Rol (rol que ocupa en el sistema)
- Área (departamento docente al cual pertenece el usuario)

RF3 Modificar usuario:

El sistema debe ser capaz de permitir al administrador modificar los datos (Remitirse a RF2) de un usuario.

RF4 Eliminar usuario:

El sistema debe brindar la opción al administrador de eliminar un usuario ya creado.

RF5 Mostrar usuario:

El sistema debe permitir ver los datos (Remitirse a RF2) que presenta un usuario determinado.

RF6 Cambiar rol del usuario:

El sistema debe ser capaz de permitir al administrador cambiar el rol de un usuario.

- Rol (rol que ocupará el usuario en el sistema)

RF7 Insertar rol:

El sistema debe permitir al administrador añadir un nuevo rol.

- Nombre del rol

RF8 Modificar rol:

El sistema debe ser capaz de permitir al administrador modificar el nombre del rol una vez creado.

- Nombre del rol

RF9 Mostrar rol:

El sistema debe permitir ver un rol determinado.

RF10 Eliminar rol:

El sistema debe brindar la opción al administrador de eliminar un rol ya creado.

RF11 Insertar plan de trabajo mensual de la facultad:

El sistema debe permitir a un usuario con los permisos necesarios añadir un nuevo plan de trabajo mensual de la facultad.

- Mes
- Facultad
- Curso
- Actividades(Grupo de actividades al que pertenece , actividad, hora, lugar, día o duración, dirige, participan)

RF12 Modificar plan de trabajo mensual de la facultad:

El sistema debe permitir a un usuario con los permisos necesarios modificar un plan de trabajo mensual de la facultad. Datos a modificar (Remitirse a RF11).

RF13 Buscar plan de trabajo mensual de la facultad:

El sistema debe permitir a un usuario con los permisos necesarios ver los datos (Remitirse a RF11) de un plan de trabajo mensual de la facultad.

RF14 Eliminar plan de trabajo mensual de la facultad:

El sistema debe brindar a un usuario con los permisos necesarios la opción de eliminar un plan de trabajo mensual de la facultad ya creado.

RF15 Publicar plan de trabajo mensual de la facultad:

El sistema debe brindar a un usuario con los permisos necesarios la opción de publicar un plan de trabajo mensual de la facultad.

RF16 Enviar notificación de plan de trabajo mensual de la facultad:

El sistema debe emitir un aviso a todos los usuarios con el rol de jefe de departamento a través de una notificación al correo electrónico.

RF17 Insertar plan de trabajo mensual del departamento:

El sistema debe permitir a un usuario con los permisos necesarios añadir un nuevo plan de trabajo mensual del departamento.

- Mes
- Curso
- Actividades(Actividad, hora de inicio, hora del fin, día, dirige, participan, observaciones)

RF18 Modificar plan de trabajo mensual del departamento:

El sistema debe permitir a un usuario con los permisos necesarios modificar los datos (Remitirse a RF17) de un plan de trabajo mensual del departamento.

RF19 Eliminar plan de trabajo mensual del departamento:

El sistema debe brindar la opción de eliminar un plan de trabajo mensual del departamento ya creado.

RF20 Buscar plan de trabajo mensual del departamento.

El sistema debe brindar a un usuario con los permisos necesarios la opción de ver un plan de trabajo mensual del departamento.

RF21 Publicar plan de trabajo mensual del departamento.

El sistema debe brindarle al usuario con los permisos necesarios la opción de publicar un plan de trabajo mensual del departamento.

RF22 Enviar notificación a profesores del departamento.

El sistema debe emitir un aviso a todos los profesores pertenecientes al departamento a través de una notificación al correo electrónico.

RF23 Insertar plan de trabajo mensual individual.

El sistema debe permitir a un usuario con los permisos necesarios añadir un nuevo plan de trabajo mensual individual.

- Mes
- Curso
- Cuantitativo
- Cualitativo
- Observaciones generales
- Actividad (Descripción, día, hora de inicio, hora fin)
- Tarea (Descripción)

RF24 Modificar plan de trabajo mensual individual.

El sistema debe permitir a un usuario con los permisos necesarios modificar los datos (Remitirse a RF24) de un plan de trabajo mensual individual.

RF25 Eliminar plan de trabajo mensual individual.

El sistema debe permitir a un usuario con los permisos necesarios eliminar un plan de trabajo mensual individual ya existente.

RF26 Buscar plan de trabajo mensual individual.

El sistema debe permitir a un usuario con los permisos necesarios ver un plan de trabajo mensual individual determinado.

RF27 Publicar plan de trabajo mensual individual.

El sistema debe brindar a un usuario con los permisos necesarios la opción de publicar un plan de trabajo mensual individual.

RF28 Enviar notificación de plan de trabajo individual.

El sistema debe emitir un aviso al jefe del departamento al cual pertenece el usuario que creo el plan individual a través de una notificación al correo electrónico.

RF29 Insertar plan de trabajo anual.

El sistema debe permitir a un usuario con los permisos necesarios añadir un nuevo plan de trabajo anual.

- Curso
- Actividad anual (Índice, trimestre, descripción)

RF30 Modificar plan de trabajo anual.

El sistema debe permitir a un usuario con los permisos necesarios modificar los datos (Remitirse a RF29) de un plan de trabajo anual ya existente.

RF31 Eliminar plan de trabajo anual.

El sistema debe permitir a un usuario con los permisos necesarios eliminar un plan de trabajo anual.

RF32 Buscar plan de trabajo anual.

El sistema debe permitir a un usuario con los permisos necesarios ver un plan de trabajo anual.

RF33 Publicar plan de trabajo anual.

El sistema debe brindar a un usuario con los permisos necesarios la opción de publicar un plan de trabajo anual.

RF34 Enviar notificación de plan de trabajo anual.

El sistema debe emitir un aviso al profesor al cual pertenece el plan de trabajo anual a través de una notificación al correo electrónico.

RF35 Insertar evaluación trimestral.

El sistema debe permitir a un usuario con los permisos necesarios añadir una nueva evaluación trimestral.

- Trimestre
- Evaluación
- Curso
- Observaciones
- Observaciones generales
- Tarea asignada(Tarea, evaluación, puntos que trata)

Impuntualidades y ausencias

- Clases
- Colectivo de año
- Sesiones de trabajo metodológico
- Visita a beca
- Reuniones de facultad
- Reunión trabajo educativo
- Reuniones UCI

Actividades de custodia en el centro

- Ausencias
- Asistencia

RF36 Modificar evaluación trimestral.

El sistema debe permitir a un usuario con los permisos necesarios modificar los datos (Remitirse a RF35) de una evaluación trimestral.

RF37 Eliminar evaluación trimestral.

El sistema debe permitir a un usuario con los permisos necesarios eliminar una evaluación trimestral.

RF38 Buscar evaluación trimestral.

El sistema debe permitir a un usuario con los permisos necesarios ver una evaluación trimestral.

RF39 Publicar evaluación trimestral.

El sistema debe brindar a un usuario con los permisos necesarios la opción de publicar una evaluación trimestral.

RF40 Enviar notificación de evaluación trimestral.

El sistema debe emitir un aviso al profesor, al cual pertenece la evaluación trimestral, a través de una notificación al correo electrónico.

RF41 Insertar resultados de control a clase.

El sistema debe permitir a un usuario con los permisos necesarios añadir un nuevo control a clase.

- Asignatura
- Grupo

- Título de la clase
- Tipo de clase
- Visitante 1
- Categoría 1
- Visitante 2
- Categoría 2
- Principales logros
- Deficiencias
- Recomendaciones
- Evaluación
- Fecha

RF42 Modificar resultados de control a clase.

El sistema debe permitir a un usuario con los permisos necesarios modificar los datos (Remitirse a RF41) de un control a clase.

RF43 Eliminar resultados de control a clase.

El sistema debe permitir a un usuario con los permisos necesarios eliminar un control a clase.

RF44 Buscar resultados de control a clases.

El sistema debe permitir a un usuario con los permisos necesarios ver un control a clases.

RF45 Publicar resultados de control a clase.

El sistema debe brindar a un usuario con los permisos necesarios la opción de publicar los resultados de un control a clase.

RF46 Enviar notificación de resultados de control a clase.

El sistema debe emitir un aviso al profesor al cual pertenece el control a clase a través de una notificación al correo electrónico.

RF47 Insertar área.

El sistema debe permitir al administrador añadir una nueva área.

- Nombre
- Facultad

RF48 Modificar área.

El sistema debe permitir al administrador modificar los datos (Remitirse a RF47) de un área.

RF49 Eliminar área.

El sistema debe permitir al administrador eliminar un área ya creada.

RF50 Mostrar área.

El sistema debe permitir al administrador mostrar las áreas existentes.

RF51 Insertar artículos.

El sistema debe permitir a un usuario añadir un nuevo artículo.

- Título
- Resumen
- Contenido

RF52 Modificar artículos.

El sistema debe permitir a un usuario modificar los datos (Remitirse a RF51) de un artículo.

RF53 Mostrar artículos.

El sistema debe permitir a un usuario mostrar los datos (Remitirse a RF51) de un artículo.

RF54 Eliminar artículos.

El sistema debe brindar la opción de eliminar un artículo ya creado

RF55 Publicar artículos.

El sistema debe permitir al administrador publicar un artículo.

Requisitos no funcionales

Los requisitos no funcionales del sistema no se capturan en la metodología XP en forma de historias de usuario, puesto que los clientes generalmente no conocen la terminología técnica que se utiliza para describir este tipo de requisitos.

La captura de requisitos no funcionales es por tanto una tarea del equipo de desarrollo completo, que haciendo un intercambio profundo de ideas con el cliente, determinan las mejores soluciones para crear un sistema que además de implementar correctamente las historias de usuario, cumpla con determinados estándares de calidad y con las características propias del negocio que se pretende automatizar.

Usabilidad

La aplicación debe ser diseñada de forma tal, que los usuarios que hagan uso de la misma sean capaces de aprender su funcionamiento en el menor tiempo posible.

Confidencialidad

Toda la información manejada en el sistema está protegida de acceso no autorizado. El usuario se autentica con los datos del dominio UCI, en caso contrario la aplicación debe mostrar un error de autenticación.

Integridad

La integridad se refiere a la corrección y completitud de los datos en la base de datos, garantizando la calidad de los mismos. Se debe garantizar que al eliminar o actualizar un tipo de contenido se modifiquen igualmente los asociados a él. El sistema no podrá dar acceso a usuarios que no estén autenticados en el mismo.

Restricciones de diseño

Son aquellos que especifican o restringen la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente. Se deben realizar validaciones de las entradas de los datos para que el usuario cometa el mínimo de errores, mostrar mensajes de envío y hacer uso del estándar de codificación que propone el framework Symfony.

Interfaz

El diseño de la interfaz visual no debe tener animaciones ni imágenes pesadas, que repercutan en la rapidez de la aplicación. Debe ser intuitiva para los usuarios que interactúan con la aplicación, permitiendo el fácil entendimiento de las

funcionalidades que el mismo brinda, además de poseer colores refrescantes para una mejor interacción entre el usuario y la aplicación.

Interfaz externa

Es importante destacar que no se trata del diseño de la interfaz en detalle, sino que especifican cómo se pretende que sea la interfaz externa del producto. Se podrá realizar las funciones en cualquier página pudiendo llegar siempre a la página principal. A todas las funcionalidades se podrá acceder desde cualquier parte del sistema. El color de los textos debe contrastar con el del fondo y el tamaño de fuente debe ser lo suficientemente grande para que la información contenida en la misma sea fácil de leer.

Requisitos de software

En las computadoras de los usuarios solo se requiere un navegador Web moderno (Mozilla Firefox 2 o superior recomendado), pueden usar cualquier sistema operativo. En el servidor de base de datos se requiere de Windows NT en adelante o cualquier distribución de Linux. Es necesario un gestor de base de dato MySQL 5.0. En el servidor Web se requiere Apache 2 o superior y PHP 5.2 con las extensiones php_mysql, php_pdo, php_pdo_mysql y php_ldap activadas.

Requisitos de hardware

En el cliente se requiere una máquina con 128 MB de RAM como mínimo, el servidor web junto con el servidor de base de datos debe tener 256 MB de RAM y 20 GB de disco duro mínimo y todas las máquinas implicadas en el funcionamiento de la aplicación deben estar conectadas a la red.

Portabilidad

Son aquellos que especifican los atributos que debe presentar el software para facilitar su traslado a otras plataformas o entornos de desarrollo. Permitir que el sistema se ejecute sobre el Sistema Operativo Windows XP o superior y GNU/LINUX.

Rendimiento

La aplicación debe ejecutarse con el mínimo de recursos para el que fue prevista y además el tiempo de respuesta a las diferentes peticiones de los usuarios debe ser menor de 30 segundos.

2.3.3 Fase de exploración

La fase de exploración es la primera etapa del ciclo de vida de proyecto realizado con XP, esta recoge las HU las cuales describen las funcionalidades del sistema. Además se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología

Historias de Usuario

Las HU son el primer paso de cualquier proyecto que siga la metodología XP. Tienen la misma finalidad que los casos de uso (CU) utilizados por la metodología RUP, pero con la diferencia que constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin muchos detalles, existiendo diferencias entre éstas y la especificación de requisitos, como por ejemplo el nivel de detalles, ya que las HU solo contienen información sobre la estimación del riesgo y el tiempo que requerirá su implementación.

Clasificación de las Historias de Usuario

Las HU están compuestas por tablas divididas en las siguientes secciones:

Número: número de la HU.

Nombre: nombre que identifica la HU.

Usuario: nombre de usuarios involucrados en la HU.

Prioridad en el negocio: se clasifican en alta (funcionalidades fundamentales en el desarrollo del sistema), media (funcionalidades a tener en cuenta pero que no tienen una afectación sobre el sistema) y baja (sirven de ayuda a los desarrolladores pero no tienen nada que ver con el sistema).

Nivel de complejidad: también se clasifican en alta (un error en la implementación, lleva a la inoperatividad del código), media (un error en la implementación retrasa la entrega de la versión) y baja (el error se puede tratar sin que traiga problemas mayores)

Estimación: tiempo estimado que se demorará el desarrollo de la HU.

Iteración asignada: número de la iteración en la que será realizada la HU.

Descripción: breve descripción de la HU.

Observaciones: a qué requisito hace referencia la HU.

Durante este proceso se identifican 11 HU. A continuación se muestran las mismas ya desarrolladas.

| Historia de Usuario | |
|---|------------------------------------|
| Número: 1 | Nombre: Autenticar usuario |
| Usuario: Administrador, Vicedecano de formación, Jefe departamento, Jefe de asignatura, Profesor | |
| Prioridad: Alta | Nivel de complejidad: Media |

| | |
|--|------------------------------|
| Estimación: 1 semana | Iteración asignada: 1 |
| Descripción: El usuario introduce sus datos en el formulario, el sistema verifica que los datos entrados sean correctos y que se trate de un usuario con privilegios. | |
| Observaciones: Los usuarios involucrados se autentican en el dominio UCI. | |

Tabla 2.2 HU Autenticar usuario.

| Historia de Usuario | |
|--|------------------------------------|
| Número: 2 | Nombre: Gestionar usuario |
| Usuario: Administrador | |
| Prioridad: Alta | Nivel de complejidad: Media |
| Estimación: 1 semana | Iteración asignada: 1 |
| Descripción: El sistema debe permitir las siguientes funcionalidades: insertar, modificar, eliminar, mostrar y cambiar rol del usuario. | |
| Observaciones: El sistema debe insertar automáticamente a un nuevo usuario que cumpla con los requisitos validos que se obtienen del servidor LDAP de la UCI, otra forma puede ser por la vía del administrador el cual tiene los permisos de insertar, modificar, mostrar, eliminar un usuario | |

Tabla 2.3 HU Gestionar usuario.

| Historia de Usuario | |
|---|------------------------------------|
| Número: 3 | Nombre: Gestionar rol |
| Usuario: Administrador | |
| Prioridad: Alta | Nivel de complejidad: Media |
| Estimación: 1 semana | Iteración asignada: 1 |
| Descripción: El sistema debe permitir realizar las siguientes funcionalidades: insertar, modificar, mostrar y eliminar rol. | |
| Observaciones: Los roles que los usuarios pueden desempeñar son los siguientes: Administrador, Vicedecano de formación, Jefe de departamento, Jefe de asignatura y | |

Profesor.

Tabla 2.4 HU Gestionar rol.

| Historia de Usuario | |
|--|---|
| Número: 4 | Nombre: Gestionar plan de trabajo mensual de la facultad |
| Usuario: Vicedecano de formación | |
| Prioridad: Alta | Nivel de complejidad: Alta |
| Estimación: 1 semana | Iteración asignada: 2 |
| Descripción: El sistema debe permitir al vicedecano de formación y al administrador realizar las siguientes funcionalidades: insertar, modificar, eliminar y buscar el plan de trabajo mensual de la facultad así como darle la opción de publicar en el sistema. Además cuando el plan de trabajo sea publicado el sistema debe de enviar notificación vía correo a todos los jefes de departamento. | |
| Observaciones: El plan de la facultad solo podrá ser visto por los jefes de departamento una vez publicado. | |

Tabla 2.5 HU Gestionar plan de trabajo mensual de la facultad.

| Historia de Usuario | |
|--|---|
| Número: 5 | Nombre: Gestionar plan de trabajo mensual del departamento |
| Usuario: Jefe de departamento | |
| Prioridad: Alta | Nivel de complejidad: Media |
| Estimación: 1 semana | Iteración asignada: 2 |
| Descripción: El sistema debe permitir al jefe del departamento y al administrador realizar las siguientes funcionalidades: insertar, eliminar, modificar, buscar y publicar el plan de trabajo mensual del departamento, además debe permitir enviar notificación vía correo de la publicación del plan a los profesores del departamento al que pertenece el plan. | |
| Observaciones: Una vez publicado el plan de trabajo mensual del departamento se crean automáticamente los planes individuales de los profesores pertenecientes al mismo, estos planes creados contienen a su vez las actividades docentes. Los profesores del | |

departamento y también el vicedecano de formación podrán acceder a ver el plan de trabajo mensual publicado.

Tabla 2.6 HU Gestionar plan de trabajo mensual del departamento

| Historia de Usuario | |
|---|---|
| Número: 6 | Nombre: Gestionar plan de trabajo mensual individual |
| Usuario: Profesor | |
| Prioridad: Alta | Nivel de complejidad: Media |
| Estimación: 1 semana | Iteración asignada: 2 |
| Descripción: El sistema debe permitir al profesor y al administrador realizar las siguientes funcionalidades: insertar, modificar, eliminar, buscar y publicar el plan de trabajo mensual individual, además debe permitir enviar notificación vía correo de la publicación al jefe de departamento. | |
| Observaciones: La funcionalidad insertar se hace automáticamente una vez creado el plan del departamento, pero en caso de fallo el usuario si tiene los permisos requerido también puede insertar el plan individual. El jefe de departamento y el vicedecano de formación podrán acceder a ver el plan de trabajo mensual individual una vez publicado. | |

Tabla 2.7 HU Gestionar plan de trabajo mensual individual.

| Historia de Usuario | |
|--|--|
| Número: 7 | Nombre: Gestionar plan de trabajo anual |
| Usuario: Jefe de departamento | |
| Prioridad: Alta | Nivel de complejidad: Media |
| Estimación: 1 semana | Iteración asignada: 2 |
| Descripción: El sistema debe permitir al jefe de departamento y al administrador realizar las siguientes funcionalidades: insertar, modificar, eliminar, buscar y publicar el plan de trabajo anual, además debe permitir enviar notificación vía correo de la publicación del plan al profesor que pertenezca. | |

Observaciones: El profesor y el vicedecano de formación pueden acceder a ver el plan de trabajo anual una vez publicado.

Tabla 2.8 HU Gestionar plan de trabajo anual.

| Historia de Usuario | |
|--|--|
| Número: 8 | Nombre: Gestionar evaluación trimestral |
| Usuario: Jefe de departamento | |
| Prioridad: Alta | Nivel de complejidad: Media |
| Estimación: 1 semana | Iteración asignada: 3 |
| Descripción: El sistema debe permitir al jefe de departamento y al administrador realizar las siguientes funcionalidades: insertar, modificar, eliminar, buscar y publicar la evaluación trimestral de un profesor, además debe permitir enviar notificación vía correo de la publicación de la evaluación al profesor. | |
| Observaciones: El profesor y el vicedecano de formación pueden acceder a ver la evaluación trimestral una vez publicada. Previo a esto el profesor debe de enviar una autoevaluación a su jefe de departamento. | |

Tabla 2.9 HU Gestionar evaluación trimestral.

| Historia de Usuario | |
|--|---|
| Número: 9 | Nombre: Gestionar resultados de control a clases |
| Usuario: Jefe de asignatura, jefe de departamento | |
| Prioridad: Alta | Nivel de complejidad: Media |
| Estimación: 1 semana | Iteración asignada: 3 |
| Descripción: El sistema debe permitir al jefe de departamento, al jefe de asignatura y al administrador realizar las siguientes funcionalidades: insertar, modificar, eliminar, publicar y buscar los resultados obtenidos por los profesores en los controles a clases, además debe permitir enviar notificación de la publicación de la evaluación del control vía correo al profesor visitado. | |
| Observaciones: El profesor y también el vicedecano de formación pueden acceder a ver | |

el control a clase una vez publicado.

Tabla 2.10 HU Gestionar resultados de control a clases.

| Historia de Usuario | |
|---|-----------------------------------|
| Número: 10 | Nombre: Gestionar área |
| Usuario: Administrador | |
| Prioridad: Media | Nivel de complejidad: Baja |
| Estimación: 1 semana | Iteración asignada: 3 |
| Descripción: El sistema debe permitir insertar, modificar, eliminar y mostrar las áreas (departamentos docentes de la facultad). | |
| Observaciones: | |

Tabla 2.11 HU Gestionar área.

| Historia de Usuario | |
|---|------------------------------------|
| Número: 11 | Nombre: Gestionar artículos |
| Usuario: Administrador, Vicedecano de formación, Jefe departamento, Jefe de asignatura, Profesor | |
| Prioridad: Baja | Nivel de complejidad: Baja |
| Estimación: 1 semana | Iteración asignada: 3 |
| Descripción: El sistema debe permitir realizar las siguientes funcionalidades: insertar, modificar, eliminar, mostrar y publicar artículo. | |
| Observaciones: Los artículos, una vez creados, son revisados por el administrador el cual decide si se publica o no en el sitio. | |

Tabla 2.12 HU Gestionar artículos

2.3.4 Planificación

Durante la planificación el cliente establece la prioridad de cada HU las cuales son organizadas en iteraciones. Además se acuerda el alcance de la entrega y se realiza una estimación del esfuerzo que costará implementar cada HU.

Plan de iteraciones

La metodología XP se divide en iteraciones relativamente cortas. Las HU son divididas en tareas de entre 1 y 3 días que son asignadas a los programadores, creándose el plan de iteraciones donde se especifican el orden y la duración. Para el desarrollo del sistema propuesto se han definido 3 iteraciones, las cuales se describen a continuación:

| Iteraciones | Orden de las Historias de Usuario a implementar |
|-------------|--|
| 1 | La primera iteración del desarrollo del sistema tendrá como objetivo, la implementación de las funcionalidades básicas para la puesta en marcha del mismo. Se tendrán en cuenta las historias de usuario 1, 2 y 3. De manera general al finalizar esta iteración se habrá podido cumplir con funcionalidades como la gestión de usuarios con sus respectivos roles y permisos, la autenticación de los mismos, obteniéndose la primera versión del software. |
| 2 | Esta iteración está centrada en la implementación de las HU que tienen prioridad alta para el cliente; estas son las números 4, 5, 6 y 7. Al finalizar esta iteración se habrán desarrollado las funcionalidades gestionar plan de trabajo mensual de la facultad, gestionar plan de trabajo mensual del departamento, gestionar plan de trabajo mensual individual y gestionar plan de trabajo anual. |
| 3 | La tercera iteración prioriza la implementación de las HU que tienen prioridad media y baja para el cliente; estas son las números 8, 9, 10 y 11. Gestionar evaluación trimestral, gestionar controles a clases, gestionar área y gestionar artículo. Finalizando esta iteración se tendrá una versión 1.0 del producto final, donde a partir de ese momento, el sistema será puesto a pruebas por un período de tiempo para evaluar su desempeño. |

Tabla 2.13. Plan de iteraciones

Planificación de la duración de iteraciones.

Para conocer el tiempo real de la duración de las iteraciones del proyecto, así como las HU que serán implementadas en cada una de las iteraciones, la metodología XP propone la creación del Plan de Duración de las Iteraciones. Su propósito fundamental es conocer el tiempo que abarca cada iteración y el orden en que son implementadas las HU dentro de cada período.

| Iteraciones | Orden de las Historias de Usuario a implementar | Cantidad de tiempo de trabajo |
|-------------|---|-------------------------------|
| 1 | <ol style="list-style-type: none">1. Autenticar usuario2. Gestionar usuario3. Gestionar rol | 1 semana |

| | | |
|----------|---|-----------|
| 2 | <ul style="list-style-type: none"> 4. Gestionar plan de trabajo mensual de la facultad 5. Gestionar plan de trabajo mensual del departamento 6. Gestionar plan de trabajo mensual individual 7. Gestionar plan de trabajo anual | 4 semanas |
| 3 | <ul style="list-style-type: none"> 8. Gestionar evaluación trimestral de profesores 9. Gestionar resultados de control a clases 10. Gestionar área 11. Gestionar artículos | 3 semanas |

Tabla 2.14: Plan de duración de Iteraciones

Planificación de las Entregas

En el plan de entrega se agrupan las HU de manera ordenada con el objetivo de conformar una entrega. El plan se conforma en base a las estimaciones de tiempo que emplean los programadores en el desarrollo de las funcionalidades.

| Sistema | Final iteración 1 | Final iteración 2 | Final iteración 3 |
|--------------------|-------------------|-------------------|-------------------|
| Versión 1.0 | 7/04/2013 | 7/05/2013 | 28/05/2013 |

Tabla 2.15: Planificación de las Entregas

2.4 Conclusiones Parciales

En este capítulo se plasman las bases para el desarrollo del sistema. Se definen los requisitos funcionales y no funcionales a tener en cuenta, obteniéndose así una idea general del funcionamiento de la aplicación. Se obtiene una planificación del tiempo de desarrollo de las iteraciones y el orden en que se implementan las historias de usuario de acuerdo con la prioridad que le es asignada.

Capítulo 3: Diseño e implementación del sistema

3.1 Introducción

En el presente capítulo se analiza la arquitectura de software, según el framework de desarrollo Symfony, utilizado para la confección del sistema que se presenta. Se describe el proceso de diseño de la aplicación a través de varios artefactos propuestos por la metodología XP. Son confeccionadas las tarjetas Clase-Responsabilidad-Colaboración (CRC) y las tareas de ingeniería. Es descrito el modelo físico de la base de datos y el diagrama de despliegue del sistema.

3.2 Arquitectura del sistema

En la construcción de la aplicación se utiliza la arquitectura cliente/servidor. En esta arquitectura las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario. Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente. (35)

3.3 Diseño

Patrones de diseño

Symfony es un completo framework 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. Además, automatiza las tareas más comunes, permitiendo 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 nueva aplicación web. (36). Seguidamente se mencionan los patrones más significativos que utiliza Symfony.

Modelo Vista Controlador

Symfony está basado en el patrón de diseño web conocido como arquitectura MVC, que está formado por tres niveles:

- La capa **Modelo** define la lógica de negocio (la base de datos pertenece a esta capa).
- La **Vista** es con lo que el usuario interactúa.
- El **Controlador** es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente.

A continuación se presenta el funcionamiento interno de Symfony 2.

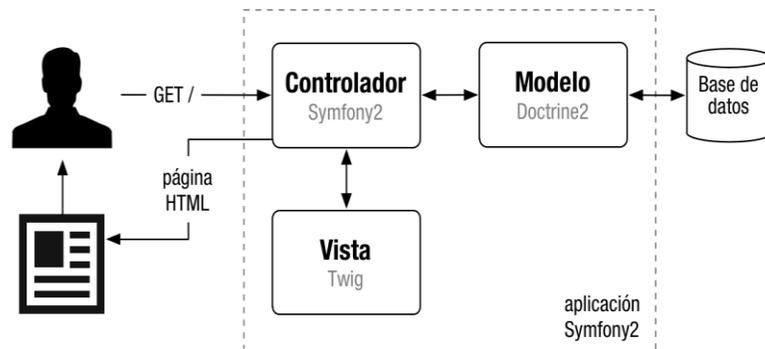


Figura 1. Patrón MVC

El patrón representa un mecanismo de mejora de procesos de desarrollo de software, fácil de comprender y aplicar. Permite obtener un sistema claro y bien especificado. La separación de la interfaz del resto del código elimina problemas como:

- Que a la complejidad de los cálculos del programa se le suma la complejidad de la interfaz de usuario, lo que proporcionaría un código de mayor calidad y de menor dificultad.
- Menor rigidez en la interfaz con el usuario, dado que el código HTML no estaría entremezclado con el lenguaje de programación y sería una tarea más fácil cambiar el diseño de la interfaz.

Patrones GRASP (General Responsibility Assignment Software Patterns)

Experto: Se evidencia este patrón puesto que Doctrine es la librería externa que utiliza Symfony 2 para realizar su capa de abstracción al modelo de datos, encapsulando toda la lógica de los datos y generando las clases con funcionalidades comunes de las entidades. Por tanto, cada clase creada por Doctrine a partir de una entidad es experta en manejar su información.

Controlador: Todas las peticiones web son manejadas por un solo controlador frontal (Dispatcher), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

Creador: El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que:

- Tiene la información necesaria para realizar la creación del objeto.
- Usa directamente las instancias creadas del objeto.
- Almacena o maneja varias instancias de la clase.
- Contiene o agrega la clase. (37)

El uso de este patrón se evidencia en las clases controladoras como ControlClase, permitiendo el acceso a la información almacenada en la clase entidad.

Bajo Acoplamiento: El bajo acoplamiento fomenta el aumento de la reutilización y la eliminación de las redundancias, creando clases más independientes y con mayor resistencia al impacto de los cambios, que aumentan la productividad y la posibilidad de reutilización. Este patrón se basa en la idea de tener las clases lo menos ligadas entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Alta cohesión: La cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase, además de que una alta cohesión garantiza que clases con responsabilidades estrechamente relacionadas no realicen un trabajo enorme, y que cada elemento de nuestro diseño debe realizar una labor única dentro del sistema.

La herramienta destinada a la gestión de actividades docentes sigue los principios de alta cohesión ya que no se hace uso de clases saturadas de métodos.

Patrones GOF (Gang of Four)

Abstrac Factory (Fábrica Abstracta): Se utiliza este patrón al trabajar con objetos de distintas familias de manera que no se mezclen entre sí, haciendo transparente el tipo de familia concreta que se esté usando. Cuando el marco de trabajo necesita, por ejemplo crear un nuevo objeto, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

Singlenton (Instancia Única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. El acceso a la Base de Datos en el sistema será mediante una única instancia para así impedir que se creen múltiples conexiones innecesarias.

Decorator (Decorador): Añade funcionalidad a una clase, dinámicamente. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el Layout decorando la misma.

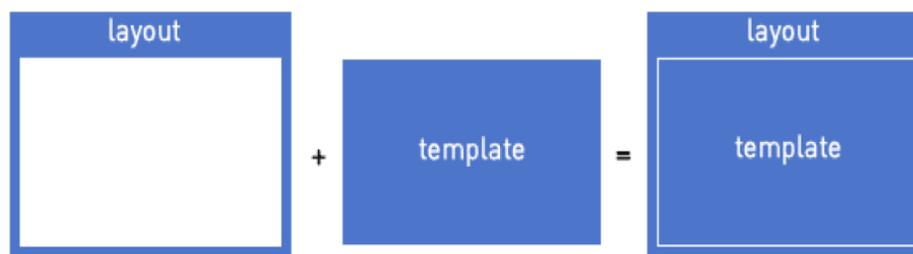


Figura 2. Patrón Decorator

Data Mapper: Este patrón pertenece a la capa de persistencia. Data mapper, actúa como nexo entre objetos en memoria y su representación en la base de datos, siendo responsable de la transferencia de datos entre ambos lugares. Este patrón es implementado en el ORM Doctrine 2, el cual es utilizado por Symfony 2 para la gestión de sus datos.

3.3.1 Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) se dividen en tres secciones que contienen la información del nombre de las clases, sus responsabilidades y sus colaboraciones. Una clase es cualquier persona, evento, concepto,

pantalla o reporte y sus responsabilidades son los métodos relacionados con esta clase y sus atributos. Las colaboraciones de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

La utilización de estas tarjetas reduce al mínimo la complejidad del diseño. Esto logra que el programador se centre en el diseño de las bases fundamentales de la clase y le impide entrar en detalles de su funcionamiento interno que en un momento pueden llegar a ser contraproducentes. Estas tarjetas propician que los diseñadores no otorguen demasiadas responsabilidades a las clases. A continuación se muestran las tarjetas CRC del sistema.

| Tarjeta: Autenticar usuario | |
|-----------------------------|--------------------|
| Funcionalidades | Colaboraciones |
| Autenticar usuario | Autenticar usuario |

Tabla 3.1. Tarjeta 1 CRC Autenticar usuario.

| Tarjeta: Gestionar área | |
|-------------------------|--------------------|
| Funcionalidades | Colaboraciones |
| Insertar área | Gestionar área |
| Modificar área | Autenticar usuario |
| Eliminar área | |
| Mostrar área | |

Tabla 3.2. Tarjeta 2 CRC Gestionar área.

| Tarjeta: Gestionar plan anual | |
|-------------------------------|----------------------|
| Funcionalidades | Colaboraciones |
| Insertar plan anual | Gestionar plan anual |
| Modificar plan anual | Gestionar usuario |
| Eliminar plan anual | Gestionar rol |
| Mostrar plan anual | Autenticar usuario |
| Publicar plan anual | |
| Enviar notificación | |

Tabla 3.3. Tarjeta 3 CRC Gestionar plan anual.

| Tarjeta: Gestionar control a clase | |
|------------------------------------|---------------------------|
| Funcionalidades | Colaboraciones |
| Insertar control a clase | Gestionar control a clase |
| Modificar control a clase | Gestionar usuario |
| Eliminar control a clase | Gestionar rol |
| Mostrar control a clase | Autenticar usuario |
| Publicar control a clase | |

Enviar notificación

Tabla 3.4. Tarjeta 4 CRC Gestionar control a clase.

| Tarjeta: Gestionar artículo | |
|-----------------------------|--------------------|
| Funcionalidades | Colaboraciones |
| Insertar artículo | Gestionar artículo |
| Modificar artículo | Gestionar usuario |
| Eliminar artículo | Autenticar usuario |
| Mostrar artículo | |
| Publicar artículo | |

Tabla 3.5. Tarjeta 5 CRC Gestionar artículos.

| Tarjeta: Gestionar evaluación trimestral | |
|--|----------------------|
| Funcionalidades | Colaboraciones |
| Insertar evaluación trimestral | Gestionar evaluación |
| Modificar evaluación trimestral | Gestionar usuario |
| Eliminar evaluación trimestral | Autenticar usuario |
| Mostrar evaluación trimestral | Gestionar rol |
| Publicar evaluación trimestral | |

Tabla 3.6. Tarjeta 6 CRC Gestionar evaluación trimestral.

| Tarjeta: Gestionar plan de la facultad | |
|--|--|
| Funcionalidades | Colaboraciones |
| Insertar plan de trabajo de la facultad | Gestionar plan de trabajo de la facultad |
| Modificar plan de trabajo de la facultad | Gestionar usuario |
| Eliminar plan de trabajo de la facultad | Autenticar usuario |
| Mostrar plan de trabajo de la facultad | Gestionar rol |
| Publicar plan de trabajo de la facultad | |

Tabla 3.7. Tarjeta 7 CRC Gestionar plan de la facultad.

| Tarjeta: Gestionar plan del departamento | |
|--|--|
| Funcionalidades | Colaboraciones |
| Insertar plan de trabajo del departamento | Gestionar plan de trabajo del departamento |
| Modificar plan de trabajo del departamento | Gestionar usuario |
| Eliminar plan de trabajo del departamento | Autenticar usuario |
| Mostrar plan de trabajo del departamento | Gestionar rol |
| Publicar plan de trabajo del departamento | |

Tabla 3.8. Tarjeta CRC 8 Gestionar plan del departamento.

| Tarjeta: Gestionar plan individual | |
|--------------------------------------|--------------------------------------|
| Funcionalidades | Colaboraciones |
| Insertar plan de trabajo individual | Gestionar plan de trabajo individual |
| Modificar plan de trabajo individual | Gestionar usuario |
| Eliminar plan de trabajo individual | Autenticar usuario |
| Mostrar plan de trabajo individual | Gestionar rol |
| Publicar plan de trabajo individual | Gestionar plan del departamento |

Tabla 3.9. Tarjeta 9 CRC Gestionar plan individual.

| Tarjeta: Gestionar usuario | |
|----------------------------|--------------------|
| Funcionalidades | Colaboraciones |
| Insertar usuario | |
| Modificar usuario | Autenticar usuario |
| Eliminar usuario | Gestionar usuario |
| Mostrar usuario | Gestionar área |

Tabla 3.10. Tarjeta 10 CRC Gestionar usuario.

| Tarjeta: Gestionar rol | |
|------------------------|--------------------|
| Funcionalidades | Colaboraciones |
| Insertar rol | Autenticar usuario |
| Modificar rol | Gestionar rol |
| Eliminar rol | Gestionar usuario |
| Mostrar rol | |
| Asignar rol a usuarios | |
| Quitar rol a usuarios | |

Tabla 3.11. Tarjeta 11 CRC Gestionar rol.

3.4 Implementación

En esta fase se plantea la implementación de las historias de usuario en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. Para ello se descomponen las historias de usuario en tareas de desarrollo para ser realizadas, describiendo cada una de éstas.

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario que fueron seleccionadas en cada una de ellas. Al principio se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte del plan se descomponen las historias en tareas de ingeniería. Para definir las tareas de la ingeniería se cuenta con una plantilla que permite definir cada una de las actividades a las que están asociadas las historias de usuario y que permiten su implementación. A continuación se muestran las tareas asignadas por iteraciones.

3.4.1 Tareas de ingeniería

Las tareas de ingeniería son un conjunto de acciones que se realizan con el objetivo de resolver las historias de usuarios, una historia de usuario puede tener una o más tareas de ingeniería en dependencia de la complejidad de la funcionalidad a desarrollar. A continuación se muestran las tareas de ingeniería pertenecientes a la primera iteración (ver restantes en anexo 1)

Iteración 1

HU Autenticar usuario.

| Tarea de ingeniería | |
|---|----------------------------|
| Número de la tarea: 1 | Número de la HU: 1 |
| Nombre de la tarea: Autenticar usuario | |
| Tipo de tarea: Desarrollo | |
| Fecha de inicio: 1/4/2013 | Fecha Fin: 2/4/2013 |
| Programador responsable: Jorge Ernesto Bauzá Becerra | |
| Descripción: El sistema debe crear un formulario con los datos de inicio de sesión (usuario y contraseña). | |

Tabla 3.12. Tarea 1 Autenticar Usuario.

| Tarea de ingeniería | |
|--|----------------------------|
| Número de la tarea: 2 | Número de la HU: 1 |
| Nombre de la tarea: Validar datos | |
| Tipo de tarea: Desarrollo | |
| Fecha de inicio: 2/4/2013 | Fecha Fin: 3/4/2013 |
| Programador responsable: Jorge Ernesto Bauzá Becerra | |
| Descripción: Se implementan las funcionalidades que permiten la captura de los datos, se verifican que sean correctos con la base de datos de la UCI y la de la aplicación. En caso de que sean correctos se redirecciona el usuario a la página correspondiente según su privilegio. | |

Tabla 3.13. Tarea 2 Validar datos.

HU Gestionar usuario.

| Tarea de ingeniería | |
|--|----------------------------|
| Número de la tarea: 1 | Número de la HU: 2 |
| Nombre de la tarea: Gestionar usuario | |
| Tipo de tarea: Desarrollo | |
| Fecha de inicio: 3/4/2013 | Fecha Fin: 5/4/2013 |
| Programador responsable: Jorge Ernesto bauza Becerra | |
| Descripción: Se implementan las funcionalidades que permiten obtener los datos (nombre, usuario, solapín, | |

área, rol, correo, cargo, categoría) del usuario a insertar. Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la manera esperada, sin dejarse datos nulos. Se implementan funcionalidades tales como listar los usuarios existentes en el sistema dado un criterio, modificar los datos del usuario o eliminarlos.

Tabla 3.14. Tarea 1 Gestionar usuario.

| Tarea de ingeniería | |
|---|----------------------------|
| Número de la tarea: 2 | Número de la HU: 2 |
| Nombre de la tarea: Cambiar rol | |
| Tipo de tarea: Desarrollo | |
| Fecha de inicio: 6/4/2013 | Fecha Fin: 6/4/2013 |
| Programador responsable: Jorge Ernesto Bauzá Becerra | |
| Descripción: Se implementan las funcionalidades que permiten cambiarle el rol a un usuario especificado. | |

Tabla 3.15. Tabla 2 Tarea Cambiar rol.

HU Gestionar rol.

| Tarea de ingeniería | |
|---|----------------------------|
| Número de la tarea: 1 | Número de la HU: 3 |
| Nombre de la tarea: Gestionar rol | |
| Tipo de tarea: Desarrollo | |
| Fecha de inicio: 7/4/2013 | Fecha Fin: 7/4/2013 |
| Programador responsable: Jorge Ernesto Bauzá Becerra | |
| Descripción: Se implementan las funcionalidades que permiten la captura del dato (nombre) del rol a insertar. Luego se realiza la validación del dato, de forma tal que el mismo sea insertado correctamente. Se implementan funcionalidades tales como listar los roles existentes en el sistema, modificar y eliminar rol. | |
| Descripción: Se implementa las funcionalidades que permiten quitar un rol a un usuario especificado. | |

Tabla 3.16. Tarea 1 Gestionar rol.

3.4.2 Diseño de la Base de Datos del sistema

En cualquier aplicación en la cual se gestione información la base de datos desempeña un papel fundamental. Es la encargada de almacenar de forma coherente y organizada los datos para evitar que dicha información se pierda. Es por esto que se realiza el diseño de la base de datos en cuestión, para lograr la persistencia de los datos de los usuarios, así como la información de los recursos que se gestionan por los mismos, controlando que esta información no tenga redundancia y no existan datos innecesarios que no cumplan ningún objetivo al ser guardados.

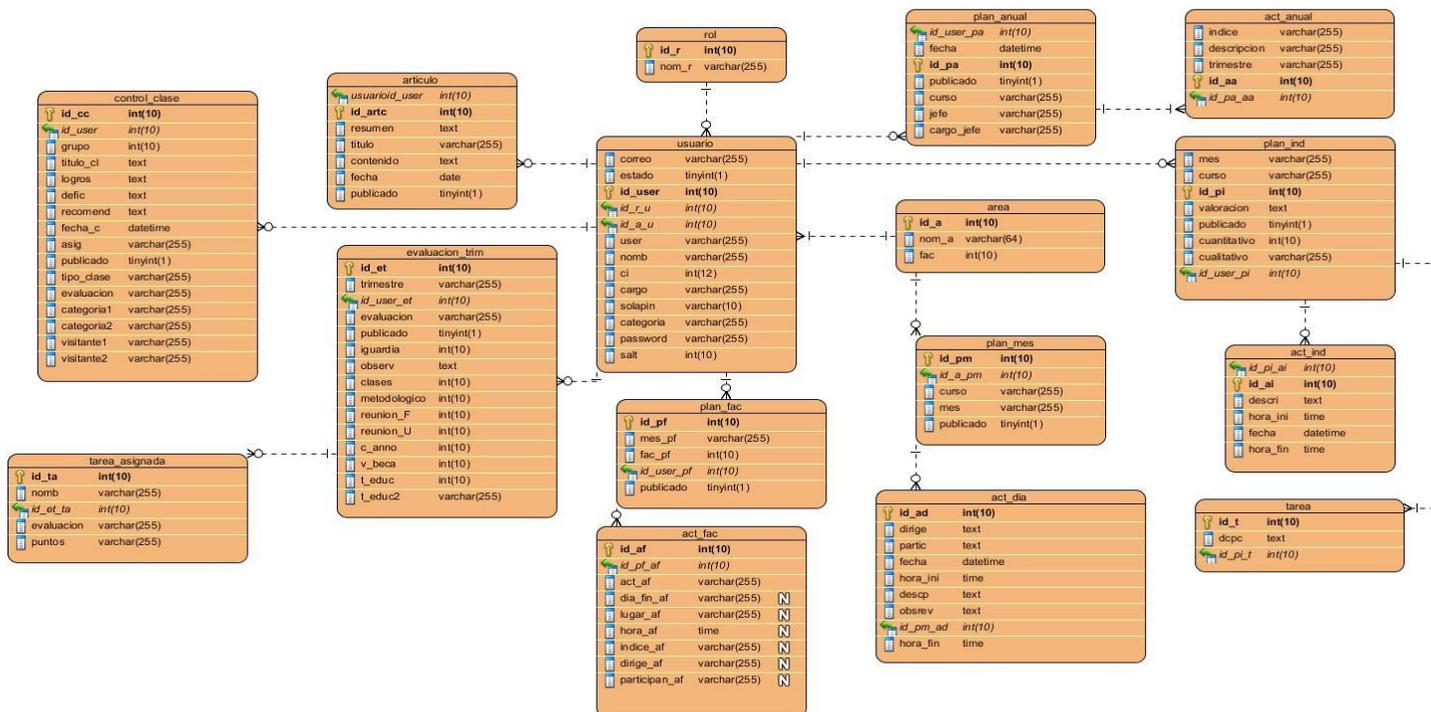


Figura 3. Modelo de datos

3.4.3 Diagrama de despliegue

Los diagramas de despliegue muestran nodos, conexiones, componentes y objetos. Los nodos representan objetos físicos con recursos computacionales como procesadores y periféricos; pueden mostrarse como una clase o una instancia, por lo que su nombre sigue la misma sintaxis establecida para clases y objetos. Las conexiones son asociaciones de comunicación entre los nodos, y se etiquetan con un estereotipo que identifica el protocolo de comunicación o la red utilizada. Los componentes son archivos de código ejecutable, que residen y se ejecutan dentro de un nodo; se pueden representar relaciones de dependencia entre los componentes que, de manera similar a las dependencias entre paquetes, corresponden al uso de servicios. (38)

El diagrama de despliegue de la propuesta de solución estará compuesto por 4 nodos fundamentales:

- **PC_cliente** (este nodo representa la computadora con que el usuario se conecta y trabaja con el sistema).
- **Servidor Apache** (en este nodo es donde está desplegado o instalado el servidor del sistema web).
- **Servidor LDAP** (este nodo representa el servidor LDAP utilizado por la UCI para obtener los datos de un usuario determinado).
- **BD MySQL** (este nodo contiene la base de datos con la que interactúa el sistema).

El proceso comienza cuando un usuario accede a la aplicación por una PC Cliente, esta se conecta a su vez con el servidor web Apache a través del protocolo HTTP. Se procede a verificar si el usuario y contraseña introducidos son correctos, mediante una petición hecha al servidor LDAP utilizando el protocolo SOAP, los resultados de este paso serán nulo o los datos de dicho usuario. Con estos datos personales el servidor Apache valida que exista el usuario en el sistema. Una vez validado el usuario, se procede a atender las peticiones del mismo, comunicándose para esto con la

base de datos (BD MySQL) mediante el protocolo SSL de manera que los datos sean encriptados y viajen seguros por la red. En la Figura 4 se describe las relaciones entre cada nodo involucrado.

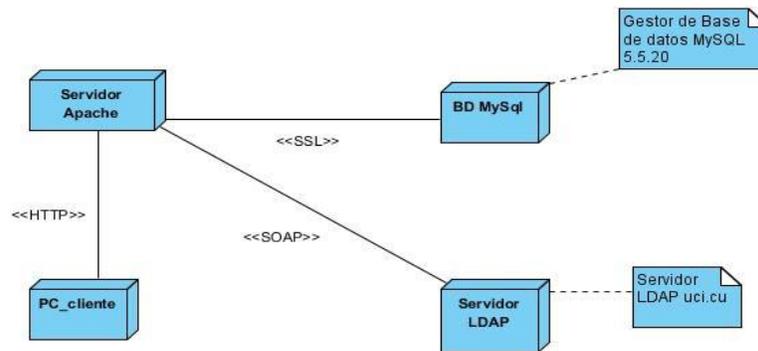


Figura 4. Diagrama de despliegue del sistema

3.5 Conclusiones Parciales

El diseño de un software es un paso decisivo en la construcción del mismo, por esta razón en el presente capítulo se crearon las bases para la implementación del sistema en cuestión, desarrollando para ello todos los artefactos definidos por la metodología XP.

Al concluir el presente capítulo se obtienen resultados significativos en la investigación, ellos son:

- Las tarjetas CRC con las clases que intervienen en la solución, que ayudan a los programadores crear diseños de clases orientadas a responsabilidades.
- Las tareas de ingeniería las cuales permiten a los desarrolladores llevar a cabo su tarea con mayor facilidad.
- El diseño de la base de datos, que posibilita un mejor entendimiento de cómo se relacionan las tablas en la base de datos.
- El diagrama de despliegue a través del cual se representan las relaciones entre los diferentes recursos computacionales.

Capítulo 4: Validación del Sistema

4.1 Introducción

En el presente capítulo se desarrollan las pruebas a la aplicación según la metodología XP, desarrollando para ello los artefactos que esta metodología define. Se describen las pruebas de aceptación creadas para ayudar a saber qué es exactamente lo que tiene que hacer el código a implementar y así una vez implementado lograr que pase dichas pruebas sin problemas.

4.2 Tipos de pruebas

Esta metodología divide las pruebas del sistema en dos grupos: 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, estas pruebas son diseñadas por el cliente final.

Para la confección de las pruebas unitarias se utiliza el sistema de pruebas que brinda Symfony. Las pruebas unitarias están orientadas a caja blanca. La realización de las pruebas de aceptación se efectúa de forma manual incluyendo el diseño de casos de prueba, este tipo es considerado como de caja negra. Las pruebas de caja negra son aquellas que se llevan a cabo sobre la interfaz del software. Las mismas pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto.

Pruebas unitarias

Estas pruebas se centran principalmente en los requisitos funcionales del software. Permiten obtener un conjunto de condiciones de entrada que ejerciten completamente los requisitos funcionales. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. (39)

Symfony brinda su propio sistema de pruebas (test). Los test unitarios son aquellos en los que se verifica que un método se comporta como debería, sin tener en cuenta su entorno.

Pruebas de aceptación

Estas pruebas funcionan como una caja negra, pues cada una de ellas representa una salida esperada del sistema. El objetivo final de las mismas es lograr que los requerimientos sean cumplidos y que el sistema sea aceptable. Una vez que todas las historias de usuario hayan pasado sus pruebas de aceptación se considera entonces terminada la aplicación.

Las pruebas de aceptación son creadas a partir de las historias de usuario. Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación.

Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas. La realización de este tipo de pruebas y la publicación de los resultados tiene que ser lo más rápido posible, para que los desarrolladores puedan realizar con la mayor rapidez los cambios que sean necesarios. (39)

Los campos que presenta una prueba de aceptación son los siguientes:

Número: Número de la HU a la que corresponde.

Código: Número único que se le asigna a cada caso de prueba que pertenece a una HU determinada con el fin de lograr una mejor organización de estos.

Nombre: Nombre del caso de prueba. Debe ser descriptivo, en la medida de las posibilidades, de lo que se comprobará y no muy extenso.

Descripción: Se describe qué es lo que se desea probar de manera corta y precisa.

Condiciones de ejecución: Condiciones especiales que deben tenerse en cuenta para ejecutar el caso de prueba.

Entradas: Entradas al caso de prueba en caso de necesitarlas.

Resultado esperado: Resultado que se desea tenga el caso de prueba. Descripción breve de lo que debe suceder.

Evaluación: Se evalúa si el caso de prueba tuvo éxito o no. En caso de ser exitoso se asigna un resultado de satisfactorio, en caso contrario insatisfactorio.

A continuación se presentan las pruebas de aceptación perteneciente a la segunda iteración de pruebas.

HU “Autenticar usuario”

| Prueba de Aceptación | |
|--|--|
| Código: HU1_p1 | Historia de usuario (número): 1 |
| Nombre: Autenticar con datos no válidos | |
| Descripción: Probar que no se pueda autenticar un usuario en el sistema con al menos un dato no válido. | |
| Condiciones de Ejecución: Usuario y/o contraseña no puede(n) ser válido(s) | |
| Entrada / Pasos de ejecución: Acceder al sistema con al menos un dato no válido (campos vacíos o erróneos) o con un usuario que no pertenezca al dominio UCI. | |
| Resultado Esperado: El sistema informa al usuario que sus datos no son válidos, dándole la posibilidad de intentarlo nuevamente. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.1. Prueba 1 de la HU Autenticar usuario

| Prueba de Aceptación | |
|--|--|
| Código: HU1_p2 | Historia de usuario (número): 2 |
| Nombre: Autenticar con datos válidos | |
| Descripción: Probar que se pueda autenticar un usuario en el sistema con datos válidos | |
| Condiciones de Ejecución: Usuario y contraseña deben ser válidos. | |
| Entrada / Pasos de ejecución: Acceder al sistema con datos válidos. | |
| Resultado Esperado: Se le muestra una página al usuario según sus privilegios, con las funcionalidades correspondientes a su rol. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.2. Prueba 2 de la HU Autenticar usuario

HU Gestionar usuario

| Prueba de Aceptación | |
|--|--|
| Código: HU2_.p1 | Historia de usuario (número): 2 |
| Nombre: Insertar usuario | |
| Descripción: Probar la funcionalidad insertar un nuevo usuario en la aplicación | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Insertar un nuevo usuario con sus datos válidos. | |
| Resultado Esperado: El usuario se inserta en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.3. Prueba 1 de la HU Gestionar usuario

| Prueba de Aceptación | |
|--|--|
| Código: HU2_.p2 | Historia de usuario (número): 2 |
| Nombre: Eliminar usuario | |
| Descripción: Probar la funcionalidad eliminar un usuario. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Eliminar el usuario seleccionado. | |
| Resultado Esperado: El usuario se elimina del sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.4. Prueba 2 de la HU Gestionar usuario

| Prueba de Aceptación | |
|--|--|
| Código: HU2_.p3 | Historia de usuario (número): 2 |
| Nombre: Modificar usuario | |
| Descripción: Probar la funcionalidad modificar un usuario. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Modificar los datos de un usuario. | |
| Resultado Esperado: Se modifican los datos del usuario. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.5. Prueba 3 de la HU Gestionar usuario

| Prueba de Aceptación | |
|------------------------|--|
| Código: HU2_.p4 | Historia de usuario (número): 2 |

| |
|--|
| Nombre: Buscar usuario |
| Descripción: Probar la funcionalidad buscar usuario. |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. |
| Entrada / Pasos de ejecución: Buscar un usuario. |
| Resultado Esperado: Se muestra el usuario. |
| Evaluación de la Prueba: Evaluación satisfactoria |

Tabla 4.6. Prueba 4 de la HU Gestionar usuario

| Prueba de Aceptación | |
|--|--|
| Código: HU2_.p5 | Historia de usuario (número): 2 |
| Nombre: Cambiar el rol del usuario | |
| Descripción: Probar la funcionalidad cambiar rol del usuario. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Cambiar el rol del usuario. | |
| Resultado Esperado: Se cambiar el rol del usuario. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.7. Prueba 5 de la HU Gestionar usuario

HU Gestionar rol

| Prueba de Aceptación | |
|--|--|
| Código: HU3_.p1 | Historia de usuario (número): 3 |
| Nombre: Insertar rol | |
| Descripción: Probar la funcionalidad insertar rol. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Insertar un nuevo rol con sus datos válidos. | |
| Resultado Esperado: Se inserta el rol en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.8. Prueba 1 de la HU Gestionar rol

| Prueba de Aceptación | |
|--|--|
| Código: HU3_.p2 | Historia de usuario (número): 3 |
| Nombre: Eliminar rol | |
| Descripción: Probar la funcionalidad eliminar un rol existente. | |

| |
|--|
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. |
| Entrada / Pasos de ejecución: Eliminar el rol seleccionado. |
| Resultado Esperado: El rol se elimina del sistema. |
| Evaluación de la Prueba: Evaluación satisfactoria |

Tabla 4.9. Prueba 2 de la HU Gestionar rol

| Prueba de Aceptación | |
|--|--|
| Código: HU3_p3 | Historia de usuario (número): 3 |
| Nombre: Modificar rol | |
| Descripción: Probar la funcionalidad modificar rol. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Modificar los datos de un rol. | |
| Resultado Esperado: El rol se modifica en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.10. Prueba 3 de la HU Gestionar rol

| Prueba de Aceptación | |
|--|--|
| Código: HU3_p4 | Historia de usuario (número): 3 |
| Nombre: Buscar rol | |
| Descripción: Probar la funcionalidad buscar rol. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Buscar un rol. | |
| Resultado Esperado: Se muestra el rol. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.11. Prueba 4 de la HU Gestionar rol

HU Gestionar plan de trabajo mensual de la facultad

| Prueba de Aceptación | |
|---|--|
| Código: HU4_p1 | Historia de usuario (número): 4 |
| Nombre: Insertar plan de trabajo mensual de la facultad | |
| Descripción: Probar la funcionalidad insertar plan de trabajo mensual de la facultad | |
| Condiciones de Ejecución: El usuario debe estar autenticado y poseer uno de los siguientes privilegios: vicedecano de formación o administrador. | |

| |
|--|
| Entrada / Pasos de ejecución: Insertar un plan de trabajo mensual de la facultad con datos válidos. |
| Resultado Esperado: Se inserta el plan de trabajo mensual de la facultad en el sistema. |
| Evaluación de la Prueba: Evaluación satisfactoria |

Tabla 4.12. Prueba 1 de la HU Gestionar plan de trabajo mensual de la facultad

| Prueba de Aceptación | |
|---|--|
| Código: HU4_.p2 | Historia de usuario (número): 4 |
| Nombre: Eliminar plan de trabajo mensual de la facultad. | |
| Descripción: Probar la funcionalidad eliminar un plan de trabajo mensual de la facultad. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios de vicedecano de formación o administrador. | |
| Entrada / Pasos de ejecución: Eliminar el plan de trabajo mensual de la facultad seleccionado. | |
| Resultado Esperado: Se elimina el plan de trabajo mensual de la facultad del sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.14. Prueba 2 de la HU Gestionar plan de trabajo mensual de la facultad

| Prueba de Aceptación | |
|---|--|
| Código: HU4_.p3 | Historia de usuario (número): 4 |
| Nombre: Modificar plan de trabajo mensual de la facultad. | |
| Descripción: Probar la funcionalidad modificar un plan de trabajo mensual de la facultad. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios de vicedecano de formación o administrador. | |
| Entrada / Pasos de ejecución: Modificar los datos del plan de trabajo mensual de la facultad. | |
| Resultado Esperado: El plan de trabajo mensual de la facultad se modifica en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.15. Prueba 3 de la HU Gestionar plan de trabajo mensual de la facultad

| Prueba de Aceptación | |
|---|--|
| Código: HU4_.p4 | Historia de usuario (número): 4 |
| Nombre: Buscar plan de trabajo mensual de la facultad. | |
| Descripción: Probar la funcionalidad buscar plan de trabajo mensual de la facultad. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener uno de los siguientes privilegios: vicedecano de formación, jefe de departamento o administrador. | |
| Entrada / Pasos de ejecución: Buscar plan de trabajo mensual de la facultad. | |
| Resultado Esperado: Se muestra el plan de trabajo mensual de la facultad. | |

| |
|--|
| Evaluación de la Prueba: Evaluación satisfactoria |
|--|

Tabla 4.16. Prueba 4 de la HU Gestionar plan de trabajo mensual de la facultad

HU Gestionar plan de trabajo mensual del departamento

| Prueba de Aceptación | |
|--|--|
| Código: HU5_.p1 | Historia de usuario (número): 5 |
| Nombre: Insertar plan de trabajo mensual del departamento. | |
| Descripción: Probar la funcionalidad insertar plan de trabajo mensual del departamento. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios de jefe de departamento. | |
| Entrada / Pasos de ejecución: Insertar un plan de trabajo mensual del departamento con datos válidos. | |
| Resultado Esperado: Se inserta el plan de trabajo mensual del departamento en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.17. Prueba 1 de la HU Gestionar plan de trabajo mensual del departamento

| Prueba de Aceptación | |
|--|--|
| Código: HU5_.p2 | Historia de usuario (número): 5 |
| Nombre: Eliminar plan de trabajo mensual del departamento. | |
| Descripción: Probar la funcionalidad eliminar un plan de trabajo mensual del departamento. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios de jefe de departamento. | |
| Entrada / Pasos de ejecución: Eliminar el plan de trabajo mensual del departamento seleccionado. | |
| Resultado Esperado: Se elimina el plan de trabajo mensual del departamento del sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.18. Prueba 2 de la HU Gestionar plan de trabajo mensual del departamento

| Prueba de Aceptación | |
|--|--|
| Código: HU5_.p3 | Historia de usuario (número): 5 |
| Nombre: Modificar plan de trabajo mensual del departamento. | |
| Descripción: Probar la funcionalidad modificar un plan de trabajo mensual del departamento. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios de jefe de departamento. | |
| Entrada / Pasos de ejecución: Modificar los datos del plan de trabajo mensual del departamento. | |
| Resultado Esperado: El plan de trabajo mensual del departamento se modifica en sistema. | |

| |
|--|
| Evaluación de la Prueba: Evaluación satisfactoria |
|--|

Tabla 4.19. Prueba 3 de la HU Gestionar plan de trabajo mensual del departamento

| Prueba de Aceptación | |
|--|--|
| Código: HU5_.p4 | Historia de usuario (número): 5 |
| Nombre: Buscar plan de trabajo mensual del departamento. | |
| Descripción: Probar la funcionalidad buscar plan de trabajo mensual del departamento. | |
| Condiciones de Ejecución: El usuario debe estar autenticado. | |
| Entrada / Pasos de ejecución: Buscar plan de trabajo mensual del departamento. | |
| Resultado Esperado: Se muestra el plan de trabajo mensual del departamento. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.20. Prueba 4 de la HU Gestionar plan de trabajo mensual del departamento

HU Gestionar plan de trabajo mensual individual

| Prueba de Aceptación | |
|--|--|
| Código: HU6_.p1 | Historia de usuario (número): 6 |
| Nombre: Insertar plan de trabajo mensual individual. | |
| Descripción: Probar la funcionalidad insertar el plan de trabajo mensual individual. | |
| Condiciones de Ejecución: El usuario debe estar autenticado. | |
| Entrada / Pasos de ejecución: Insertar el plan de trabajo mensual individual con datos válidos. | |
| Resultado Esperado: Se inserta el plan de trabajo mensual individual en la base de datos. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.21. Prueba 1 de la HU Gestionar plan de trabajo mensual individual

| Prueba de Aceptación | |
|---|--|
| Código: HU6_.p2 | Historia de usuario (número): 6 |
| Nombre: Eliminar plan de trabajo mensual individual. | |
| Descripción: Probar la funcionalidad eliminar plan de trabajo mensual individual. | |
| Condiciones de Ejecución: El usuario debe estar autenticado. | |
| Entrada / Pasos de ejecución: Eliminar el plan de trabajo mensual individual seleccionado. | |
| Resultado Esperado: Se elimina el plan de trabajo mensual individual del sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.22. Prueba 2 de la HU Gestionar plan de trabajo mensual individual

| Prueba de Aceptación | |
|------------------------|--|
| Código: HU6_.p3 | Historia de usuario (número): 6 |

| |
|--|
| Nombre: Modificar plan de trabajo mensual individual. |
| Descripción: Probar la funcionalidad modificar plan de trabajo mensual individual. |
| Condiciones de Ejecución: El usuario debe estar autenticado. |
| Entrada / Pasos de ejecución: Modificar los datos del plan de trabajo mensual individual. |
| Resultado Esperado: El plan de trabajo mensual individual se modifica en el sistema. |
| Evaluación de la Prueba: Evaluación satisfactoria |

Tabla 4.23. Prueba 3 de la HU Gestionar plan de trabajo mensual individual

| Prueba de Aceptación | |
|---|--|
| Código: HU6_.p4 | Historia de usuario (número): 6 |
| Nombre: Buscar plan de trabajo mensual individual. | |
| Descripción: Probar la funcionalidad buscar plan de trabajo mensual individual de un profesor. | |
| Condiciones de Ejecución: El usuario debe estar autenticado | |
| Entrada / Pasos de ejecución: Buscar el plan de trabajo mensual individual de un profesor. | |
| Resultado Esperado: Se muestra el plan de trabajo mensual individual. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.24. Prueba 4 de la HU Gestionar plan de trabajo mensual individual

HU Gestionar área

| Prueba de Aceptación | |
|--|--|
| Código: HU7_.p1 | Historia de usuario (número): 7 |
| Nombre: Insertar área | |
| Descripción: Probar la funcionalidad insertar área. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Insertar una nueva área con datos válidos. | |
| Resultado Esperado: Se inserta el área en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.25. Prueba 1 de la HU Gestionar área

| Prueba de Aceptación | |
|--|--|
| Código: HU7_.p2 | Historia de usuario (número): 7 |
| Nombre: Eliminar área | |
| Descripción: Probar la funcionalidad eliminar una área existente. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |

| |
|---|
| Entrada / Pasos de ejecución: Eliminar el área seleccionada. |
| Resultado Esperado: El área se elimina del sistema. |
| Evaluación de la Prueba: Evaluación satisfactoria |

Tabla 4.27. Prueba 2 de la HU Gestionar área

| Prueba de Aceptación | |
|--|--|
| Código: HU7_.p3 | Historia de usuario (número): 7 |
| Nombre: Modificar área | |
| Descripción: Probar la funcionalidad modificar una área existente. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Modificar los datos del área. | |
| Resultado Esperado: El área se modifica en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.28. Prueba 3 de la HU Gestionar área

| Prueba de Aceptación | |
|--|--|
| Código: HU7_.p4 | Historia de usuario (número): 7 |
| Nombre: Buscar área | |
| Descripción: Probar la funcionalidad buscar área. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador. | |
| Entrada / Pasos de ejecución: Buscar el área. | |
| Resultado Esperado: Se muestra el área. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.29. Prueba 4 de la HU Gestionar área

HU Gestionar área

| Prueba de Aceptación | |
|---|--|
| Código: HU8_.p1 | Historia de usuario (número): 8 |
| Nombre: Insertar plan de trabajo anual | |
| Descripción: Probar la funcionalidad insertar plan de trabajo anual. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del jefe de departamento. | |
| Entrada / Pasos de ejecución: Insertar un nuevo plan de trabajo anual con todos sus datos válidos. | |

| |
|---|
| Resultado Esperado: Se inserta el plan de trabajo anual en el sistema. |
| Evaluación de la Prueba: Evaluación satisfactoria |

Tabla 4.30. Prueba 1 de la HU Gestionar plan de trabajo anual

| Prueba de Aceptación | |
|---|--|
| Código: HU8_.p2 | Historia de usuario (número): 8 |
| Nombre: Eliminar plan de trabajo anual | |
| Descripción: Probar la funcionalidad eliminar plan de trabajo anual existente. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del jefe de departamento. | |
| Entrada / Pasos de ejecución: Eliminar el plan de trabajo anual seleccionado. | |
| Resultado Esperado: El plan de trabajo anual se elimina del sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.31. Prueba 2 de la HU Gestionar área

| Prueba de Aceptación | |
|---|--|
| Código: HU8_.p3 | Historia de usuario (número): 8 |
| Nombre: Modificar plan de trabajo anual | |
| Descripción: Probar la funcionalidad modificar un plan de trabajo anual existente. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios del jefe de departamento. | |
| Entrada / Pasos de ejecución: Modificar los datos del plan de trabajo anual. | |
| Resultado Esperado: El plan de trabajo anual se modifica en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.32. Prueba 3 de la HU Gestionar plan de trabajo anual

| Prueba de Aceptación | |
|---|--|
| Código: HU8_.p4 | Historia de usuario (número): 8 |
| Nombre: Buscar plan de trabajo anual | |
| Descripción: Probar la funcionalidad buscar plan de trabajo anual. | |
| Condiciones de Ejecución: El usuario debe estar autenticado. | |
| Entrada / Pasos de ejecución: Buscar el plan de trabajo anual. | |
| Resultado Esperado: Se muestra el plan de trabajo anual. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.33. Prueba 4 de la HU Gestionar plan de trabajo anual

HU Gestionar evaluación trimestral

| Prueba de Aceptación | |
|--|--|
| Código: HU9_.p1 | Historia de usuario (número): 9 |
| Nombre: Insertar evaluación trimestral. | |
| Descripción: Probar la funcionalidad insertar evaluación trimestral. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios de vicedecano de formación o jefe de departamento. | |
| Entrada / Pasos de ejecución: Insertar la evaluación trimestral con datos válidos. | |
| Resultado Esperado: Se inserta la evaluación trimestral en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.34. Prueba 1 de la HU Gestionar evaluación trimestral

| Prueba de Aceptación | |
|--|--|
| Código: HU9_.p2 | Historia de usuario (número): 9 |
| Nombre: Eliminar evaluación trimestral. | |
| Descripción: Probar la funcionalidad eliminar evaluación trimestral. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios de jefe de departamento o vicedecano de formación. | |
| Entrada / Pasos de ejecución: Eliminar la evaluación trimestral seleccionada. | |
| Resultado Esperado: Se elimina la evaluación trimestral del sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.35. Prueba 2 de la HU Gestionar evaluación trimestral

| Prueba de Aceptación | |
|--|--|
| Código: HU9_.p3 | Historia de usuario (número): 9 |
| Nombre: Modificar evaluación trimestral. | |
| Descripción: Probar la funcionalidad modificar evaluación trimestral. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener los privilegios de jefe de departamento o vicedecano de formación. | |
| Entrada / Pasos de ejecución: Modificar los datos de la evaluación trimestral. | |
| Resultado Esperado: La evaluación trimestral se modifica en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.37. Prueba 3 de la HU Gestionar evaluación trimestral

| Prueba de Aceptación | |
|------------------------|--|
| Código: HU9_.p4 | Historia de usuario (número): 9 |

| |
|---|
| Nombre: Buscar evaluación trimestral. |
| Descripción: Probar la funcionalidad buscar evaluación trimestral de un profesor. |
| Condiciones de Ejecución: El usuario debe estar autenticado y tener los privilegios de jefe de departamento, vicedecano de formación o ser el profesor al cual pertenece la evaluación trimestral. |
| Entrada / Pasos de ejecución: Buscar la evaluación trimestral de un profesor. |
| Resultado Esperado: Se muestra la evaluación trimestral. |
| Evaluación de la Prueba: Evaluación satisfactoria |

Tabla 4.38. Prueba 4 de la HU Gestionar evaluación trimestral

HU Gestionar control a clase

| Prueba de Aceptación | |
|--|---|
| Código: HU10_.p1 | Historia de usuario (número): 10 |
| Nombre: Insertar control a clase | |
| Descripción: Probar la funcionalidad insertar control a clase. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener uno de los siguientes privilegios: administrador, jefe de departamento o jefe de asignatura. | |
| Entrada / Pasos de ejecución: Insertar un nuevo control a clase con datos válidos. | |
| Resultado Esperado: Se inserta el control a clase en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.39. Prueba 1 de la HU control a clase

| Prueba de Aceptación | |
|--|---|
| Código: HU10_.p2 | Historia de usuario (número): 10 |
| Nombre: Eliminar control a clase | |
| Descripción: Probar la funcionalidad eliminar un control a clase. | |
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener uno de los siguientes privilegios: administrador, jefe de departamento o jefe de asignatura. | |
| Entrada / Pasos de ejecución: Eliminar el control a clase seleccionado. | |
| Resultado Esperado: Se elimina el control a clase del sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.40. Prueba 2 de la HU Gestionar control a clase

| Prueba de Aceptación | |
|---|---|
| Código: HU10_.p3 | Historia de usuario (número): 10 |
| Nombre: Modificar control a clase | |
| Descripción: Probar la funcionalidad modificar un control a clase. | |

| |
|--|
| Condiciones de Ejecución: El usuario debe estar autenticado y debe tener uno de los siguientes privilegios: administrador, jefe de departamento o jefe de asignatura. |
| Entrada / Pasos de ejecución: Modificar los datos del control a clase. |
| Resultado Esperado: El control a clase se modifica en el sistema. |
| Evaluación de la Prueba: Evaluación satisfactoria |

Tabla 4.41. Prueba 3 de la HU Gestionar control a clase

| Prueba de Aceptación | |
|---|---|
| Código: HU10_.p4 | Historia de usuario (número): 10 |
| Nombre: Buscar control a clase | |
| Descripción: Probar la funcionalidad buscar control a clase. | |
| Condiciones de Ejecución: El usuario debe estar autenticado. | |
| Entrada / Pasos de ejecución: Buscar un control a clase. | |
| Resultado Esperado: Se muestra el control a clase. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.42. Prueba 4 de la HU Gestionar control a clase

HU Gestionar artículo

| Prueba de Aceptación | |
|--|---|
| Código: HU11_.p1 | Historia de usuario (número): 11 |
| Nombre: Insertar artículo | |
| Descripción: Probar la funcionalidad insertar artículo. | |
| Condiciones de Ejecución: El usuario debe estar autenticado. | |
| Entrada / Pasos de ejecución: Insertar un nuevo artículo con todos sus datos válidos. | |
| Resultado Esperado: Se inserta el artículo en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.43. Prueba 1 de la HU Gestionar artículo

| Prueba de Aceptación | |
|---|---|
| Código: HU11_.p2 | Historia de usuario (número): 11 |
| Nombre: Eliminar artículo | |
| Descripción: Probar la funcionalidad eliminar artículo. | |
| Condiciones de Ejecución: El usuario debe estar autenticado, si es administrador podrá eliminar cualquier artículo, si posee otro rol solo podrá eliminar los artículos publicados por él. | |
| Entrada / Pasos de ejecución: Eliminar el artículo seleccionado. | |
| Resultado Esperado: El artículo se elimina del sistema. | |

| |
|--|
| Evaluación de la Prueba: Evaluación satisfactoria |
|--|

Tabla 4.44. Prueba 2 de la HU Gestionar artículo

| Prueba de Aceptación | |
|---|---|
| Código: HU11_.p3 | Historia de usuario (número): 11 |
| Nombre: Modificar artículo | |
| Descripción: Probar la funcionalidad modificar un artículo existente. | |
| Condiciones de Ejecución: El usuario debe estar autenticado, si es administrador podrá modificar cualquier artículo, si posee otro rol solo podrá modificar los artículos publicados por él. | |
| Entrada / Pasos de ejecución: Modificar los datos de un artículo. | |
| Resultado Esperado: El artículo se modifica en el sistema. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.45. Prueba 3 de la HU Gestionar artículo

| Prueba de Aceptación | |
|---|---|
| Código: HU11_.p4 | Historia de usuario (número): 11 |
| Nombre: Buscar artículo | |
| Descripción: Probar la funcionalidad buscar artículo. | |
| Condiciones de Ejecución: El usuario debe estar autenticado. | |
| Entrada / Pasos de ejecución: Buscar un artículo. | |
| Resultado Esperado: Se muestra el artículo. | |
| Evaluación de la Prueba: Evaluación satisfactoria | |

Tabla 4.46. Prueba 4 de la HU Gestionar artículo

Conclusiones sobre las pruebas

Se realizaron dos iteraciones de pruebas de funcionalidad a nivel de sistema para validar la calidad del software, con la ayuda de los Casos de Prueba de Aceptación descritos anteriormente. En la primera iteración se detectaron un total de 5 No Conformidades (NC), de ellas 1 fue de permiso y las 4 restantes de funcionalidad. Las mismas fueron mitigadas dando como resultado que en la segunda iteración no fuera encontrada ninguna no conformidad como se muestra en la figura 5.

Algunas de las no conformidades encontradas en la primera iteración fueron:

- No se obtenía del servidor LDAP la facultad correspondiente a un profesor determinado.
- No se había considerado al jefe de departamento como profesor, por lo que no se le podía crear un plan de trabajo individual.
- No se validaban todos los requisitos que debía poseer una evaluación trimestral.



Figura 5. Resultados de las pruebas de aceptación

4.3 Conclusiones parciales

En el presente capítulo se realizó la descripción de las pruebas de aceptación a las que fue sometida la aplicación con la intención de verificar el funcionamiento de la misma, probando individualmente la implementación de cada historia de usuario en su correspondiente iteración y asignándole una evaluación. Durante las pruebas de funcionalidad realizadas a la aplicación fueron detectadas 5 NC las mismas fueron resueltas en el menor tiempo posible dando como resultado que en la segunda iteración no fuera encontrada ninguna otra.

Conclusiones

Durante el desarrollo de este trabajo se realizó un estudio que permitió tener un conocimiento de la situación actual y las tendencias de los sistemas de gestión de la planificación de las actividades docentes a nivel nacional e internacional. Este estudio demostró la necesidad de desarrollar un sistema de este tipo en la Facultad 2 de la Universidad de las Ciencias Informáticas con el fin de gestionar la planificación de las actividades docente realizadas por los profesores.

Se presentó una solución web que gestiona el proceso de planificación de las actividades que desarrollan los trabajadores docentes de la facultad. La aplicación también abarca los procesos de elaboración de los controles a clases y las evaluaciones trimestrales que presentan los profesores. El sistema brinda la ventaja de centralizar la información así como su organización y disponibilidad para las distintas personas relacionadas con la misma.

El desarrollo del trabajo utilizando las guías de la metodología XP permitió minimizar el tiempo en que los usuarios finales obtuvieron un producto listo para utilizarse, sin sacrificar la calidad de la documentación generada y dejando las puertas abiertas para futuras mejoras. Se documentó todo el trabajo desde sus inicios lo que facilitará su estudio por parte de futuros desarrolladores, permitiendo una comprensión rápida y sencilla de la concepción general del sistema.

Recomendaciones

Luego de haber implementado la propuesta para el sistema automatizado que gestiona el proceso de planificación de las actividades docentes se recomienda:

- Utilizar este trabajo de diploma como bibliografía para posteriores investigaciones.
- Se incluya el software implementado en el proceso de gestión de planificación de las actividades docentes llevado a cabo en los distintos departamentos de la Facultad 2 de la Universidad de las Ciencias Informáticas.
- Implementar otros módulos que contribuyan a facilitar aún más el trabajo en el proceso.

Trabajos citados

1. **Ullastres, Marchesi.** Presente y futuro de la Reforma educativa en España. Revista Iberoamericana. [En línea] 2001. <http://www.rieoei.org/rie27a03.htm>.
2. LA REFLEXIÓN DE LA PRÁCTICA COMO ELEMENTO DE MEJORA DEL DESEMPEÑO ACADÉMICO DE LOS COLECTIVOS DOCENTES EN EDUCACIÓN BÁSICA. **HUERTA PÉREZ, PATRICIA** . XI Congreso Nacional de Investigación Educativa / 15. Procesos de Formación / Ponencia.
3. **Méndez, R. M.** La actitud de los estudiantes hacia la universidad. Tesis doctoral. España: Universidad de Santiago de Compostela : s.n.
4. **Wass, S.** Salidas escolares y trabajo de campo en la educación primaria. Madrid, España : s.n., 1992.
5. **Gal, Le.** Los derechos del niño en la escuela. Una educación para la ciudadanía. Barcelona, España : s.n., 2005.
6. **Thurler, Gather.** Innovar en el seno de la institución escolar. Barcelona, España : s.n., 2004.
7. **Bolívar, A.,** La asesoría al colectivo docente y el trabajo en el aula. En Secretaría de Educación. México : s.n., 2006.
8. **Marcelo, C. & Vaillant, D.** Desarrollo profesional ¿Cómo se aprende a enseñar? Madrid, España: Narcea. : s.n., 2009.
9. **Contreras Jordán, O. R.** Didáctica de la educación física. Un enfoque constructivista. Barcelona, España: INDE. : s.n., 1998.
10. Planificación-Conceptos Básicos. [En línea] http://www.elprisma.com/apuntes/administracion_de_empresas/planificacion/default2.asp.
11. Concepto de planificación. [En línea] <http://definicion.de/?s=Concepto+de+planificaci%C3%B3n>.
12. **GARCÍA, MIGUEL.** PLANIFICACIÓN (DEFINICIÓN – ORIGEN – PRINCIPIOS - ETAPAS – PARADIGMAS). 2005.
13. Concepto de planificación educativa. [En línea] <http://es.scribd.com/doc/67124208/EI-Concepto-de-Planificacion-Educativa>.
14. **Romero, Yibetza.** [En línea] 9 de Marzo de 2010. <http://www.monografias.com/trabajos53/tecnologia-comunicacion/tecnologia-comunicacion.shtml>.
15. **Dr. Fernando Vera** . [En línea] 17 de julio de 2006. <http://trabajosfernandovera.blogspot.com/2006/07/gestin-escolar-precisando-el-concepto.html>.
16. Gestión docente. [En línea] <http://www.monografias.com/trabajos94/gestion-educativa/gestion-educativa.shtml>.
17. [En línea] 16 de octubre de 2009. <http://www.educando.edu.do/articulos/directivo/la-gestin-educativa/>.
18. [En línea] http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_para_el_desarrollo_de_software.
19. **S. Pressman, Roger.** Ingeniería de software. Un enfoque práctico. 6. 4.
20. **Brito Acuña, Kareny.** Por qué utilizar Scrum para desarrollar aplicaciones web. [En línea] 2009. <http://www.eumed.net/libros/2009c/584/por%20que%20utilizar%20Scrum%20para%20desarrollar%20aplicaciones%20web.htm>.
21. **Jeffries, R., Anderson, A. y Hendrickson, C.** Extreme Programming Installed. s.l. : Addison-Wesley, 2001.
22. **Macías Rodríguez, Juan y Vázquez Rodríguez, Annarelys.** Estudio de Sistemas de Gestión de Contenidos basados en lenguaje PHP. Trabajo de Diploma para optar por el título de Ingeniero en ciencias informáticas. 2007.
23. [En línea] 5 de marzo de 2012. <http://foros.monografias.com/showthread.php/60249-Caracteristicas-del-PHP...>
24. **Negrino, Tom y Smith, Dori.** Guía de aprendizaje JavaScript. 3. Madrid : PEARSON EDUCACIÓN, S.A., 2000.
25. Jquery write less, do more. [En línea] 2012. <http://jquery.com/>.
26. Pancuca. [En línea] Febrero de 2013. <http://www.pancuca.com/?topic=926>.

27. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** El Lenguaje Unificado de Modelado. Manual de referencia. [Documento] Madrid : s.n., 2000.
28. Introduction to OMG UML. [En línea] 2009. http://www.omg.org/gettingstarted/what_is_uml.htm.
29. Symfony Web Framework. [En línea] <http://www.symfony-project.org>.
30. Visual-paradigm.com. UML tool, business process modeler and database designer for software development team. [En línea] 2010. <http://www.visual-paradigm.com/>.
31. **autores, Colectivo de.** Entorno Virtual de Aprendizaje. [En línea] 2008. <http://eva.uci.cu/mod/resource/view.php?inpopup=true&id=16179>.
32. Netpecos.org. PostGreSQL. netpecos.org. [En línea] 2007. http://www.netpecos.org/docs/mysql_postgres/x15.html..
33. MySQL con Clase. Mysql-conclase. [En línea] 2005. <http://mysql.conclase.net/>.
34. Ciberneta. Conceptos básicos del servidor web. [En línea] 2009. http://www.ciberneta.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php.
35. Arquitectura. [En línea] <http://www.csi.map.es/csi/silice/Global71.html>..
36. **POTENCIER, F.; ZANINOTTO, F.** Symfony la guía definitiva. [En línea] 2008. http://www.librosweb.es/symfony_1_2.
37. **Saavedra, Jorge.** El mundo informático. [En línea] 8 de Mayo de 2007. [http://jorgesaaavedra.wordpress.com/category/patrones-grasp/...](http://jorgesaaavedra.wordpress.com/category/patrones-grasp/)
38. Diagrama de Despliegue. [En línea] abril de 2013. <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc>..
39. **Escribano, G. F.** eXtreme Programming / Programación Extrema. [En línea] 2002. <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>.

Bibliografía

1. **Ullastres, Marchesi.** Presente y futuro de la Reforma educativa en España. Revista Iberoamericana. [En línea] 2001. <http://www.rieoei.org/rie27a03.htm>.
2. **LA REFLEXIÓN DE LA PRÁCTICA COMO ELEMENTO DE MEJORA DEL DESEMPEÑO ACADÉMICO DE LOS COLECTIVOS DOCENTES EN EDUCACIÓN BÁSICA.** **HUERTA PÉREZ, PATRICIA** . XI Congreso Nacional de Investigación Educativa / 15. Procesos de Formación / Ponencia.
3. **Méndez, R. M.** *La actitud de los estudiantes hacia la universidad. Tesis doctoral.* España: Universidad de Santiago de Compostela : s.n.
4. **Wass, S.** *Salidas escolares y trabajo de campo en la educación primaria.* Madrid, España : s.n., 1992.
5. **Gal, Le.** *Los derechos del niño en la escuela. Una educación para la ciudadanía.* Barcelona, España : s.n., 2005.
6. **Thurler, Gather.** *Innovar en el seno de la institución escolar.* Barcelona, España : s.n., 2004.
7. **Bolívar, A.,.** *La asesoría al colectivo docente y el trabajo en el aula. En Secretaría de Educación.* México : s.n., 2006.
8. **Marcelo, C. & Vaillant, D.** *Desarrollo profesional ¿Cómo se aprende a enseñar?* Madrid, España: Narcea. : s.n., 2009.
9. **Contreras Jordán, O. R.** *Didáctica de la educación física. Un enfoque constructivista.* Barcelona, España: INDE. : s.n., 1998.
10. Planificación-Conceptos Básicos. [En línea] http://www.elprisma.com/apuntes/administracion_de_empresas/planificacion/default2.asp.
11. Concepto de planificación. [En línea] <http://definicion.de/?s=Concepto+de+planificaci%C3%B3n>.
12. **GARCÍA, MIGUEL.** *PLANIFICACIÓN (DEFINICIÓN – ORIGEN – PRINCIPIOS - ETAPAS – PARADIGMAS).* 2005.
13. Concepto de planificación educativa. [En línea] <http://es.scribd.com/doc/67124208/EI-Concepto-de-Planificacion-Educativa>.
14. **Romero, Yibetza.** [En línea] 9 de Marzo de 2010. <http://www.monografias.com/trabajos53/tecnologia-comunicacion/tecnologia-comunicacion.shtml>.
15. **Dr. Fernando Vera** . [En línea] 17 de julio de 2006. <http://trabajosfernandovera.blogspot.com/2006/07/gestin-escolar-precisando-el-concepto.html>.
16. Gestión docente. [En línea] <http://www.monografias.com/trabajos94/gestion-educativa/gestion-educativa.shtml>.
17. [En línea] 16 de octubre de 2009. <http://www.educando.edu.do/articulos/directivo/la-gestin-educativa/>.
18. [En línea] http://wiki.monagas.udo.edu.ve/index.php/Metodolog%C3%ADas_para_el_desarrollo_de_software.
19. **S. Pressman, Roger.** *Ingeniería de software. Un enfoque práctico.* 6. 4.
20. **Brito Acuña, Kareny.** *Por qué utilizar Scrum para desarrollar aplicaciones web.* [En línea] 2009. <http://www.eumed.net/libros/2009c/584/Por%20que%20utilizar%20Scrum%20para%20desarrollar%20aplicaciones%20web.htm>.
21. **Jeffries, R., Anderson, A. y Hendrickson, C.** *Extreme Programming Installed.* s.l. : Addison-Wesley, 2001.
22. **Macias Rodríguez, Juan y Vázquez Rodríguez, Annarelys.** *Estudio de Sistemas de Gestión de Contenidos basados en lenguaje PHP. Trabajo de Diploma para optar por el título de Ingeniero en ciencias informáticas.* 2007.
23. [En línea] 5 de marzo de 2012. <http://foros.monografias.com/showthread.php/60249-Caracteristicas-del-PHP...>
24. **Negrino, Tom y Smith, Dori.** *Guía de aprendizaje JavaScript.* 3. Madrid : PEARSON EDUCACIÓN, S.A., 2000.
25. *Jquery write less, do more.* [En línea] 2012. <http://jquery.com/>.
26. *Pancuca.* [En línea] Febrero de 2013. <http://www.pancuca.com/?topic=926>.
27. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado. Manual de referencia.* [Documento] Madrid : s.n., 2000.

28. Introduction to OMG UML. [En línea] 2009. http://www.omg.org/gettingstarted/what_is_uml.htm.
29. Symfony Web Framework. [En línea] <http://www.symfony-project.org>.
30. Visual-paradigm.com. *UML tool, business process modeler and database designer for software development team*. [En línea] 2010. <http://www.visual-paradigm.com/>.
31. **autores, Colectivo de.** Entorno Virtual de Aprendizaje. [En línea] 2008. <http://eva.uci.cu/mod/resource/view.php?inpopup=true&id=16179>.
32. Netpecos.org. *PostGreSQL*. *netpecos.org*. [En línea] 2007. <http://www.netpecos.org/docs/mysql_postgres/x15.html>..
33. MySQL con Clase. *MySQL-conclase*. [En línea] 2005. <<http://mysql.conclase.net/>>.
34. Ciberneta. Conceptos básicos del servidor web. [En línea] 2009. http://www.ciberneta.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php.
35. Arquitectura. [En línea] <http://www.csi.map.es/csi/silice/Global71.html>..
36. **POTENCIER, F.; ZANINOTTO, F.** Symfony la guía definitiva. [En línea] 2008. http://www.librosweb.es/symfony_1_2.
37. **Saavedra, Jorge.** El mundo informático. [En línea] 8 de Mayo de 2007. [http://jorgesaaavedra.wordpress.com/category/patrones-grasp/...](http://jorgesaaavedra.wordpress.com/category/patrones-grasp/)
38. Diagrama de Despliegue. [En línea] abril de 2013. <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc..>
39. **Escribano, G. F.** eXtreme Programming / Programación Extrema. [En línea] 2002. <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>.
40. **Issi Camy, Lázaro.** *JavaScript*. Madrid : ANAYA MULTIMEDIA, 2002. ISBN 84-415-1384-8.
41. **Bibeault, Bear y Katz, Yehuda.** *JQuery in Action*. Greenwich : Manning Publications Co., 2008.
42. deConceptos.com. [En línea] febrero de 2013. <http://deconceptos.com/general/informacion>.
43. Definición ABC. [En línea] <http://www.definicionabc.com/general/gestion.php>.
44. Definicion.de. *Definicion.de*. [En línea] <http://definicion.de/informacion/>.
45. Ecured. [En línea] http://www.ecured.cu/index.php/Visual_Paradigm.
46. Qué es Visual Paradigm. *Sitio de descargas de software*. [En línea] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
47. **Paez, Urdaneta.** *¿Qué es la gestión de Información?*. INFOLAC. 1990. Tesis de Licenciatura , Departamento de Bibliotecología y Ciencia de la Información, Universidad de La Habana.
48. Ecured. [En línea] http://www.ecured.cu/index.php/Gestión_de_la_Información.
49. **Ponjuán, G.** Gestión de la Información: dimensiones e implementación para el éxito organizacional. Rosario. 2004.
50. **De La Cruz Comptis, Doralis y Lorié Montalvo, Giorbis Miguel.** Sistema automatizado para la gestión de información de los cursos optativos. La Habana, Cuba : s.n. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.
51. ECURED. [En línea] 2013. <http://www.ecured.cu/index.php/Symfony>.
52. ECURED. [En línea] 2013. http://www.ecured.cu/index.php/Modelo_Vista_Controlador.
53. EcuRed. [En línea] 2013. <http://www.ecured.cu/index.php/MySQL>.
54. johanatov . [En línea] <http://johanatov.blogspot.es/>.
55. Mastermagazine. [En línea] <http://www.mastermagazine.info/termino/7006.php>.

56. **Fonseca Cabrera, José Armando.** Sistema de Gestión de las Actividades Metodológicas para la Facultad Regional “Mártires de Artemisa”. Artemisa : Universidad de las Ciencias Informáticas Facultad Regional “Mártires de Artemisa”, 2012.
57. **Cruz Iñigo, Dayron.** Sistema de gestión de información de la facultad 8. Módulo para la gestión de la información docente. 2007. Trabajo de Diploma para optar por el título de Ingeniero Informático.
58. *Patterns of Enterprise Application Architecture.* **MARTIN, F.** s.l. : Addison-Wesley Professional, 2002.
59. **Escribano, G. F.** eXtreme Programming / Programación Extrema. [En línea] 2002. <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>.
60. [En línea] <http://definicion.de/informacion/>.
61. [En línea] enero de 2013. <http://www.tiposde.com/escritos/informacion/tipos-de-informacion.html>.
62. Gestion de la información. [En línea] 12 de 2012. <http://www.buenastareas.com/ensayos/Gestion-De-La-Informacion/6783774.html>.
63. concepto de informacion. [En línea] <http://definicion.de/informacion/>.