



Universidad de las Ciencias Informáticas
Centro de Informatización Universitaria
Facultad 1

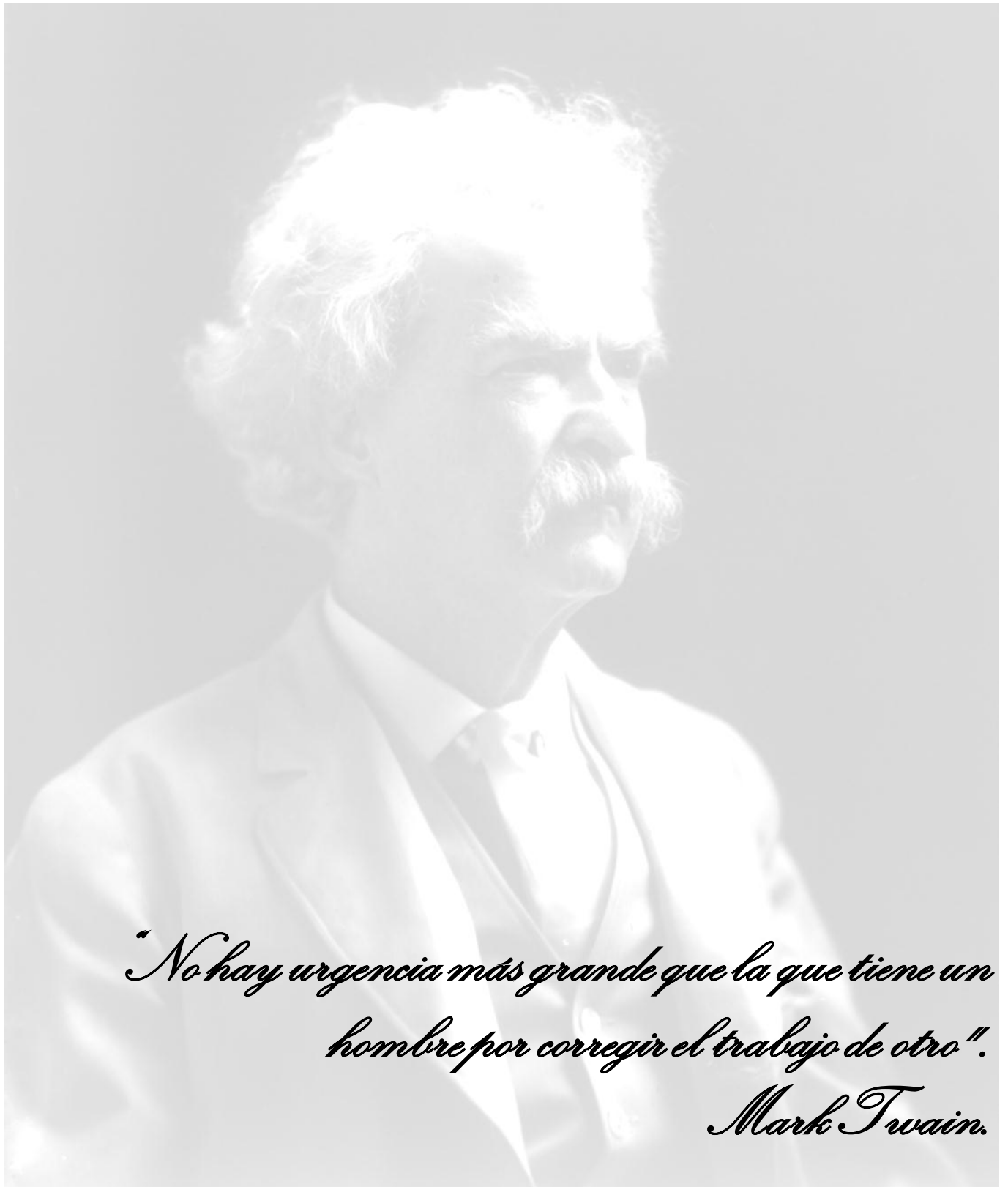
Propuesta de solución para la planificación de las actividades docentes en el Sistema de Gestión Académica de Pregrado

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Leandro Daniel Pérez Tamayo

Tutor: Ing. Osiris Perez Moya

La Habana, Cuba
Junio 2012



"No hay urgencia más grande que la que tiene un hombre por corregir el trabajo de otro".

Mark Twain.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2012.

Leandro Daniel Pérez Tamayo

Firma del autor

Ing. Osiris Perez Moya

Firma del tutor

AGRADECIMIENTOS

A toda mi familia, en especial a mis padres por ser las personas más importantes de mi vida, a mi mamá por ser tan especial conmigo y estar en cada momento difícil apoyándome en todos los sentidos, a mi papá por su apoyo y soporte, a mi hermano por ser la fuerza que me hizo salir adelante en momentos difíciles a mi abuelita Ada por ser mi refugio, por su inocente sagacidad y por la fuerza de voluntad con que se enfrenta a cada problema, por ser un ejemplo a seguir, a mi abuelo porque su presencia como hombre honrado y laborioso aún vive en mi corazón.

A mi tutor Osiris por todo su empeño, esfuerzo y dedicación, y por ser además un amigo incondicional.

A todos los amigos que supieron estar ahí brindando su apoyo tanpreciado no solo durante estos cinco años sino durante toda mi vida, en especial a Manuel Alejandra Yaneixi Manuel Enrique Lauradys Yoan Gaspar José Ángel Giselle Wendy Geny Tania Ris Enrique Danilo Dreidys Javier Adis Salvador Yerandy Daylenis Arturo Damián al claustro de profesores que me mostraron el camino a seguir durante estos cinco años y a todos los que de una forma u otra han hecho realidad este sueño.

DEDICATORIA

A toda mi familia, en especial a mis padres por ser las personas más maravillosas del mundo, por ser mi guía, mi motivación y mi ejemplo siempre. A mi abuelita que es una persona muy especial en mi vida. Gracias a todos por su amor, comprensión, sacrificio y dedicación.

RESUMEN

El proceso de planificación de la docencia, como parte del proceso docente-educativo que se realiza en la educación superior para cada carrera, es de extrema importancia. En la Universidad de las Ciencias Informáticas este se realiza en la Dirección de planificación y control del proceso de formación docente. Sin embargo, el mecanismo actual por el que se gestiona y planifica la dosificación de las asignaturas es engorroso y demora en concretarse por la cantidad de reuniones y de recursos que consume. Como consecuencia de lo anterior, se ha desarrollado una propuesta de solución informática para el Sistema de Gestión Académica de Pregrado que permite la gestión de la planificación docente y la elaboración de un balance de carga adecuado, con el objetivo de realizar su ejecución de una manera rápida. Es importante destacar que para llevar a término dicha propuesta se utilizaron las herramientas y tecnologías definidas por el Centro de Informatización Universitaria.

Palabras Claves: educación superior, gestión académica, gestión universitaria, planificación docente, programación.

ÍNDICE

Introducción.....	10
Capítulo 1: Análisis de los elementos teóricos asociados al desarrollo de sistemas para la gestión académica	13
Introducción.....	13
1.1 Marco conceptual	13
1.2 Estudio de las tendencias relacionadas con la planificación docente en soluciones informáticas.....	14
1.3 Entorno de desarrollo	17
1.3.1 Proceso de desarrollo de software	17
1.3.2 Herramientas utilizadas en el entorno de desarrollo	20
1.4 Normativas del Ministerio de Educación Superior relacionadas con la planificación docente	22
Conclusiones parciales	26
Capítulo 2: Características del sistema	28
Introducción.....	28
2.1 Caracterización del entorno	28
2.2 Propuesta de solución	29
2.3 Técnicas para la obtención de requisitos	30
2.4 Requisitos funcionales	31
2.5 Requisitos no funcionales	42
2.6 Patrón arquitectónico.....	43
2.6.1 Patrón Modelo-Vista-Controlador, especificación de su utilización	43
2.7 Patrones de diseño	45
2.7.1 Patrones de asignación de responsabilidades de software	45
2.7.2 Patrones banda de los 4	47
2.8 Estándares para el diseño de prototipos	48

2.9 Vista de despliegue	53
2.10 Relación de la propuesta de solución con el Sistema de Gestión Académica de Pregrado ..	54
Conclusiones parciales	56
Capítulo 3: Validación de la solución propuesta	57
Introducción.....	57
3.1 Técnicas para validar los requisitos	57
3.2 Criterios de validación de requisitos del cliente y del producto	58
3.2.1 Resultado de la validación de los requisitos	59
3.3 Propuesta de pruebas	60
3.3.1 Diseño de casos de caso de pruebas.....	62
Conclusiones parciales	70
Conclusiones.....	71
Recomendaciones.....	72
Referencias bibliográficas.....	73
Anexo 1: Diagrama del proceso de negocio de la planificación docente con BPMN	76

Índice de tablas

Tabla 1. Descripción del requisito crear tarea del calendario docente	33
Tabla 2. Descripción del requisito crear período lectivo.	36
Tabla 3. Descripción del requisito crear P-1 de asignatura.....	38
Tabla 4. Caso de prueba definido para el requisito crear actividad del calendario docente.	63

Índice de figuras

<i>Fig. 1 Vista gestionar plan calendario docente.</i>	34
Fig. 2 Vista crear evento del plan calendario docente.	35
Fig. 3 Vista gestionar P-1 de asignatura.	37
Fig. 4 Vista crear período lectivo.	37
Fig. 5 Vista gestionar P-1 de asignatura.	39
Fig. 6 vista de crear P-1 de asignatura.	40
Fig. 7 Funcionamiento del patrón Modelo-Vista-Controlador (MVC) en el marco de trabajo GUUD.....	44
Fig. 8 Áreas de la vista de presentación.	48
Fig. 9 Áreas de la vista de presentación.	49
Fig. 10 Áreas de la vista de presentación.	50
Fig. 11 Mensaje de información.....	51
Fig. 12 Mensaje de error	52
Fig. 13 Mensaje de advertencia.....	52
Fig. 14 Mensajes de confirmación.	53
Fig. 15 Diagrama de despliegue.....	54
Fig. 16 Relación de los componentes del subsistema de Gestión Académica de Pregrado.	56
Fig. 17 Diagrama del proceso de negocio de la planificación docente con BPMN.....	76

Introducción

La universidad es el centro encargado de formar profesionales para asumir tareas específicas en pos del desarrollo social. En su interior, tienen lugar un grupo de procesos, llamados sustantivos, que en su conjunto contribuyen con la formación integral del estudiante. Entre estos se destaca el Proceso de Enseñanza Aprendizaje (PEA) que tiene un carácter sistémico y organizado.

El PEA, en su organización como sistema, requiere de una distribución adecuada de las diferentes asignaturas por semestres, semanas, días y horas lectivas. Para una adecuada distribución se debe tener en cuenta las exigencias de las disciplinas que se imparten a los estudiantes y su importancia para la especialidad; así como el tipo de enseñanza, la forma de organización de las actividades docentes y extradocentes que se realizan.

Un desbalance de estas actividades afecta la manera en que han de ser distribuidos los contenidos a lo largo de la formación del estudiante. La planificación docente y el balance adecuado entre estos procesos son elementos claves para cumplir con los principios y normas de la higiene de la actividad docente (Cleger *et al*, 2007).

En aras de facilitar la organización de la actividad docente, en la Universidad de las Ciencias Informáticas (UCI) se desarrolla el Sistema de Gestión Universitaria (SGU). Se trata de una aplicación informática que permitirá la integración de todos los procesos sustantivos de la universidad. El SGU contiene varios subsistemas entre los que se encuentra el Sistema de Gestión Académica de Pregrado. Este sistema permite tener el control de los procesos vinculados a la docencia y mejorar la ejecución de procesos relacionados con la formación del estudiante.

Actualmente el proceso de planificación docente de la UCI se realiza de forma manual, en este se encuentran involucradas varias áreas, encargadas de definir tareas que contribuyen con la formación académica, lo que genera una gran cantidad de reuniones y encuentros. El resultado final de la información que se solicita es la creación de un plan de eventos que contribuye a la formación de un calendario docente organizado. Este recopila todas las actividades a realizarse en el próximo curso escolar, situación que trae consigo la ralentización de la elaboración de un

correcto balance de carga docente que complemente la formación que una universidad de excelencia demanda. Además genera la movilización de gran cantidad de recursos para la obtención de un balance de carga adecuado. En la actualidad el SGU no cuenta con un módulo que integre la funcionalidad relacionada con la planificación de la dosificación docente de las asignaturas.

Teniendo en cuenta la situación problemática anteriormente descrita, se define el siguiente **problema científico de la investigación**: ¿Cómo facilitar la planificación de las actividades docentes en la Universidad de las Ciencias Informáticas, de forma tal que se reduzca el tiempo y el esfuerzo que conlleva la ejecución actual de este proceso?

Se determina como el **objeto de estudio** de la investigación el proceso de planificación docente, y se define como **objetivo general**: proponer una solución que facilite la realización del proceso de planificación de las actividades docentes de la Universidad de las Ciencias Informáticas y pueda ser integrada al Sistema de Gestión Académica de Pregrado.

Para dar cumplimiento al objetivo general, se plantean los siguientes **objetivos específicos**:

- Desarrollar el marco teórico conceptual de la investigación.
- Elaborar la propuesta de solución.
- Validar las funcionalidades identificadas.

Para guiar la investigación se defiende la siguiente **idea**: con la propuesta de una solución informática que facilite el proceso de planificación de las actividades docentes de la Universidad de las Ciencias Informáticas se reducirá el tiempo dedicado a la ejecución de este proceso.

Durante la investigación fueron empleados métodos empíricos y teóricos. Dentro de los métodos empíricos se utilizó la **entrevista**. Este método permitió conocer el funcionamiento de la planificación docente en la universidad. Las entrevistas se realizaron a los trabajadores de la dirección de planificación y de control de la información de la UCI. Fue utilizado además la **modelación** en la elaboración de la propuesta de solución y la **revisión documental**.

Dentro de los métodos teóricos se utilizó el **analítico-sintético** que permitió llegar a conclusiones a partir del estudio realizado del objeto de la investigación, construyendo ideas para generalizar la solución a partir de condiciones específicas de cada solución identificada en

el estudio de los sistemas homólogos de planificación docente. El **inductivo-deductivo** se utilizó en diferentes etapas, a partir de los artefactos o documentos de entrada se pudo deducir la distribución de las principales actividades, docentes y extradocentes. Permitió obtener una visión más acertada sobre cómo planificar la cantidad de semanas y horas lectivas por cada asignatura y elaborar el calendario docente. El uso de este método permitió la definición de una idea a defender que guía la investigación. El **histórico-lógico**, fue utilizado al analizar la evolución de la planificación docente, cómo se realiza en las universidades y cuáles son las tendencias actuales para el desarrollo de *software*.

Como resultados de la investigación se obtendrán los siguientes:

- Un análisis del estado del arte de las soluciones relacionadas con la planificación docente y sobre las tendencias en el desarrollo de sistemas de gestión docente.
- El diseño de solución de la funcionalidad para la planificación de actividades de formación del SGU.

La presente investigación se estructura en 3 capítulos, introducción, conclusiones, recomendaciones y referencias bibliográficas:

El Capítulo 1 “Análisis de los elementos teóricos asociados al desarrollo de sistemas para la gestión académica”, expone los principales conceptos utilizados en el transcurso de la investigación, así como la descripción referente al estado del arte de las herramientas definidas en el CENIA como entorno de trabajo para dar solución a problemas de planificación docente.

El Capítulo 2 “Propuesta de solución informática para la planificación de las actividades docentes” expone un análisis de la solución propuesta y una descripción del flujo actual del proceso de planificación docente. Además presenta una descripción del proceso que será gestionado por el Sistema de Gestión Universitaria y se muestran las funcionalidades identificadas.

El Capítulo 3 “Validación de la solución propuesta” expone el resultado de la aplicación de los diseños de los casos de pruebas y mecanismos utilizados para la validación de la propuesta.

Capítulo 1: Análisis de los elementos teóricos asociados al desarrollo de sistemas para la gestión académica

Introducción

En este capítulo se abordan las definiciones de los conceptos asociados al dominio del problema de la investigación. Además, se esboza el estudio realizado a algunos sistemas homólogos para analizar cómo gestionan la planificación docente. Por último, se realiza una descripción del entorno de trabajo utilizado para la elaboración de la propuesta de solución, y se definen las herramientas y tecnologías que se utilizan para la elaboración de la misma.

1.1 Marco conceptual

La planificación es un proceso primario en cualquier organización. Ayuda a recoger información de varias fuentes para resolver las dependencias, cambios o necesidades que emergen e imposibilitan obtener el objetivo definido.

Cleger y otros autores (2007) plantean que *“la planificación es una función administrativa que determina por anticipado cuáles son los objetivos que deben cumplirse y qué debe hacerse para alcanzarlos; por tanto, es un modelo teórico para actuar en el futuro. La planificación comienza por establecer los objetivos y detallar los planes necesarios para alcanzarlos de la mejor manera posible. Determina a dónde se pretende llegar, qué debe hacerse además del cómo, cuándo y en qué orden deben suceder los acontecimientos.”*

El diccionario de la lengua española define la planificación como el plan general, que se organiza para obtener un objetivo determinado (Real Academia Española, 2000).

Ackoff (1981) incluye en su definición que *“la planificación es el proceso para decidir antes de que se requiera la acción. Se anticipa a la toma de decisiones.”*

Los conceptos anteriormente analizados permiten afirmar que la planificación es el plan organizado que se anticipa a la toma de decisiones para lograr objetivos que deben cumplirse y

define además qué se debe hacer para alcanzarlos.

Uno de los procesos que utiliza la planificación como técnica de organización es la planificación docente dentro de la formación académica en las universidades. En un estudio realizado por Cleger y otros autores (2007) se plantea que *“la planificación docente es la actividad encargada de organizar adecuadamente el proceso de enseñanza para cumplir la función fundamental de la universidad, relacionada con la formación del profesional que la sociedad exige.”*

Otra definición de planificación docente afirma que se trata de un proceso que permite organizar el trabajo docente y facilita su evaluación. Con esta acción se aseguran espacios temporales para la creatividad, para la recomposición de estrategias o para la confirmación de las ya planteadas (Guardado, 2011).

Los aspectos anteriormente analizados permiten asumir en la investigación que la planificación docente es la forma de organizar adecuadamente el proceso de enseñanza para cumplir la función fundamental de la universidad facilitando su evaluación.

1.2 Estudio de las tendencias relacionadas con la planificación docente en soluciones informáticas

El proceso de planificación docente es muy amplio y utilizado en el mundo educacional. Estos abarcan una extensa variedad de actividades relacionadas con los centros educacionales, desde el nivel principal de la institución hasta los estudiantes. Con el desarrollo de las tecnologías de la informática y las comunicaciones, se han desarrollado aplicaciones que apoyan el proceso de planificación docente. A continuación se analizan algunos sistemas que apoyan este proceso en su ejecución:

El “Rediseño de procesos mediante un enfoque de red conversacional del caso planificación docente de la universidad de Talca”, una de las dieciséis universidades del consorcio de universidades estatales de Chile, del consejo de rectores de las universidades chilenas, está encaminado a resolver el problema de asignar salas y horarios a todas las actividades docentes, sin superposiciones.

La intención es desarrollar una herramienta que permita modelar y proponer un rediseño de la

actual estructura para una mejor planificación de las actividades docentes. La solución actual es el **Sistema de Gestión Curricular (SGC)**, fue desarrollada en el año 1997 con un importante impacto en el ámbito de gestión. Entre los procesos que el SGC incluye, se encuentran, el registro de las actividades académicas que desarrolla el alumno desde su ingreso a la universidad hasta su egreso y el proceso de planificación docente. El sistema fue diseñado utilizando los servicios que provee el sistema de gestión de bases de datos ORACLE bajo una arquitectura cliente-servidor para las labores administrativas y arquitectura web (Garrido, 2004). En realidad, el desafío está en cómo mejorar el mecanismo actual de asignación de recursos a las actividades, para que estas funcionen apropiadamente según plantean Castro e Inzunza Quitral (2010).

El rediseño del SGC está en desarrollo. Todavía no se conoce de la existencia de resultados concretos como consecuencia de su realización. Para la propuesta que se desea elaborar se tendrán en cuenta algunas de las funcionalidades presentes en el proceso de planificación docente del SGC referentes al trabajo con el control de la dosificación docente en los P-1 de las asignaturas. Sin embargo, se decide no adaptar este sistema al SGU porque las herramientas y tecnologías utilizadas no se ajustan al entorno de desarrollo definido por el Centro de Informatización Universitaria (CENIA) para su implementación.

El “**Sistema Informático de apoyo para la planificación didáctica para los docentes de educación básica (SAPDEP)**” realizado en la Universidad del Salvador de Centroamérica, es una aplicación de escritorio realizado en plataforma privativa. Desde el punto de vista de los requerimientos técnicos, el SAPDEP ha sido construido bajo el paradigma de la Programación Orientada a Objetos (POO), aplicando el patrón Modelo-Vista-Controlador (MVC), para desarrollar el sistema y para ser desplegado sobre tecnologías web.

La implementación del SAPDEB permitió reducir los costos de implementación de la metodología de aprendizaje por competencias. Algunas funcionalidades del sistema son: agregar, eliminar y modificar plan de estudio, así como crear, modificar y eliminar una planificación (Calderón *et al*, 2009).

Este sistema no fue usado por no cumplir con los estándares de desarrollo y con las políticas de

migración al *software* libre del Centro de Informatización Universitaria. Aunque se tendrán en cuenta las funcionalidades anteriormente enunciadas para la elaboración de la solución.

La Universidad de Oriente cuenta con el **Sistema Informático para la gestión de la planificación docente**. En el desarrollo de la solución, fue utilizado *Symfony* como marco de trabajo, por las ventajas que ofrece en cuanto a la automatización de las tareas comunes, y por las buenas prácticas de programación y patrones de diseño que sigue. Entre las principales funcionalidades implementadas en el sistema se encuentra: Solicitar la generación de una planificación docente. El sistema genera una planificación docente inicial, que servirá de apoyo al planificador para la toma de decisiones (Universidad de Oriente, 2010).

Esta funcionalidad se tendrá presente para el desarrollo de la propuesta. Este sistema no fue tomado en cuenta pues aunque fue desarrollado con tecnología libre no es posible integrarlo al marco de trabajo del Centro de Informatización Universitaria.

El **Sistema de Gestión de la Nueva Universidad (SIGENU)**, es una herramienta desarrollada en *Java* en el Instituto Superior Politécnico José Antonio Echevarría (CUJAE). Está formado por diferentes módulos entre los cuales se encuentra uno encargado de facilitar la gestión de los planes de estudio. Este módulo permite realizar el ajuste de los planes de estudio y programas analíticos de las asignaturas. El SIGENU, en su concepción, se diseñó en el ámbito de la secretaría del centro de estudio, esto quiere decir que otros usuarios que forman parte del ámbito académico como los profesores y los estudiantes no pueden interactuar con él.

En la UCI no sucede así, el proceso de planificación docente es ejecutado a nivel central, por la dirección de planificación docente, y todos los usuarios pueden interactuar con el sistema. No se opta por adoptar el sistema porque la solución no se ajusta a la integración del SGU que utiliza un entorno de desarrollo diferente y los anteriores tampoco cuentan con todas las funcionalidades necesarias para realizar el proceso de planificación docente en la UCI pues el negocio que implementa no coincide con el de aquí; aunque se tendrá en cuenta la funcionalidad: gestionar los modelos de planificación y control del proceso docente (CUJAE, 2011).

1.3 Entorno de desarrollo

Un entorno de desarrollo está compuesto por un conjunto de herramientas, además de la necesaria descripción del proceso de desarrollo de *software*. Un entorno de desarrollo llega a ser indispensable para una correcta implementación de una aplicación.

1.3.1 Proceso de desarrollo de software

El proceso de desarrollo de *software* tiene como propósito la feliz producción de un producto que cumpla con los requisitos del cliente (Universidad Politécnica de Valencia, 2009). Este está formado por un conjunto de actividades llevadas a cabo por los ingenieros de *software*. Algunas de las actividades que son comunes en el desarrollo de aplicaciones informáticas son mostradas a continuación:

- Especificaciones del *software*, basadas en restricciones definidas en reunión con el cliente, representado por la Dirección de planificación y Control del proceso de formación de Vicerrectoría de Formación de la UCI.
- Desarrollo del *software*, la propuesta de solución fue diseñada para lograr una correcta integración con el resto de los procesos concernientes a la formación del estudiante.
- Evolución del *software*, donde la propuesta de solución se ha modificado para adaptarlo a los cambios requeridos por el cliente en momentos determinados (Sommerville, 2005).

Estas actividades están cumplimentadas por varias fases entre las que se encuentran las propuestas por Pressman: la fase de definición de especificaciones que intenta identificar la información que ha de ser procesada, qué función y rendimiento se desea, comportamiento que tendrá el sistema, interfaces a ser establecidas y qué criterios de validación se necesitan para definir un sistema correcto.

La segunda fase es la de desarrollo, donde se define cómo han de diseñarse las estructuras de datos, traducir la propuesta a un lenguaje de programación y cómo han de realizarse las pruebas.

La última fase que propone Pressman es la de mantenimiento. En esta, el enfoque está dado según los cambios asociados a las correcciones de errores destacados, y a las adaptaciones

requeridas a medida que evoluciona el entorno del *software* (Pressman, 2005).

Algunos de los artefactos que se generan en el desarrollo de un *software* son:

- Definición del proceso del negocio utilizando notación BPMN como lenguaje de notación de marcado.
- Especificación de requisitos de *software*.
- Diseño de casos de prueba (Pressman, 2005).

Enfoque ágil en el desarrollo de software

La utilización del enfoque ágil queda evidenciada mediante uno de sus cuatro principios fundamentales, a su vez propósitos del Manifiesto Ágil:

- **El desarrollo se centra más en el individuo y las interacciones del equipo de desarrollo que en el proceso y las herramientas.** Las personas son el principal factor de éxito de un proyecto software. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que este configure su propio entorno de desarrollo según sus necesidades.
- **Centrarse en el desarrollo del software que funciona y no en conseguir una buena documentación.** La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos.
- **La colaboración con el cliente es más importante que la negociación a través de un contrato.** Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- **Responder a los cambios que ocurran antes de seguir estrictamente un plan.** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta (Canós *et al*, 2009).

Modelo de Madurez y Capacidad Integrado

El Modelo de Madurez y Capacidad Integrado (CMMI®, *Capability Maturity Model Integration*, según sus siglas en inglés) es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Lo que se pretende con el nivel dos de CMMI® es conseguir que en los proyectos de la organización exista una gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados.

Este nivel ayuda a asegurar que las prácticas existentes se mantienen durante tiempos de estrés. Cuando estas prácticas están en su lugar, los proyectos se realizan y gestionan de acuerdo a sus planes documentados. Las siete áreas de procesos que exige el nivel dos son:

Administración de Requerimientos (REQM). Tiene como propósito mantener bajo control los requerimientos del producto.

Planificación del Proyecto (PP). Esta área de proceso tiene como propósito establecer y mantener el plan que será empleado para ejecutar y monitorear el proyecto. El plan se desarrolla sobre la base de los requerimientos administrados por el área REQM.

Dentro de esta área de proceso se incluyen todas las actividades necesarias para determinar el alcance del proyecto, estimar el esfuerzo y los costos, establecer el cronograma, identificar riesgos, y obtener el compromiso de todos los involucrados respecto al plan de proyecto.

Monitoreo y Control del Proyecto (PMC). Esta área de proceso es complementaria y es una consecuencia de la Planificación del Proyecto (PP). Su propósito es monitorear la ejecución del proyecto empleando el plan, y gestionar acciones correctivas en el caso de detectarse desvíos.

Para poder cumplir con estos objetivos es necesario implementar prácticas de seguimiento, tales como: el reporte de horas trabajadas en el proyecto, el informe de avance periódico y las revisiones en puntos particulares del proyecto.

Medición y Análisis (MA). Esta área de proceso apunta a desarrollar y mantener capacidades de medición que permitan satisfacer las necesidades de información de la organización.

Aseguramiento de la Calidad de Productos y Procesos (PPQA). Debe garantizarse un nivel apropiado de independencia entre los productores y los evaluadores. El objetivo de esta área es proveer una evaluación objetiva de los procesos y de los artefactos producidos. Las prácticas de

esta área de proceso implican:

- Evaluar los procesos ejecutados, los artefactos producidos y los servicios provistos según los estándares y descripciones de proceso aplicables.
- Identificar las no conformidades, comunicarlas a los responsables, y asegurar su resolución.
- Informar a los interesados el resultado de las actividades de aseguramiento de la calidad.

Administración de la Configuración (CM). Esta área de proceso tiene como propósito mantener la integridad de todos los artefactos producidos por el proyecto, lo cual implica identificar los elementos de configuración, realizar sobre ellos cambios de manera controlada, generar y mantener las líneas bases, y proveer la información precisa sobre el estado de la configuración a todos los interesados.

Administración de Acuerdos con Proveedores (SAM). Esta área de proceso está originalmente pensada para todo lo relacionado con la adquisición de productos que vayan a ser incorporados en la solución a entregar al cliente, las prácticas incluidas aquí también sirven para todo aquello que sea necesario comprar pero que no será finalmente entregado al cliente, como por ejemplo las herramientas de desarrollo (Chrissis *et al*, 2009).

1.3.2 Herramientas utilizadas en el entorno de desarrollo

Las herramientas CASE son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de un *software*. Permiten modelar y documentar sus artefactos, y cubren todo el ciclo de vida del proceso de desarrollo.

Lenguaje de Modelado Unificado 2.0

El modelado visual permite manejar la complejidad de los sistemas a analizar o diseñar. Un modelo es expresado en un lenguaje de modelado compuesto por vistas, elementos de modelos y un conjunto de mecanismos generales o reglas que indican cómo utilizarlos.

El Lenguaje de Modelado Unificado (UML, *Unified Modeling Language*, según sus siglas en inglés), es un lenguaje para especificar, construir, visualizar y documentar artefactos generados en procesos de desarrollo de *software*. Este lenguaje de modelado permite la modelación de sistemas con tecnología orientada a objetos.

Su utilización es independiente del lenguaje de programación y de las características de los

proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos informáticos, de arquitectura o de cualquier otra rama.

UML es el lenguaje de modelado propuesto a utilizar debido a que presenta grandes ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado (Lamarca, 2009).

Visual Paradigm 8.4

El *Visual Paradigm* es una herramienta profesional que permite representar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Esta herramienta ofrece un entorno para crear diagramas utilizando los estereotipos de *UML 2.0*¹ lo que se refiere a un lenguaje estándar para todo el equipo de desarrollo, que permite la ingeniería directa e inversa, y está disponible para múltiples plataformas, permite generar imágenes y reportes de buena calidad (*Visual Paradigm Company, 2011*).

Algunas de las características de esta herramienta son las siguientes:

- *Visual Paradigm* utiliza la versión 2 de UML que utiliza los estereotipos de BPMN para modelar procesos, notación seleccionada para el modelado del negocio.
- Fácil de instalar y actualizar.
- Permite el diseño de prototipo de interfaz de usuario. Permite insertar información adicional a los diagramas mediante notas y comentarios para describir sus elementos lo que facilita la revisión de los prototipos (Lamarca, 2009).

Evolus Pencil 1.3.4

Es una herramienta libre y de código abierto para crear diagramas y prototipos de interfaz gráfica de usuario. *Evolus Pencil* permite diseñar las ventanas de los prototipos para *Windows* o *Linux*. Puede agregarse como complemento para el navegador *Mozilla Firefox*. Algunas de las características de esta herramienta son que permite:

¹ Se refiere al lenguaje de modelado unificado

- La construcción de prototipos.
- Tipo de funcionamiento: arrastrar y soltar (*Pencil Project*, 2010).

Notación de modelado de proceso de negocio

La Notación de Modelado de Proceso de Negocio (BPMN, por sus siglas en inglés). Es un lenguaje formal que permite modelar, simular y eventualmente ejecutar procesos de negocios. Proporciona además una notación gráfica para expresar los mismos mediante un diagrama de proceso de negocio.

También concreta el grupo de especificaciones propuestas por Iniciativa de Gestión de Procesos del Negocio (BPML, por sus siglas en inglés). No es utilizado para modelar aplicaciones sino procesos que corren dentro de esas aplicaciones. El modelado en BPMN se realiza mediante diagramas muy simples con un conjunto de elementos gráficos. Con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil comprender el flujo y el proceso. Las cuatro categorías básicas de elementos son:

- Objetos de flujo: eventos, actividades, rombos de control de flujo (decisiones y/o uniones).
- Objetos de conexión: flujo de secuencia, flujo de mensaje, asociación.
- Carriles: participantes, carriles.
- Artefactos: objetos de datos, grupos, anotación de texto.

Esta notación ha sido diseñada especialmente para coordinar la secuencia de procesos y mensajes que fluyen entre participantes de actividades distintas (Bizagi, 2012).

1.4 Normativas del Ministerio de Educación Superior relacionadas con la planificación docente

El Ministerio de Educación Superior, en la resolución 210 del Reglamento Metodológico del 2006 establece:

- En el **artículo 21**: El plan de estudio es el documento fundamental de carácter estatal que establece la dirección general y el contenido principal de la preparación del profesional. Está integrado por: el Modelo del Profesional, el Plan del Proceso Docente y

los Programas de Disciplina, definidos en los artículos 66, 67 y 68 del presente Reglamento.

El plan de estudio se elabora para cada carrera en correspondencia con las necesidades sociales existentes en el país, los avances científicos y tecnológicos de la época actual y las particularidades de la profesión en cuestión.

- **Artículo 65:** los documentos de carácter estatal o rectores que componen el plan de estudio son los siguientes:
 - El Modelo del Profesional.
 - El Plan del Proceso Docente.
 - Los Programas de las Disciplinas.

Además de estos tres documentos, las comisiones nacionales de carrera elaborarán también Indicaciones metodológicas y de organización en las que se precisen otros aspectos que contribuyan a su mejor interpretación y ejecución por los centros homólogos.

- **Artículo 77:** la planificación y la organización del proceso docente educativo constituyen dos de las funciones principales del trabajo metodológico en la educación superior. Se realizan antes del inicio del curso académico y de cada período docente en los centros de educación superior. Los planes de estudio de las carreras constituyen la base para la planificación y la organización del proceso docente educativo.

Esta labor garantizará la integración y sistematización de las actividades docentes para alcanzar los objetivos generales de la formación del profesional, para contribuir al adecuado balance de la carga docente del estudiante, y para el uso racional de los recursos humanos y materiales.

- **Artículo 78:** las fechas de inicio y de culminación del curso académico se establecen por el Ministerio de Educación Superior previa consulta con la dirección del país, comunicándolo a cada organismo de la administración central del estado con centros de educación superior adscritos.

A partir de esta información, los rectores de los centros de educación superior se responsabilizarán con la planificación y organización de cada curso y período académico. Con este fin se precisarán las tareas a desarrollar, los documentos a elaborar, los plazos para su

cumplimiento y la fecha correspondiente a cada nivel, garantizando que todo el proceso organizativo y de planificación esté concluido antes de comenzar cada período docente.

Cada organismo dictará las indicaciones específicas que se requieran para la planificación y organización del curso y período académico.

Los directores de sedes universitarias se responsabilizarán con la instrumentación de la planificación y organización de cada curso y período académico a partir de las indicaciones que emanen de la sede central.

- **Artículo 79:** la organización del proceso docente educativo se realizará por períodos, que pueden ser, entre otros: semestres, trimestres, bloques, módulos y estancias. Es potestad del Rector de cada centro de educación superior definir la forma de organizar el proceso docente educativo para cada carrera, tanto en la sede central como en las sedes universitarias.
- **Artículo 80:** los profesores, con independencia y creatividad, a partir del programa analítico de la asignatura, desarrollarán la preparación de cada una de las actividades docentes.
- **Artículo 156:** la planificación, organización y elaboración de las evaluaciones frecuentes será responsabilidad del profesor, quien decidirá qué evaluaciones serán calificadas y registradas en la libreta de control a clases.
- **Artículo 157:** la planificación, organización y elaboración de las evaluaciones parciales se proponen por el profesor o el colectivo de asignatura, en caso de existir, y se aprueban en el departamento docente correspondiente.

El balance de las evaluaciones parciales propuestas en las asignaturas que se desarrollan en cada período lectivo se ajustará en la facultad o sede universitaria, para lograr un balance adecuado y serán aprobadas por el decano o director de la sede, siempre con la participación de la representación estudiantil (MES, 2006).

1.5 Sistema de Gestión Universitaria. Subsistema de Gestión Académica de

Pregrado

El Subsistema de Gestión de Pregrado contiene los procesos de diseño y gestión de la carrera, actividades de secretaría, registro y control docente, la gestión de los trabajos de diploma y los títulos así como la planificación y control del proceso docente. Algunos de los procesos de apoyo que abarca son: estructura y composición de las diferentes áreas de la entidad o centro de estudios, gestión del personal, seguridad, configuración del sistema, reportes y estadísticas asociadas al resultado de la gestión de información de los diferentes procesos. Entre las facilidades que presenta el sistema están:

- Que involucra a todos los actores del proceso en el sistema.
- Es posible gestionar más de una carrera en la universidad.
- Permitirá la gestión flexible de varios planes de estudio.
- Es un sistema más flexible, configurable y fácil de usar.
- Permitirá la gestión de la estructura del área de formación.
- Permitirá la gestión de los trabajos de diplomas y los títulos.
- Permitirá la gestión de los perfiles de la carrera.
- Permitirá la integración de los sistemas que gestionan los procesos de apoyo separados.
- Permitirá la gestión de profesores guías, jefes de brigada y alumnos ayudantes.
- Permitirá la evaluación de alumnos ayudantes y profesores.
- Permitirá la caracterización pedagógica por los estudiantes a los profesores.
- Permitirá la gestión de bajas, traslados y licencias.
- Permitirá la gestión de eventos.
- Permitirá la integración y control de actividades de formación.

Algunos de los módulos que integrarán el SGU son:

Carrera: permitirá gestionar el proceso de organización de las carreras definidas por el MES, acorde a las necesidades y características de la entidad, (planes de estudio, homologación de plan de estudio y solicitudes de cambio) además de realizar ajustes personalizados a un estudiante o grupo de estudiantes.

Personal y secretaría: permitirá gestionar todo el personal vinculado con los procesos de

pregrado. Posibilitará la realización de movimientos de un estudiante (traslados, bajas, licencias), así como acciones de secretaría (promoción de estudiantes, registro de datos docentes para los profesores, solicitudes de movimiento, inscripción de alumnos ayudantes).

Registro y control docente: permitirá gestionar todo el registro de las evaluaciones docentes y la asistencia de los estudiantes, asignar profesores a las diferentes estructuras docentes, establecer un balance de la carga docente de los profesores así como posibilitar un mejor control de los profesores por facultad y grupo.

Tesis y títulos: permitirá gestionar todo el proceso de desarrollo de una tesis de pregrado (asignación, seguimiento y control) y el otorgamiento de los títulos.

Planificación docente: permitirá gestionar la planificación macro de las actividades predeterminadas que ocurren regularmente en un curso académico. Posibilitará asignarle a cada actividad un período de tiempo en semanas, de forma tal que quede conformada una planificación general del curso donde se vean reflejados los momentos de mayor envergadura. Dentro de las funcionalidades del módulo están:

- Planificación del horario docente.
- Planificación de actividades de formación. La presente investigación se centra en esta funcionalidad.

Reportes: permitirá brindar información tanto nominal, cuantitativa y gráfica a partir de los datos generados de todo el proceso de gestión a nivel general, quedando identificados en cuatro grandes grupos como son: reportes de matrícula, indicadores cuantitativos, resultados docentes y movimientos.

Conclusiones parciales

El estudio del proceso de planificación de las actividades docentes en la UCI, permitió conocer las principales características del mismo, que comprende como uno de sus elementos el control docente para el profesional que la sociedad cubana requiere. El estudio de los sistemas homólogos analizados permitió adquirir el conocimiento sobre las características fundamentales que estos poseen, los cuales no satisfacen todas las necesidades y condiciones requeridas, sin

embargo, constituyen una base de estudio para llegar a la obtención de la propuesta de la nueva solución. El estudio de las herramientas, lenguajes y proceso de desarrollo definidos por el CENIA, permitió la familiarización con los elementos del entorno de desarrollo y adquirir los conocimientos necesarios, para poder utilizarlos en el desarrollo de la propuesta de solución.

Capítulo 2: Características del sistema

Introducción

El presente capítulo tiene como objetivo describir la solución propuesta de acuerdo con el problema planteado. Se detalla la necesidad de desarrollar el sistema. Se explica el modelo de dominio y los conceptos asociados a este. Además, se definen los requisitos funcionales y los no funcionales que deberá cumplir la aplicación, así como las reglas que se seguirán.

2.1 Caracterización del entorno

Actualmente el proceso para planificar las actividades docentes inicia cuando el gestor de la planificación docente recibe a través de la Vicerrectoría de Formación indicaciones que el Ministerio de Educación Superior define para el próximo curso escolar. Generando un documento con las principales indicaciones sobre las actividades del próximo curso escolar. Recibe además otras definiciones de la universidad como son el pase de fin de año, la planificación del Trabajo Socialmente Útil (TSU), la Reunión de Estudios militares (REM), con esto se elabora una versión preliminar del calendario docente.

Con esta primera versión del calendario docente, la Vicerrectoría de Formación convoca a una reunión con las principales áreas de la universidad. Estas áreas son Vicerrectoría de Extensión Universitaria, Vicerrectoría de Investigación, representantes de la Federación Estudiantil Universitaria (FEU) y Unión de Jóvenes Comunistas (UJC). En esta reunión queda organizado el plan de eventos docentes, políticos, deportivos y culturales para el próximo curso, con este plan se trata de que las fechas definidas tengan el menor número de coincidencias.

Una vez que el calendario docente es aprobado se pone a disposición de los jefes de departamentos docentes centrales junto a un documento que recoge algunas precisiones sobre el plan calendario docente a tener en cuenta y el cronograma para la planificación del próximo semestre. Estas precisiones están asociadas con el ajuste de las dosificaciones de las asignaturas. A partir de aquí se ajustan las versiones iniciales de la dosificación docente en el

modelo de planificación y control del proceso docente (P1). El cronograma se envía a los departamentos docentes centrales, donde los jefes convocan a una reunión con los asesores de asignaturas de los departamentos docentes centrales y los jefes de años para ajustar el balance del contenido y las evaluaciones parciales de las asignaturas por años. De esta reunión se genera el sistema de evaluaciones de las asignaturas ajustado para la versión final de P1, que es enviado posteriormente al gestor de planificación docente.

El gestor de la planificación docente elabora el balance de carga. En el anexo 1 se representa el proceso anteriormente descrito.

Reglas del negocio

- El proceso de planificación inicia con las indicaciones del MES.
- El calendario docente es aprobado por la Vicerrectoría de formación.
- Las actividades no docentes del curso las definen las áreas de la universidad: Vicerrectoría de Extensión Universitaria, FEU, UJC.
- Si hay más de dos evaluaciones en una misma semana la dosificación docente de las asignaturas deberá ser modificada.

2.2 Propuesta de solución

La propuesta de solución incluye la definición de los requerimientos para una agrupación funcional de planificación docente, centrándose específicamente en la planificación de las actividades docentes que generan como salida el plan calendario docente y el balance de carga de las asignaturas. Para la identificación de los requisitos se analizó el reglamento docente del MES en los basamentos de la resolución ministerial 210. Se identificaron 41 requisitos funcionales, puestos a disposición del cliente para llegar a un acuerdo, para esto se realizaron varias entrevistas abiertas mediante las cuales se pudo obtener una vista acertada sobre la ejecución del proceso docente en la UCI.

2.3 Técnicas para la obtención de requisitos

La ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación, y administrar los requisitos conforme son transformados en un sistema operacional (Pressman, 2005). A continuación se explican las características de las técnicas que se aplicaron para identificar los requisitos de la solución que se propone:

Entrevista

Las entrevistas formales e informales con los *stakeholders*² del sistema son una parte importante en la mayoría de los procesos de la ingeniería de requerimientos. En estas entrevistas, el equipo de la ingeniería de requerimientos hace preguntas a los *stakeholders* sobre el sistema que utilizan y sobre el sistema a desarrollar. Los requerimientos provienen de las respuestas a estas preguntas. El tipo de entrevistas utilizadas en la presente investigación fue: **Abiertas**, en las que no existió un programa predefinido. Fueron examinados una serie de cuestiones con los *stakeholders* del sistema y por lo tanto, fue posible desarrollar una mejor comprensión de sus necesidades (Sommerville, 2005).

Prototipos

Durante la actividad de extracción de requerimientos, puede ocurrir que algunos requerimientos no estén claros o que no se esté muy seguro de haber entendido correctamente los requerimientos obtenidos hasta el momento, estos elementos pueden llevar al traste el producto final.

Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiendo conseguir una importante retroalimentación en cuanto a si el sistema diseñado con base a los requerimientos recolectados le permite al usuario realizar su trabajo.

El desarrollo del prototipo comienza con la captura de requerimientos. Desarrolladores y clientes

² Se refiere a los involucrados en el proceso de desarrollo de software.

se reúnen y definen los objetivos generales del *software*, identifican todos los requerimientos que son conocidos, y señalan áreas en las que será necesaria la profundización en las definiciones. Luego tiene lugar un “diseño rápido”. El diseño rápido se centra en una representación de aquellos aspectos del *software* que serán visibles al usuario. El diseño rápido lleva a la construcción de un prototipo (Chaves, 2006).

2.4 Requisitos funcionales

Los requisitos funcionales son la definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada de datos y cómo se debe comportar ante situaciones particulares. Describen el funcionamiento del sistema. Los requisitos funcionales del usuario pueden ser frases muy generales sobre lo que el sistema debería hacer. Se suelen expresar como objetivos del sistema. Los requisitos funcionales del sistema deben describir los servicios que hay que proporcionar con todo detalle (Pressman, 2005).

Los requisitos funcionales están descritos en el artefacto Especificación de Requisitos de *Software* en el expediente de proyecto del SGU. Este documento tiene en cuenta un grupo de aspectos que ayudan en la descripción de los requisitos. La descripción de los requisitos le facilita al programador entender lo que debe hacer para implementar las funcionalidades. Entre los aspectos presentes en el documento se encuentran:

- **Número:** utilizado para reconocer los requisitos.
- **Nombre:** nombre que va a adoptar el requisito.
- **Descripción:** breve descripción del requisito.
- **Pre-condiciones:** condiciones que deben ocurrir antes para que se pueda ejecutar la funcionalidad.
- **Complejidad:** puede ser alta, media o baja según el grado de complejidad que tenga el requisito.
- **Prioridad para el cliente:** puede ser alta, media o baja, lo cual da la medida de cuán importante es el requerimiento para el sistema, siendo lógicamente los que tienen prioridad alta, que son imprescindibles para el funcionamiento del mismo, media que no

son tan necesarias, mientras que los que tienen prioridad baja son condiciones que aún sin ellas el sistema podría funcionar.

- **Prototipo:** donde se incluye la interfaz de usuario.
- **Campos:** van a ser los campos existentes en el prototipo.
- **Tipo de datos, las reglas o restricciones del negocio:** donde se describe lo que debe cumplirse o tenerse en cuenta, además de los valores válidos de los datos.

A continuación se listan los requisitos funcionales identificados.

RF 1: crear tarea del plan calendario docente.

RF 2: modificar tarea del plan calendario docente.

RF 3: eliminar tarea del plan calendario docente.

RF 4: buscar tarea del plan calendario docente.

RF 5: mostrar tarea del plan calendario docente.

RF 6: mostrar plan calendario docente.

RF 7: exportar a PDF plan calendario docente.

RF 8: publicar plan calendario docente.

RF 9: finalizar plan calendario docente.

RF 10: aprobar plan calendario docente por el Vicerrector de Formación Docente.

RF 11: aprobar plan calendario docente por el Rector.

RF 12: exportar a excel plan calendario docente.

RF 13: permitir realizar la búsqueda avanzada de evento del plan calendario docente.

RF 14: crear año académico.

RF 15: eliminar año académico.

RF 16: crear período lectivo.

RF 17: eliminar período lectivo.

RF 18: crear P1 de asignatura.

RF 19: modificar P1 de asignatura.

RF 20: eliminar P1 de asignatura.

RF 21: mostrar P1 de asignatura.

- RF 22:** crear tema asignatura.
- RF 23:** modificar tema asignatura.
- RF 24:** eliminar tema asignatura.
- RF 25:** mostrar tema asignatura.
- RF 26:** crear actividad docente.
- RF 27:** modificar actividad docente.
- RF 28:** mostrar actividad docente.
- RF 29:** eliminar actividad docente.
- RF 30:** editar sistema de evaluación.
- RF 31:** exportar a PDF (P-1) de asignatura.
- RF 32:** mostrar balance de carga docente.
- RF 33:** exportar a PDF balance de carga docente.
- RF 34:** exportar a excel balance de carga docente.
- RF 35:** imprimir balance de carga docente.
- RF 36:** publicar balance de carga docente.
- RF 37:** guardar observación.
- RF 38:** publicar (P-1) de asignatura.
- RF 39:** mostrar distribución de las semanas del curso.
- RF 40:** mostrar distribución del TSU.
- RF 41:** publicar distribución de semanas por año.

A continuación se muestran algunos ejemplos de descripciones de requisitos seleccionados a partir del criterio de prioridad para el cliente, tomándose como ejemplo los requisitos con prioridad alta:

Tabla 1. Descripción del requisito crear tarea del calendario docente

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF1.]	Crear tarea del calendario docente.	1. El usuario debe estar autenticado con el rol	Media	Alta

		<p>“Gestor de la planificación docente”.</p> <p>2. El usuario debe encontrarse en la ventana “Gestionar Calendario Docente”.</p> <p>3. El usuario debe haber seleccionado la opción “Crear tarea del calendario docente”.</p>	
--	--	---	--

Prototipo

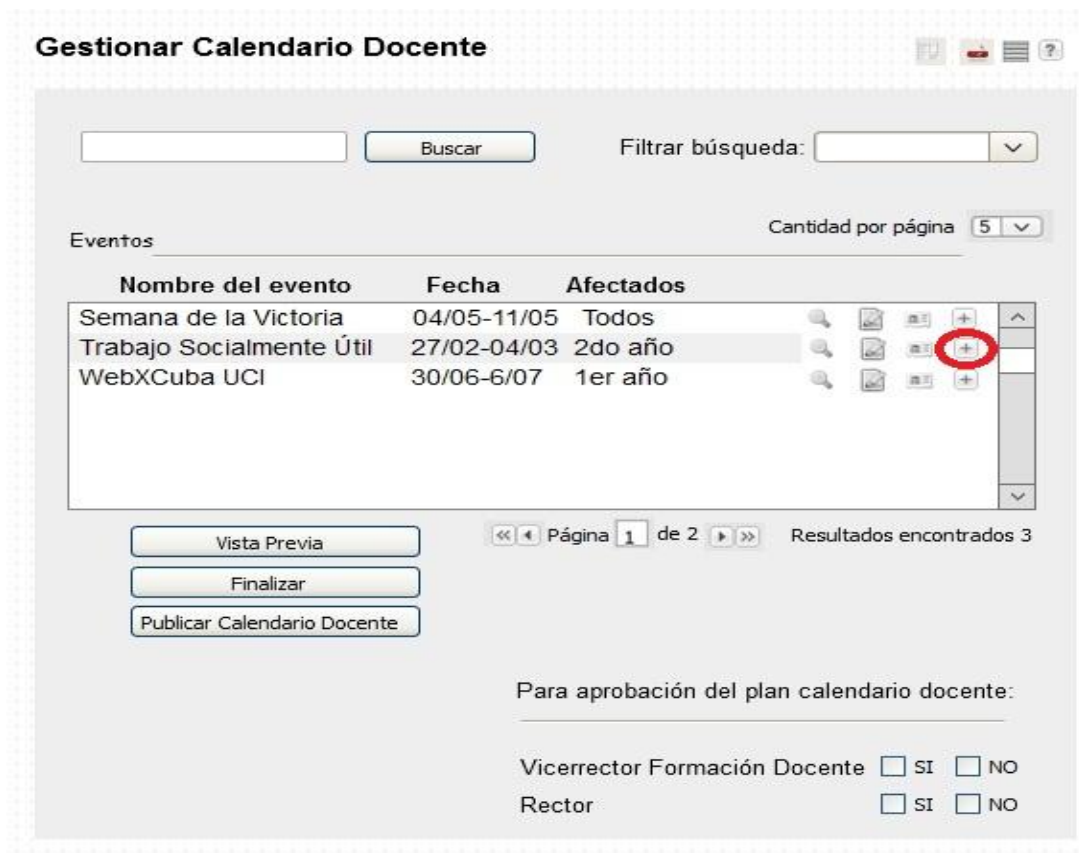


Fig. 1 Vista gestionar plan calendario docente.



Fig. 2 Vista crear evento del plan calendario docente.

Campos	Tipos de Datos	Reglas o Restricciones
Nombre evento	<i>character varying</i>	Admite de 2 a 50 caracteres entre letras, números y <i>underscores</i> . La cantidad de caracteres permitidos para una palabra es de 30. Campo obligatorio.
Fecha Inicio	<i>Datetime</i>	Campo obligatorio.
Fecha Fin	<i>Datetime</i>	Campo obligatorio.
Año académico	<i>character varying</i>	Campo obligatorio
Asignado por	<i>character varying</i>	Campo obligatorio
Observaciones		
<ol style="list-style-type: none"> 1. En caso que el elemento ya exista se muestra un mensaje de error “La tarea ya existe”. 2. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de cancelar la acción?”. 3. En caso de que introduzca datos incorrectos se muestra un mensaje de error 		

	<p>“El elemento se ha introducido incorrectamente”.</p> <p>4. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo “Campo requerido” en el campo que debe ser llenado obligatoriamente.</p> <p>5. Interactúa con esta acción el administrador y gestor de planificación docente.</p> <p>6. En el campo asignado, si se selecciona MES o Rectoría la tarea se convierte en fija, no puede ser modificada ni movida de lugar, por lo que en su representación se mostrará con un color diferente que la identifica, las demás tareas el sistema no se le asigna color. En cualquiera de los otros casos puede ser modificada con posterioridad según sea conveniente.</p>
--	--

Tabla 2. Descripción del requisito crear período lectivo.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RF	Crear período lectivo.	<p>El usuario debe estar autenticado con el rol “Jefe departamento central”.</p> <p>El usuario debe encontrarse en la ventana “Gestionar P-1 de asignaturas”.</p> <p>El usuario debe haber seleccionado la opción “Crear período lectivo”.</p>	Media	Alta
Prototipo				

Gestionar P-1 de asignaturas

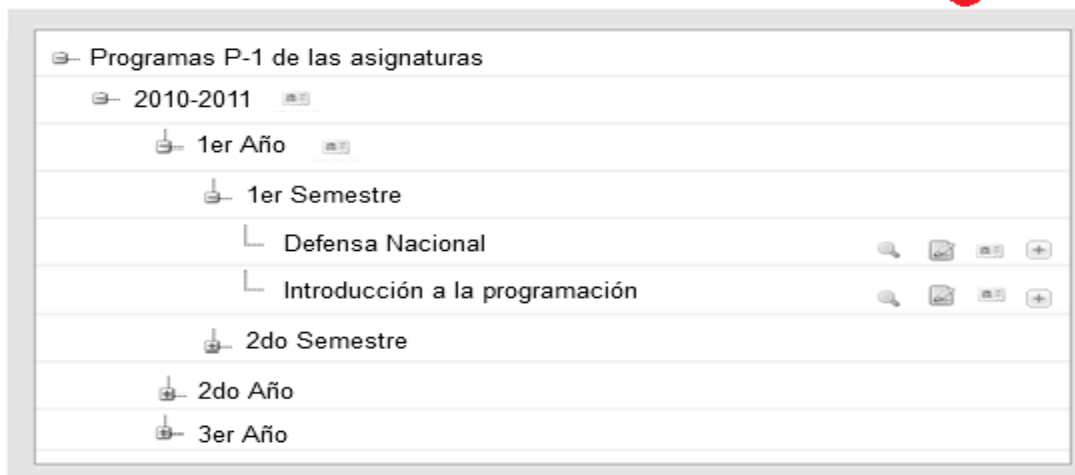


Fig. 3 Vista gestionar P-1 de asignatura.

Crear período lectivo

Fig. 4 Vista crear período lectivo.

Campos	Tipos de Datos	Reglas o Restricciones
Fecha inicio	<i>datetime</i>	Desplegable para selección. Valor inicial: en blanco.
Fecha fin	<i>datetime</i>	Desplegable para selección. Valor inicial: en blanco.
Observaciones		
1. Interactúa con esta acción los Jefes de Departamento Centrales y el		

	<p>administrador.</p> <ol style="list-style-type: none"> 2. En caso que el elemento ya exista se muestra un mensaje de error “El período lectivo ya existe”. 3. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de cancelar la acción? “. 4. En caso de que introduzca datos incorrectos se muestra un mensaje de error “El elemento se ha introducido incorrectamente”. 5. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo “Campo requerido” en el campo que debe ser llenado obligatoriamente.
--	---

Tabla 3. Descripción del requisito crear P-1 de asignatura.

	Nombre	Descripción	Complejidad	Prioridad para cliente
[RF18.]	Crear (P-1) de asignatura	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado con el rol “Jefe departamento central”. 2. El usuario debe encontrarse en la ventana “Gestionar (P-1) asignatura”. 3. Debe haber seleccionado Crear (P-1) asignatura en el icono flotante marcado 	Alta	Alta

Prototipo

Gestionar P-1 de asignaturas

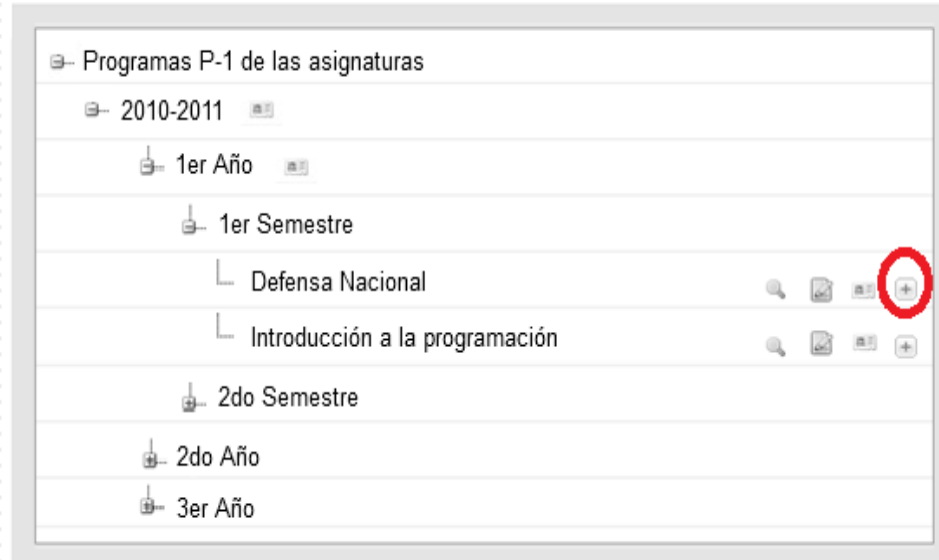


Fig. 5 Vista gestionar P-1 de asignatura.

Crear programa analítico de asignatura (P-1).

P1- **DATOS GENERALES**

Asignatura:

Disciplina:

Carrera:

Año Acad.: **Curso:**

DISTRIBUCION DE HORAS

Nuevo tema	C	CP	H/NP.	S	T	L	Eval	Total
Tema 1	2	2		2	4		X	14
Tema 2	2				6		X	14
Tema 3	4						X	16
Total								44

Segundo Semestre:

Sem.	Tema	No. Activ.	Tipo de Activ.	Contenido	Medios
1	1	1	Conferencia Presencial Presencial	Surgimiento de la sociedad criolla cubana. Orientación del Taller	PC. EVA, Pizarra, M.A.
2	1	2	Conferencia Presencial Presencial	Surgimiento de la sociedad criolla cubana.	PC. EVA Pizarra, M.A.
		3	Taller Presencial		PC. EVA. Pizarra, M.A.

Sistema de Evaluación: Habilitar Edición

La asignatura tiene una PP en el primer semestre, otra PP en el segundo semestre y un examen final en el segundo ...

Semana	Tema	P.E.	Taller	Semin	CP.	P.P.
1	1					
2	1	X	X			
3	1				X	
4	1			X		
5	1				X	
6	2					
7	2	X				
8	2				X	
9	2				X	
10	2					
11	2				X	
12	1/2					X
13	3					
14	3	X				
15	TI	TI	TI	TI	TI	TI

Examen Final: Si No:

Elaborado por:

Aprobado por:

Fig. 6 vista de crear P-1 de asignatura.

Campos

Tipos de Datos

Reglas o

		Restricciones
P1	<i>Varchar</i>	Obligatorio
Asignatura	<i>VarChar</i>	Obligatorio
Disciplina	<i>VarChar</i>	Obligatorio
Carrera	<i>VarChar</i>	Obligatorio
Habilitar edición del sistema de evaluación.	<i>bool</i>	
Clases	<i>int</i>	Opcional
Clases prácticas	<i>int</i>	Opcional
Horas no presenciales	<i>int</i>	Opcional
Seminarios	<i>int</i>	Opcional
Talleres	<i>int</i>	Opcional
Laboratorios	<i>int</i>	Opcional
Evaluación	<i>bool</i>	Opcional
Examen final	<i>bool</i>	Opcional
Elaborado por	<i>Varchar</i>	Obligatorio
Aprobado por	<i>Varchar</i>	Obligatorio
Observaciones		
<ol style="list-style-type: none"> 1. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de cancelar la acción? “. 2. En caso de que introduzca datos incorrectos se muestra un mensaje de error “El elemento se ha introducido incorrectamente”. 3. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje de error de color rojo “Campo requerido” en el campo que debe ser llenado 		

	<p>obligatorio.</p> <p>4. Interactúa con esta acción el administrador, jefes de departamentos centrales y asesores centrales de asignaturas.</p> <p>5. En caso de nunca haber sido agregado un P-1, se pondrá un campo en blanco por defecto.</p> <p>6. El semestre, año, y período lectivo los toma del árbol jerárquico.</p>
--	--

2.5 Requisitos no funcionales

Los requerimientos no funcionales, como su nombre lo sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como finalidad, el tiempo de respuesta y la capacidad de almacenamiento (Sommerville, 2005).

Usabilidad

RNF1: la interfaz del sistema debe ser de fácil manejo, posibilitando a los usuarios una rápida adaptación.

RNF2: el sistema debe adaptarse al lenguaje y términos utilizados por los usuarios en el proceso de planificación docente con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.

RNF3: el sistema debe presentar enlaces a las funcionalidades tanto laterales como en barra de iconos flotantes que permitan el acceso rápido a la información por parte de los usuarios, para aprovechar las potencialidades de las estructuras de la información.

Seguridad

RNF4: deben existir distintos roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de estos al sistema, contribuyendo además la protección de la información.

RNF5: el sistema debe ser capaz de registrar el accionar del usuario, así como permitir auditorías y exámenes de las trazas tanto en tiempo real como en históricos. Se precisa un monitor de incidencia para la visualización y tratamiento de las mismas.

Restricciones del diseño

RNF6: servidor de base de datos con *PostgreSQL* 8.4 o superior bajo el sistema operativo *Ubuntu Server* 10.4 o superior.

RNF7: servidor de aplicaciones *web*: *Apache* 2.2 o superior.

RNF8: navegador web: *Internet Explorer* 7 o superior, *Mozilla Firefox* 2.3 o superior.

Interfaz

RNF9: la interfaz debe ser sencilla utilizando colores claros y sin cúmulo de imágenes u objetos que puedan distraer al cliente del objetivo.

Interfaces hardware

RNF10: la comunicación entre el cliente y el servidor de aplicaciones se lleva a través del protocolo HTTP o HTTPS.

2.6 Patrón arquitectónico

La arquitectura de *software* es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo y la conducta de estos. Según se percibe desde el resto del sistema, los patrones se coordinan para alcanzar la misión del sistema, al igual que las formas en que los componentes interactúan. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión del detalle inherente a la mayor parte de las abstracciones (Reynoso, 2004).

En la solución que se elabora se propone la utilización del Modelo-Vista-Controlador, porque este es el patrón de arquitectura base que utiliza el marco de trabajo GUUD del SGU.

2.6.1 Patrón Modelo-Vista-Controlador, especificación de su utilización

El Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, permitiendo realizar una programación multicapa (Guglielmetti, 2004).

- Modelo: representa la lógica del negocio y la información del sistema. La lógica de datos asegura la integridad y permite derivar nuevos datos.

- Vista: presenta el modelo en un formato adecuado para interactuar, usualmente es la interfaz de usuario.
- Controlador: responde a las acciones del usuario e invoca cambios en el modelo o genera la vista apropiada dependiendo de las peticiones del usuario. Sirve como un intermediario entre el modelo, la vista y cualquier otro recurso necesario para procesar una petición y generar una página *web* (Pressman, 2005).

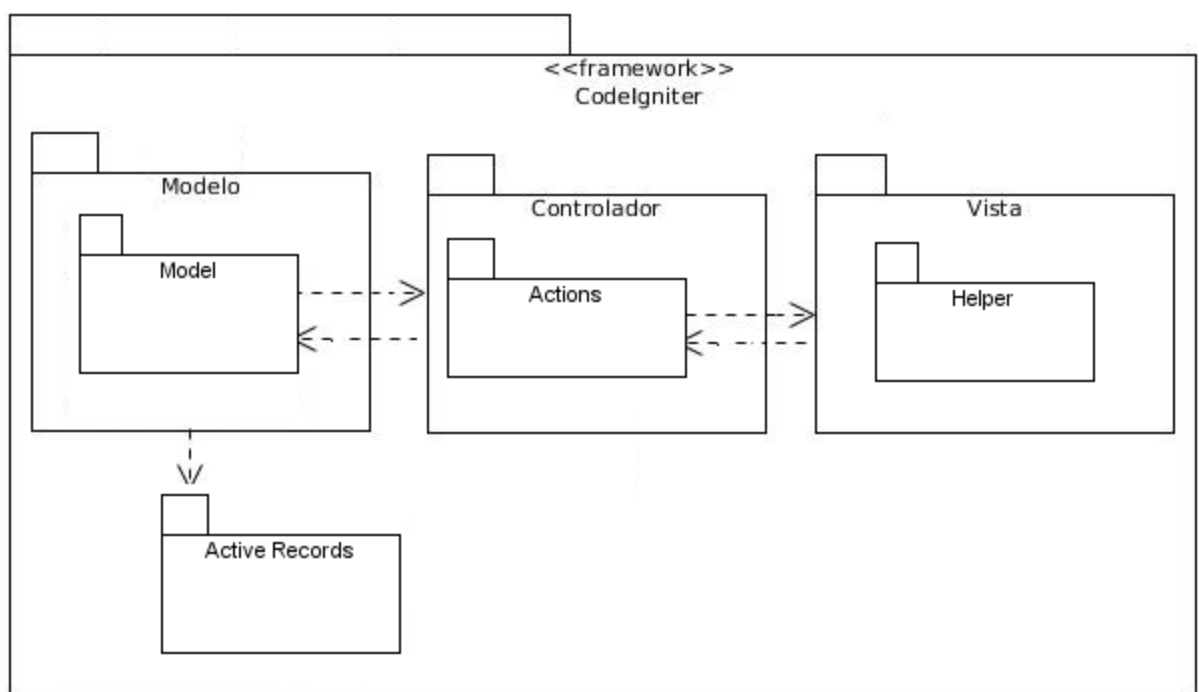


Fig. 7 Funcionamiento del patrón Modelo-Vista-Controlador (MVC) en el marco de trabajo GUUD

El marco de trabajo GUUD definido por el CENIA se basa en el patrón de desarrollo Modelo-Vista-Controlador. Por lo general sus clases del modelo contendrán las funciones que le ayudan a recuperar, insertar y actualizar la información en su base de datos, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la lógica de negocio.

El mismo integra a su vez los *frameworks* CodeIgniter en su versión 1.7.3 y JQuery 1.3.2 en una sola infraestructura, razón por la cual posee las mismas características que estos. En esta integración se incluyen además un conjunto de novedades o mejoras y algunas modificaciones he-

chas específicamente al *CodeIgniter*, además creación de *plugins* y componentes de interfaz de usuario en *jQuery*.

2.7 Patrones de diseño

Los patrones de diseño son los principios generales de soluciones que aplican ciertos estilos que ayudan a la creación de *software*. Una descripción de un problema y la solución a la que le da el nombre y que se puede aplicar en nuevos contextos (Pressman, 2005).

2.7.1 Patrones de asignación de responsabilidades de software

Los patrones GRASP (Patrones de Asignación de Responsabilidades de *Software*) constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos. En el marco de trabajo GUUD se manifiestan los siguientes patrones: experto, creador, controlador, alta cohesión y bajo acoplamiento. A continuación se detallan cada uno de los usos de estos patrones.

Experto: la responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada (Visconti y Astudillo, 2011).

Ejemplo: se propone su uso en la implementación de las clases librerías, que son las que cuentan con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio.

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, se asigna la responsabilidad de que una clase B cree un objeto de la clase A solamente cuando:

- B contiene a A.
- B es una agregación (o composición) de A.
- B almacena a A.
- B tiene los datos de inicialización de A (datos que requiere su constructor).
- B usa a A.

Este patrón se utilizó para identificar qué clase A debe crear elementos de una clase B, apoyándose en que la clase A debería: contener, agregar, registrar, utilizar y tener los datos de

inicialización de la clase B (Visconti y Astudillo, 2011).

Ejemplo: se propone su uso en la implementación de la clase *loader* que es el objeto *load* de las clases controladoras, encargada de cargar los elementos del marco de trabajo dígase, librerías, modelos.

Controlador: es un evento generado por actores externos. Se asocian con operaciones del sistema, operaciones del sistema como respuestas a los eventos del sistema, tal como se relacionan los mensajes y los métodos. Normalmente un controlador delega en otros objetos el trabajo que se necesita hacer; coordina o controla la actividad. No realiza mucho trabajo por sí mismo (Visconti y Astudillo, 2011).

El patrón controlador se propone utilizar para que sirva como intermediario entre cada una de las capas, de forma tal que se garantice la comunicación entre los eventos externos del sistema en la capa de presentación y los componentes de la capa de negocio. Permite asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.

Bajo acoplamiento: el acoplamiento es una medida de fuerza con que un elemento tiene conocimiento o confía en otros elementos. Este patrón es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Esto facilita la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema (Visconti y Astudillo, 2011).

Este patrón se propone se utilice con la idea de tener las clases lo menos ligadas entre sí y asignar una responsabilidad para mantener bajo acoplamiento. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Alta cohesión: la cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento, cada elemento debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto

identificable. Una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo (Visconti y Astudillo, 2011).

La propia implementación de *CodeIgniter* contiene estos patrones nivelados pues permite el uso de los componentes de forma individual, evidenciando el bajo acoplamiento y la dependencia entre ellos o alta cohesión.

2.7.2 Patrones banda de los 4

Se describen algunos de los patrones banda de los 4 (GOF) para su uso en la propuesta de solución, utilizados además actualmente en el marco de trabajo GUUD:

Fábrica abstracta: permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando (Gamma *et al*, 1995).

Ejemplo: se evidencia en el módulo seguridad al cual se debe acceder mediante la integración al SGU.

Singleton (Instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia (Gamma *et al*, 1995).

Ejemplo: se propone se utilice en todas las clases controladoras, son instancias únicas. También en la IOC para la interacción entre módulos.

Mediator (Mediador): definirá un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto (Gamma *et al*, 1995).

Ejemplo: se propone se utilice mediante el uso de las librerías que funcionan como mediadoras entre las clases controladoras y los modelos o acceso a datos.

Observer (Observador): define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él (Gamma *et al*, 1995).

Ejemplo: se propone se utilice mediante la implementación de la clase *loader* que es el objeto *load* de las clases controladoras, encarga de cargar los elementos del marco de trabajo *dígase*, librerías, modelos y se encarga de actualizar la controladora instanciada.

2.8 Estándares para el diseño de prototipos

Para realizar el diseño de los prototipos se tuvo como guía las pautas del diseño visual establecidas en el Manual de Directrices del Sistema de Gestión Universitaria, que permite la uniformidad en todas las páginas *web* que lo componen.

2.8.1 Distribución de la información en la propuesta de información

La vista de presentación es la primera vista que se le muestra a cualquier usuario, mediante la que podrá autenticarse y acceder al sistema.

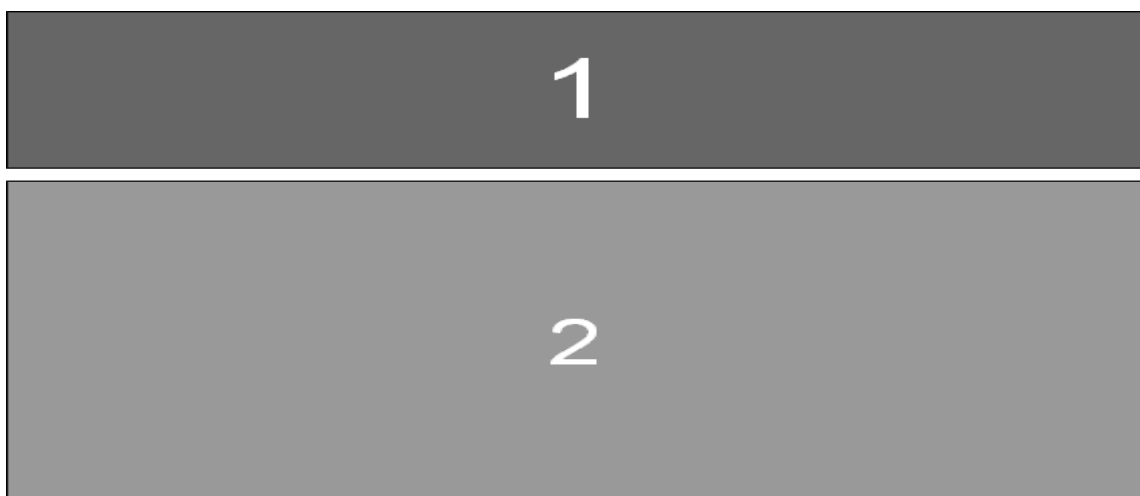


Fig. 8 Áreas de la vista de presentación.

- 1) Cabezal o área de identificación.
- 2) Área de entrada de datos.

La vista de escritorio se muestra luego de la autenticación del usuario, en la que podrá seleccionar el subproceso horizontal, módulo o servicio al que desee acceder y tenga los permisos requeridos.

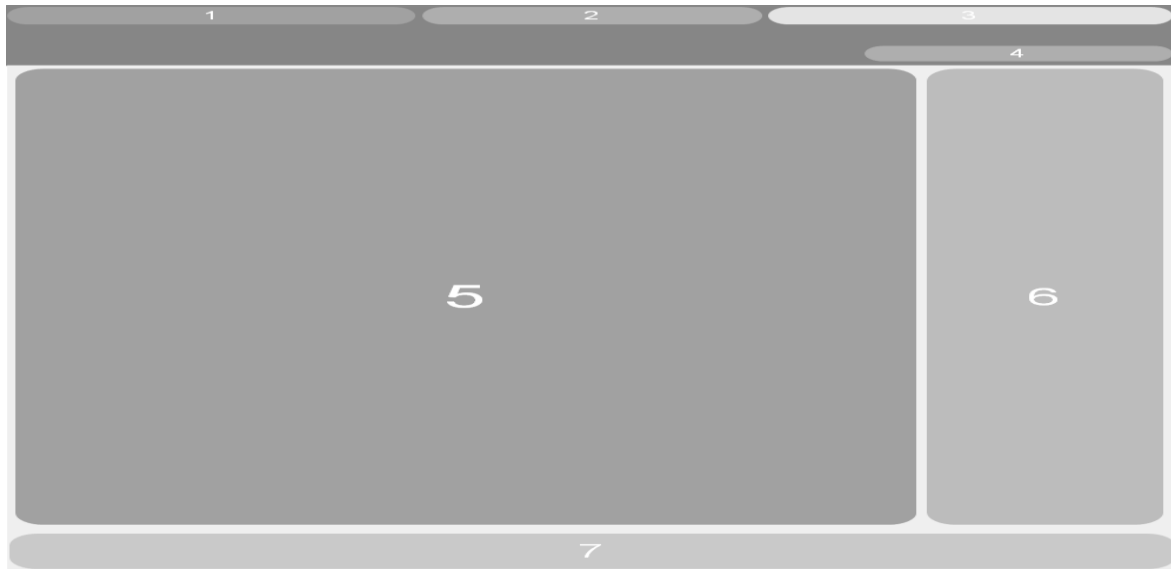


Fig. 9 Áreas de la vista de presentación.

- 1- Área de subprocesos horizontales.
- 2- Área del nombre de la aplicación.
- 3- Área del buscador.
- 4- Área de nombre de usuario.
- 5- Área de líneas de procesos.
- 6- Área de servicios horizontales.
- 7- Área de pie de página.

La vista de gestión de procesos permite el acceso a los módulos del subsistema seleccionado por el usuario y las funcionalidades que poseen.

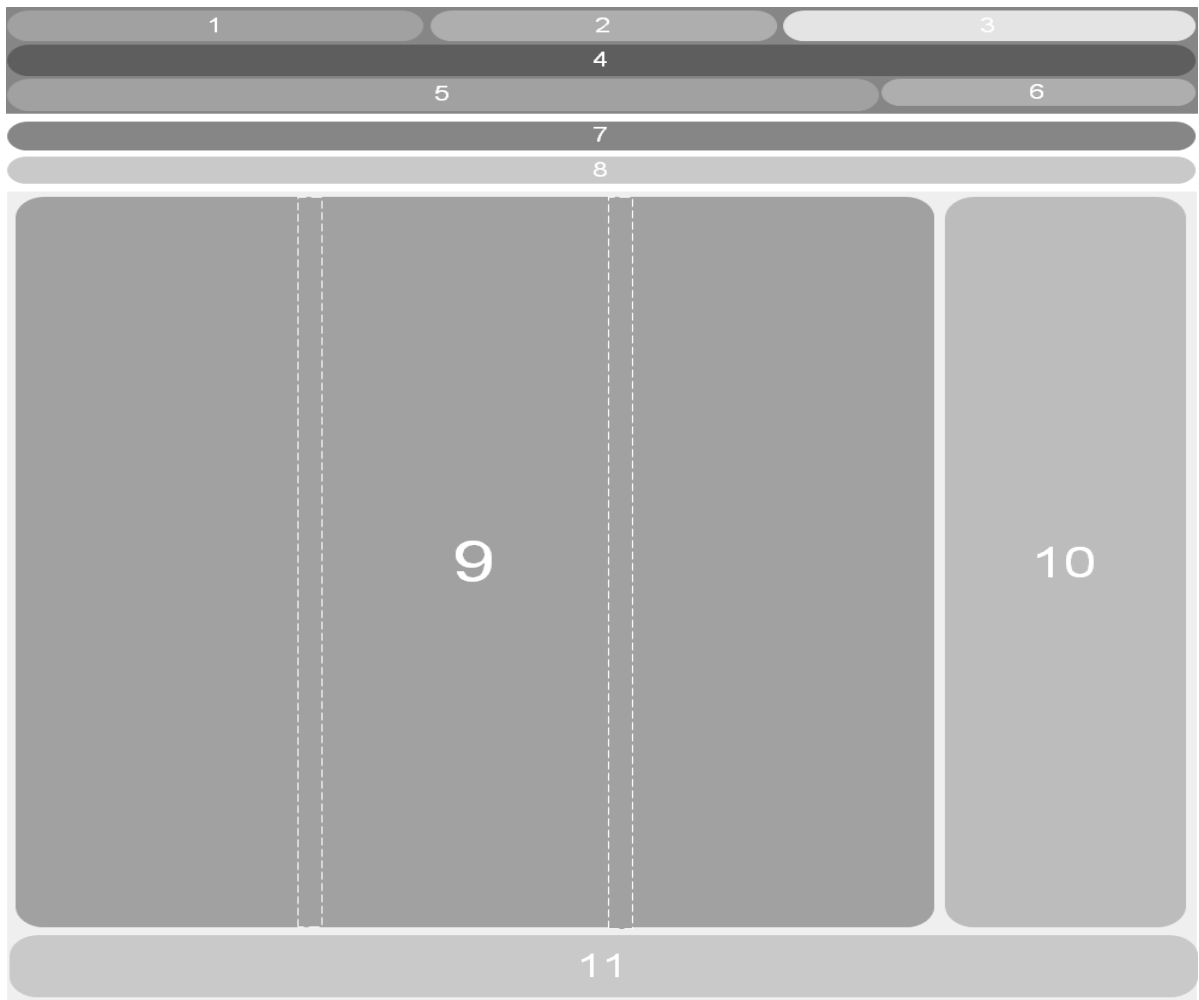


Fig. 10 Áreas de la vista de presentación.

- | | |
|--------------------------------------|-------------------------------|
| 1- Área de subprocesos horizontales. | 7- Área de traza. |
| 2- Área del nombre de la aplicación. | 8- Área de línea de progreso. |
| 3- Área del buscador. | 9- Área de contexto. |
| 4- Área de líneas de procesos. | 10- Área de menú de módulos. |
| 5- Área de módulos. | 11- Área de pie de página. |
| 6- Área de nombre de usuario. | |

En el área del menú de módulos aparecerán las opciones a seleccionar, al seleccionar la del módulo de Planificación Docente se mostrarán los procesos fundamentales que éste módulo abarca:

- Planificación del horario docente.
- Planificación de actividades de formación.

Dentro de esta funcionalidad se encontrarán las vistas relacionadas con las acciones:

- a) Gestionar plan calendario docente.
- b) Gestionar P-1 de asignaturas.
- c) Gestionar balance de carga docente.

Tipos de mensajes

Se definieron 4 tipos de mensajes:

- Mensajes de información.
- Mensajes de error.
- Mensajes de advertencia.
- Mensajes de confirmación.

Mensajes de información para el cliente

Se utilizan para brindar una información al usuario, cuando se crea un elemento, se modifica, se eliminar o se asocia:

- “El elemento ha sido activado satisfactoriamente.”
- “El elemento ha sido desactivado satisfactoriamente.”
- “El elemento ha sido creado satisfactoriamente.”
- “El elemento ha sido modificado satisfactoriamente.”
- “El elemento ha sido eliminado satisfactoriamente.”



Fig. 11 Mensaje de información.

Mensajes de error

Se utilizan para mostrar al usuario cuando ha realizado una acción incorrecta:

- “El elemento ya existe.”
- “Ha dejado campos obligatorios vacíos.”
- “Se han introducido datos incorrectamente.”

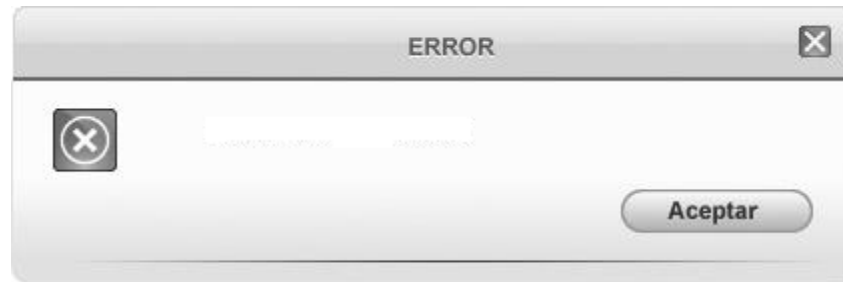


Fig. 12 Mensaje de error

Mensajes de advertencia

Se utilizan cuando es necesario advertir al usuario de algún suceso si ejecuta la acción que está solicitando.

- “Perderá la información que no ha sido guardada.”



Fig. 13 Mensaje de advertencia

Mensajes de confirmación

Son los que se utilizan cuando es necesario asegurarse que el usuario desea realizar una acción deseada, por ejemplo cuando va a eliminar un elemento, es necesario asegurarse que eso es lo que desea.

- ¿Está seguro que desea eliminar el elemento?



Fig. 14 Mensajes de confirmación.

2.9 Vista de despliegue

El diagrama de despliegue es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. Describen la arquitectura física del sistema durante la ejecución, en términos de procesadores, dispositivos y componentes de *software*.

A continuación se describen los componentes del diagrama de despliegue:

- Nodos: elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos (impresoras, lectores de códigos de barras o dispositivos de comunicación).
- Dispositivos: nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela. Generalmente se usan para identificar el tipo de dispositivo. Los nodos se conectan mediante asociaciones de comunicación que indican:
 - Algún tipo de ruta de comunicación entre los nodos.
 - Los nodos intercambian objetos o envían mensajes a través de esta ruta.
- Conectores: expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

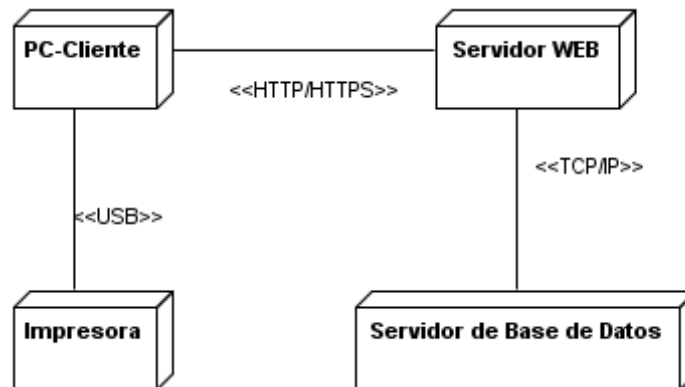


Fig. 15 Diagrama de despliegue.

- PC_cliente: computadora que cuente con un navegador actualizado y que siga los estándares *web* (*Mozilla Firefox* v 2 o superior). Se recomiendan estaciones de trabajo con sistema operativo *GNU/Linux*.
- Servidor *web*: representa una estación donde estará montado el servidor *Apache* sobre el cual correrá la aplicación.
- Servidor de base de datos: representa el servidor donde estará el SGBD *PostgreSQL* que dará respuesta a las peticiones hechas por la aplicación.

2.10 Relación de la propuesta de solución con el Sistema de Gestión Académica de Pregrado

El Sistema de Gestión Académica de Pregrado, se encuentra integrado al Sistema de Gestión Universitaria (SGU), lo que permite el uso de las facilidades provistas por dicha integración, como la reutilización de varios componentes que están implementados por otros sistemas, la centralización de archivos e información y la eliminación de redundancias en el código. De la misma forma se comporta la relación entre los módulos de Pregrado, que se encuentran en continua colaboración para lograr el mejor funcionamiento y la calidad de la gestión del proceso docente. La solución propuesta para el módulo de planificación docente utiliza elementos necesarios de otros módulos del sistema y del núcleo de SGU.

Seguridad: el módulo de seguridad, implementado a nivel central para todo el Sistema de

Gestión Universitaria, garantiza el acceso a la información dados los niveles de privilegio de cada usuario, haciendo uso de la arquitectura sobre la cual está desarrollado el sistema. En planificación docente la seguridad permitiría que el gestor de la planificación docente, los jefes de departamentos centrales u otros involucrados con este proceso, sólo sean capaces de acceder a las funcionalidades en las que tengan establecidos los permisos, evitando el mal uso, corrupción y modificación de la información por personas no autorizadas.

Configuración: la configuración del sistema, también gestionada de forma centralizada, facilita un mejor control sobre los posibles cambios que pueden ocurrir en el sistema, lo cual provee robustez y adaptabilidad al mismo. Dicho módulo permite gestionar la información de cada sistema, sus módulos y las funcionalidades asociadas.

Trazas: gestiona la información referente a las incidencias de un usuario sobre el sistema, registrando el usuario, la acción realizada y el momento en que se ejecutó la misma. Permite el registro de las actividades que se realicen sobre el módulo.

Carrera: se encarga de la gestión de los elementos que conforman los planes de estudios definidos por el Ministerio de la Enseñanza Superior (MES), acorde a las necesidades y características de la entidad. De él se obtienen elementos importantes para el control docente como son: estructura de los modelos de planificación y control del proceso docente, además de las reglas del proceso de planificación docente.

Reportes: permite obtener información en forma de gráficas o listados sobre elementos de la gestión académica incluyendo los de planificación docente, como reportes sobre los períodos críticos en la realización de actividades evaluativas que obtiene de la funcionalidad de planificación de las actividades de formación docente.

El SGU, está compuesto por 13 subsistemas, entre los que se encuentra el Sistema de Gestión Académica de Pregrado, estos subsistemas se relacionan entre sí, mediante llamadas remotas a la base de datos del núcleo a través de *plproxy* que es el encargado de establecer las conexiones.

El SGU se encuentra compuesto por varios módulos relacionados. La figura que a continuación se muestra refleja la integración de los componentes en el Subsistema de Gestión Académica

de Pregrado:

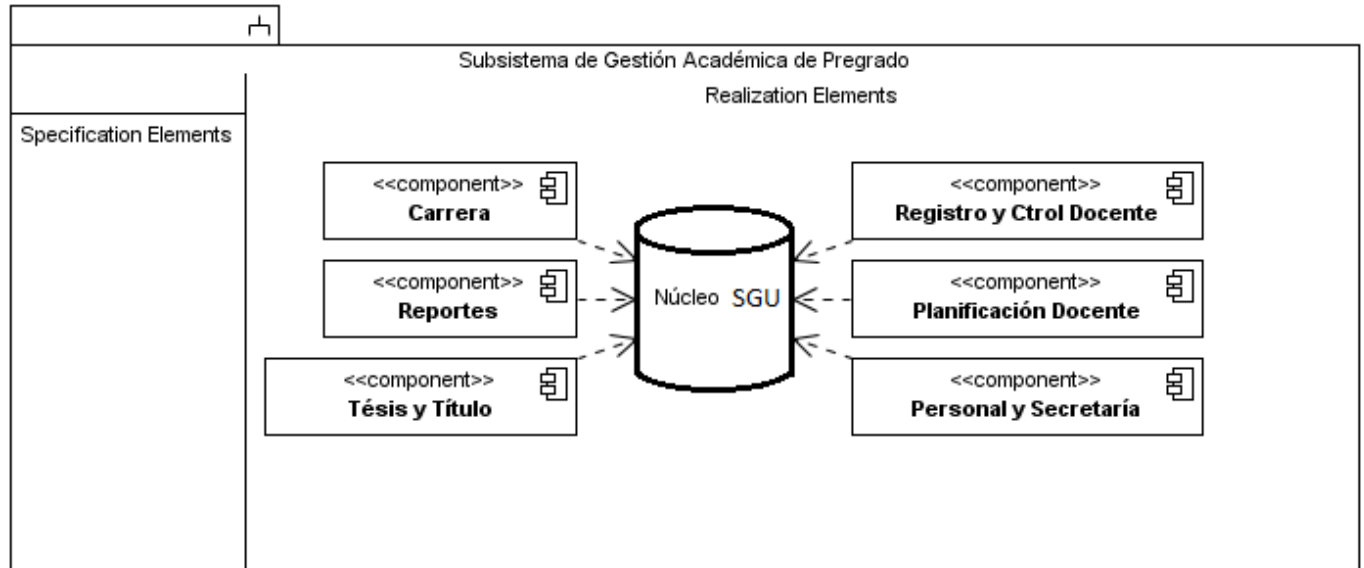


Fig. 16 Relación de los componentes del subsistema de Gestión Académica de Pregrado.

Conclusiones parciales

Al realizar una valoración de los procesos involucrados en el campo de acción, se analizó cómo se lleva a cabo actualmente el proceso de planificación docente, y además cómo se deberá realizar una vez que se informatice. Se identificaron los requisitos funcionales, describiendo cada uno de estos, con sus respectivos prototipos de interfaz como forma de validación de los mismos. Se realizó un análisis de cómo la propuesta de solución se debería relacionar con el Sistema de Gestión Académica de Pregrado, del SGU.

Capítulo 3: Validación de la solución propuesta

Introducción

Durante el desarrollo de un sistema se pueden presentar errores en el mismo. La validación de la propuesta de solución es esencial para el desarrollo saludable de una futura aplicación, una estrategia de prueba del *software* debe ser lo suficientemente flexible como para promover un enfoque personalizado. Al mismo tiempo, debe ser adecuadamente rígido como para promover una planeación razonable y un seguimiento administrativo del avance del proyecto. La recomendación de pruebas a realizarle al sistema luego de su implementación e integración al Sistema de Gestión Universitaria de Pregrado está encaminada a minimizar la cantidad de posibles errores a cometer.

3.1 Técnicas para validar los requisitos

La validación de requisitos examina la especificación para asegurar que todos los requisitos de *software* se han establecido de manera precisa; que se han detectado las inconsistencias, omisiones y errores y que éstos han sido corregidos, así como que los productos de trabajo cumplen con los estándares establecidos para el proceso (Pressman, 2005). Para la presente propuesta fueron puestas en prácticas las siguientes técnicas para la validación de los requisitos:

- **Revisiones de requerimientos.** En esta técnica, los requerimientos son analizados por los revisores y el cliente.
- **Construcción de prototipos.** En este enfoque de validación, se muestra un modelo del sistema a los clientes.
- **Generación de casos de prueba.** Los requerimientos deben poder probarse. Si las pruebas para éstos se conciben como parte del proceso de validación, a menudo revelan los problemas en los requerimientos. Si una prueba es difícil o imposible de diseñar, normalmente significa que los requerimientos serán difíciles de implementar y deberían ser considerados nuevamente. Desarrollar pruebas para los requerimientos del usuario

antes de que se escriba código es una parte fundamental de la programación extrema (Sommerville, 2005). Fue aplicado en la investigación mediante la realización de los diseños de casos de prueba y la propuesta de las pruebas que a realizar una vez finalizada la elaboración de la propuesta de solución.

3.2 Criterios de validación de requisitos del cliente y del producto

Según CMMI®, la validación demuestra que el producto o servicio adquirido satisface las necesidades de las partes interesadas en el negocio y los requisitos del cliente. El proceso de desarrollo con enfoque ágil basado en el nivel 2 de CMMI® propone criterios que permiten validar los requisitos del cliente y del producto, donde se responde a varias interrogantes que validan si el requisito se aprueba o no. Las actividades de validación son llevadas a cabo por el adquiridor, el proveedor o por ambas partes en función de lo establecido en el acuerdo con el proveedor (INTECO, 2009).

Las pruebas de validación del *software* logran que se cumplan con los requisitos. Una correcta planeación de estas delinea la clase de ensayos que se aplicarán. El plan se diseña para asegurar que se satisfagan todos los requisitos funcionales, que se alcanzan en todas las características de comportamiento, que se cumple con todos los requisitos de desempeño y que la documentación es correcta (Pressman, 2005).

Preguntas para validar requisitos del cliente

- ¿El proveedor del requisito es un proveedor válido?
- ¿El requisito está identificado como único?
- ¿El requisito es modificable?
- ¿El requisito no es ambiguo?
- ¿El requisito está completo?
- ¿El requisito es congruente con otros requisito relacionados?
- ¿El requisito puede ser implementado?
- ¿El requisito puede ser probado?
- ¿El resultado de la evaluación de impacto es positivo?

- ¿El requisito está correcto?

Preguntas para validar requisitos del producto

- ¿Están identificados los elementos de entrada?
- ¿Están identificados los elementos de salida?
- ¿El requisito es dado por el superior determinado en el organigrama del proyecto?
- ¿El requisito no es ambiguo?
- ¿Es técnicamente factible?
- ¿Puede ser verificado?
- ¿Está correcto?
- ¿El resultado de la evaluación de impacto es positivo?
- ¿El requisito es traceable?

3.2.1 Resultado de la validación de los requisitos

La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea (Universidad de Sevilla, 2002). Examina la especificación para asegurar que todos los requisitos de *software* se han establecido de manera precisa; que se han detectado las inconsistencias, omisiones o errores y que éstos han sido corregidos, además que los productos de trabajo cumplen con los estándares establecidos para el proceso, proyecto y producto (Pressman, 2005).

La validación de los requisitos permite asegurar que los requisitos:

- Fueron atendidos correctamente.
- Definen el producto de *software* correcto.
- Cumplen con los estándares definidos.

Existen algunas técnicas que permiten que el proceso de validación tenga una mejor calidad, de ellas se utilizaron:

- Revisión de requerimientos: se revisaron los documentos de especificación de requisitos y se detectaron un número de inconsistencias, que fueron anomalías, omisiones u otros tipos de errores, a las que se les dio solución.

- Generación de casos de prueba: se realizan los diseños de casos de pruebas para cada uno de los requisitos especificados, permitiendo verificar que todos se pudieran probar, e identificar en medida de la complejidad del diseño de caso de prueba, requisitos que deberían ser reconsiderados y cuáles son los más fáciles de implementar con vista a lograr una mejor planificación de la implementación de cada caso de prueba.

Algunas de las inconsistencias identificadas como resultado de las revisiones realizadas en este proceso se encuentran:

- En el área de íconos flotantes que se muestra en algunos prototipos aparecen íconos que no se encuentran en la descripción.
- La complejidad en algunos casos no estuvo acorde con la dada en el documento Evaluación de Requisitos.
- No están especificados algunos campos que son requeridos.
- No está definida la posición de los P-1 de asignaturas, así como nivel en que se deben ubicar en el árbol jerárquico.

Para consultar el resto de las inconsistencias resueltas remitirse al expediente de proyecto del SGU.

3.3 Propuesta de pruebas

Los objetivos de realizar una prueba son:

- Detectar un error.
- Tener un buen caso de prueba, es decir que tenga más probabilidad de mostrar un error no descubierto antes.
- Descubrir un error no descubierto antes (éxito de la prueba).

Por lo que se propone realizar las pruebas en los siguientes niveles:

Prueba de unidad: suele considerarse adyacente al paso de la codificación, el diseño de las pruebas de unidad puede realizarse antes de que empiece la codificación o después de que se ha generado el código fuente. Una revisión de la información proporciona una guía para establecer casos de prueba que probablemente descubrirán errores en cada una de las

categorías analizadas. Cada caso de prueba debe relacionarse con un conjunto de resultados esperados.

Prueba de integración: una técnica sistemática para construir la arquitectura del *software* mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar componentes a los que se aplicó una prueba de unidad y construir una estructura de programa (Pressman, 2005).

Se propone se utilicen los métodos de Caja blanca y de Caja negra.

Prueba de caja blanca: la prueba de caja blanca se basa en el diseño de casos de prueba. Mediante la prueba de la caja blanca el ingeniero del *software* puede obtener casos de prueba que:

1. Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
2. Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de caja blanca como uno de los tipos de pruebas más importantes que se debe aplicar a la solución propuesta, para lograr como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad, así como comprobar exhaustivamente la lógica del programa y ejercitar estructuras de datos internos para asegurar su validez (Pressman, 2005).

Prueba de caja negra: se refiere a las pruebas que se llevan a cabo sobre la interfaz del *software*, por lo que los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del *software*.

La prueba de caja negra se centra principalmente en los requisitos funcionales del *software*. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten

completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

Estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien (Pressman, 2005).

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

1. Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*.

Permitiendo derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de la propuesta. Tratando de encontrar errores en las siguientes categorías: 1- funciones incorrectas o faltantes, 2- errores de interfaz, 3- errores en estructuras de datos o en acceso a bases de datos externas, 4- errores de comportamiento o desempeño, 5- errores de inicialización y término (Pressman, 2005).

3.3.1 Diseño de casos de caso de pruebas

Para verificar que el sistema cumpla con todos los requerimientos especificados por el cliente es necesario realizarle pruebas con el fin de detectar si existe algún error.

Las pruebas basadas en escenario o basadas en fallas soslaya dos tipos importantes de errores: 1- especificaciones incorrectas, 2- interacciones entre subsistemas. Cuando ocurren errores asociados con especificaciones incorrectas, el producto no hace lo que el cliente quiere.

Podría hacer lo incorrecto, u omitir funcionalidades importantes (Pressman, 2005). Los diseños de casos de pruebas (DCP) son una prueba basada en escenarios y se realizan para cada funcionalidad especificada. El objetivo fundamental de los DCP es poseer documentación por la cual se pueda guiar el proceso de prueba que se le debe realizar al sistema para su posterior validación (Sommerville, 2005). El caso de prueba mostrado fue escogido de forma aleatoria.

Tabla 4. Caso de prueba definido para el requisito crear actividad del calendario docente.

Esce- nario	Descripción	Nombre evento	Fecha Inicio	Fecha Fin	Año aca- démico	Orientado por	Res- puesta del sis- tema	Flujo central
EC 1.1 Insertar datos correc- tamen- te.	Mediante este escenario se insertan en el sistema un evento del ca- lendaro docen- te	V	V	V	V	V	El siste- ma ac- tualiza el listado y muestra el men- saje "El elemento ha sido creado satisfac- toria- mente".	El usuario una vez autentica- do en el sis- tema selec- ciona el Sub- sistema de Pregrado y luego el mó- dulo Control docente. El sistema mues- tra las opcio- nes de menú y el usuario seleccionó en la agrupación funcional "Elementos de
		TSU	12/05/ 2012	19/05/ 2012	1ro	Rectoría		
		V	V	V	V	V		
		REM	13/04/ 2012	28/04/ 2012	3ro, 4to	MES		
		V	V	V	V	V		

								configuración" la funcionalidad "Conceptos de evaluación". Seguidamente selecciona en el área de iconos flotantes la opción de Crear El usuario llena los datos correctamente y presiona el botón Aceptar
EC 1.2	Mediante este escenario se introducen datos para insertar un concepto de evaluación que ya existe en el sistema.	I	I	I	I	I	El sistema muestra un mensaje de error indicando: El elemento ya existe.	El usuario una vez autenticado en el sistema selecciona el Sub-sistema de Pregrado y luego el módulo Control docente. El sistema muestra las opcio-
Insertar elementos repetidos.		TSU	12/05/2012	19/05/2012	1ro	Rectoría		

							<p>nes de menú y el usuario seleccionó en la agrupación funcional "Elementos de configuración" la funcionalidad "Conceptos de evaluación". Seguidamente selecciona en el área de iconos flotantes la opción Crear. El usuario llena los datos de un elemento que ya esté en el sistema y presiona el botón Aceptar.</p>
--	--	--	--	--	--	--	---

EC 1.3 Insertar datos incompletos.	Mediante este escenario no se insertan en el sistema un evento del calendario docente	V	V	V	V	I	El sistema muestra encima del componente un mensaje en color rojo indicando: Campo requerido.	El usuario una vez autenticado en el sistema selecciona el Sub-sistema de Pregrado y luego el módulo Control docente. El sistema muestra las opciones de menú y el usuario seleccionó en la agrupación funcional "Elementos de configuración" la funcionalidad "Conceptos de evaluación". Seguidamente selecciona en el área de iconos flotantes la opción
		TSU	12/05/2012	19/05/2012	1ro	Vacío		
		I	V	V	V	V		
		Vacío	13/04/2012	28/04/2012	3ro, 4to	MES		

								Crear El usuario llena los datos de un elemento que ya esté en el sistema y presiona el botón Aceptar.
EC 1.4	Mediante este escenario se introducen datos incorrectos para crear un evento en el plan calendario docente	I	V	V	V	V	El sistema muestra en color rojo encima del componente: Solo se permiten letras, números espacios y <i>underscores</i> .	El usuario una vez autenticado en el sistema selecciona el Sub-sistema de Pregrado y luego el módulo Control docente. El sistema muestra las opciones de menú y el usuario seleccionó en la agrupación funcional "Elementos de configuración"
Insertar datos incorrectos.		"@#%256^"	12/05/2012	19/05/2012	1ro	Rectoría		

								la funcionalidad "Conceptos de evaluación". Seguidamente selecciona en el área de iconos flotantes la opción Crear. El usuario llena los datos de forma incorrecta y presiona el botón Aceptar.
EC 1.5 Cancelar operación.	Se cancela la creación del evento en el plan.	NA	NA	NA	NA	NA	El sistema muestra el mensaje de confirmación ¿Está seguro de realizar la acción?	El usuario una vez autenticado en el sistema selecciona el Sub-sistema de Pregrado y luego el módulo Control docente. El sistema muestra las opcio-

								<p>nes de menú y el usuario seleccionó en la agrupación funcional "Elementos de configuración" la funcionalidad "Conceptos de evaluación". Seguidamente selecciona en el área de iconos flotantes la opción Crear. El usuario modifica o no los datos y presiona el botón Cancelar.</p>
--	--	--	--	--	--	--	--	---

En la tabla mostrada anteriormente la V indica válido, I indica inválido, y NA que no es necesario proporcionar un valor del dato en este caso, ya que es irrelevante. Para consultar el resto de los diseños de los casos de prueba consultar la documentación asociada al expediente del proyecto de pregrado del CENIA.

Conclusiones parciales

En este capítulo fueron abordadas las pruebas realizadas a la propuesta de solución y las que se consideran necesarias realizar posterior y durante la implementación de la funcionalidad, guiado por el proceso de desarrollo basado en las prácticas ágiles del nivel 2 de CMMI. Además se proponen un conjunto de pruebas para lograr el correcto funcionamiento del sistema. Lo cual permitió concluir que la detección de inconsistencias es un paso fundamental para obtener una correcta implementación de la propuesta de solución, así como el correcto diseño de casos de prueba permiten tener una visión más acertada sobre cuáles funcionalidades no deben dejar de ser implementadas, ha sido un paso imprescindible para definir los patrones de conducta que debe asumir la aplicación en el futuro.

Conclusiones

A partir del estudio del proceso de planificación docente, se conocieron las principales características de este proceso, que engloba como uno de sus elementos la planificación de actividades de formación. Este análisis permitió conocer cómo se ejecuta actualmente el proceso de planificación docente, y cómo se debe realizar una vez que se informatice.

El estudio de las normativas establecidas por el MES y los aspectos señalados por el cliente permitieron identificar los requisitos funcionales, describiendo cada uno de estos con sus respectivos prototipos de interfaz como forma de validación de los mismos; así como las pruebas para lograr su correcto funcionamiento en el sistema.

Con el análisis de algunos sistemas homólogos se comprobó que estos no satisfacen todas las necesidades y condiciones requeridas, sin embargo constituyeron una base de estudio para llegar a la obtención de la propuesta de la nueva solución. El estudio de las herramientas, lenguajes y proceso de desarrollo definido por el CENIA, permitió la familiarización con los elementos del entorno de desarrollo y adquirir los conocimientos necesarios sobre los mismos, para poder utilizarlos en la construcción de la propuesta de solución.

Recomendaciones

Aunque se dio cumplimiento a los objetivos definidos en la investigación se recomienda:

- Realizar la implementación de la propuesta de solución para la planificación de las actividades docentes.
- Realizar las pruebas definidas en la propuesta de solución.
- Realizar pruebas de integración a la solución propuesta implementada con el Sistema de Gestión Académica de Pregrado.
- Obtener informes que resuman el balance de carga docente del profesorado y de los departamentos según el semestre, curso, año académico y plan docente deseado.

Referencias bibliográficas

- Ackoff. (1981). *Apuntes sobre Gestión*. Obtenido de <http://www.apuntesgestion.com/2008/08/13/definicion-planificacion/>
- Bizagi. (2012). *Bizagi. BPMN Business Process Modeling Notation* obtenido de <http://wiki.bizagi.com/es/index.php?title=BPMN>
- Calderón Serrano, J. D., Montano Hernández, R., Núñez Abarca, J. A., y Oviedo Martínez, M. A. (2009). *Sistema informático de apoyo para la planificación didáctica para los docentes de educación básica*. Obtenido de <http://debian.ues.edu.sv/compartidos/gato/documentos/tareas%20en%20la%20ues/sapdeb/sapdeb/recursos/SAPDEBmanual/index.htm>
- Castro, N., Inzunza Quitral, M. (2010). *Rediseño de proceso mediante un enfoque de red conversacional – caso Planificación Docente Universidad de Talca*. Obtenido de <http://www.librospdf.net/>
- Chaves, M. A. (2006). *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software*.
- Cleger Tamayo, S., Pérez Campaña, M., y Rodríguez Expósito, F. (2007). *Alternativa para el proceso de planificación de horarios docentes de*. *Revista Ciencias Holguín* (13), 8.
- CodeIgniter. (2011). *CodeIgniter User Guide Version 2.1.0*. Obtenido de http://codeigniter.com/user_guide/overview/mvc.html
- CUJAE. (2011). *Manual de Usuario del SIGENU Sistema de Gestión de la Nueva Universidad*. Recuperado el 30 de abril de 2012, de <http://www.isch.edu.cu/sigenu>
- EcuRed. (2009). *Patrones de diseño y arquitectura*. Obtenido de http://www.ecured.cu/index.php/Patrones_de_dise%C3%B1o_y_arquitectura
- Company, V. P. (2011). *UML, BPMN and Database Tool for Software Development*. Retrieved from <http://www.visual-paradigm.com>
- Gamma, E., Richard Helm, Johnson, R., y Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*.

Guardado, J. L. (2011). *Investigaciones sobre la educación de la República de Guatemala*
Obtenido de <http://beto.stormpages.com/le/index.html>

Garrido, L. (2004). *Un Sistema De Gestión Docente Universitario: una experiencia en acción.*
Obtenido de <http://contenidosweb.usalca.cl/index.php?option=content&task=view&id=68>

Guglielmetti, M. (2004). *MASTERMAGAZINE. Definición de Cliente / Servidor* obtenido de
<http://www.mastermagazine.info/termino/4294.php>

Gutiérrez, P. P. (2009). *El proceso de desarrollo de software* Obtenido de <http://ingenieria-en-sistemas-de-software.blogspot.com/2009/02/el-proceso-de-desarrollo-de-software.html>

José H. Canós, P. L. (2009). *Métodologías Ágiles en el Desarrollo de Software.* Valencia.

María, L. L. (2009). *Lenguaje UML.* Obtenido de <http://www.hipertexto.info/documentos/uml.htm>.

Mary Beth Chrissis, M. K. (2009). *CMMI® Guía para la integración de procesos.*

Pressman, R. S. (2005). *Ingeniería dle Software, Un enfoque práctico.[6ta Edición] s.l. : Mc Graw Hill.*

Project, P. (2010). *Get the Latest Release:*Obtenido de <http://pencil.evolus.vn/en-US/Home.aspx>

Real Academia Española. (2000). *Diccionario de la Lengua Española.* Obtenido de
http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=planificaci%C3%B3n

Reynoso, C. (2004). *Introducción a la Arquitectura de Software.* Obtenido de
<http://carlosreynoso.com.ar/archivos/carlos-reynoso-introduccion-a-la-arquitectura-de-software.pdf>

Sommerville, I. (2005). *Ingeniería de Software.* Madrid.

Sommerville, I. (2005). *Ingeniería del Software.*Septima Edición. Madrid: Pearson Education.

Universidad de Oriente. (Diciembre de 2010). *Sistema informático para la gestión de la planificación docente.* Obtenido de
<http://tallertematico2010.fordes.co.cu/public/site/104.pdf>

Universidad de Sevilla. (2002). *Ingeniería de Requisitos para Aplicaciones web-Un estudio comparativo.* Sevilla.

UML, BPMN and Database Tool for Software Development. (2011). Obtenido de
<http://www.visual-paradigm.com>

Universidad politécnica de Valencia. (2009). Proceso de desarrollo de software. Valencia. España.

Visual Paradigm Company. (2011). UML, BPMN and Database Tool for Software Development. Obtenido de <http://www.visual-paradigm.com>

Visconti, M., y Astudillo, H. (2011). Fundamentos de Ingeniería de Software. Retrieved from Universidad Técnica Federico Santa María: <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/08-Patrones.pdf>

Zabalza, M. Á. (2005). Planificación Didáctica de la Docencia Universitaria en el Marco de la Convergencia Europea. (U. P. Navarra, Ed.) Navarra, España. Obtenido de www.alu.ua.es/p/psp4/historia/index.html

Resolución 210 del Ministerio de Educación Superior 2006

Resolución 120 del Ministerio de Educación Superior 2010

Anexo 1: Diagrama del proceso de negocio de la planificación docente con BPMN

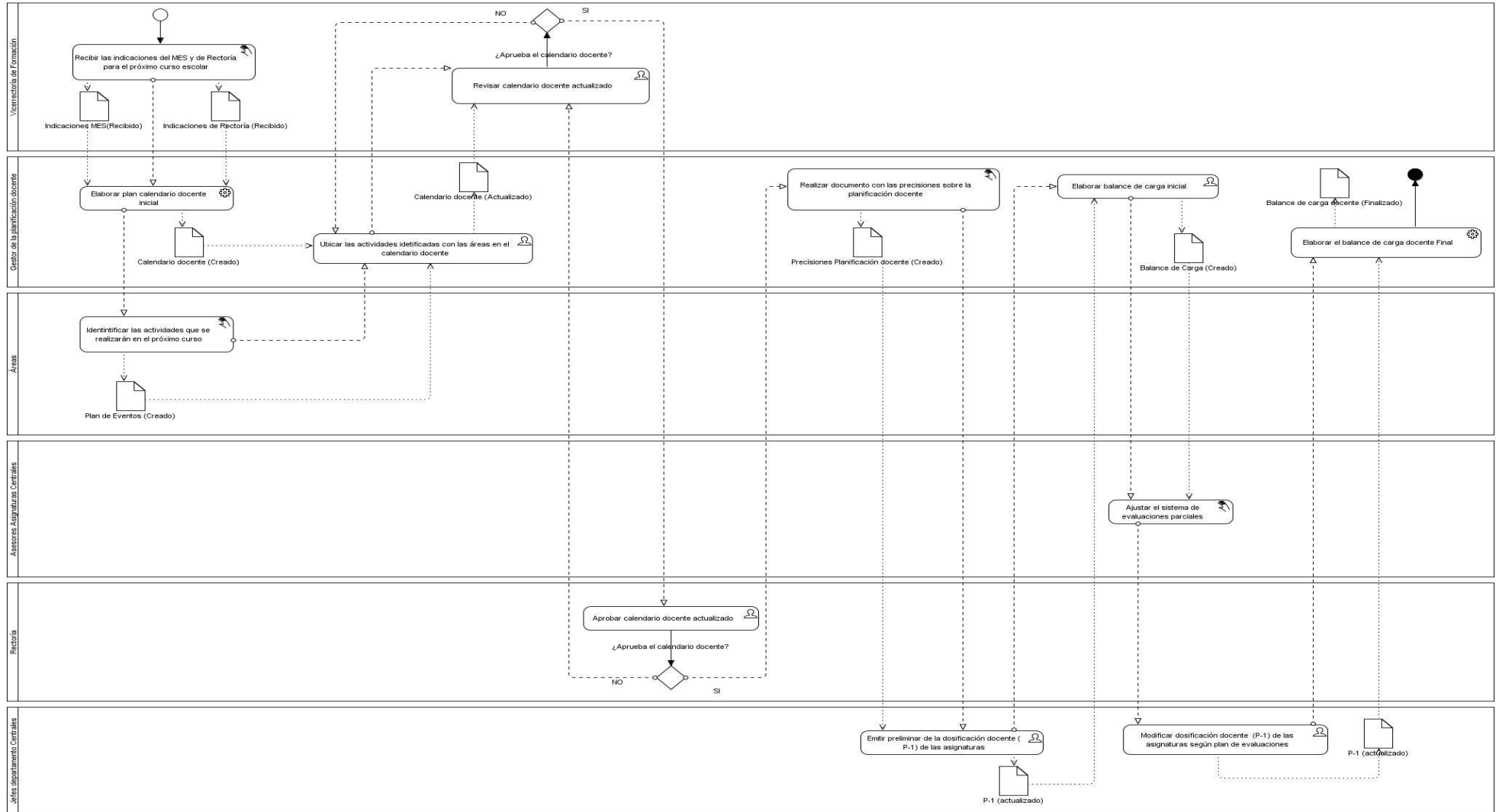


Fig. 17 Diagrama del proceso de negocio de la planificación docente con BPMN

