



FACULTAD 4

Universidad de las Ciencias Informáticas

**“Sistema para la gestión de la información, el control y la
evaluación de los profesores y especialistas de la
Facultad 4.”**

**Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas**

Autores:

Rosalía Suárez Núñez
Frank Ernesto Lorenzo Suárez

Tutores

Ing. Yusdel Meriño Almaguer
Ing. Angel Alberto Vazquez Sánchez

“Año 56 de la Revolución”

La Habana, Junio 2014



“Los grandes sueños se hacen realidad, con inmensos sacrificios”

Che



Declaración de Autoría

Declaramos ser los únicos autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de ella para lo que necesite, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

Firma del autor
Rosalia Suárez Núñez

Firma del autor
Frank Ernesto Lorenzo Suárez

Firma del tutor
Ing. Yusdel Meriño Almaguer

Firma del cotutor
Ing. Angel Alberto Vazquez Sánchez



Del tutor:

ymerino@uci.cu

Del cotutor:

aavazquez@uci.cu



De Rosalía:

Mi primer agradecimiento es para mis padres por haberme dado la vida.

A mi papá por siempre haberme cuidado y confiado en mí, por apoyarme y brindarme todo su cariño, por haber sido padre y madre a la vez, por estar siempre que lo he necesitado.

A mi mamá, que aunque desafortunadamente ya no se encuentre físicamente presente siempre la llevo en mi corazón.

A mi hermanita por siempre estar ahí, por escucharme y aconsejarme.

A mis abuelos paternos y maternos por haberme criado con tanto cariño.

A mis tíos por ser tan atentos y siempre haberme dado lo mejor de ellos.

A toda mi familia por darme tanto cariño y apoyo.

A mi novio por haberme acompañado desde mis inicios en la universidad, compartiendo buenos y malos momentos y a su familia por acogerme como su hija.

A todas mis amistades por ser las personas con las que siempre he podido contar en todo momento: Yanet, Elena, Analiet, Aymeé, Gretel, Aimara, Dainelys y Ariannis, por ser unas de mis amigas más allegadas, a todas las quiero mucho!!!

A todos los profesores que de una forma u otra me han ayudado en mi paso por la universidad.

A todas las personas que me han ayudado y aconsejado para llegar hasta aquí.

Un agradecimiento muy especial para mi compañero de tesis por todo el trabajo que pasamos juntos y porque sin él no hubiera sido posible llegar hasta aquí.

A los tutores por su dedicación y brindarnos parte de su tiempo.

A todos mis más sinceros agradecimientos...

De Frank Ernesto:

Agradezco primeramente a mis padres que sin ellos esto no hubiera sido posible. A toda mi familia. A mi compañera de tesis por todo el trabajo que hemos pasado juntos. A mis tutores por todos los consejos y ayuda que me brindaron y a todos los amigos que tengo y he tenido en la UCI.

De Rosalía:

Dedico esta tesis a toda mi familia que siempre me ha apoyado en especial a mi mamá que aunque ya no se encuentre presente físicamente, siempre fue mi sostén y mi mejor amiga.

A mi papá que siempre me ha sabido guiar durante toda mi vida estudiantil, por ser un padre ejemplar y siempre estar ahí.

A mi hermana por siempre estar pendiente de cada paso que doy y apoyarme en cada decisión que he tomado en la vida.

A toda mi familia por brindarme tanto cariño y comprensión en todo momento.

De Frank Ernesto:

Dedico esta tesis a mis padres por todo el apoyo que me han dado y por todo el sacrificio que han hecho por mí.

A mi hermana y a mi hermano que está por nacer.

A mis abuelos, que ya no están físicamente conmigo y mis abuelas.

A todos mis tíos, tías y primos.

A mi gente del barrio y a todo el piquete de la UCI.



A raíz de la necesidad de informatizar el país surge la Universidad de las Ciencias Informáticas (UCI), con el principal objetivo de preparar y graduar ingenieros que cumplan con esta importante misión. Para esto, la UCI cuenta con una gran cantidad de profesores, que son distribuidos por departamentos docentes en cada facultad. Por tal razón, es necesario informatizar muchos de los procesos que se gestionan en la misma.

Uno de los procesos anteriormente mencionados es la informatización de la información de los profesores y especialistas de la Facultad 4. Con ese propósito se realiza el presente trabajo, que tiene como objetivo realizar un sistema informático para la gestión de la información, el control y la evaluación de los profesores y especialistas de dicha facultad.

Este sistema permitirá la gestión y almacenamiento de la información y los planes de trabajo de los profesores y especialistas de la Facultad 4, la realización de reportes estadísticos sobre el cumplimiento de los planes de trabajo correspondientes a cada nivel y los jefes de departamento podrán darle seguimiento a la evaluación de los profesores.

Palabras clave: evaluación profesoral, plan de trabajo de profesores

<i>Introducción</i>	1
<i>Capítulo 1: Fundamentación Teórica</i>	5
1.1 Introducción.....	5
1.2 Principales conceptos asociados al dominio del problema	5
1.2.1 Planeación Estratégica	5
1.2.2 Plan de trabajo.....	7
1.2.3 Evaluación profesoral	8
1.3 Sistema de gestión de la información	10
1.4 Soluciones similares.....	10
1.4.1 Internacional	10
1.4.2 Nacional.....	11
1.4.3 Universidad de las Ciencias Informáticas.....	12
1.5 Posibilidad de integración con otras aplicaciones.....	13
1.6 Aplicaciones Web.....	13
1.7 Metodología de desarrollo	14
1.7.1 Proceso Unificado de Desarrollo de <i>Software</i>	14
1.7.2 Programación Extrema	15
1.8 Lenguaje de modelado	15
1.9 Herramienta CASE.....	16
1.9.1 Visual Paradigm-UML	17
1.10 Lenguaje de programación.....	17
1.10.1 Lenguaje de programación del lado del servidor.....	18
1.10.2 Lenguaje de programación del lado del cliente	18
1.11 <i>Framework</i> de desarrollo.....	19
1.12 Entorno de desarrollo	20
1.13 Sistema Gestor de Base de Datos.....	21
1.14 Servidor web	22
1.15 Conclusiones parciales.....	23
<i>Capítulo 2: "Propuesta de solución"</i>	24
2.1 Introducción.....	24
2.2 Descripción del sistema propuesto	24
2.3 Patrón arquitectónico	25

2.4 Objetivos del sistema	25
2.5 Usuarios relacionados con el sistema	26
2.6 Exploración	27
2.6.1 Historias de Usuario.....	27
2.7 Planificación	30
2.7.1 Lista de Reserva del Producto	30
2.7.2 Estimación de esfuerzos por HU.....	35
2.8 Plan de Iteraciones.....	36
2.9 Plan de duración de las iteraciones.....	37
2.10 Plan de entrega.....	38
2.11 Prototipo de interfaz de usuario no funcional.....	39
2.12 Tarjetas CRC	40
2.13 Conclusiones parciales.....	42
<i>Capítulo 3: Implementación y pruebas</i>	43
3.1 Introducción.....	43
3.2 Módulos del sistema.....	43
3.3 Modelo de datos.....	43
3.4 Implementación.....	45
3.5 Tareas de ingeniería	47
3.6 Patrones de diseño usados en la solución.....	52
3.7 Estándares de codificación.....	53
3.8 Pruebas.....	54
3.8.1 Pruebas unitarias	55
3.8.2. Pruebas de aceptación	55
3.9 Conclusiones parciales.....	63
<i>Conclusiones generales</i>	64
<i>Recomendaciones</i>	65
<i>Referencias bibliográficas</i>	66
<i>Anexos</i>	70
<i>Glosario de términos</i>	116

Ilustración 1: Partes del proceso de planeación	5
Ilustración 2: UNIVERSITAS XXI-Recursos Humanos	11
Ilustración 3: Prototipo de interfaz de usuario no funcional	40
Ilustración 4: Estructura de las tablas en la base de datos.....	44
Ilustración 5: Resultado de las pruebas de aceptación realizadas por el cliente	62
Ilustración 6: Resultado de las pruebas de aceptación realizadas por el grupo de calidad	62

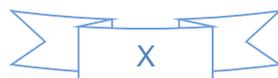


Tabla 1: Usuarios del sistema.....	26
Tabla 2: HU Editar perfil de usuario	28
Tabla 3: HU Gestionar departamento	28
Tabla 4: HU Gestionar actividad	28
Tabla 5: HU Buscar profesor.....	29
Tabla 6: HU Gestionar plan de trabajo de la facultad	29
Tabla 7: Funcionalidades del sistema	31
Tabla 8: Restricciones funcionales del sistema.....	33
Tabla 9: Estimación de esfuerzo por HU.....	35
Tabla 10: Plan de duración de las iteraciones.....	37
Tabla 11: Plan de entrega de las HU	38
Tabla 12: Tarjeta CRC UsuarioController.....	41
Tabla 13: Tarjeta CRC ProfesorController	41
Tabla 14: Tarjeta CRC DepartamentoController	41
Tabla 15: Tarjeta CRC PlanIndividualController	41
Tabla 16: Tarjeta CRC ParametroController	42
Tabla 17: Funcionalidades abordadas en la primera iteración	45
Tabla 18: Funcionalidades abordadas en la segunda iteración.....	46
Tabla 19: Funcionalidades abordadas en la tercera iteración	46
Tabla 20: Funcionalidades abordadas en la cuarta iteración	47
Tabla 21: TI Editar datos de usuario	48
Tabla 22: Tarea de ingeniería Buscar información de un profesor	49
Tabla 23: TI Realizar un control del cumplimiento de las actividades.....	50
Tabla 24: TI Importar plan de trabajo	51
Tabla 25: CP de aceptación P1	56
Tabla 26: CP de aceptación P8	57
Tabla 27: CP de aceptación P17	58

Introducción

En la actualidad la informática se encuentra en constante desarrollo a nivel internacional por lo que ha alcanzado un gran auge mundialmente. A raíz de la necesidad de informatizar la sociedad, Cuba decide dar los primeros pasos para lograr el cumplimiento de este objetivo. Con este propósito surge la Universidad de las Ciencias Informáticas (UCI), creada al calor de la Batalla de ideas como un proyecto de la Revolución.

La UCI cuenta con centros productivos que tienen como principal misión desarrollar la industria del *software* en Cuba, para contribuir con el desarrollo económico del país. Para esto cuenta con riquezas tanto en tecnologías como en capital humano.

En la UCI, como en muchas universidades del país, se deben realizar evaluaciones periódicas para determinar en qué medida se cumplen o no los objetivos que se propone. La información obtenida de dichas evaluaciones es primordial para trazar líneas de actuación dirigidas a mejorar las actividades docentes con el fin de elevar la calidad de la actividad universitaria.

Como parte del capital humano, la universidad posee un gran número de profesores los cuales no solo realizan actividades docentes, sino otras que tienen que ver con su categoría docente, formación, producción, superación y extensión universitaria. La realización continua de estas actividades hace engorroso el control de la información de los profesores y especialistas, así como de los planes de trabajo por el gran número de indicadores que se generan.

Actualmente la información tiene un valor primordial para cualquier entidad. Por tal razón, es necesario administrarla y gestionarla de la mejor manera posible con el fin de posibilitar conocimiento a todo el que la consulte.

La Facultad 4 de la UCI, no cuenta con un sistema que permita gestionar la información de los profesores y especialistas por departamentos docentes, en la actualidad existe una descentralización de datos e informaciones importantes de los docentes, lo que trae como consecuencia afectaciones a los directivos a la hora de tomar cualquier decisión con respecto a los profesores, teniendo en cuenta que los datos de los mismos son almacenados en procesadores de texto y hojas de cálculo, lo que dificulta su actualización, en caso de ocurrir un cambio en algún profesor solo se actualiza en la documentación del Jefe de departamento.

Toda la información que se genera en la Facultad 4 referente a los profesores se tramita vía correo electrónico debido a que no se encuentra centralizada para su consulta y actualización.

Existen dificultades en la evaluación de los profesores y especialistas pues estos no poseen un espacio para llevar un registro del cumplimiento de las actividades planificadas en su plan de trabajo y que tributan a su evaluación trimestral y anual, que pueda ser consultada por el jefe de departamento

para evaluarlas oportunamente y evitar errores por el tiempo que transcurre entre una evaluación y otra, en que muchas actividades pueden quedar olvidadas.

La institución establece tres tipos de planes de trabajo: Plan de trabajo individual de cada profesor o especialista, plan de trabajo por departamento docente y plan de trabajo de la facultad. El proceso de gestión de los mismos se realiza de forma manual, con el apoyo de herramientas informáticas tales como *Microsoft Office* con los programas de procesadores de texto, hojas de cálculo y clientes de calendario, lo cual hace engorroso este proceso. Se torna complejo la posibilidad de realizar cambios en los planes de trabajo y que sean del conocimiento inmediato de todos los involucrados, actualmente esta información se envía por correo electrónico y son numerosos los mensajes que se envían con las actualizaciones.

A partir del análisis realizado de la situación problemática descrita, se identifica el siguiente **problema a resolver**: ¿Cómo contribuir a la gestión de la información, el control y la evaluación de los profesores y especialistas de la Facultad 4?

Para dar solución al problema a resolver se trazó como **objetivo general**: Desarrollar una aplicación informática para la gestión de la información, el control y la evaluación de los profesores y especialistas de la Facultad 4.

La presente investigación tiene como **objeto de estudio**: Proceso de gestión de la información, el control y la evaluación de profesores y especialistas, enmarcándose en el **campo de acción**: Sistemas informáticos para la gestión de la información, el control y la evaluación de los profesores y especialistas de la Facultad 4.

A partir del objetivo general se generaron los siguientes **objetivos específicos**:

- Realizar estudio del estado del arte sobre sistemas para la planificación, evaluación y control de recursos humanos.
- Seleccionar la metodología, herramientas y tecnologías a utilizar para el desarrollo del sistema.
- Implementar el sistema según la metodología seleccionada.
- Validar mediante pruebas de *software* el funcionamiento del sistema.

Por lo anteriormente planteado se enuncia como **idea a defender**: con la implementación del sistema de gestión propuesto se podrá realizar un mejor control de la información y evaluación de los profesores y especialistas de la Facultad 4.

Para dar cumplimiento a los objetivos específicos se plantean las siguientes **tareas**:

- Estudio del estado del arte:
 - Análisis de sistemas similares a nivel internacional, nacional y en la UCI.

- Selección de metodología, herramientas y tecnologías a utilizar para el desarrollo de la solución propuesta.
- Confección de los artefactos ingenieriles.
- Definición del modelo de datos.
- Implementación de la solución propuesta.
- Realización de pruebas para detección de errores.

Métodos Científicos de Investigación

Los métodos científicos de investigación representan una forma de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y relaciones. Se clasifican en teóricos y empíricos, los cuales están dialécticamente relacionados. (1)

Los métodos teóricos permiten estudiar las características del objeto de investigación que no son observables directamente, crean las condiciones para ir más allá de las características superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis. (1)

Para la realización del presente trabajo se utilizaron una combinación de métodos de la investigación científica, los que responden a los objetivos planteados.

- **Histórico-Lógico:** Permitió el análisis de la trayectoria del proceso, brindando información relevante sobre el mismo. Durante la investigación, se utilizó para realizar un análisis de la documentación relacionada con el proceso de gestión y control de la información de los profesores y especialistas de la Facultad 4, así como de la gestión de los planes de trabajo de cada uno de ellos.
- **Analítico-Sintético:** Permitió dividir el problema en pequeñas partes para así realizar un análisis más profundo de cada una de ellas, tomando sus aspectos esenciales enfocados a la situación problemática. Durante la investigación, se utilizó en el análisis del proceso de gestión de los planes de trabajo y la evaluación de los profesores y especialistas, los cuales se analizaron por separado para entender mejor su funcionamiento. La utilización de este método facilitó extraer aspectos fundamentales presentes en la bibliografía consultada durante la investigación y que son esenciales en el desarrollo de la misma.
- **Modelación:** Se utilizó para la realización de modelos que representen las relaciones existentes entre las definiciones y características del objeto de estudio, para mejor comprensión del sistema.

Para lograr el cumplimiento de los objetivos propuestos el documento se estructura en tres capítulos que se describen a continuación:

Capítulo 1: “Fundamentación teórica”

Este capítulo contiene la fundamentación teórica del tema, donde se exponen una serie de conceptos asociados al dominio del problema. Se realiza un estudio de las soluciones similares al campo de acción y se selecciona la metodología, herramientas y tecnologías más apropiadas para el desarrollo del sistema.

Capítulo 2: “Propuesta de la solución”

En este capítulo se realiza la descripción de la solución propuesta. Se determinan los servicios que brindará el sistema una vez desarrollado y se definen las funcionalidades que debe cumplir. Se generan los artefactos que propone la metodología seleccionada.

Capítulo 3: “Implementación y prueba”

En este capítulo se recoge la información relacionada con la implementación del sistema y las pruebas realizadas al mismo. Se implementan todas las funcionalidades anteriormente definidas y se detallan las pruebas realizadas al sistema una vez desarrollado, con el objetivo de asegurar la calidad de la solución.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En este capítulo se tratan una serie de elementos relacionados con el marco teórico en el que se enmarca la investigación, para un mejor entendimiento del trabajo. Se realiza un estudio detallado de los principales conceptos, así como de las metodologías de desarrollo de *software* existentes, las herramientas y las tecnologías, para así seleccionar las más eficientes a utilizar en el desarrollo del sistema.

1.2 Principales conceptos asociados al dominio del problema

En la problemática de la presente investigación se ven reflejados varios **conceptos** que por su importancia son enunciados a continuación:

1.2.1 Planeación Estratégica

La planeación es una etapa que forma parte del proceso administrativo mediante el cual se definen estrategias y se seleccionan alternativas en función de las metas y objetivos principales de una empresa. (2)



Ilustración 1: Partes del proceso de planeación

El carácter estratégico de la planeación no solo trata de prever un camino por el cual transitar, si no que busca anticipar su rumbo y si es posible cambiar su destino.

Luego de realizado un análisis de la bibliografía consultada, que tratan el término “Planeación estratégica”, se extrajeron varios conceptos que fueron emitidos por diferentes autores, lo que queda evidenciado en las siguientes definiciones:

Según el licenciado en economía y editorialistas Acle Tomasini, la planeación estratégica se define como un conjunto de acciones que deben ser desarrolladas para lograr objetivos estratégicos; lo que implica definir y priorizar los problemas a resolver, plantear soluciones, determinar los responsables para realizarlos, asignar recursos para llevarlos a cabo y establecer la forma y periodicidad para medir los avances. (3)

Menguzzato y Renau la definen como un análisis racional de las oportunidades y amenazas que presenta el entorno para la organización, de los puntos fuertes y débiles de la misma frente a este entorno y la selección de un compromiso estratégico. (3)

A partir de las investigaciones realizadas los autores de la presente investigación coinciden con el profesor Henry Mintzberg, cuando define la planeación estratégica como: proceso de relacionar las metas de una organización, determinar las políticas y programas necesarios para alcanzar objetivos específicos en camino hacia esas metas y establecer los métodos necesarios para asegurar que las políticas y los programas sean ejecutados, o sea, es un proceso formulado de planeación a largo plazo que se utiliza para definir y alcanzar metas organizacionales. (3)

El estudio realizado permitió a los autores de la presente investigación extraer aspectos en común que presentan los conceptos antes mencionados y otros que fueron analizados, como son:

- Es un proceso que se utiliza para definir y alcanzar las metas organizacionales.
- Se deben establecer los mecanismos necesarios para poder evaluar el cumplimiento de lo acordado.
- Es un proceso de planeación a largo plazo.
- Se realiza sobre la base de un análisis del ambiente.

Algunas de las **principales características** con las que cuenta el proceso de planeación estratégica son las siguientes: (2)

- Está proyectada a varios años, con efectos y consecuencias futuristas.
- Ampara a la empresa como una totalidad, abarca todos los recursos y áreas de actividades y se preocupa por trazar los objetivos a nivel de organizaciones.
- Es definida por la cima de la organización y corresponde al plan mayor, al cual están subordinados todos los demás.

La planeación estratégica es un término que tiene asociados varios **beneficios** que por su importancia se enuncian a continuación: (4)

- Las organizaciones que la utilizan son más exitosas que las que no la usan.
- Las empresas que tienen altos rendimientos reflejan una orientación más estratégica y enfoque a largo plazo.
- Aumenta la capacidad de prevención de problemas.
- Provee las bases para clasificar las responsabilidades individuales.

1.2.2 Plan de trabajo

Los planes de trabajo son esenciales en los centros educacionales, estos son la base fundamental para llevar el control de las actividades asignadas a los profesores a lo largo de los cursos docentes.

La bibliografía consultada, permitió extraer varios conceptos de plan de trabajo, que se encuentran estrechamente relacionados al dominio del problema:

- Un plan de trabajo es el compromiso que tienen los profesores de realizar actividades en los campos de la investigación, docencia, extensión universitaria y la administración académica, sin perjudicar las demás tareas relacionadas con su condición de miembro de la comunidad universitaria. Este conjunto de actividades permitirá subsiguientemente evaluar al profesorado durante su desempeño particular. Debe contener metas, objetivos, actividades, período de tiempo para realizar las mismas, además de contar con indicadores de logros y resultados. (5)
- Un plan de trabajo no es más que un instrumento de planificación que ordena y sistematiza determinada información de modo que pueda tenerse una visión del trabajo a realizar, así nos indica: objetivos, metas, actividades, responsables y fecha de las actividades. Se escribe para planear las actividades de un período de tiempo dado, primero para convencer de su aprobación a los que toman las decisiones, luego como documento guía de las actividades a efectuar durante ese período. (6)

El estudio realizado de diversos conceptos de plan de trabajo permitió a los autores adoptar el primer concepto de los antes mencionados para desarrollar el presente trabajo de diploma, debido a que es el que guarda más relación con la problemática presente.

Una de los objetivos fundamentales por los que se realizan los planes de trabajo es porque son prácticos, permiten realizar el seguimiento de la ejecución y ayuda a la reprogramación de actividades, facilita el proceso de evaluación del profesorado, puesto que la misma dependerá en gran medida del cumplimiento de las actividades que queden reflejadas en los planes de trabajo. Estos objetivos deben cumplir con varios criterios, entre los que se encuentran: estar fundamentados en la misión, visión y metas institucionales y deben hacer referencia a datos cualitativos y cuantitativos existentes que

puedan apoyar a nuevos objetivos que se proponga la organización. (7)

Los planes de trabajo cuentan con un grupo de **elementos fundamentales** que se detallan seguidamente: (6)

- Metas: son expresiones generales sobre lo que se quiere lograr por parte de una organización para cumplir con una misión.
- Objetivos: definen el cómo lograr las metas.
- Actividades: son las expresiones específicas de las tareas a llevarse a cabo para alcanzar los objetivos trazados. Indican la persona responsable de realizar la actividad y las fechas en que se llevarán a cabo las mismas.
- Período de tiempo para realizar las actividades.
- Indicadores de logros, resultados o productos.

Entre los **beneficios** de la planeación del trabajo se encuentran:

- Crear una atmósfera de trabajo en equipo y cooperación.
- Informar al personal sobre el rango y tipo de actividades de los demás.
- Motivar al personal a trabajar con objetivos que suponen un desafío y que sean realistas.
- Infundir en el personal un sentido de compromiso en la obtención de los objetivos a largo y a corto plazo.
- Crear un formato que sea lo suficientemente flexible para incorporar y responder a cambios inesperados.

1.2.3 Evaluación profesoral

Una de las cuestiones más defendidas en la actualidad es la necesidad prioritaria de elevar la calidad de la educación, donde la evaluación del desempeño de los docentes es una variable de gran trascendencia.

Como sostiene el doctor Antonio Medina Rivilla, “la evaluación es la actividad reflexiva que nos permite conocer la calidad de los procesos y los logros alcanzados en el desarrollo del proyecto. La evaluación es una valoración sistemática que facilita el conocimiento minucioso de los procesos aplicados y fundamentalmente las decisiones futuras de cambio que nos proponemos llevar a cabo”. (8)

La evaluación del profesorado universitario es un tema de gran importancia para el desarrollo de la educación superior, debido a que compromete la calidad de los procesos sustantivos de las universidades a partir de la consideración de las categorías de competencia y desempeño profesoral.

Luego de un estudio parcial realizado a varios conceptos enunciados por diferentes autores del término evaluación, se resumen los siguientes resultados:

Según Valle Lima, la evaluación “es el proceso sistemático durante el cual se analizan de manera cualitativa las transformaciones que tienen lugar en la institución como consecuencia de un sistema de influencias permitiendo obtener juicios de valor sobre ese desarrollo y posibilitando la toma de decisiones”. (9)

Los asesores del Ministerio de Educación en Cuba definen la evaluación del docente como “una función del dirigente en busca de la excelencia. Es un proceso sistemático y sistémico que tiene un doble carácter: el valorativo y el desarrollador”. (10)

Después de analizados los conceptos anteriormente expuestos, los autores del presente trabajo consideran, que la definición de evaluación profesoral más relacionada al marco de la investigación es:

La evaluación del desempeño de los trabajadores es la mediación sistemática de la actividad laboral que realizan durante un período de tiempo y de su potencial desarrollo en el ámbito de la entidad laboral. Además de ser un proceso permanente que se consolida cada año, mediante la ponderación de las calificaciones obtenidas por el profesor en las diferentes funciones y actividades consignadas en el plan de trabajo, la evaluación profesoral deberá ser objetiva, imparcial, formativa e integral, y valorará el cumplimiento y la calidad de las actividades desarrolladas por el profesor, ponderadas según la importancia de ellas y el grado de responsabilidad del profesor en cada una. (11)

Los **objetivos fundamentales** de la evaluación del desempeño de los profesores en las universidades son: (12)

- Valorar el cumplimiento de las tareas de su plan de trabajo y la calidad alcanzada en cada una, sus principales aciertos y desaciertos.
- Revisar el grado de cumplimiento de las recomendaciones realizadas en la evaluación anterior y establecer los objetivos individuales, que debe alcanzar en el período de tiempo que media entre dos evaluaciones.
- Evaluar su disciplina y comportamiento, sus relaciones con sus compañeros y estudiantes, así como su contribución al cumplimiento de los objetivos de su área y de la institución en general.
- Determinar el grado de desarrollo de sus motivaciones profesionales, sus valores éticos y revolucionarios.
- Sentirse reconocido y estimulado moral y materialmente por los resultados de su trabajo y méritos alcanzados.

En la UCI, los especialistas y los profesores son evaluados por el jefe de departamento trimestralmente con las categorías de Deficiente, Adecuado y Superior, que es su evaluación de desempeño, lo cual influye en el pago de salario. Los profesores también son evaluados anualmente de Excelente, Bien, Regular o Mal, en dependencia del cumplimiento de las actividades reflejadas en su plan de trabajo.

1.3 Sistema de gestión de la información

La información es un elemento fundamental para el desarrollo, con el cursar de los años la gestión de la información ocupa cada vez más, un espacio mayor en la economía de los países a escala mundial. Los autores de la presente investigación coinciden con la Lic. Lourdes Aja Quiroga cuando expresa que la información es “la forma social de existencia del conocimiento consolidada en una fuente determinada.” (13)

De igual forma plantea que la gestión de la información “comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada.” (13)

Los sistemas de gestión de la información surgen entonces tras la necesidad de controlar los grandes volúmenes de información de una forma fácil, sencilla, confiable y además para obtener el mayor rendimiento de este recurso.

1.4 Soluciones similares

Ante el interés que presentan empresas u organizaciones de mantener un control de las actividades que respectan a sus trabajadores y de esta forma tener un punto de partida para otorgar sus evaluaciones, surge la necesidad de incrementar el desarrollo de sistemas informáticos capaces de cumplir con este objetivo. Para el desarrollo de la propuesta de solución se hizo necesario realizar un estudio de los sistemas de información similares que existen a nivel internacional, nacional y en la UCI. Los resultados obtenidos se muestran a continuación:

1.4.1 Internacional

Luego de un estudio realizado a varios sistemas de gestión es posible enunciar que internacionalmente existe un sistema con características similares a las que se pretenden desarrollar, el cual es detallado a continuación:

UNIVERSITAS XXI - Recursos Humanos

Sistema implementado en la universidad de Oviedo, Asturias, España, que se encarga de la gestión de los recursos humanos y nóminas para universidades y otras administraciones públicas que cubren las

necesidades de administración, gestión, control e información en todos los colectivos (funcionarios, laborales y eventuales) y garantiza la integridad y coherencia de la información. Entre sus módulos se encuentra “Expediente Administrativo” que tiene como funcionalidad la gestión de datos personales y curriculares de los empleados, méritos, grados, cargos, ausencias y situaciones administrativas. (14) Este sistema es solo de carácter informativo.



Ilustración 2: UNIVERSITAS XXI-Recursos Humanos

1.4.2 Nacional

En Cuba también se encuentran sistemas que por sus similitudes se les realizaron los estudios pertinentes para ver en qué medida contribuyen con la propuesta de solución:

✚ Gestión de recursos humanos

Una investigación realizada en la Universidad de Cienfuegos tiene como objetivo documentar el desarrollo de un *software* para la gestión de información de profesores y cuadros que permita perfeccionar la gestión de los Recursos Humanos en dicha entidad, y a raíz de esto, el desarrollo pleno del proceso docente - educativo. Se desarrolló en la Dirección de Recursos Humanos, departamento de Capacitación, Cuadros y Estimulación, específicamente cuadros y profesores.

Se llevaron a cabo las etapas del proceso de desarrollo de *software*, según lo especificado en la metodología de Proceso Unificado de Desarrollo (RUP). Como resultado se obtuvo un sistema

automatizado para la Gestión de Información de profesores y cuadros que permite perfeccionar la gestión de los Recursos Humanos de la Universidad de Cienfuegos, basado en una interfaz web con un ambiente amigable, legible y de fácil navegación.

1.4.3 Universidad de las Ciencias Informáticas

Por su parte la UCI, cuenta con varios sistemas para la gestión de la información y los planes de trabajo de los profesores. Algunos de estos sistemas son:

✚ “Sistema de Gestión del Plan de Trabajo de los Profesores de la Facultad 1 de la Universidad de las Ciencias Informáticas”

El objetivo de este trabajo es el análisis, diseño e implementación de un sistema para la gestión del plan de trabajo de los profesores de la facultad 1. La realización de este sistema proporcionó un ahorro de tiempo a la hora de gestionar los planes de trabajo de los profesores de la Facultad 1, los jefes de departamento no tuvieron que depender exclusivamente de los profesores de su departamento docente para la creación o modificación de los planes de trabajo, se garantizó la seguridad de todos los datos necesarios de los profesores vinculados al plan, se pudo verificar el cumplimiento de las tareas realizadas por los profesores y una vez que el sistema fue puesto en práctica en todos los departamentos docentes de la facultad 1, se presentó la posibilidad de reutilizarlo en todas las demás facultades de la universidad y otros centros docentes que pudieran necesitar de un sistema informático para la gestión de los planes de trabajo de sus profesores.

El sistema de gestión expuesto anteriormente solamente se enfoca en los planes de trabajo de los profesores, no destacando así otros procesos que son importantes en los departamentos docentes, como son: el proceso de evaluación de los profesores y especialistas, que está dado por el cumplimiento de las actividades que componen sus planes de trabajo, así como el debido seguimiento y control que se le debe dar a los mismos.

✚ “Sistema para el control de la información y evaluación de los profesores en los departamentos docentes de la facultad 8”

Este sistema tiene como objetivo almacenar la información de los medios básicos y profesores, obtener reportes y además brindar a los jefes de departamento la posibilidad de evaluar a los profesores según las actividades realizadas y del cumplimiento del plan de trabajo anual. En esta solución no se toman en cuenta los planes de trabajo de la facultad y por departamentos docentes.

Del estudio realizado de las soluciones similares se concluye que las mismas poseen características que les permiten resolver un gran número de problemas. Sin embargo, ninguna satisface las necesidades vigentes en la facultad 4, puesto que no se detecta la existencia de un sistema de gestión

que agrupe la gestión de la información de los profesores y especialistas de dicha facultad, así como llevar a cabo el control del cumplimiento de las actividades reflejadas en los planes de trabajo y que tributan a su evaluación trimestral y anual.

Los sistemas mencionados anteriormente no se están poniendo en práctica en la actualidad, lo que mantiene la necesidad de implementar un sistema que permita la gestión, el control de la información y la evaluación de los profesores y especialistas de la Facultad 4, así como de los planes de trabajo individuales de cada uno, por departamentos docentes y de la Facultad. Sin embargo, al realizar un análisis de los mismos, se decide por parte del equipo de desarrollo tomar algunos aspectos positivos que ayuden a la confección de la propuesta de solución:

- La gestión de los planes de trabajo facilita el seguimiento y control que deben dar los jefes de departamento al cumplimiento de los mismos.
- La gestión de la información de los profesores y especialistas proporciona beneficios a los directivos a la hora de tomar cualquier decisión con respecto a los mismos.

1.5 Posibilidad de integración con otras aplicaciones

La informática es una rama de la ciencia que ha ido evolucionando a través de la historia hasta el punto de ser una de las más usadas por el hombre en la actualidad. Por esta razón es importante que cada vez que se realice un proyecto informático, se tengan en cuenta las posibilidades de integrarlo con otras aplicaciones que les puedan servir de apoyo.

Los autores de la presente investigación estudiaron esta posibilidad y se decidió integrar el presente sistema de gestión con el LDAP¹, el cual es un protocolo estándar que permite el acceso a directorios para buscar diversa información en un entorno de red mediante protocolos TCP/IP². LDAP también se considera una base de datos, aunque su sistema de almacenamiento puede ser diferente. En la aplicación se utilizará para la obtención de datos de usuarios.

1.6 Aplicaciones Web

Las aplicaciones Web forman un sistema íntegro de intercambio de datos, procesos e información entre el servidor (servidor Web) y los clientes (navegador). El usuario solicita una página Web existente en un servidor http mediante el navegador, el servidor envía entonces la información requerida al ordenador del usuario, el navegador interpreta los códigos HTML y presenta la información contenida en la página Web en la pantalla del ordenador del usuario. (15)

¹ LDAP *Lightweight Directory Access Protocol*, en español Protocolo Ligero de Acceso a Directorios.

² TCP/IP: Protocolo de Control de Transmisión/ Protocolo de Internet. Es el protocolo común utilizado por todos los ordenadores conectados a Internet, de manera que éstos puedan comunicarse entre sí.

Para elaborar el sistema se propone el desarrollo de una **aplicación web**, lo cual permitirá que se pueda utilizar desde cualquier área de la universidad con solo una computadora conectada a la red con navegador web.

Fundamentación de su selección

Las ventajas de las aplicaciones web avalan su selección: (15)

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

1.7 Metodología de desarrollo

Las metodologías de desarrollo de *software* surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto. (16)

Dentro del proceso de desarrollo de *software* se han implementado una serie de metodologías para el modelado de aplicaciones, que apuntan a resolver distintos problemas existentes en el desarrollo de este tipo de *software*. Estas se clasifican en dos grandes grupos: las metodologías pesadas, dentro del que se encuentra Proceso Unificado de Desarrollo de *Software* (RUP) y las metodologías ágiles, al que pertenece Programación Extrema o XP (*Extreming Programing*).

El primer grupo de los antes mencionados se basa fundamentalmente en el uso exhaustivo de la documentación, mientras que el segundo se preocupa más por las relaciones con el cliente y por la capacidad de respuesta a los cambios.

Luego de un estudio bibliográfico realizado sobre las metodologías de desarrollo de *software*, se arribó a la conclusión de que las dos metodologías mencionadas anteriormente son de las más usadas en el desarrollo de *software*, las cuáles serán analizadas a continuación:

1.7.1 Proceso Unificado de Desarrollo de *Software*

Este tipo de metodología se caracteriza fundamentalmente por realizar un mayor énfasis en la planificación y control del proyecto que se desee realizar, además de especificar de forma precisa los requisitos y el modelado de los flujos de trabajo que se planteen. Sin embargo, atendiendo a las características del ambiente de desarrollo de la presente investigación, el uso de esta metodología presentaría dificultades tales como: (17)

- El consumo excesivo de tiempo por la gran multitud de artefactos propios de la metodología, es un inconveniente para el equipo de desarrollo, puesto que no se dispone de mucho tiempo para la implementación del sistema.
- Hay poco personal de desarrollo, ya que solo dos personas están a cargo del desarrollo de la aplicación.
- En este tipo de metodología al aparecer un nuevo requisito o de ocurrir un cambio en alguno de ellos hay que comenzar una nueva iteración para dar cumplimiento a la funcionalidad, lo que representa una dificultad, debido a que se está en presencia de requisitos cambiantes.

Todos estos inconvenientes pueden ser eliminados usando la metodología XP.

1.7.2 Programación Extrema

Esta metodología es la más destacada dentro los procesos ágiles de desarrollo de *software*. Los programadores que la utilizan consideran que los cambios en los requisitos durante el desarrollo del *software* son un aspecto natural e inevitable. Algunas de las ventajas que presenta XP es que permite la retroalimentación, la presión está a lo largo de todo el proceso y no en una entrega final, permite definir en cada iteración cuales son los objetivos de la siguiente, es apropiada para entornos volátiles.

(17)

Luego de realizar el análisis de estas metodologías y teniendo en cuenta que tanto el sistema que se desea implementar como el equipo de desarrollo son pequeños, se seleccionó XP como metodología de desarrollo.

Fundamentación de la selección

Se decide seleccionar XP debido a que cuenta con características que están acorde al ambiente de desarrollo, tales como: el equipo de desarrollo está compuesto por grupos pequeños y trabajando en un mismo sitio, los diseños de *software* son sencillos y libres de complejidad, tiene una gran capacidad adaptiva al cambio, ya que los mismos se implementan rápidamente tal y como fueron sugeridos, por tanto, el manejo de los cambios se convierte en una parte sustantiva del proceso. Otra de las características principales que presenta esta metodología es que la prioridad es satisfacer al cliente mediante tempranas y s entregas de *software*, el cliente forma parte del equipo de desarrollo y existe una comunicación efectiva entre ambos lados. Además, el proceso de pruebas en XP es continuo, lo que propicia que la cantidad de errores sea pequeña. (18)

1.8 Lenguaje de modelado

Los lenguajes de modelado reducen la complejidad de un sistema al abstraerlo de detalles innecesarios. Se representan por notaciones en su mayoría visuales, que intentan representar un

sistema de *software* a un nivel mucho más alto que los lenguajes de programación, representándolo en forma más intuitiva para personas sin especialización en Informática. (19)

Uno de los lenguajes de modelado más usado en la actualidad es el UML³, el cual es una notación visual que representa elementos como: los requerimientos del sistema, estructura general y comportamiento del mismo. (19)

Ya que el UML proviene de técnicas orientadas a objetos, se crea con la fuerte intención de que este permita un correcto modelado orientado a objetos. El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático. Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones en: visualizar, especificar, construir y documentar. (19)

El lenguaje UML, es el lenguaje de modelado de *software* más usado en el mundo. Actualmente muchas de las herramientas CASE⁴ y de desarrollo de *software* han adoptado al UML, como lenguaje de modelado. En su creación y desarrollo participaron estudiosos e investigadores de gran prestigio y fue apoyado por la mayoría de las instituciones más importantes del mundo de la Informática. (19)

1.9 Herramienta CASE

El concepto de Herramienta CASE ha sido emitido a lo largo de la historia por diferentes autores. Uno de estos conceptos es el de los autores Kendall & Kendall:

“Las Herramientas son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del *software* en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras”. (20)

Algunos de los **beneficios** obtenidos por la tecnología CASE son: facilidad para la revisión de aplicaciones, soporte para el desarrollo de prototipos de sistema, así como la generación de código. También brinda un soporte interactivo para el proceso de desarrollo de los sistemas informáticos y la generación de documentación técnica. (20)

De la bibliografía consultada sobre las herramientas CASE, se resume que las mismas tienen como **objetivos fundamentales** mejorar la productividad en el desarrollo y mantenimiento de *software* para

³ UML: Lenguaje unificado de modelado.

⁴ CASE: *Computer Aided Software Engineering*, lo que en español significa Ingeniería de *software* asistida por computadora.

aumentar su calidad, mejorar la planificación de un proyecto, así como el tiempo y coste de desarrollo y mantenimiento de sistemas informáticos. Este tipo de herramienta aumenta la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos, ayuda a la reutilización del *software*, portabilidad y estandarización de la documentación y facilita el uso de las distintas metodologías propias de la ingeniería del *software*. (20)

1.9.1 Visual Paradigm-UML

Entre las herramientas CASE más usadas actualmente se encuentra Visual Paradigm. Es una herramienta UML profesional considerada en el mercado mundial como muy completa y fácil de usar, es multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones, la misma soporta el ciclo de vida completo del desarrollo de *software*. El *software* de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Tiene disponibilidad de múltiples versiones para cada necesidad y disponibilidad de integrarse en los principales IDEs. Apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. (21)

Fundamentación de la selección

Se escogió como herramienta CASE Visual Paradigm 8.0, debido a que presenta varias ventajas, tales como: emplea una rápida respuesta con poca memoria utilizando moderadamente los tiempos del procesador, lo que le permite manejar grandes y complicadas estructuras de un proyecto en una forma muy eficiente. Además a diferencia del Rational Rose que es *software* propietario el Visual Paradigm es *software* libre y el equipo de desarrollo posee experiencia en el trabajo de esta herramienta. Por estas razones será utilizada para el diseño del prototipo de interfaz de usuario no funcional y el modelo de datos del sistema.

1.10 Lenguaje de programación

Los ordenadores son una de las herramientas más versátiles que existen en la actualidad. Son capaces de simplificar y realizar tareas, automatizar procesos tediosos, facilitan el intercambio de la información, entre otras funciones. Estas acciones deben ser descritas en lenguajes entendibles para ellos. Para mejorar esta comunicación se han desarrollado una gran variedad de lenguajes de programación para simplificar este proceso.

1.10.1 Lenguaje de programación del lado del servidor

Para realizar la implementación del sistema se utilizó el lenguaje PHP, pues se trata de un tipo de lenguaje de programación de código del lado del servidor, diseñado fundamentalmente para el desarrollo web de contenido dinámico.

Entre los aspectos positivos de PHP se encuentra que su código fuente es invisible al navegador y al cliente, ya que es el propio servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Es también multiplataforma, sencillo y ofrece una gran variedad de funciones para la explicación de bases de datos sin complicaciones. (22)

PHP 5.3

La versión de PHP que se utilizó es la 5.3, porque posee entre sus características que es un lenguaje multiplataforma, consume pocos recursos, fácil de aprender y posee un alto rendimiento.

1.10.2 Lenguaje de programación del lado del cliente

JavaScript es un lenguaje de programación de script⁵ orientado a objetos que se utiliza principalmente para crear páginas web. A pesar de su nombre JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente es una marca registrada de la empresa Sun Microsystems, con la inclusión en su código fuente de la librería o *framework* JQuery, que será utilizado por el equipo de desarrollo para elaborar la propuesta de solución. (23)

JQuery: es un producto con una aceptación por parte de los programadores muy buena y un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Además, es un producto estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del *framework*, todas las ventajas que presenta, sin duda son de agradecer ya que se obtienen de manera gratuita, porque tiene licencia para uso en cualquier tipo de plataforma, personal o comercial. Implementa una serie de clases (de programación orientada a objetos) que permite programar sin preocupaciones del navegador con el que está utilizando el usuario, ya que funcionan de exacta forma en todas las plataformas más habituales. Para la implantación del sistema se utilizará JQuery en su versión 1.9. (24)

HTML5: lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos

⁵ Script: secuencia de comando.

más populares que existen para la construcción de documentos y también de los más fáciles de aprender. (25)

Las hojas de estilo en cascada (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML y por extensión en XHTML. Como *framework* CSS se utilizará Bootstrap para el desarrollo del sistema.

Bootstrap: es un *framework* diseñado para facilitar la creación de sitios web. Es un tipo de técnica de remuestreo de datos que permite resolver problemas relacionados con la estimación de los intervalos de confianza o la prueba de significación estadística. Resulta más accesible y simple de comprender. Para la implementación del sistema se escogió Bootstrap en su versión 3.03. (26)

1.11 Framework de desarrollo

Un *framework* para aplicaciones web es un *software* o conjunto de bibliotecas, que está diseñado para dar soporte al desarrollo de sitios y en general a la construcción de cualquier aplicación web. Entonces un *framework* trata de facilitar aquellas actividades comunes realizadas durante el desarrollo de la aplicación, como por ejemplo: acceso a la base de datos, uso de plantillas, manejo de sesiones, separación de aspectos de programación; además de promover la reutilización de código. (27)

El análisis de la bibliografía consultada permite confirmar que usar un *framework* de desarrollo trae consigo facilidades como son: (27)

Seguridad: Permite identificar los usuarios de la aplicación, y restringir el acceso a funciones basadas en algún criterio definido o mediante listas de control de acceso.

Acceso a Base de Datos: Permite acceder a varios manejadores de bases de datos sin realizar cambios en el código.

Utilización de Plantillas: Aplicando plantillas o temas y el uso de variables para aquellas partes dinámicas donde se insertan los datos extraídos de una base de datos, se puede reducir dramáticamente el número de páginas en un sitio.

El estudio bibliográfico realizado sobre los *framework* de desarrollo muestra que uno de los más usados en la actualidad es Symfony, puesto que separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Automatiza las tareas más comunes que permiten al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP 5 y es compatible con la mayoría de los gestores de bases de datos como: MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. (28)

Fundamentación de la selección

Entre los aspectos que propiciaron la selección de Symfony en su versión 2.3.7 para el desarrollo del presente sistema informático, se encuentran sus características, entre las que se destacan: su desarrollo es muy activo, puesto que incluye mejoras constantes, combinando la flexibilidad con la rapidez de ejecución, tiene una amplia documentación, su reutilización de usuarios es muy activa, debido a que la comunidad de usuarios crece cada día y ofrece un gran soporte de forma gratuita. Además, posee una gran flexibilidad tanto en el diseño global del *framework* como en su sistema de configuración. (29)

1.12 Entorno de desarrollo

Un IDE⁶ es un programa informático compuesto por un conjunto de herramientas de programación. Es un entorno que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un GUI⁷.

- **Editor de código fuente:** Editor de texto que sirve para editar el código fuente de aplicaciones informáticas.
- **Un compilador:** Es un traductor de código fuente, lo traduce a un lenguaje que sea legible para las máquinas.
- **Un depurador:** Es una aplicación que tiene como función probar y eliminar posibles errores en un programa en desarrollo.
- **Constructor de interfaz gráfica:** Herramienta que sirve para crear y diseñar las interfaces con las cuales habrá interacción entre la aplicación y el usuario. (30)

Como parte de los IDE se encuentran: JCreator, Eclipse y el NetBeans por citar algunos ejemplos. Para la realización del sistema se utilizó el NetBeans, que es un IDE de código abierto y una plataforma de aplicaciones que permite a los desarrolladores crear rápidamente la Web, con el uso de la tecnología Java. Dispone de soporte para crear interfaces gráficas de forma visual. Desarrollo de aplicaciones web, así como la creación de aplicaciones compatibles con teléfonos móviles. (29)

A pesar de que el NetBeans está desarrollado fundamentalmente en Java y el lenguaje de programación que será utilizado para la implementación del sistema es PHP, se decide su utilización ya que es de código abierto fácilmente portable en plataformas como Unix, Linux y Windows.

⁶ IDE: Entorno de desarrollo integrado.

⁷ GUI: Constructor de interfaz gráfica.

Fundamentación de la selección

Para el desarrollo de la propuesta de solución se usará el NetBeans. Algunas de las características que propiciaron su selección se muestran a continuación: (31)

- Presenta un rendimiento óptimo en tiempo de ejecución y optimización de recursos.
- NetBeans IDE proporciona diferentes vistas de los datos, de múltiples ventanas del proyecto a herramientas útiles para la creación de las aplicaciones y la gestión de manera eficiente, lo que le permite profundizar en los datos de forma rápida y sencilla.
- Puede ejecutarse en una gran variedad de sistemas operativos, incluyendo Windows y Linux.

La versión 7.3.1, posee un mejor soporte para las tecnologías Java más recientes, una edición de código de forma rápida e inteligente, una administración de proyectos más rápida y eficiente, además de proporcionar un desarrollo rápido de la interfaz de usuario.

1.13 Sistema Gestor de Base de Datos

Un SGBD⁸ es un *software* cuyo objetivo es proporcionar una interfaz entre la base de datos (BD), el usuario y las aplicaciones que la utilizan. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la BD, el control de acceso, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. (32)

Un SGBD debe permitir: (33)

- Definir una BD: especificar tipos, estructuras y restricciones de datos.
- Construir la BD: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipula la BD: realizar consultas, actualizarlas y generar informes.

Una de las características principales características que presentan los SGBD son: (33)

- Abstracción de la información: los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos.
- Independencia: consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven en ella.
- Consistencia: consiste en que todos los datos repetidos se actualicen de forma simultánea.
- Seguridad: los SGBD deben garantizar que la información almacenada en la base de datos se encuentre segura frente a usuarios malintencionados que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información.

⁸ SGBD: Sistema gestor de base de datos.

- **Integridad:** se trata de adoptar todas las medidas necesarias para garantizar la validez de los datos almacenados.

Como SGBD se utilizará MySQL, que es relacional, multihilo y multiusuario. Su popularidad como aplicación web está muy ligada a PHP, que a menudo está en combinación con MySQL.

En las aplicaciones web, MySQL presenta una baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en la lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. (33)

Fundamentación de la selección

La selección de este SGBD en su versión 5.2 está fundamentada principalmente por las ventajas que se muestran a continuación: (34)

- **Velocidad:** es mucho más rápido que otros SGBD.
- **Portabilidad:** se ejecuta en la mayoría de los sistemas operativos y, en la mayoría de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- **Funcionalidad:** dispone de muchas de las funciones que exigen la mayor parte de los desarrolladores profesionales. Así mismo, se desarrolla y actualiza de forma mucho más rápida.
- **Facilidad de uso:** resulta fácil de usar y de administrar. Sus herramientas son potentes y flexibles, sin sacrificar su capacidad de uso.

1.14 Servidor web

Un servidor web es un programa que se ejecuta continuamente en una computadora, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. (17)

Apache es uno de los servidores web más populares en la actualidad. Es un servidor de código abierto y gratuito, por lo que se puede decir que corre sobre cualquier plataforma. Una de sus características es que muestra mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Es configurable, robusto y estable, lo que contribuye a que hoy en día sea el servidor web más utilizado en el mundo.

Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta Apache, optimizando el rendimiento y la rapidez del código. Los módulos de Apache se pueden clasificar en tres categorías:

Módulos Base: Módulo con las funciones básicas de Apache.

Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones.

Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor. (17)

Fundamentación de su selección

La selección de Apache 2.2.22 como servidor web para el desarrollo del sistema, se basa fundamentalmente en sus ventajas, como son: son personalizables, permiten la implementación de los últimos y más nuevos protocolos y corren en gran número de sistemas operativos. Además, hoy en día existen diversos módulos para este tipo servidor, que pueden ser usados cuando sea necesario. El servidor puede ser administrado vía comandos, lo que hace la administración remota muy conveniente. (35)

1.15 Conclusiones parciales

Los métodos científicos y las técnicas de recopilación de datos empleadas en el estudio que se presenta en este capítulo permitieron desarrollar la teoría que sustenta la presente investigación. Los sistemas de gestión estudiados no cumplen las expectativas del presente sistema, sin embargo se tomaron aspectos positivos para que ayuden en el desarrollo del *software*.

Luego de un estudio realizado sobre las metodologías, herramientas y lenguajes de programación existentes para el desarrollo de una herramienta web, se seleccionaron las más adecuadas para el cumplimiento del objetivo general propuesto. Se decide realizar la aplicación sobre la base de la metodología XP como guía del proceso de desarrollo, pues provee un conjunto de principios y prácticas que se ajustan al entorno y condiciones del proyecto.

Capítulo 2: "Propuesta de solución"

2.1 Introducción

Para obtener éxito en el proceso de desarrollo de un *software*, se debe adaptar la metodología al contexto del proyecto, puesto que no existe ninguna que sea universal para hacer frente a esta situación. La metodología seleccionada para el desarrollo del sistema es Programación Extrema, por tal razón todos los artefactos generados se encuentran condicionados al uso de la misma.

En el presente capítulo se detallan las fases de Exploración y Planificación definidas en el ciclo de vida de la metodología XP, de igual manera serán presentados algunos artefactos importantes, tales como: Historias de Usuario (HU), que son la técnica utilizada para especificar las funcionalidades del *software* y tarjetas CRC (Clase-Responsabilidad-Colaborador), que permiten brindar un mayor enfoque orientado a objetos. Además, se determinan los servicios que debe brindar la solución.

2.2 Descripción del sistema propuesto

El sistema propuesto tiene como objetivo permitir la gestión de los planes de trabajo, el control de las actividades contenidas en los mismos y la evaluación de los profesores y especialistas de la Facultad 4. Los usuarios podrán desempeñar 6 roles: usuario, jefe de departamento, vicedecano, asesor, decano y administrador. Inicialmente, al autenticarse con el usuario y contraseña del dominio UCI se tiene asignado el rol usuario.

El sistema permite la creación de planes de trabajo de facultad, de departamentos docentes e individuales, en los que se recogen las actividades planificadas mensualmente. A los planes de trabajo por departamentos docentes e individuales se le pueden incorporar actividades pertenecientes al plan de facultad y si el usuario pertenece a algún departamento puede agregar actividades del plan de trabajo de su departamento docente. El sistema brinda la posibilidad de crear reportes de cumplimiento de los planes de trabajo de facultad y departamentos docentes, además los reportes podrán ser exportados en formato "pdf".

Los planes de trabajos pueden ser exportados en archivos de formato "ics", los cuales podrán ser utilizados en programas como *Microsoft Outlook*, *Mozilla Thunderbird* y en aplicaciones calendario para móviles que utilicen este tipo de archivo. Además brinda la posibilidad de crear nuevos planes de trabajo importando los archivos "ics".

Los profesores y especialistas de la Universidad son evaluados trimestral o anualmente. La aplicación permite al decano evaluar a los vicedecanos, al asesor y a los jefes de departamentos, estos últimos evalúan a los profesores que pertenecen a su departamento. Para otorgar una evaluación anual se debe tener en cuenta el cumplimiento de los parámetros definidos en el plan de resultados del usuario

a evaluar y para asignar una evaluación trimestral se debe valorar el cumplimiento de las actividades contenidas en los planes de trabajo individuales de ese trimestre. El sistema brinda la posibilidad de exportar las evaluaciones en formato “pdf”.

2.3 Patrón arquitectónico

Christopher Alexander, un reconocido arquitecto inglés dice que “cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema, de tal modo que se pueda aplicar esta solución un millón de veces, sin hacer lo mismo dos veces”. (36)

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC⁹, que está formado por tres niveles:

- El Modelo: representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista: transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador: se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. (37)

2.4 Objetivos del sistema

Un objetivo consiste en plantear una meta o un propósito a alcanzar y que de acuerdo al ámbito donde sea utilizado, tiene cierto nivel de complejidad. Se plantean de manera abstracta en un principio pero luego, pueden o no concretarse en la realidad, si el proceso de realización ha sido exitoso o no.

Los Objetivos de Investigación son la guía del estudio; expresan de manera muy sintética qué se pretende con la investigación y guardan relación directa con las actividades, comenzando con un verbo en infinitivo o señalando una intención de cambio o afectación de algún aspecto de interés en particular. (38)

Una de las primeras tareas que se debe llevar a cabo para la realización de un sistema informático con éxito es la definición de los objetivos. Con la implementación de la propuesta de solución, se persiguen los objetivos siguientes:

- Contribuir a la gestión y almacenamiento de la información de los profesores y especialistas de la Facultad 4.

⁹ MVC: Modelo- Vista- Controlador.

- Contribuir a la gestión de los planes de trabajo de facultad, los departamentos docentes y los planes de trabajo individuales de profesores y especialistas.
- Realizar reportes estadísticos sobre el cumplimiento de los planes de trabajo de facultad y departamentos docentes.
- Dar seguimiento a las evaluaciones de los profesores por parte de los jefes de departamento y el decano.

2.5 Usuarios relacionados con el sistema

En la tabla siguiente se muestran los usuarios que tendrán acceso al sistema y sus responsabilidades.

Tabla 1: Usuarios del sistema

Usuario	Descripción
Administrador	Es la persona autorizada para la gestión del sistema, encargado de modificar y eliminar usuarios del sistema. Puede crear, modificar y eliminar departamentos, parámetros generales y las áreas de resultados claves.
Decano	Es el usuario con permisos para agregar actividades contenidas dentro del plan de trabajo de la facultad. Puede realizar un control del cumplimiento de las actividades del vicedecano y de los jefes de departamento, controlar sus planes de resultados y otorgarles una evaluación anual y trimestral. Controla el cumplimiento de las actividades del plan de la facultad agregadas en los planes de trabajo de los departamentos docentes.
Vicedecano	Es el usuario con permisos para agregar, editar y eliminar actividades contenidas dentro del plan de trabajo individual.
Asesor	Es el usuario con permisos para crear y eliminar el plan de trabajo de la facultad, así como agregar, editar y eliminar actividades del mismo. Tiene la posibilidad de crear, modificar y eliminar el reporte del cumplimiento del plan de trabajo de la facultad. Además, crea y elimina su evaluación anual.
Jefe de Departamento	Es la persona autorizada para gestionar los planes de trabajo por departamentos docentes y las actividades contenidas dentro del mismo. Realiza el control del cumplimiento de las actividades de sus subordinados, crea el reporte del cumplimiento de los mismos, controla el cumplimiento de los parámetros del plan de resultados y les otorga una evaluación anual y trimestral. Gestiona su plan de resultados y evaluación anual.

Usuario	Gestiona su plan individual, evaluación anual y plan de resultados.
---------	---

2.6 Exploración

La exploración es la primera fase que plantea la metodología XP, donde los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema. (39)

2.6.1 Historias de Usuario

Son la técnica utilizada para especificar las funcionalidades del *software*. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, ya sean funcionalidades del sistema o restricciones funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada una es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración. (40)

Respecto a la información contenida en la HU, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos solo se propone utilizar un nombre y una descripción. (39)

Para la elaboración de las HU del presente trabajo se tomarán los parámetros siguientes:

- **Número:** Posee el número asignado a la HU (número consecutivo a partir del 1).
- **Nombre HU:** Atributo que identifica la HU.
- **Usuario:** ¿Quién ejecuta la HU?
- **Prioridad de Negocio:** Define la relevancia o impacto de la HU para el negocio de acuerdo a las necesidades del usuario.
- **Riesgo de desarrollo:** Evidencia el nivel de riesgo en caso de no realizarse la HU. (Alto / Medio / Bajo).
- **Puntos estimados:** Permite estimar la duración de la implementación, representando con 1 una semana de trabajo. Por lo que cuando el valor de dicho atributo es de 0.5 equivale a 2 días y medio de trabajo.
- **Iteración asignada:** Precisa la iteración en la que será desarrollada la HU.
- **Descripción:** Explica en qué consiste la HU, teniendo en cuenta las acciones realizadas por el usuario y la respuesta brindada por el sistema.
- **Observaciones:** Brinda información extra que se estime agregar para hacer más comprensible la HU.

Tabla 2: HU Editar perfil de usuario

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Editar perfil de usuario
Usuario: Administrador	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 0.5	Iteración asignada: 1
Descripción: El sistema debe permitir editar el perfil del usuario que se va a registrar. Se pueden modificar datos obtenidos del LDAP como son el nombre y los apellidos y añadir otros que son necesarios para registrarse, tales como: carnet de identidad, número de solapín, departamento docente al que pertenece, militancia (si es militante o no), categoría docente, categoría científica y residencia (si reside o no en la universidad).	
Observaciones: Da cumplimiento a la funcionalidad F1 "Editar perfil de usuario".	

Tabla 3: HU Gestionar departamento

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Gestionar departamento
Usuario: Administrador	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir crear, modificar y eliminar un departamento. Para crear un departamento solo se necesita introducir el nombre, que además puede ser modificado.	
Observaciones: Para que se muestre la lista con los departamentos, así como para eliminar uno, debe existir al menos un departamento previamente creado en el sistema. Da cumplimiento a la funcionalidad F2 "Gestionar departamento".	

Tabla 4: HU Gestionar actividad

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Gestionar actividad

Usuario: Todos	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
<p>Descripción: El sistema debe permitir adicionar, mostrar, editar y eliminar una actividad. Para adicionar una actividad a un plan de trabajo de la facultad o a un plan de trabajo por departamento docente se requiere de los datos siguientes: fecha (se selecciona la fecha y hora de la actividad), descripción, lugar, prioridad, quién la dirige y quién participa, además de una valoración de la misma. Si la actividad pertenece a un plan de trabajo individual se requieren los siguientes datos: fecha (se selecciona la fecha y hora de la actividad), descripción, lugar y prioridad. Estos datos también pueden ser editados.</p>	
<p>Observaciones: Para mostrar, editar y eliminar una actividad debe existir como mínimo una en el sistema. Da cumplimiento a la funcionalidad F3 "Gestionar actividad".</p>	

Tabla 5: HU Buscar profesor

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Buscar profesor
Usuario: Todos	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
<p>Descripción: El sistema debe permitir buscar los datos de un profesor determinado a partir de diferentes criterios de búsqueda, como son: nombre y apellidos, número de solapín, carnet de identidad, departamento docente al que pertenece, militancia (se selecciona UJC o PCC), categoría docente, categoría científica y residencia (si el profesor es interno o externo).</p>	
<p>Observaciones: Da cumplimiento a la funcionalidad F4 "Buscar profesor".</p>	

Tabla 6: HU Gestionar plan de trabajo de la facultad

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Gestionar plan de trabajo de la facultad
Usuario: Asesor, vicedecano y decano	

Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Descripción: El sistema debe permitir adicionar, mostrar y eliminar un plan de trabajo de la facultad. Para adicionar un plan de trabajo se selecciona la fecha en que se desea crear (mes y año), el plan puede ser mostrado junto a las actividades que lo componen y además puede ser eliminado.	
Observaciones: Para mostrar una lista con los planes de trabajo, así como para poder realizar la opción de eliminar, deben existir planes de trabajo de este tipo en el sistema. Da cumplimiento a la funcionalidad F5 "Gestionar plan de trabajo de la facultad".	

2.7 Planificación

En esta fase de la metodología XP, el cliente establece la prioridad de cada HU y los programadores realizan una estimación del esfuerzo necesario en cada una de ellas. (39)

Además, se realiza la lista de reserva del producto con las funcionalidades y las restricciones funcionales, aunque estos no forman parte de los artefactos que genera esta metodología, una descripción detallada podría facilitar, conjuntamente con las HU, el desarrollo del sistema. Finalmente se planifica el tiempo de duración del desarrollo mediante un plan de iteraciones. (41) (42)

2.7.1 Lista de Reserva del Producto

A través de las LRP¹⁰ se definen y priorizan las funcionalidades que tendrá el sistema, así como las restricciones funcionales del *software*.

2.7.1.1 Funcionalidades del sistema

Según lo expresado por Sommerville en el año 2005, las funcionalidades de un sistema "son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, también pueden declarar explícitamente lo que el sistema no debe hacer." (43)

Las funcionalidades a implementar se muestran en la tabla presentada a continuación, que contiene los siguientes parámetros:

- **Código:** Números consecutivos a partir del 1 que identifica las funcionalidades.
- **Descripción:** Contiene la descripción de las funcionalidades.
- **Prioridad:** Contiene los niveles de prioridad de las funcionalidades.

¹⁰ LPR: Lista de Reserva del Producto.

Tabla 7: Funcionalidades del sistema

Código	Descripción	Prioridad
F.1	Editar perfil de usuario.	Alta
F.2	Gestionar departamento. F2.1 Crear departamento. F2.2 Modificar departamento. F2.3 Eliminar departamento.	Alta
F.3	Gestionar actividad. F3.1 Adicionar actividad. F3.2 Mostrar actividad F3.2 Editar actividad. F3.3 Eliminar actividad.	Alta
F.4	Buscar profesor.	Alta
F.5	Gestionar plan de trabajo de la facultad. F5.1 Adicionar plan de trabajo de la facultad. F5.2 Mostrar plan de trabajo de la facultad. F5.3 Eliminar plan de trabajo de la facultad.	Alta
F.6	Gestionar plan de trabajo por departamento docente. F6.1 Adicionar plan de trabajo por departamento docente. F6.2 Mostrar plan de trabajo por departamento docente. F6.3 Eliminar plan de trabajo por departamento docente F6.4 Agregar actividad del plan de trabajo de la facultad en el plan de trabajo por departamento docente. F6.5 Quitar actividad del plan de trabajo de la facultad del plan de trabajo por departamento docente.	Alta
F.7	Gestionar plan de trabajo individual (profesor y especialista). F7.1 Adicionar plan de trabajo por departamento docente.	Alta

	<p>F7.2 Mostrar plan de trabajo por departamento docente.</p> <p>F7.3 Eliminar plan de trabajo por departamento docente</p> <p>F7.4 Agregar actividad del plan de trabajo de la facultad y del plan de trabajo por departamento docente en el plan de trabajo individual.</p> <p>F7.5 Quitar actividad del plan de trabajo de la facultad y del plan de trabajo por departamento docente del plan de trabajo individual.</p>	
F.8	Realizar control del cumplimiento de las actividades de los usuarios por departamento docente.	Alta
F.9	<p>Gestionar plan de resultados.</p> <p>F9.1: Crear plan de resultados.</p> <p>F9.2: Eliminar plan de resultados.</p> <p>F9.3: Mostrar plan de resultados.</p>	Media
F.10	<p>Gestionar área.</p> <p>F10.1 Crear área.</p> <p>F10.2 Editar área.</p> <p>F10.3 Eliminar área.</p>	Media
F.11	<p>Gestionar parámetros generales.</p> <p>F11.1 Crear parámetro general.</p> <p>F11.2 Editar parámetro general.</p> <p>F11.3 Eliminar parámetro general.</p>	Media
F.12	<p>Gestionar parámetro.</p> <p>F12.1 Crear parámetro.</p> <p>F12.2 Editar parámetro.</p> <p>F12.3 Eliminar parámetro.</p>	Media
F.13	<p>Asignar evaluación.</p> <p>F13.1 Asignar evaluación anual.</p> <p>F13.2 Asignar evaluación trimestral.</p>	Media
F.14	Importar plan de trabajo en formato iCalendar.	Baja

F.15	Exportar plan de trabajo en formato iCalendar.	Baja
F.16	Gestionar reportes del cumplimiento de los planes de trabajo. F16.1 Crear reportes del cumplimiento de los planes de trabajo. F16.2 Modificar reportes del cumplimiento de los planes de trabajo. F16.3 Mostrar reportes del cumplimiento de los planes de trabajo. F16.4 Eliminar reportes del cumplimiento de los planes de trabajo.	Baja

2.6.1.2 Restricciones funcionales

Sommerville en el año 2005 definió que las restricciones funcionales de un sistema “son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares.” (43)

Las restricciones funcionales del sistema se muestran en la tabla presentada a continuación, que contiene los siguientes parámetros:

Código: Números consecutivos a partir del 1 que identifica las restricciones funcionales.

Clasificación: Contiene la clasificación de las restricciones funcionales y la descripción de las mismas.

Tabla 8: Restricciones funcionales del sistema

Código	Clasificación
	Usabilidad
1	La interfaz de usuario de la aplicación debe ser sencilla, intuitiva, que le permita al usuario que interactúa con el sistema realizar cualquier labor en el menor tiempo posible y con pocas acciones.
2	En el momento que ocurra un error con los datos de entrada a la aplicación le será informado al usuario de manera detallada.
	Eficiencia
3	Los tiempos de respuestas en el procesamiento de datos y solicitud de estos en el sistema tienen que ser breves. En las ocasiones en que el sistema esté realizando mayor carga de trabajo por encontrarse procesando y almacenando datos, estos tiempos no deben exceder de los 10 segundos.

	Seguridad
4	Al sistema se accederá a través de la autenticación convencional: usuario y contraseña.
5	Cada usuario debe tener solo los permisos necesarios para realizar las operaciones que le sean permitidas.
	Portabilidad
6	El sistema podrá utilizarse en cualquier sistema operativo.
	Requerimientos de <i>hardware</i>
7	<p>Para las computadoras del cliente:</p> <p>Se requiere tengan tarjeta de red.</p> <p>Se requiere tengan al menos 128 MB de memoria RAM.</p> <p>Procesador Pentium 500 MHz como mínimo.</p>
8	<p>Para el servidor:</p> <p>Se requiere tarjeta de red.</p> <p>Se requiere tenga al menos 1GB de RAM.</p> <p>Procesador Pentium 1.3 GHz como mínimo.</p>
	Software
9	<p>La solución propuesta debe poderse visualizar en los navegadores:</p> <p>Mozilla Firefox 5.0 o superior</p> <p>Internet Explorer 9 o superior</p> <p>Google Chrome 10.0 o superior</p> <p>Opera 11.0 o superior</p> <p>Safari 5.0 o superior.</p>

2.7.2 Estimación de esfuerzos por HU

Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. (39)

Tabla 9: Estimación de esfuerzo por HU

Código	Descripción	Estimación (semanas)
F.1	Editar perfil de usuario.	0.5
F.2	Gestionar departamento.	1
F.3	Gestionar actividad.	1
F.4	Buscar profesor.	0.5
F.5	Gestionar plan de trabajo de la facultad.	1
F.6	Gestionar plan de trabajo por departamento docente.	1
F.7	Gestionar plan de trabajo individual (profesor y especialista).	1
F.8	Realizar control del cumplimiento de las actividades de los usuarios por departamento docente.	1
F.9	Gestionar plan de resultados.	1
F.10	Gestionar área.	0.5
F.11	Gestionar parámetros generales.	0.5
F.12	Gestionar parámetro.	0.5
F.13	Asignar evaluación anual.	0.5
F.14	Asignar evaluación trimestral.	0.5
F.15	Importar plan de trabajo en formato iCalendar	1

F.16	Exportar plan de trabajo en formato iCalendar	1
F.17	Gestionar reportes del cumplimiento de los planes de trabajo	0.5

2.8 Plan de Iteraciones

La creación de un sistema, según la metodología seleccionada se divide en iteraciones, cada una con una duración ideal de 1 a 3 semanas. En el plan de iteraciones, las HU seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. (44) Al finalizar las iteraciones la aplicación contará con todas las funcionalidades implementadas para darle cumplimiento a los objetivos propuestos.

Iteración 1: En esta iteración se entregarán las funcionalidades que tienen prioridad alta para el cliente, correspondiendo a las HU:

- HU1: Editar perfil de usuario.
- HU2: Gestionar Departamento.
- HU3: Gestionar Actividad.

Iteración 2: En esta iteración se realizarán las restantes funcionalidades que son importantes para el cliente, siendo estas:

- HU4: Buscar profesor.
- HU5: Gestionar plan de trabajo de la facultad.
- HU6: Gestionar plan de trabajo por departamento docente.
- HU7: Gestionar plan de trabajo individual (profesor y especialista).

Iteración 3: En la tercera iteración se recogerán las funcionalidades que son de prioridad media para el cliente:

- HU8: Realizar control del cumplimiento de las actividades de los usuarios por departamento docente.
- HU9: Gestionar plan de resultados.
- HU10: Gestionar área.
- HU11: Gestionar parámetros generales.
- HU12: Gestionar parámetro.
- HU13: Asignar evaluación anual.
- HU14: Asignar evaluación trimestral.

Iteración 4: En la cuarta iteración se recogerán las funcionalidades que son de prioridad baja para el cliente, pero no menos importante para los desarrolladores, correspondiendo a las HU siguientes:

- HU15: Importar plan de trabajo en formato iCalendar.
- HU16: Exportar plan de trabajo en formato iCalendar.
- HU17: Gestionar reportes del cumplimiento de los planes de trabajo.

2.9 Plan de duración de las iteraciones

El plan de duración de las iteraciones se realiza luego de tener el estimado en días que demora implementar cada HU. Se tendrá en cuenta además la prioridad que el cliente le asigna a cada historia y el nivel de complejidad que estas poseen.

La tabla propuesta para el plan de duración de las iteraciones cuenta con los siguientes parámetros:

- **Iteración:** Contiene el identificador de la iteración que se va a realizar.
- **Orden de las HU:** Contiene las HU que se van a realizar en la iteración en el mismo orden en que deben ser implementadas.
- **Duración total:** Cantidad de semanas que durará realizar la iteración, la que depende del tiempo estimado establecido en las HU propuestas.

Tabla 10: Plan de duración de las iteraciones

Iteración	Orden de las HU	Duración total
1	Editar perfil de usuario.	2.5 semanas
	Gestionar departamento.	
	Gestionar Actividad.	
2	Buscar profesor.	3.5 semanas
	Gestionar plan de trabajo de la facultad.	
	Gestionar plan de trabajo por departamento docente.	
	Gestionar plan de trabajo individual (profesor y especialista).	
3	Realizar control del cumplimiento de las actividades de los usuarios por departamento docente.	

	Gestionar plan de resultados.	4.5 semana
	Gestionar área.	
	Gestionar parámetros generales.	
	Gestionar parámetro.	
	Asignar evaluación anual.	
	Asignar evaluación trimestral.	
4	Importar plan de trabajo en formato iCalendar.	2.5 semanas
	Exportar plan de trabajo en formato iCalendar.	
	Gestionar reportes del cumplimiento de los planes de trabajo.	

2.10 Plan de entrega

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto. Se confecciona en base a las estimaciones de tiempo de desarrollo realizadas por los desarrolladores.

Tabla 11: Plan de entrega de las HU

Historia de Usuario	Primera iteración	Segunda iteración	Tercera iteración	Cuarta iteración
Editar datos de usuario.	V 1.0	Finalizado	Finalizado	Finalizado
Gestionar departamento.	V 1.0	Finalizado	Finalizado	Finalizado
Gestionar Actividad.	V 1.0	Finalizado	Finalizado	Finalizado
Buscar profesor.	-	V 1.0	Finalizado	Finalizado
Gestionar plan de trabajo de la facultad.	-	V 1.0	Finalizado	Finalizado
Gestionar plan de trabajo por departamento.	-	V 1.0	Finalizado	Finalizado
Gestionar plan de trabajo individual	-	V 1.0	Finalizado	Finalizado

(profesor y especialista).				
Realizar control del cumplimiento de las actividades de los usuarios por departamento docente.	-	-	V 1.0	Finalizado
Gestionar plan de resultados.	-	-	V 1.0	Finalizado
Gestionar área.	-	-	V 1.0	Finalizado
Gestionar parámetros generales.	-	-	V 1.0	Finalizado
Gestionar parámetro.	-	-	V 1.0	Finalizado
Asignar evaluación anual.	-	-	V 1.0	Finalizado
Asignar evaluación trimestral.	-	-	V 1.0	Finalizado
Importar plan de trabajo en formato iCalendar.	-	-	-	V 1.0
Exportar plan de trabajo en formato iCalendar.	-	-	-	V 1.0
Gestionar reportes del cumplimiento de los planes de trabajo.	-	-	-	V 1.0

2.11 Prototipo de interfaz de usuario no funcional

Un prototipo "...es un modelo del comportamiento del sistema que puede ser usado para entenderlo completamente o ciertos aspectos de él y así clarificar los requerimientos... Un prototipo es una representación de un sistema, aunque no es un sistema completo, posee las características del sistema final o parte de ellas". (45)

El prototipo muestra una ventana con un encabezado azul que dice "Inicio de sesión". Debajo del encabezado, hay un campo de texto etiquetado "Usuario", un campo de texto etiquetado "Contraseña" y un botón etiquetado "Aceptar".

Ilustración 3: Prototipo de interfaz de usuario no funcional

2.12 Tarjetas CRC

En la metodología XP, no se requiere de la representación del sistema mediante diagramas de clases utilizando notación UML, en su lugar usa tarjetas CRC (Clase-Responsabilidad-Colaboración). Una tarjeta CRC no es más que una ficha de papel o cartón que representa una entidad del sistema.

Establece 3 dimensiones, las cuales identifican el rol de un objeto en análisis y/o diseño: nombre de la clase, responsabilidades y colaboraciones. (46)

- **Una clase** representa a una colección de objetos similares.
- **Una responsabilidad** es aquello que la clase sabe o hace. Una clase puede cambiar el valor de lo que sabe pero no puede cambiar el valor de lo que saben otras clases. Algunas veces una clase tiene una responsabilidad que cumplir pero no tiene toda la información para hacerlo. Esto hace que tenga que interactuar con otras clases para obtener su colaboración.
- **Las colaboraciones** toman una de dos formas: un pedido de información o un pedido de que se realice una operación.

Las principales **características** de las tarjetas CRC son:

- Identificación de clases y asociaciones que participan del diseño del sistema.
- Obtención de las responsabilidades que debe cumplir cada clase.
- Establecimiento de cómo una clase colabora con otras clases para cumplir con sus responsabilidades.

Tabla 12: Tarjeta CRC UsuarioController

Nombre: UsuarioController.	
Responsabilidad	Colaboraciones
updateAction	PlanIndividual
	PlanDepartamento
editAction	
deleteAction	
asignarDepartamentoAction	

Tabla 13: Tarjeta CRC ProfesorController

Nombre: ProfesorController	
Responsabilidad	Colaboraciones
buscarAction	Usuario

Tabla 14: Tarjeta CRC DepartamentoController

Nombre: DepartamentoController	
Responsabilidad	Colaboraciones
createAction	
editAction	
deleteAction	Reporte
	PlanDepartamento
indexAction	
newAction	
updateAction	
usuariosDepAction	Usuario
quitarDepAction	Usuario
	PlanIndividual
	Actividad

Tabla 15: Tarjeta CRC PlanIndividualController

Nombre: PlanIndividualController	
Responsabilidad	Colaboraciones
mostrarAction	Actividad
actualizarAction	Actividad
agregarAction	Actividad
eliminarAction	Actividad
quitarAction	Actividad
editAction	Actividad

actividadesFacAction	Actividad
actividadesDepAction	Actividad
listarAction	Actividad
adicionarAction	Actividad
exportarAction	Actividad

Tabla 16: Tarjeta CRC ParametroController

Nombre: ParametroController	
Responsabilidad	Colaboraciones
listaAction	PlanResultado
createAction	PlanResultado
indexAction	
newAction	
editAction	
updateAction	
deleteAction	
controlarParametroAction	
actualizarAction	

2.13 Conclusiones parciales

Al finalizar el presente capítulo se concluye que la metodología Programación Extrema, seleccionada para el desarrollo de las funcionalidades del sistema de gestión de información, el control y la evaluación de los profesores y especialistas de la Facultad 4, posibilitó realizar la propuesta del sistema detallándola con la ayuda de los artefactos propuestos por la misma. Después de un estudio de la audiencia a la que va dirigida la solución, se delimitaron 6 tipos de roles. Se definieron las funcionalidades del sistema, para las cuales se elaboró el plan de iteraciones, el cual cuenta con 4 iteraciones para un total de 17 HU, que describen los aspectos principales a tener en cuenta para el desarrollo de la solución. Asociado a estas HU se construyó el plan de entregas, se determinó un cronograma que especifica las entregas que deben hacerse, del cual se estimaron un total de 4 entregas. Se elaboró el prototipo de interfaz no funcional de la solución y las tarjetas CRC que representan las funcionalidades a implementar definiendo las responsabilidades y los colaboradores.

Capítulo 3: Implementación y pruebas

3.1 Introducción

Este capítulo está enmarcado en las fases de implementación y prueba de la metodología XP. En la primera fase de las antes mencionadas, la programación que es considerada uno de los pilares fundamentales dentro de XP es guiada por parejas, lo cual es una de las buenas prácticas que propone la metodología seleccionada. Esta se realiza a partir del análisis y diseño realizado durante las iteraciones hasta la entrega final del producto. Durante este período el cliente mantiene una constante comunicación con el equipo de desarrollo, no solo para la aclaración de posibles dudas sino para formar parte de este.

En la fase de prueba, el cliente realiza las pruebas funcionales para verificar que el funcionamiento del sistema es correcto y de ocurrir errores se solucionarán en iteraciones posteriores.

Durante el desarrollo del capítulo, se abordarán elementos esenciales relacionados con la implementación del sistema, estándares de codificación empleados y patrones de diseño aplicados durante su desarrollo. Además, se realizarán las pruebas de *software* con el objetivo de lograr una aplicación que cumpla con las funcionalidades previamente definidas.

3.2 Módulos del sistema

En Symfony los módulos son equivalentes a los *bundles*, que no son más que un conjunto estructurado de archivos que se encuentran en un directorio. Los *bundles* son la parte más importante de Symfony2. Permiten utilizar funcionalidades construidas por otras personas o las propias del programador para distribuirlas y reutilizarlas en otros proyectos.

A continuación se especifican los *bundles* usados en la solución y sus funcionalidades:

InformacionBundle: Es donde se gestiona la información de los profesores.

UsuarioBundle: Se encarga de la gestión de los usuarios, los departamentos y los archivos ics.

EvaluacionBundle: Gestiona las evaluaciones de los usuarios y los planes de resultados.

PlanBundle: Gestiona los planes de trabajo, las actividades y los reportes de cumplimiento.

BackendBundle: Se encarga de la administración del sistema.

3.3 Modelo de datos

El diseño de la base de datos es algo esencial para el buen funcionamiento de cualquier *software*. Ayuda a comprender mejor la estructura del sistema y sus funcionalidades. La estructura de los datos en la base de datos es modelada utilizando la herramienta CASE DBDesigner 4, reflejándose así los

conceptos, relaciones y reglas del dominio estudiado, con el objetivo de darle respuesta al problema afrontado por el equipo.

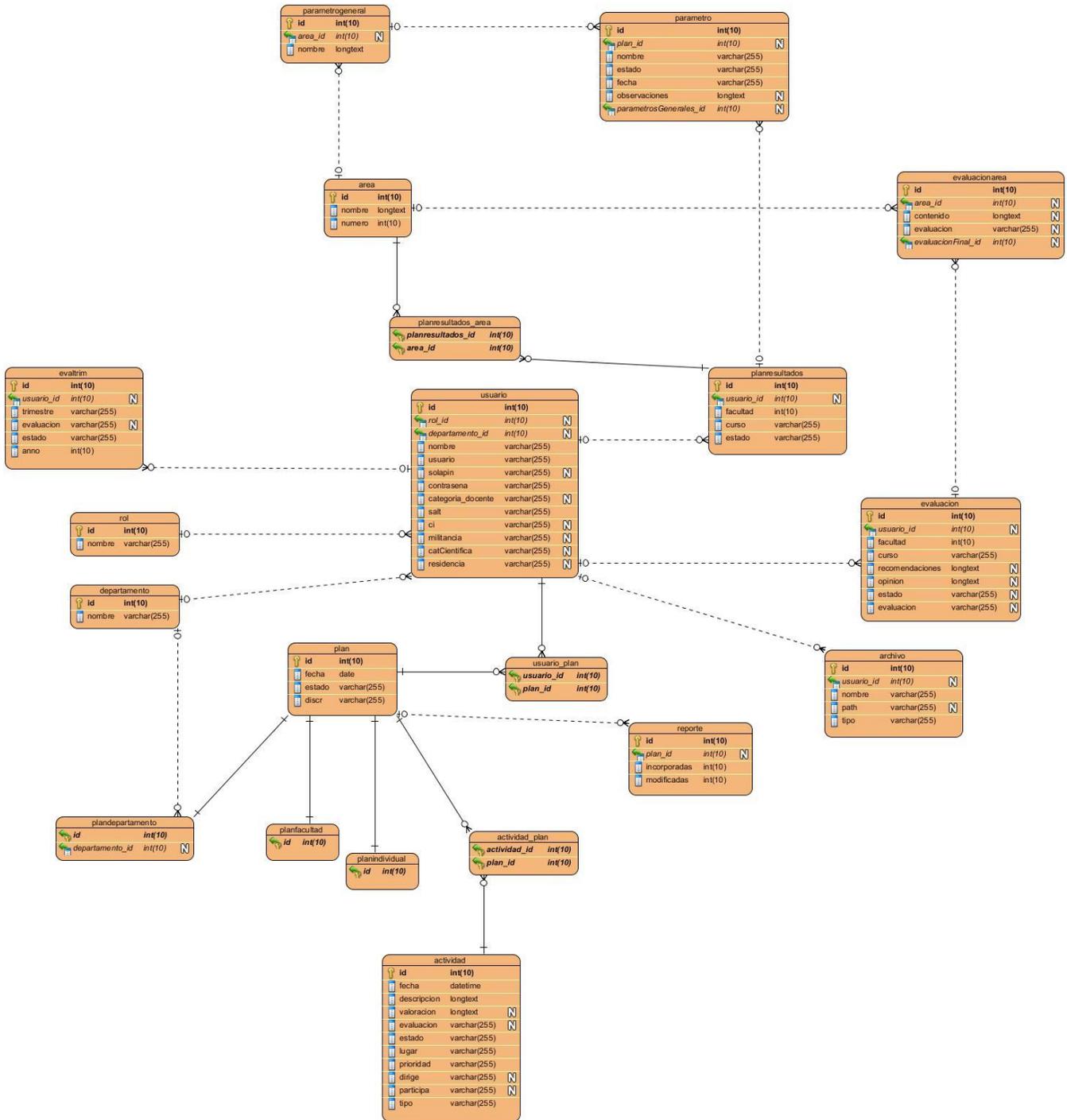


Ilustración 4: Estructura de las tablas en la base de datos

3.4 Implementación

La metodología XP propone que los proyectos sean desarrollados por pares de programadores, esto propicia que se reduzcan los errores y se logren mejores diseños. Usando esta práctica el producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales. (44)

Adicionalmente, la programación en pares tiene las siguientes ventajas: (44)

- La mayoría de los errores se descubren en el momento en que se codifican, ya que el código es permanentemente revisado por dos personas.
- La cantidad de defectos encontrados en las pruebas es estadísticamente menor.
- Los diseños son mejores y el código más corto.
- El equipo resuelve problemas de forma más rápida.
- Las personas aprenden significativamente más acerca del sistema y del desarrollo de *software*.
- El proyecto termina con más personas que conocen los detalles de cada parte del código.

En esta fase, las HU serán implementadas de acuerdo a la planificación realizada en las iteraciones. Por esta razón se procede a verificar el plan realizado para comenzar la implementación.

Para la elaboración del presente trabajo fueron definidas 4 iteraciones que serán detalladas a continuación:

3.4.1 Iteración 1

En la primera iteración se implementaron las HU que son de prioridad alta para el cliente:

Tabla 17: Funcionalidades abordadas en la primera iteración

Nombre de las HU	Tiempo de implementación (semanas)	
	Estimación	Real
Editar perfil de usuario.	0.5	0.2
Gestionar departamento.	1	1.2
Gestionar actividad.	1	1.2

3.4.2 Iteración 2

En la segunda iteración se implementaron las restantes HU que son importantes para el cliente:

Tabla 18: Funcionalidades abordadas en la segunda iteración

Nombre de las HU	Tiempo de implementación (semanas)	
	Estimación	Real
Buscar profesor.	0.5	0.4
Gestionar plan de trabajo de la facultad.	1	1.4
Gestionar plan de trabajo por departamento docente.	1	2.4
Gestionar plan de trabajo individual (profesor y especialista)	1	2.2

3.4.3 Iteración 3

En la tercera iteración se implementaron las HU que son de prioridad media para el cliente:

Tabla 19: Funcionalidades abordadas en la tercera iteración

Nombre de las HU	Tiempo de implementación (semanas)	
	Estimación	Real
Realizar control del cumplimiento de las actividades de los usuarios por departamento docente.	0.5	0.4
Gestionar plan de resultado.	1	0.6
Gestionar área.	0.5	0.8
Gestionar parámetros generales.	0.5	0.8
Gestionar parámetros.	0.5	0.8
Asignar evaluación.	0.5	0.8
Asignar evaluación trimestral.	0.5	0.8

3.4.4 Iteración 4

En la cuarta iteración se implementaron las HU que son de prioridad baja para el cliente:

Tabla 20: Funcionalidades abordadas en la cuarta iteración

Nombre de las HU	Tiempo de implementación (semanas)	
	Estimación	Real
Importar plan de trabajo en formato iCalendar.	1	0.4
Exportar plan de trabajo en formato iCalendar.	1	0.2
Gestionar reportes del cumplimiento de los planes de trabajo.	0.5	0.8

3.5 Tareas de ingeniería

Según la metodología XP, dentro de la etapa de planificación de la iteración, el cliente elige el conjunto de historias de usuario de mayor valor para que sean implementadas en la iteración planeada. A continuación son divididas en tareas más específicas denominadas tareas de ingeniería o tareas de programación (se plasman en tarjetas de papel donde se describe qué se debe realizar y son muy dinámicas y flexibles, pueden ser cambiadas por otras más generales o más específicas, agregarse nuevas o modificarse según las necesidades existentes, cada una de estas tareas podrá ser comprobada a través de los casos de prueba). Luego, cada programador elige las tareas que desea implementar, las analiza en mayor detalle y realiza una estimación de su tiempo de desarrollo. Finalmente el cliente ordena, según sus necesidades, las historias estimadas, dejando para iteraciones posteriores las que sobrepasan la capacidad productiva de la iteración. (47)

Para la elaboración de las Tareas de ingeniería (TI) del presente trabajo, se tomarán los parámetros siguientes:

- **Número tarea:** Número consecutivo a partir del 1.
- **Número historia:** Identifica la HU.
- **Nombre tarea:** Identifica la TI.
- **Tipo de tarea:** Desarrollo/ Corrección/ Mejora/ Otra.
- **Puntos estimados:** Permite estimar la duración de la implementación, donde un 1 equivale a una semana (0.2 equivale a 1 día).
- **Fecha inicio:** Comienzo de la tarea.
- **Fecha fin:** Fin de la tarea.
- **Programador responsable:** Persona encargada de la realización de la TI.

- **Descripción:** Explica en qué consiste la tarea de ingeniería.

Para la **primera iteración** se definieron las siguientes TI:

HU.1: Editar perfil de usuario

- TI.1: Editar datos de usuario.

HU.2: Gestionar departamento.

- TI.2: Crear departamento docente.
- TI.3: Mostrar departamento docente.
- TI.4: Eliminar departamento docente.

HU.3: Gestionar actividad.

- TI.5: Adicionar actividad.
- TI.6: Mostrar actividad.
- TI.7: Editar actividad.
- TI.8: Eliminar actividad.

Tabla 21: TI Editar datos de usuario

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre de la tarea: Editar datos de usuario.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio:20/03/2014	Fecha fin:21/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad editar datos de usuario haciendo uso de un CRUD ¹¹ (Gestionar de Symfony 2) de la entidad Usuario.	

En el Anexo III se muestran las restantes TI definidas para esta iteración.

Para la **segunda iteración** se definieron las siguientes TI:

HU4: Buscar profesor.

- TI.8 Buscar información de un profesor.

HU5. Gestionar plan de trabajo de la facultad.

- TI.10: Adicionar plan de trabajo de la facultad.

¹¹ CRUD: *Create, Read, Update, and Delete*, en español Crear, Mostrar, Modificar y Eliminar

- TI.11: Mostrar plan de trabajo de la facultad.
- TI.12: Eliminar plan de trabajo de la facultad.

HU6. Gestionar plan de trabajo por departamento docente.

- TI.13: Adicionar plan de trabajo por departamento docente.
- TI.14: Mostrar plan de trabajo por departamento docente.
- TI.15: Eliminar plan de trabajo por departamento docente.
- TI.16: Agregar actividad del plan de trabajo de la facultad en el plan de trabajo por departamento docente.
- TI.17: Quitar actividad del plan de trabajo de la facultad del plan de trabajo por departamento docente.

HU7. Gestionar plan de trabajo individual (profesor y especialista).

- TI.18: Adicionar plan de trabajo por departamento docente.
- TI.19: Mostrar plan de trabajo por departamento docente.
- TI.20: Eliminar plan de trabajo por departamento docente.
- TI.21: Agregar actividad del plan de trabajo de la facultad y del plan de trabajo por departamento docente en el plan de trabajo individual.
- TI.22: Quitar actividad del plan de trabajo de la facultad y del plan de trabajo por departamento docente del plan de trabajo individual.

Tabla 22: Tarea de ingeniería Buscar información de un profesor

Tarea	
Número de tarea: 8	Número de HU: 4
Nombre de la tarea: Buscar información de un profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 03/02/2014	Fecha fin: 05/02/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad que permite buscar información de los profesores haciendo uso de la tecnología Ajax para realizar la búsqueda y el paginado de los datos.	

En el Anexo IV se muestran las restantes TI definidas para esta iteración.

Para la **tercera iteración** se definieron las siguientes TI:

HU8. Realizar control del cumplimiento de las actividades de los usuarios por departamento docente.

- TI.23: Realizar un control del cumplimiento de las actividades.

HU9. Gestionar plan de resultados.

- TI.24: Crear plan de resultados.
- TI.25: Eliminar plan de resultados.
- TI.26: Mostrar plan de resultados.

HU10. Gestionar área.

- TI.27: Crear área.
- TI.28 Listar área.
- TI.29 Modificar área.
- TI.30 Eliminar área.

HU11. Gestionar parámetros generales.

- TI.31 Crear parámetros generales.
- TI.32 Listar parámetros generales.
- TI.33 Modificar parámetros generales.
- TI.34 Eliminar parámetros generales.

HU12. Gestionar parámetro.

- TI.35: Crear parámetro.
- TI.36: Listar parámetro.
- TI.37: Modificar parámetro.
- TI.38: Eliminar parámetro.

HU13. Asignar evaluación anual.

- TI.39: Crear evaluación anual.
- TI.40: Mostrar evaluación anual.
- TI.41: Eliminar evaluación anual.
- TI.42: Asignar evaluación anual.

HU14. Asignar evaluación trimestral.

- TI.43.Crear evaluación trimestral.
- TI.44: Mostrar evaluación trimestral.
- TI.45: Eliminar evaluación trimestral
- TI.46: Asignar evaluación trimestral.

Tabla 23: TI Realizar un control del cumplimiento de las actividades

Tarea

Número de tarea: 22	Número de HU: 8
Nombre de la tarea: Realizar un control del cumplimiento de las actividades.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 09/03/2014	Fecha fin: 11/03/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar las funcionalidades que permitan listar los usuarios del departamento seleccionado, que tienen esas actividades en sus planes de trabajo individuales, así como llevar a cabo el control del cumplimiento de las mismas. Se hace uso de las entidades Usuario, Actividad y PlanIndividual.	

En el Anexo V se muestran las restantes TI definidas para esta iteración.

Para la **cuarta iteración** se definieron las siguientes TI:

HU14: Importar plan de trabajo en formato iCalendar.

- TI.47: Importar plan de trabajo.

HU15: Exportar plan de trabajo en formato iCalendar.

- TI.48: Exportar plan de trabajo.

HU16: Gestionar reportes del cumplimiento de los planes de trabajo.

- TI.49: Crear reportes del cumplimiento de los planes de trabajo.
- TI.50: Mostrar reportes del cumplimiento de los planes de trabajo.
- TI.51: Eliminar reportes del cumplimiento de los planes de trabajo.
- TI.52: Modificar reporte del cumplimiento de los planes de trabajo.

Tabla 24: TI Importar plan de trabajo

Tarea	
Número de tarea: 47	Número de HU: 14
Nombre de la tarea: Importar plan de trabajo.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 01/02/2014	Fecha fin: 03/02/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad que permita importar un plan de trabajo.	

En el Anexo VI se muestran las restantes TI definidas para esta iteración.

3.6 Patrones de diseño usados en la solución

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de *software*.” En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares. Se deben tener presente los siguientes elementos de un patrón: su nombre, el problema (cuándo aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios). (48)

Los patrones GRASP¹² son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. (49)

Los patrones GOF¹³ se descubren a raíz del libro “*Design Patterns: Elements of Reusable Object-Oriented Software*” de Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides, en 1994. Se denominan así ya que el libro fue escrito por los cuatro autores antes mencionados. Según su propósito pueden clasificarse en creacionales, estructurales y de comportamiento.

Patrones GRASP usados en la solución:

Alta cohesión: El patrón alta cohesión plantea que una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. (50)

Las clases controladoras están formadas por varias funcionalidades que están estrechamente relacionadas, siendo las mismas las responsables de proporcionar datos a las plantillas además de realizar diferentes operaciones como instanciar objetos y acceder a sus propiedades.

Bajo Acoplamiento: Este patrón plantea que debe haber poca dependencia entre las clases para poder extraerlas de un modo independiente y reutilizarlas. (50)

La clase controladora hereda únicamente de *Controller* para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Controlador: Se basa en asignar la responsabilidad de todos los eventos realizados a una clase específica que constituye el único punto de entrada para cada evento. (50)

En la solución propuesta se define un controlador por cada entidad que recibe las peticiones realizadas, asociándolas a acciones predefinidas que permiten ejecutar la petición enviada.

¹² GRASP: General Responsibility Assignment *Software Patterns*, en español: Patrones Generales de *Software* para Asignar Responsabilidades.

¹³ GOF: Gang of Four, en español Banda de los cuatro.

Experto: Es el más usado al asignar responsabilidades. Pretende asignar una responsabilidad al experto en información, o sea, la clase que cuenta con la información necesaria para cumplir la responsabilidad, como crear un objeto o implementar un método.

Su utilización se hace evidente pues mediante el empleo del ORM Doctrine, Symfony2 crea entidades para manipular la información de la base de datos. Estas entidades contienen la información necesaria de la tabla que representan, al mismo tiempo que cuentan con las funcionalidades necesarias para el trabajo con los datos.

Creador: Consiste en asignar responsabilidades para la creación de objetos. El patrón trata de resolver el problema de diseño de quién debería ser responsable de crear una nueva instancia de alguna clase. (50)

Es utilizado en la mayoría de las clases controladoras para crear instancias de formularios y entidades.

Patrones GOF usados en la solución:

Estructurales

Decorator (decorador): Este patrón añade funcionalidad a una clase de forma dinámica y transparente. En la aplicación el archivo *base.html.twig* o plantilla global almacena el código HTML que es común a todas las páginas de la herramienta, de ella hereda el archivo *frontend.html.twig* que es el que contiene el diseño general de la herramienta para no tener que repetirlo en cada página.

Comportamiento

Strategy (estrategia): Un ejemplo de su utilización es cuando se listan los usuarios para realizar la evaluación anual; si el usuario autenticado tiene el rol decano se mostrarán los usuarios, jefes de departamento, los vicedecanos y el asesor, por otra parte, si tiene el rol jefe de departamento, entonces se mostrarán los miembros del departamento al que pertenece.

Chain of responsibility (cadena de responsabilidad): Su aplicación principal en el diseño del sistema es en el tratamiento de excepciones. Está concebido que ante la ocurrencia de un error al realizarse una determinada consulta a la base de datos el mismo sea manejado por el modelo, creando una nueva excepción de tipo *NotFoundException*. Dicha excepción debe ser propagada al controlador, el cual será el encargado de capturarla y enviarla a la vista ya traducida. Esta última por su parte mostrará un mensaje al usuario en un lenguaje entendible notificando el error y sin especificar detalles del mismo, distribuyéndose de esta manera las responsabilidades entre los diferentes componentes.

3.7 Estándares de codificación

Los estándares de codificación son reglas que se siguen para la escritura del código fuente. De tal manera que a los programadores se les facilite entender el código escrito por otros. Su propósito

fundamental es que el código tenga un estilo consistente, independiente del autor, con lo cual el sistema resulte fácil de entender y por supuesto fácil de mantener. (51) Forman parte de una de las buenas prácticas que propone la metodología XP, donde todos los programadores deben escribir y documentar el código de la misma manera.

Los estándares de codificación son un complemento a la programación por pares que debe permitir:

- Clarificar más que confundir.
- Promover la intención del código.
- Permitir que los programas se acerquen lo mejor posible al lenguaje natural.
- Incorporar las mejores prácticas de la codificación.

Para la implementación del sistema se siguió un estándar de codificación que se encuentra dentro de los elementos generales que conforman un estilo de código que es la notación *camel*. Este tipo de notación cuenta con dos variantes:

- *UpperCamelCase*, *CamelCase* o *PascalCase*: En esta variante la primera letra también es mayúscula.

Este estándar se evidencia en el nombre de todas las clases, las cuales comienzan con la primera letra en mayúscula y las demás en minúscula, por ejemplo: Usuario. En caso de ser un nombre compuesto se escribe el segundo nombre seguido del primero, también con la primera letra mayúscula y el resto en minúscula, por ejemplo: PlanDepartamento.

- *lowerCamelCase*, *camelCase* o *dromedaryCase*: La primera letra es minúscula.

Este estándar se evidencia en el nombre de los métodos, los cuales comienzan con la primera letra en minúscula. En caso de ser un nombre compuesto se escribe el segundo nombre seguido del primero con la primera letra mayúscula y el resto en minúscula, por ejemplo: actividadesFacAction.

3.8 Pruebas

“Las pruebas de *Software* son la ejecución del código usando combinaciones de entradas, en un determinado estado, para revelar defectos.” (52) Su objetivo fundamental es verificar los requisitos del sistema, por lo que son los propios requisitos la principal fuente de información a la hora de construir las pruebas del sistema.

Algunos objetivos de las pruebas expresados por Pressman en el año 2002 son (53):

- La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta posibilidad de mostrar un error no descubierto hasta entonces.

- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Uno de los pilares fundamentales de la metodología XP es el proceso de prueba. XP anima a probar constantemente tanto como sea posible para asegurar todo el proceso de desarrollo de un producto. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. (54)

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñada por el cliente final. (54)

3.8.1 Pruebas unitarias

Las pruebas unitarias consisten en comprobaciones manuales o automatizadas desarrolladas por los programadores, tienen como característica que son escritas antes que el código y son las encargadas de verificar el mismo de forma automática. Se realizan para verificar que el código correspondiente al sistema se comporta de manera esperada. Uno de los beneficios que propicia es que permite saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo. (55)

Symfony 2 utiliza la librería PHPUnit para realizar los *test* (pruebas), la cual es una herramienta para realizar pruebas de caja blanca a las aplicaciones web desarrolladas en PHP. Requiere de un amplio conocimiento del lenguaje, pues son escritas a código. De esta forma los *test* unitarios de Symfony 2 combinan la potencia de PHPUnit con las utilidades y facilidades proporcionadas por dicho *framework*.

Esta herramienta constituye sin dudas una ayuda a los programadores de la presente investigación al permitirles probar y corregir cada método al momento de ser elaborado. La cantidad de pruebas realizadas por entidad fueron las siguientes: PlanResultados (2), Parametro (8), ParametroGeneral (3), Evaluacion (5), EvaluacionArea (2), Reporte (6), PlanIndividual (3), PlanFacultad (3), PlanDepartamento (3), Actividad (13). Se realizaron un total de 48 pruebas unitarias al código, para un total de 48 pruebas satisfactorias.

3.8.2. Pruebas de aceptación

Las pruebas de aceptación son una parte integral del desarrollo incremental según lo practicado por XP. Todas las historias de los usuarios se apoyan en este tipo de pruebas, las cuales son más importantes que las pruebas unitarias, debido a que significan la satisfacción del cliente con el producto.

A continuación, se detallan los casos de prueba (CP) de aceptación por cada una de las iteraciones definidas. La tabla propuesta para los CP contiene los siguientes parámetros:

- **Código:** Contiene el identificador del CP.
- **Historia de usuario:** Contiene el nombre de la HU correspondiente a este CP.
- **Nombre:** Contiene el nombre de la funcionalidad que se prueba.
- **Descripción:** Contiene una breve descripción de la prueba realizada.
- **Condiciones de ejecución:** Incluye las condiciones necesarias para que se pueda realizar la prueba.
- **Pasos de ejecución:** Contiene los pasos a seguir para poder realizar la prueba.
- **Resultado esperado:** Contiene la descripción de lo que se espera después de realizada la prueba.
- **Evaluación de la prueba:** Muestra si la prueba fue satisfactoria o insatisfactoria.

Pruebas de aceptación para la iteración 1:

Tabla 25: CP de aceptación P1

Caso de prueba de aceptación	
Código: P1-01	Historia de usuario: Editar perfil de usuario.
Nombre: Editar perfil.	
Descripción: Prueba para la funcionalidad que permite editar el perfil de usuario.	
Condiciones de ejecución:	
✓ El usuario no debe estar autenticado con el rol administrador.	
Pasos de ejecución:	
<p>El usuario selecciona la opción "Editar perfil" que se encuentra en el menú principal, se le brinda la posibilidad de editar los siguientes datos: nombre y apellidos (no pueden ser nulo y admiten letras únicamente), carnet de identidad (no puede ser nulo y admite solamente un número de 11 dígitos), solapín (no puede ser nulo y admite una cadena de texto que comience con una letra mayúscula seguida de 6 números), departamento (permite escoger el departamento del usuario y admite valor nulo escogiendo la opción "Ninguno"), militancia (permite escoger la militancia del usuario que puede ser: UJC o PCC, admite valor nulo escogiendo la opción "Ninguna"), categoría docente (permite escoger la categoría docente del usuario que puede ser: Profesor auxiliar, Instructor, Asistente y admite valor nulo escogiendo la opción "Ninguna"), categoría científica (permite escoger la categoría científica del usuario que puede ser: Máster, Doctor y admite valor nulo escogiendo la opción "Ninguna"), residencia (permite escoger si el usuario es: Interno o Externo, no admite valor nulo). Luego se acciona sobre el</p>	

botón "Aceptar" para guardar los datos y si se acciona en el botón "Cancelar" se muestra la vista anterior. En caso de una entrada de datos no válida se muestra un mensaje de error.

Resultado esperado: Los datos son registrados satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Los restantes CP se encuentran en el Anexo VII.

Pruebas de aceptación para la iteración 2:

Tabla 21: CP de aceptación P4

Caso de prueba de aceptación	
Código: P4-01	Historia de usuario: Buscar profesor.
Nombre: Buscar profesor.	
Descripción: Prueba para la funcionalidad que permite buscar un profesor en la aplicación.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ Debe existir en el sistema al menos un usuario registrado. ✓ Para obtener algún resultado se debe ingresar al menos un dato en el filtro seleccionado. 	
Pasos de ejecución: <p>El usuario selecciona la opción "Datos de profesor" que se encuentra en el menú principal. Luego se selecciona la opción "Buscar profesores" y el sistema brinda la posibilidad de filtrar la búsqueda por diferentes criterios, mostrando una lista con los resultados de la misma. La búsqueda se puede filtrar por los siguientes datos: nombre y apellidos (admite solamente letras), carnet de identidad (admite solamente un número de 11 dígitos), solapín (admite una cadena de texto que comience con una letra mayúscula seguida de 6 números), militancia (UJC, PCC), categoría docente (Profesor auxiliar, Instructor, Asistente), categoría científica (Máster, Doctor), residencia (Interno, Externo). En caso de ocurrir un error en la entrada de datos se muestra el mensaje "¡Error! Datos incorrectos". En caso de buscar sin haber insertado dato alguno se mostrará el mensaje "¡Advertencia! Debe agregar datos a la búsqueda".</p>	
Resultado esperado: Los datos son encontrados satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Los restantes CP se encuentran en el Anexo VIII.

Pruebas de aceptación para la iteración 3:

Tabla 26: CP de aceptación P8

Caso de prueba de aceptación

Código: P8-01	Historia de usuario: Realizar control del cumplimiento de las actividades de los usuarios por departamento docente.
Nombre: Realizar control del cumplimiento de las actividades.	
Descripción: Prueba para la funcionalidad que permite realizar un control de las actividades de los usuarios por departamento docente.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de jefe de departamento. ✓ Los usuarios del departamento deben tener actividades asignadas. 	
Pasos de ejecución:	
<p>Se selecciona la opción “Usuarios” del menú principal y se escoge “Controlar actividades”, mostrándose la lista de usuarios del departamento. Luego se selecciona el usuario deseado y se muestra el listado de sus planes individuales. Al seleccionar un plan individual se muestra una lista con sus actividades. El jefe de departamento tiene la posibilidad de seleccionar las actividades cumplidas y acciona sobre el botón “Aceptar”, posteriormente se muestra un mensaje de confirmación: “La actividad de los usuarios ha sido modificada” y el estado de la misma se actualiza a “Cumplida”. Las actividades que no estén marcadas se consideran incumplidas.</p>	
Resultado esperado: Los datos se pueden controlar de forma satisfactoria.	
Evaluación de la prueba: Prueba satisfactoria.	

Los restantes CP se encuentran en el Anexo IX.

Pruebas de aceptación para la iteración 4:

Tabla 27: CP de aceptación P17

Caso de prueba de aceptación	
Código: P17-01	Historia de usuario: Gestionar reportes del cumplimiento de los planes de trabajo.
Nombre: Crear reporte.	
Descripción: Prueba para la funcionalidad que permite crear los reportes del cumplimiento de los distintos planes de trabajo.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de asesor o jefe de departamento. 	
Pasos de ejecución:	

Se selecciona el plan de trabajo requerido, al ser mostrado se selecciona la opción "Crear reporte", aparece un formulario que permite llenar los datos referentes a la cantidad de actividades incorporadas (el campo no puede ser nulo y solo admite números entre 0 y 50) y cantidad modificadas (el campo no puede ser nulo y solo admite números entre 0 y 50). Luego de llenado el formulario se acciona sobre el botón "Aceptar" y se muestra un mensaje de confirmación: "El reporte ha sido creado". En caso de una entrada de datos no válida se muestra un mensaje de error.

Resultado esperado: El reporte es creado satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: P17-02

Historia de usuario: Gestionar reportes del cumplimiento de los planes de trabajo.

Nombre: Modificar reporte.

Descripción: Prueba para la funcionalidad que permite modificar los reportes del cumplimiento de los distintos planes de trabajo.

Condiciones de ejecución:

- ✓ El usuario debe estar autenticado con el rol de asesor o jefe de departamento.
- ✓ Debe existir al menos un reporte en la aplicación.

Pasos de ejecución:

Cuando se muestra el reporte del cumplimiento de los planes de trabajo se accede a la opción "Modificar reporte" y se permite modificar las actividades incorporadas (el campo no puede ser nulo y solo admite números entre 0 y 50) y las modificadas (el campo no puede ser nulo y solo admite números entre 0 y 50). Luego se presiona el botón "Aceptar" y se muestra el mensaje de confirmación: "El reporte ha sido modificado". En caso de una entrada de datos no válida se muestra un mensaje de error.

Resultado esperado: El reporte es modificado satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: P17-03

Historia de usuario: Gestionar reportes del cumplimiento de los planes de trabajo.

Nombre: Mostrar reporte.
Descripción: Prueba para la funcionalidad que permite mostrar los reportes del cumplimiento de los distintos planes de trabajo.
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con uno de los siguientes roles: decano, vicedecano, asesor o jefe de departamento. ✓ Debe existir al menos un reporte en la aplicación.
Pasos de ejecución: Al ser mostrado un plan de trabajo de facultad o un plan de trabajo por departamentos docentes, se accede a la opción "Mostrar reporte" y se muestra un gráfico de pastel con los siguientes datos: actividades cumplidas, incumplidas, modificadas e incorporadas, además un listado de las actividades que incluye la fecha, hora y valoración del cumplimiento de cada actividad.
Resultado esperado: El reporte es mostrado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P17-04	Historia de usuario: Gestionar reportes del cumplimiento de los planes de trabajo.
Nombre: Eliminar reporte.	
Descripción: Prueba para la funcionalidad que permite eliminar los reportes del cumplimiento de los distintos planes de trabajo.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de asesor o jefe de departamento. ✓ Debe existir al menos un reporte en la aplicación. 	
Pasos de ejecución: Cuando se muestra el reporte del cumplimiento del plan de trabajo se presiona el botón "Eliminar" y aparece un mensaje de confirmación: "El reporte ha sido eliminado".	
Resultado esperado: El reporte es eliminado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Los restantes CP se encuentran en el Anexo X.

3.8.1.1 Resultados de las pruebas de aceptación

Para conocer el grado de satisfacción del cliente atendiendo a sus necesidades, se realizaron 4 iteraciones de las pruebas de aceptación, una por cada entrega realizada. Como resultado se obtuvo que en la primera iteración se detectaron un total de 4 no conformidades, de ellas 1 significativa y las restantes no significativas, que fueron resueltas antes de la segunda iteración. Cuando se iteró por segunda vez se detectaron 3 no conformidades no significativas y estas fueron solucionadas. En la tercera iteración solamente se encontró 1 no conformidad significativa, que fue resuelta antes de ejecutarse la cuarta iteración, reportando esta última 0 no conformidades, razón por la cual la aplicación quedó aprobada por parte del cliente, quien evaluó el 100 % de los casos de prueba de forma satisfactoria.

A petición del cliente y los desarrolladores, se propuso que el *software* fuera validado por el Grupo de calidad del Centro de Tecnologías para la Formación (FORTES) de la Facultad 4, realizándole pruebas de caja negra. Se realizaron 4 iteraciones y se encontraron varios tipos de errores, entre los que sobresalen que algunos botones cancelar no cerraban las ventanas, algunos botones aceptar no funcionaban correctamente y algunos campos de entrada estaban sin validar.

Se realizó una primera iteración donde se detectaron un total de 5 no conformidades: 3 significativas y dos no significativas, que fueron solucionadas. En la segunda iteración se detectaron 2 no conformidades: 1 significativa y 1 no significativa, que también fueron resueltas. En la tercera iteración solamente se detectó 1 no conformidad no significativa, que igualmente fue solucionada. Cuando se realizó la cuarta iteración no se encontraron no conformidades. Se pudo observar que los errores encontrados en las iteraciones anteriores fueron erradicados y se comprobó que existe correspondencia entre las funcionalidades implementadas y las propuestas. Estos resultados se muestran en el siguiente gráfico:

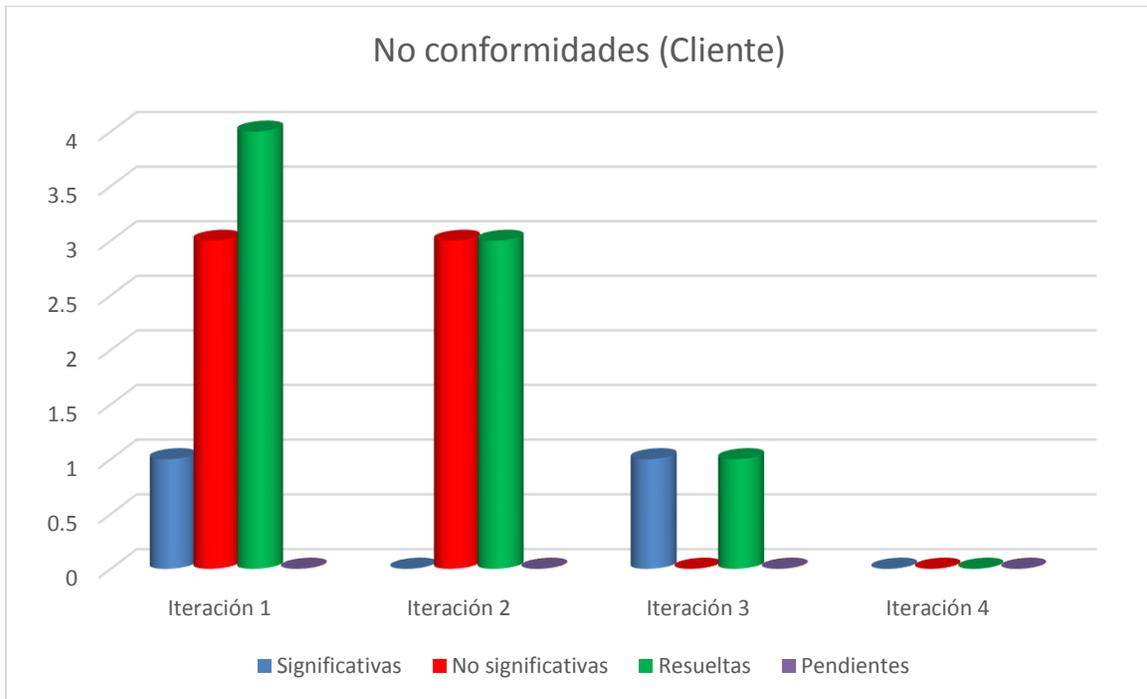


Ilustración 5: Resultado de las pruebas de aceptación realizadas por el cliente

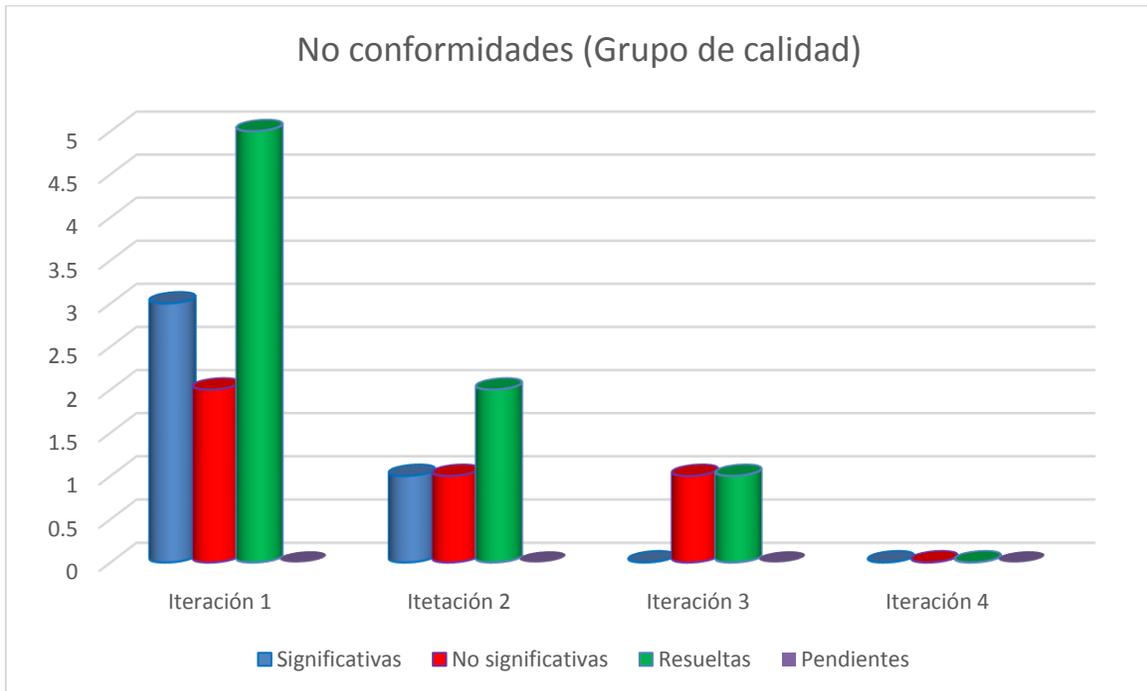


Ilustración 6: Resultado de las pruebas de aceptación realizadas por el grupo de calidad

3.9 Conclusiones parciales

El diseño propuesto en el capítulo anterior permitió desarrollar las funcionalidades descritas a partir de las herramientas, lenguajes, metodologías y tecnologías seleccionadas. En el transcurso de este capítulo se detallaron los patrones de diseño, las tareas de ingeniería, así como la ejecución de las pruebas unitarias y de aceptación que son las que propone la metodología XP, para comprobar en cada una de las iteraciones del desarrollo que se han alcanzado los objetivos propuestos al inicio de estas.

Conclusiones generales

- Los métodos científicos empleados para investigar el objeto de estudio posibilitaron identificar la teorías y los conceptos que permiten un mayor entendimiento del mismo, sustentando así la presente investigación.
- El estudio realizado durante la investigación permitió seleccionar correctamente las herramientas y tecnologías para el desarrollo del sistema, el cual se sustentó en las bases de la metodología XP.
- La selección de la metodología permitió generar los artefactos fundamentales que posibilitaron el desarrollo de la aplicación.
- El diseño realizado permitió realizar una aplicación que cumpliera con las funcionalidades definidas y que dan solución al objeto de estudio.
- Luego de haberse realizado el proceso de pruebas de aceptación se demostró la correcta implementación de sus funcionalidades dando cumplimiento a las necesidades del cliente.

Como resultado de la investigación se logró desarrollar una aplicación con la que se da cumplimiento a las especificidades de los objetivos propuestos. Con el principal propósito de mejorar el proceso de distribución de la información, así como la gestión de las actividades que se realizan en los departamentos docentes de la facultad 4 y además para contribuir a elevar la calidad del desarrollo del trabajo en los mismos.

Recomendaciones

A partir del trabajo realizado y después de haber analizado los resultados obtenidos se recomienda:

- Aplicar el *software* en los departamentos docentes de la facultad 4 y socializar su uso en el resto de las facultades de la universidad.
- Añadir otras funcionalidades acorde a las nuevas necesidades que surjan en los departamentos.

Referencias bibliográficas

1. **Hernández León, Rolando Alfredo y Coello González, Sayda.** *El proceso de Investigación Científica.*
2. **Mintzberg, Henry.** *Planeación Estratégica.* Colombia : s.n., 2007.
3. **Díaz Gavilán, Yasmín Karín.** *Desarrollo de imagen institucional del Colegio Santa María de Matellini de Chorritos a través de la planificación estratégica.* 2001.
4. **Cerreto, Julio.** Planeación-Estratégica. *Planeación-Estratégica.* [En línea] [Citado el: 02 01, 2014.] [http://planeacion-estrategica.blogspot.com/..](http://planeacion-estrategica.blogspot.com/)
5. **Montero Mosqueda, Mailis y Rodríguez Favier, Erislandy.** *Sistema de Gestión para la planificación y control de las actividades de los profesores.* Ciudad Habana : s.n., 2008.
6. **Battle, Phil.** Orientaciones para preparar un plan de trabajo. *Orientaciones para preparar un plan de trabajo.* [En línea] [Citado el: 02 10, 2014.] <http://www.scn.org/mpfc/modules/pm-plns.htmScCont>.
7. **Lugo Pérez, Dr Javier.** *Guía para elaboración de plan de trabajo e informe anual de oficinas y decanatos de la UPR-Utuado relacionados al plan estratégico.* Utado, Puerto Rico : s.n., 2012.
8. **Rizo Moreno, Héctor Eli.** *Evaluación del docente universitario. Una visión institucional.* Colombia : s.n.
9. **Valle Lima, Alberto.** *La dirección en la educación. Soporte magnético.* 2003.
10. **autores, colectivo de.** *Seminario Nacional para el Personal Docente.* La Habana : Pueblo y Educación, 1999.
11. **Salinas, Marta Lorena.** *La evaluación profesoral.*
12. **Desiderio Rodríguez, Rodrigo.** *Metodología para la evaluación del desempeño de los docentes universitarios.*
13. **Aja Quiroga, Lic. Lourdes.** *Gestión de la información, gestión del conocimiento y gestión de la calidad en las organizaciones.* Ciudad de la Habana : s.n., Octubre-2002.
14. **ocu.es. ocu.es.** [En línea] [Citado el: 02 15, 2014.]
15. **Pérez Neira, Yaneisy y Ruíz Rodríguez, Rolian.** *Sistema para el control de la información y evaluación de los profesores en los departamentos docentes de la facultad 8.* Ciudad de la Habana : s.n., 2009.
16. **Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, Aurelio.** *Metodologías de desarrollo del software.* 2008.
17. **Pérez Neira, Yaneisy y Ruiz Rodríguez, Rolian.** *Sistema para el control de la información y evaluación de los profesores en los departamentos docentes de la Facultad 8.* Ciudad Habana : s.n., 2009.
18. **Joskowicz, José.** *XP1.* 2008.

19. **González Blanco, Yaneisy.** *Sistema de UML para los proyectos de la línea Soluciones Integrales.* La Habana : s.n., 2011.
20. **De la Cruz Rodríguez, Elizabet y Guzmán Hernández, Dalianny.** *Propuesta de herramienta CASE para los proyectos del centro de Desarrollo de la Informática Industrial (CEDIN).* La Habana : s.n., 2010.
21. **Ruiz Leyva, Ilen.** *Análisis y diseño de un sistema para automatizar la gestión de la información de los profesores de la facultad 4 de la UCI.* Ciudad de La Habana : s.n., 2008.
22. **González Martins, Miguel Angel.** *Los lenguajes de programación actuales.* Universidad de Alcalá de Henares : s.n., 2013.
23. **Eguíluz Pérez, Javier.** *Introducción a JavaScript.* 2008.
24. **Maqueira González, Jeykel.** *Cliente JQuery para la integración del Servidor de Inteligencia de Negocio con el Sistema Informativo (SI) de la Administración Provincial de Artemisa.* Artemisa : s.n., 2012.
25. **Cruz Iñigo, Dayron.** *Sistema Gestión de la Información de la Facultad 8. Módulo para la Gestión de la Información Docente.* Ciudad Habana : s.n., 2006-2007.
26. **Ledesma, Rubén.** *Introducción al Bootstrap. Desarrollo de un ejemplo acompañado de software de aplicación.* Mar del Plata, Argentina : s.n.
27. **Ing. Mendoza Vázquez, Iván.** *Definición de un framework para las aplicaciones web con navegación sensible a concers.* República Argentina : s.n.
28. **Codelgniter.** *Guía del Usuario de Codelgniter Versión 2.1.4. Guía del Usuario de Codelgniter Versión 2.1.4.* [En línea] 2013. [Citado el: 01 2014, 17.] [http://escodeigniter.com/guía_usuario/..](http://escodeigniter.com/guía_usuario/) [http://escodeigniter.com/guía_usuario/..](http://escodeigniter.com/guía_usuario/)
29. **Ramis Suárez, Regla María y Pereda Trujillo, Junior.** *Sistema de Gestión del Plan de Trabajo de los profesores de la Facultad 1 de la Universidad de las ciencias Informáticas.* Ciudad Habana : s.n., 2009.
30. **alan2793.** *Entorno de Desarrollo Integrado(IDE). Entorno de Desarrollo Integrado(IDE).* [En línea] 2013. [Citado el: 02 2014, 03.] [http://alanss18.wordpress.com/2013/01/25/entorno-de-desarrollo-integradoide/.](http://alanss18.wordpress.com/2013/01/25/entorno-de-desarrollo-integradoide/) [http://alanss18.wordpress.com/2013/01/25/entorno-de-desarrollo-integradoide/.](http://alanss18.wordpress.com/2013/01/25/entorno-de-desarrollo-integradoide/)
31. **Up to Down.** *Up to Down.* [En línea] 2011. [Citado el: 01 2014, 20.] [http://netbeans-ide.uptodown.com/..](http://netbeans-ide.uptodown.com/) [http://netbeans-ide.uptodown.com/..](http://netbeans-ide.uptodown.com/)
32. **Pérez Garzón, María Teresa.** *Innovación y experiencias educativas.* 2011.
33. *¿ Qué es un Sistema Gestor de Base de Datos o SGBD? ¿ Qué es un Sistema Gestor de Base de Datos o SGBD?* [En línea] 2008. [Citado el: 03 05, 2014.] [http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-base-de-datos-o-sgbd/.](http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-base-de-datos-o-sgbd/)
34. **Peña Cruz, Yamile y Mustelier Ramírez, Franklin.** *Sistema Recomendador de noticias para el portal Octavitos.* La Habana : s.n., 2013.

35. Apache. *Apache*. [En línea] [Citado el: 01 2014, 15.] http://httpd.apache.org/docs/2.0/new_features_2_0.html.
http://httpd.apache.org/docs/2.0/new_features_2_0.html.
36. juandarodriguez.es. *juandarodriguez.es*. [En línea] [Citado el: 03 2014, 20.] <http://juandarodriguez.es/cursosf14/unidad7.html>. <http://juandarodriguez.es/cursosf14/unidad7.html>.
37. librosweb.es. *librosweb.es*. [En línea] [Citado el: 03 2014, 20.] <http://librosweb.es/symfony-1-2/capitulo-2/el-patron-mvc.html>. <http://librosweb.es/symfony-1-2/capitulo-2/el-patron-mvc.html>.
38. **Coronel Montaguth, Yessy Lorena y Díaz Manzano, Margarita Rosa**. *Ensayo sobre metodología de la investigación cuantitativa en las ciencias sociales*. Barranquilla : s.n., 2012.
39. **Letelier, Patricio y Penadés, Ma.Carmen**. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. [En línea] [Citado el: 03 2014, 03.] <http://www.cyta%28xp%29.html/>.
40. **Letelier Torres, Patricio y Sánchez López, Emilio A**. *Metodologías ágiles en el desarrollo de software*. España : s.n., 2003.
41. **Bagarotti Acebo, Yadira**. *Gestión de la Calidad en el ciclo de Desarrollo del Software de proyectos que usan metodologías ágiles*. Granma : s.n., Agosto, 2012.
42. **Peñalver Romero, Gladys Marsi, García de la Puente, Sergio Jesús y Meneses Abad, Abel**. *SXP, metodología de desarrollo de software SXP, software development methodology*. s.l. : Serie Científica de la Universidad de las Ciencias Informáticas, 2011.
43. **Sommerville, Ian**. *Ingeniería de Software. Séptima edición*. Madrid : PEARSON ADUCACION.S.A, 2005.
44. **Ing. Jaskowicz, Jose**. *Reglas y Prácticas en eXtreme Programming*. España : s.n., 2008.
45. **Reyes R., Leydis, Ruiz, María del Carmen y Vivanco E., Mónica**. Prototipos informáticos. *Prototipos informáticos*. [En línea] 2009. [Citado el: 04 2014, 01.] <http://sistemas2009unl.wordpress.com/prototipos-informaticos/>.
46. **Casas, Sandra y Reinaga, Héctor**. *Identificación y Modelado de los Aspectos Tempranos por Tarjetas de Responsabilidad y Colaboraciones*. Santa Cruz, Argentina : s.n.
47. **Erljman Piwen, Ariel y Goyén Fros, Alejandro**. *Problemas y soluciones en la implementación de Extreme Programming*. Uruguay : s.n., 2001.
48. **Tedeschi, Nicolás**. ¿Qué es un Patrón de diseño? ¿Qué es un Patrón de diseño? [En línea] [Citado el: 04 20, 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
<http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
49. **Visconti, Marcello y Astudillo Hernán**. *Fundamentos de la Ingeniería de software*. inf.utfsm.cl.
50. **LAMAN, C**. *UML y patrones. Una introducción al análisis y diseño orientado a objetos y el proceso unificado. 2da Edición*. España : s.n., 2002.
51. **Arias Calleja, Manuel**. *Carmen. Estándares de codificación*.

52. **Prof. Carolina, Ivette.** *Clase11: Pruebas de software.* Universidad Simón Bolívar : s.n.
53. **Ruiz Tenorio, Roberto.** *Las pruebas de software y su importancia en las organizaciones.* Universidad de Veracruzana : s.n., Agosto,2010.
54. **J.J. Gutiérrez, M.J. Escalona, M. Mejías, J. Torres.** *Pruebas del sistema en la Programación Extrema.* Sevilla : s.n.
55. **Presman, Roger S.** *Ingeniería de software. Un enfoque práctico. Quinta edición.*
56. **Calabria, Luis y Píriz, Pablo.** *Metodología XP.* Uruguay : s.n., 2003.
57. **Pozo Hidalgo, Sergio.** *Técnicas Automáticas para la Diagnósis de Consistencia y Conformidad en las Políticas de Control de Acceso.* Research Report submitted to the Department of Computer Languages and Systems of the University of Sevilla.
58. **Blanco Bueno, Carlos.** *Ingeniería de software II. Tema 1: Construcción y pruebas de software.* Universidad de Cantabria : s.n.

Anexos

Anexo I

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Gestionar plan de trabajo por departamento docente
Usuario: Jefe de departamento	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
<p>Descripción: El sistema debe permitir adicionar, mostrar y eliminar un plan de trabajo según el departamento docente al que pertenezca el usuario registrado. Para adicionar el plan de trabajo se selecciona la fecha en que va a ser creado (mes y año), el mismo puede ser mostrado junto a las actividades que lo componen y ser eliminado. El jefe de departamento podrá adicionar y quitar actividades pertenecientes al plan de trabajo de la facultad en el plan de trabajo del departamento.</p>	
<p>Observaciones: Para mostrar una lista con los planes de trabajo, así como para poder realizar la opción de eliminar, deben existir planes de trabajo de este tipo en el sistema. Da cumplimiento a la funcionalidad F6 "Gestionar plan de trabajo por departamento docente".</p>	

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Gestionar plan de trabajo individual (profesor y especialista)
Usuario: Todos.	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
<p>Descripción: El sistema debe permitir adicionar, mostrar y eliminar su plan de trabajo. Para adicionar el plan de trabajo se selecciona la fecha en que va a ser creado (mes y año), el mismo puede ser mostrado junto a sus actividades y ser eliminado. El usuario podrá adicionar y quitar actividades pertenecientes al plan de trabajo de la facultad y al plan de trabajo del departamento en su plan de trabajo individual.</p>	

Observaciones: Para mostrar una lista con los planes de trabajo y eliminar uno de ellos, deben existir planes de trabajo de este tipo en el sistema. Da cumplimiento a la funcionalidad F7 "Gestionar plan de trabajo individual (profesor y especialista)".

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Realizar control del cumplimiento de las actividades de los usuarios por departamento docente
Usuario: Jefe de Departamento	
Prioridad de Negocio: Media.	Riesgo de desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 2
Descripción: El sistema debe permitir llevar a cabo un control del cumplimiento de las actividades relegadas en los planes de trabajo de cada uno de los profesores por departamentos docentes. Los jefes de departamentos llevan el control del cumplimiento de las actividades del plan de trabajo de la facultad y del plan de trabajo por departamento docente que fueron agregadas al plan de trabajo individual de sus subordinados.	
Observaciones: Responde a la funcionalidad F8 " Realizar control del cumplimiento de las actividades de los usuarios por departamento docente."	

Historia de Usuario	
Número: 9	Nombre Historia de Usuario: Gestionar plan de resultados
Usuario: Todos	
Prioridad de Negocio: Media	Riesgo de desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 3
Descripción: El sistema debe permitir crear, mostrar, exportar a "pdf" y eliminar un plan de resultados. Para crear un plan de resultados se debe agregar el curso en que va a ser creado.	
Observaciones: Para mostrar y eliminar un plan de resultados debe existir al menos uno en el sistema. Da cumplimiento a la funcionalidad F9 " Gestionar plan de resultados".	

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Gestionar área
Usuario: Administrador	
Prioridad de Negocio: Media	Riesgo de desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 3
Descripción: El sistema debe permitir crear, modificar y eliminar las áreas de resultados claves. Para crear una actividad se deben escoger el nombre y el número del área.	
Observaciones: Para modificar y eliminar un área del plan de resultados debe existir al menos una en el sistema. Da cumplimiento a la funcionalidad F10 " Gestionar área".	

Historia de Usuario	
Número: 11	Nombre Historia de Usuario: Gestionar parámetros generales
Usuario: Administrador	
Prioridad de Negocio: Media	Riesgo de desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 3
Descripción: El sistema debe permitir crear, modificar y eliminar los parámetros generales de los planes de resultados. Para crear un parámetro general se deben escoger el nombre y el área a la que va a pertenecer el parámetro general.	
Observaciones: Para modificar y eliminar un parámetro general del plan de resultados debe existir al menos uno en el sistema. Da cumplimiento a la funcionalidad F11 " Gestionar parámetros generales".	

Historia de Usuario	
Número: 12	Nombre Historia de Usuario: Gestionar parámetros
Usuario: Todos	
Prioridad de Negocio: Media	Riesgo de desarrollo: Medio

Puntos estimados: 0.5	Iteración asignada: 3
Descripción: El sistema debe permitir crear, modificar y eliminar los parámetros del plan de resultados. Para crear un parámetro se debe insertar el nombre la fecha y escoger el parámetro general al que va a pertenecer el parámetro, el estado y se puede poner una observación. Los datos pueden ser editados por el jefe de departamento o el decano.	
Observaciones: Para modificar y eliminar un parámetro del plan de resultados debe existir al menos uno en el sistema. Da cumplimiento a la funcionalidad F12 " Gestionar parámetro ".	

Historia de Usuario	
Número: 13	Nombre Historia de Usuario: Asignar evaluación anual
Usuario: Decano y jefe de departamento	
Prioridad de Negocio: Media	Riesgo de desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 3
Descripción: El sistema debe permitir crear, mostrar, eliminar y asignar evaluación anual a los profesores. Para crear una evaluación anual se selecciona el curso docente que se va a evaluar. Teniendo en cuenta el desempeño del profesor en el curso especificado se le asigna una evaluación (Excelente, Bien, Regular y Mal) y se brinda la posibilidad de redactar una recomendación.	
Observaciones: Da cumplimiento a la F13 "Asignar evaluación".	

Historia de Usuario	
Número: 14	Nombre Historia de Usuario: Asignar evaluación trimestral
Usuario: Decano y jefe de departamento	
Prioridad de Negocio: Media	Riesgo de desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 3
Descripción: El sistema debe permitir crear, mostrar, eliminar y asignar evaluación trimestral a los profesores. Para crear una evaluación trimestral se selecciona el trimestre que se va a evaluar. Teniendo en cuenta el desempeño del profesor en el trimestre especificado se le asigna una evaluación (Superior, Adecuado o	

Deficiente).
Observaciones: Da cumplimiento a la F13 "Asignar evaluación".

Historia de Usuario	
Número: 15	Nombre Historia de Usuario: Importar plan de trabajo en formato iCalendar
Usuario: Todos	
Prioridad de Negocio: Baja	Riesgo de desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 3
Descripción: El sistema debe permitir importar un archivo ".ics" con el cual se puede crear un plan de trabajo o puede ser eliminado. Para subir el archivo al servidor debe insertarse los siguientes datos: nombre y tipo (se escoge el tipo de plan de trabajo que se creará). Además, se brinda la opción para subir el archivo.	
Observaciones: Da cumplimiento a la funcionalidad F14 "Importar plan de trabajo en formato iCalendar".	

Historia de Usuario	
Número: 16	Nombre Historia de Usuario: Exportar plan de trabajo en formato iCalendar
Usuario: Todos	
Prioridad de Negocio: Baja.	Riesgo de desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 3
Descripción: El sistema debe permitir exportar un plan de trabajo determinado en formato ".isc".	
Observaciones: Da cumplimiento a la funcionalidad F15 "Exportar plan de trabajo en formato iCalendar".	

Historia de Usuario	
Número: 17	Nombre Historia de Usuario: Gestionar reporte del cumplimiento de los planes de trabajo.

Usuario: Jefe de Departamento y asesor	
Prioridad de Negocio: Baja	Riesgo de desarrollo: Bajo
Puntos estimados: 0.5	Iteración asignada: 3
Descripción: El sistema debe permitir crear, mostrar, modificar y eliminar los reportes del cumplimiento de las actividades reflejadas en los planes de trabajo de cada uno de los profesores. Para crear un reporte de cumplimiento se deben insertar los siguientes datos: actividades incorporadas y actividades modificadas, los cuales pueden ser modificados. Además, el reporte puede ser exportado a "pdf".	
Observaciones: Para mostrar, modificar y eliminar un reporte debe existir al menos uno en el sistema. Da cumplimiento a la funcionalidad F16 " Gestionar reporte del cumplimiento de los planes de trabajo".	

Anexo II

Nombre: ActividadController	
Responsabilidad	Colaboraciones
eliminarAction	Plan
showAction	

Nombre: PlanDepartamentoController	
Responsabilidad	Colaboraciones
listarAction	
crearAction	
exportarAction	
eliminarAction	Actividad
	Reporte
	PlanIndividual
mostrarAction	Actividad
adicionarAction	Actividad
editAction	Actividad
updateAction	Actividad
actividadesFacAction	Actividad
agregarAction	Actividad
quitarAction	Actividad
reporteAction	Reporte
reporteNuevoAction	Reporte

reporteCrearAction	Reporte
reporteActualizarAction	Reporte
reporteEditarAction	Reporte
eliminarReporteAction	Reporte
usuariosDepAction	Actividad
	PlanIndividual
usuariosCumplieronAction	Actividad
	PlanIndividual

Nombre: PlanDepartamentoController	
Responsabilidad	Colaboraciones
listarAction	
crearAction	
exportarAction	
eliminarAction	Actividad
	Reporte
	PlanIndividual
mostrarAction	Actividad
nuevoAction	Actividad
editAction	Actividad
updateAction	Actividad
reporteAction	Reporte
reporteNuevoAction	Reporte
reporteCrearAction	Reporte
reporteActualizarAction	Reporte
reporteEditarAction	Reporte
eliminarReporteAction	Reporte

Nombre: FileControler	
Responsabilidad	Colaboraciones
listarAction	
uploadAction	
parsearAction	
eliminarAction	

Nombre: AreaControler

Responsabilidad	Colaboraciones
indexAction	
createAction	
newAction	
editAction	
updateAction	
deleteAction	

Nombre: PlanResultadosController	
Responsabilidad	Colaboraciones
planesReDepAction	Usuario
planUserDepAction	Usuario
createAction	Area
deleteAction	Parametro
indexAction	
showAction	
exportarAction	

Nombre: ParametroGeneralController	
Responsabilidad	Colaboraciones
createAction	Area
editAction	Area
updateAction	Area
indexAction	
deleteAction	

Anexo III

Tarea	
Número de tarea: 2	Número de HU: 2
Nombre de la tarea: Crear departamento docente	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 13/01/2014	Fecha fin: 15/01/2014
Programador responsable: Rosalia Suárez Núñez	

Descripción: Implementar la funcionalidad crear departamento docente haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Departamento.

Tarea	
Número de tarea: 3	Número de HU: 2
Nombre de la tarea: Modificar departamento docente.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 15/01/2014	Fecha fin: 17/01/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad modificar departamento docente haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Departamento.	

Tarea	
Número de tarea: 4	Número de HU: 2
Nombre de la tarea: Eliminar departamento docente.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 17/01/2014	Fecha fin: 19/01/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad eliminar departamento docente haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Departamento.	

Tarea	
Número de tarea: 5	Número de HU: 3
Nombre de la tarea: Adicionar actividad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 26/01/2014	Fecha fin: 28/01/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para adicionar una actividad haciendo uso de un CRUD (Gestionar	

de Symfony 2) de la entidad Actividad.

Tarea	
Número de tarea: 6	Número de HU: 3
Nombre de la tarea: Mostrar actividad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 26/01/2014	Fecha fin: 28/01/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para mostrar una actividad haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Actividad.	

Tarea	
Número de tarea: 7	Número de HU: 3
Nombre de la tarea: Editar actividad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 28/01/2014	Fecha fin: 30/01/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para editar una actividad haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Actividad.	

Tarea	
Número de tarea: 8	Número de HU: 3
Nombre de la tarea: Eliminar actividad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 01/02/2014	Fecha fin: 03/02/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para eliminar una actividad haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Actividad.	

Anexo IV

Tarea	
Número de tarea: 10	Número de HU: 5
Nombre de la tarea: Adicionar plan de trabajo de la facultad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 05/02/2014	Fecha fin: 07/02/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para adicionar un plan de trabajo de la facultad haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanFacultad.	

Tarea	
Número de tarea: 11	Número de HU: 5
Nombre de la tarea: Mostrar plan de trabajo de la facultad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 07/02/2014	Fecha fin: 10/02/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para mostrar un plan de trabajo de la facultad haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanFacultad.	

Tarea	
Número de tarea: 12	Número de HU: 5
Nombre de la tarea: Eliminar plan de trabajo de la facultad.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 10/02/2014	Fecha fin: 12/02/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para eliminar un plan de trabajo de la facultad haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanFacultad.	

Tarea

Número de tarea: 13	Número de HU: 6
Nombre de la tarea: Adicionar plan de trabajo por departamento docente.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 12/02/2014	Fecha fin: 14/02/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para adicionar un plan de trabajo por departamento docente haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanDepartamento.	

Tarea	
Número de tarea: 14	Número de HU: 6
Nombre de la tarea: Mostrar plan de trabajo por departamento docente.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 14/02/2014	Fecha fin: 17/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para mostrar un plan de trabajo por departamento docente haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanDepartamento.	

Tarea	
Número de tarea: 15	Número de HU: 6
Nombre de la tarea: Eliminar plan de trabajo por departamento docente.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 17/02/2014	Fecha fin: 19/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para eliminar un plan de trabajo por departamento docente haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanDepartamento.	

Tarea	
Número de tarea: 16	Número de HU: 6

Nombre de la tarea: Agregar actividad perteneciente al plan de trabajo de la facultad en el plan de trabajo del departamento.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 19/02/2014	Fecha fin: 22/02/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para agregar una actividad perteneciente al plan de trabajo de la facultad en el plan de trabajo por departamento docente haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanDepartamento.	

Tarea	
Número de tarea: 17	Número de HU: 6
Nombre de la tarea: Quitar actividad perteneciente al plan de trabajo de la facultad en el plan de trabajo del departamento.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 22/02/2014	Fecha fin: 24/02/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para quitar una actividad perteneciente al plan de trabajo de la facultad en el plan de trabajo por departamento docente haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanDepartamento.	

Tarea	
Número de tarea: 18	Número de HU: 7
Nombre de la tarea: Adicionar plan de trabajo individual.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 24/02/2014	Fecha fin: 26/02/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para adicionar un plan de trabajo individual haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanIndividual.	

Tarea	
Número de tarea: 19	Número de HU: 7
Nombre de la tarea: Mostrar plan de trabajo individual.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 26/02/2014	Fecha fin: 29/02/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para mostrar un plan de trabajo individual haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanIndividual.	

Tarea	
Número de tarea: 20	Número de HU: 7
Nombre de la tarea: Eliminar plan de trabajo individual.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 01/03/2014	Fecha fin: 03/03/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para eliminar un plan de trabajo individual haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanIndividual.	

Tarea	
Número de tarea: 21	Número de HU: 7
Nombre de la tarea: Agregar una actividad perteneciente al plan de trabajo de la facultad y al plan de trabajo del departamento en el plan de trabajo individual.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 03/03/2014	Fecha fin: 05/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para agregar una actividad perteneciente al plan de trabajo de la facultad y al plan de trabajo por departamento docente en el plan de trabajo individual haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanIndividual.	

Tarea	
Número de tarea: 22	Número de HU: 7
Nombre de la tarea: Quitar una actividad perteneciente al plan de trabajo de la facultad y al plan de trabajo del departamento del plan de trabajo individual.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 06/03/2014	Fecha fin:08/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para quitar una actividad perteneciente al plan de trabajo de la facultad y al plan de trabajo por departamento docente en el plan de trabajo individual haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanIndividual.	

Anexo V

Tarea	
Número de tarea: 24	Número de HU:9
Nombre de la tarea: Crear plan de resultados.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 12/03/2014	Fecha fin:13/03/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para crear un plan de resultados haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanResultados.	

Tarea	
Número de tarea: 25	Número de HU:9
Nombre de la tarea: Eliminar plan de resultados.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 14/03/2014	Fecha fin:15/03/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para eliminar un plan de resultados haciendo uso de un CRUD	

(Gestionar de Symfony 2) de la entidad PlanResultados.

Tarea	
Número de tarea: 26	Número de HU: 9
Nombre de la tarea: Mostrar plan de resultados.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 16/03/2014	Fecha fin: 17/03/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para mostrar un plan de resultados haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad PlanResultados.	

Tarea	
Número de tarea: 27	Número de HU: 10
Nombre de la tarea: Crear área.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 19/03/2014	Fecha fin: 20/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para crear un área haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Área.	

Tarea	
Número de tarea: 28	Número de HU: 10
Nombre de la tarea: Listar área.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 21/03/2014	Fecha fin: 22/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para listar área haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Área.	

Tarea	
Número de tarea: 29	Número de HU: 10
Nombre de la tarea: Modificar área.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 25/03/2014	Fecha fin: 26/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para modificar un área haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Área.	

Tarea	
Número de tarea: 30	Número de HU: 10
Nombre de la tarea: Eliminar área.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 27/03/2014	Fecha fin: 28/03/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para eliminar área haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Area.	

Tarea	
Número de tarea: 31	Número de HU: 11
Nombre de la tarea: Crear parámetros generales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 01/04/2014	Fecha fin: 02/04/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para crear parámetros generales haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad ParametroGeneral.	

Tarea

Número de tarea: 32	Número de HU: 11
Nombre de la tarea: Listar parámetros generales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 03/04/2014	Fecha fin: 04/04/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para listar parámetros generales haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad ParametroGeneral.	

Tarea	
Número de tarea: 33	Número de HU: 11
Nombre de la tarea: Modificar parámetros generales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 05/04/2014	Fecha fin: 06/04/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para modificar parámetros generales haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad ParametroGeneral.	

Tarea	
Número de tarea: 34	Número de HU: 11
Nombre de la tarea: Eliminar parámetros generales.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 07/04/2014	Fecha fin: 08/04/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad para eliminar parámetros generales haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad ParametroGeneral.	

Tarea	
Número de tarea: 35	Número de HU: 12

Nombre de la tarea: Crear parámetros.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 09/04/2014	Fecha fin: 10/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para crear parámetros haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Parametro.	

Tarea	
Número de tarea: 36	Número de HU: 12
Nombre de la tarea: Listar parámetros.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 12/04/2014	Fecha fin: 13/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para listar parámetros haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Parametro.	

Tarea	
Número de tarea: 37	Número de HU: 12
Nombre de la tarea: Modificar parámetros.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 15/04/2014	Fecha fin: 16/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para modificar parámetros haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Parametro.	

Tarea	
Número de tarea: 38	Número de HU: 12
Nombre de la tarea: Eliminar parámetros.	

Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 18/04/2014	Fecha fin: 19/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad para eliminar parámetros haciendo uso de un CRUD (Gestionar de Symfony 2) de la entidad Parametro.	

Tarea	
Número de tarea: 39	Número de HU: 13
Nombre de la tarea: Crear evaluación anual.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 20/04/2014	Fecha fin: 21/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad que permita crear una evaluación anual.	

Tarea	
Número de tarea: 40	Número de HU: 13
Nombre de la tarea: Crear evaluación trimestral.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 21/04/2014	Fecha fin: 22/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad que permita crear una evaluación trimestral.	

Tarea	
Número de tarea: 41	Número de HU: 13
Nombre de la tarea: Mostrar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 20/04/2014	Fecha fin: 21/04/2014

Programador responsable: Rosalia Suárez Núñez
Descripción: Implementar la funcionalidad que permita mostrar una evaluación

Tarea	
Número de tarea: 42	Número de HU: 13
Nombre de la tarea: Eliminar evaluación.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 20/04/2014	Fecha fin: 21/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad que permita asignar una evaluación anual a un profesor.	

Tarea	
Número de tarea: 43	Número de HU: 13
Nombre de la tarea: Asignar evaluación anual a los profesores.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 20/04/2014	Fecha fin: 21/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad que permita asignar una evaluación anual a un profesor.	

Tarea	
Número de tarea: 44	Número de HU: 13
Nombre de la tarea: Asignar evaluación trimestral a los profesores y especialistas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 22/04/2014	Fecha fin: 23/04/2014
Programador responsable: Rosalia Suárez Núñez	
Descripción: Implementar la funcionalidad que permita asignar una evaluación trimestral a un profesor o especialista.	

Anexo VI

Tarea	
Número de tarea: 46	Número de HU: 15
Nombre de la tarea: Exportar plan de trabajo.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 21/04/2014	Fecha fin: 22/04/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad que permite exportar un plan de trabajo haciendo uso de la librería ICS.	

Tarea	
Número de tarea: 47	Número de HU: 16
Nombre de la tarea: Crear reporte del cumplimiento de los planes de trabajo.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 25/04/2014	Fecha fin: 36/04/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad que permita crear el reporte del cumplimiento de los planes de trabajo, haciendo uso del CRUD (Gestionar de Symfony 2) de la entidad Reporte y de la librería chart.js para la visualización de los diagramas.	

Tarea	
Número de tarea: 48	Número de HU: 16
Nombre de la tarea: Mostrar reporte del cumplimiento de los planes de trabajo.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 01/05/2014	Fecha fin: 02/05/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	

Descripción: Implementar la funcionalidad que permita mostrar el reporte del cumplimiento de los planes de trabajo, haciendo uso del CRUD (Gestionar de Symfony 2) de la entidad Reporte.

Tarea	
Número de tarea: 49	Número de HU: 16
Nombre de la tarea: Eliminar reporte del cumplimiento de los planes de trabajo.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 02/05/2014	Fecha fin: 03/05/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad que permita eliminar el reporte del cumplimiento de los planes de trabajo, haciendo uso del CRUD (Gestionar de Symfony 2) de la entidad Reporte.	

Tarea	
Número de tarea: 50	Número de HU: 16
Nombre de la tarea: Modificar reporte del cumplimiento de los planes de trabajo.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 03/05/2014	Fecha fin: 04/05/2014
Programador responsable: Frank Ernesto Lorenzo Suárez	
Descripción: Implementar la funcionalidad que permita modificar el reporte del cumplimiento de los planes de trabajo, haciendo uso del CRUD (Gestionar de Symfony 2) de la entidad Reporte.	

Anexo VII

Caso de prueba de aceptación	
Código: P2-01	Historia de usuario: Gestionar departamento docente.
Nombre: Crear departamento.	
Descripción: Prueba para la funcionalidad que permite crear un departamento docente.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador. 	

<p>Pasos de ejecución:</p> <p>El administrador selecciona en el menú principal la opción “Departamento”, llena los datos en un formulario con el nombre del departamento (es único, no admite valor nulo y solo se pueden introducir letras que conformen una cadena de texto con menos de 50 caracteres), se presiona el botón “Crear”, luego se muestra un mensaje de confirmación: “El departamento ha sido creado” y se agrega a la lista de los departamentos existentes. En caso de una entrada de datos no válida se muestra un mensaje de error.</p>
<p>Resultado esperado: El departamento es creado satisfactoriamente.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Caso de prueba de aceptación	
Código: P2-02	Historia de usuario: Gestionar departamento docente.
Nombre: Modificar departamento.	
Descripción: Prueba para la funcionalidad que permite modificar un departamento docente.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un departamento en el sistema. 	
<p>Pasos de ejecución:</p> <p>El administrador selecciona en el menú principal la opción “Departamento”. De la lista de departamentos, se escoge el que se quiere modificar, se selecciona la opción “Editar” y se brinda la posibilidad de modificar el nombre del departamento (es único, no admite valor nulo y solo se pueden introducir letras que conformen una cadena de texto con menos de 50 caracteres). Luego se acciona en el botón “Aceptar” y se muestra el mensaje de confirmación: “El departamento ha sido modificado” y se muestra la lista de los departamentos con los cambios guardados. En caso de una entrada de datos no válida se muestra un mensaje de error.</p>	
Resultado esperado: El departamento es modificado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P2-03	Historia de usuario: Gestionar departamento docente.

Nombre: Eliminar departamento.
Descripción: Prueba para la funcionalidad que permite eliminar un departamento docente.
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un departamento en el sistema.
Pasos de ejecución: <p>El administrador selecciona en el menú principal la opción “Departamento” mostrándose un listado con los departamentos existentes en la aplicación hasta el momento, se selecciona el que se desea eliminar, se presiona el botón “Eliminar” y aparece un mensaje de confirmación: “El departamento ha sido eliminado”.</p>
Resultado esperado: El departamento es eliminado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P3-01	Historia de usuario: Gestionar actividad.
Nombre: Mostrar actividad	
Descripción: Prueba para la funcionalidad que permite mostrar una actividad.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador. ✓ Deben existir actividades creadas. 	
Pasos de ejecución: <p>Las actividades se muestran conjuntamente con los planes de trabajo.</p>	
Resultado esperado: Las actividades se muestran satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P3-02	Historia de usuario: Gestionar actividad.
Nombre: Editar actividad	
Descripción: Prueba para la funcionalidad que permite editar una actividad.	

<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador. ✓ Deben existir actividades creadas.
<p>Pasos de ejecución:</p> <p>Se selecciona la opción “Plan de trabajo” del menú principal. Al mostrar un plan de trabajo se escoge la actividad que se desea modificar, se accede a la opción “Editar” y aparece un formulario con los datos de la actividad, los mismos son modificados y se muestra un mensaje de confirmación: “La actividad ha sido modificada”. Cuando la actividad pertenece a un plan de trabajo de la facultad o a un plan de trabajo por departamento docente, los datos a editar son los siguientes: fecha (se selecciona la fecha y la hora de la actividad, no admite valor nulo y en caso de existir una actividad programada para esa hora se muestra un mensaje de error), descripción (no admite valor nulo, admite una cadena de texto con 200 caracteres como tamaño máximo), lugar (no debe ser nulo, admite una cadena de texto formada por números y letras con un tamaño máximo de 50 caracteres), prioridad (se selecciona la prioridad que puede ser alta, media o baja y no admite valor nulo), quién dirige (puede tener valor nulo y admite solamente letras), quién participa (puede tener valor nulo y admite solamente letras) y valoración (puede tener valor nulo y admite una cadena de texto con un tamaño máximo de 200 caracteres). Cuando la actividad pertenece a un plan de trabajo individual se pueden editar los siguientes datos: fecha, descripción, lugar y prioridad. En caso de una entrada de datos no válida se muestra un mensaje de error.</p>
<p>Resultado esperado: La actividad es editada satisfactoriamente.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Caso de prueba de aceptación	
Código: P3-03	Historia de usuario: Gestionar actividad.
Nombre: Eliminar actividad	
Descripción: Prueba para la funcionalidad que permite eliminar una actividad.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador. ✓ Deben existir actividades creadas. 	
<p>Se selecciona la opción “Plan de trabajo” del menú principal. Al mostrar un plan de trabajo se escoge la actividad que se desea eliminar, se accede a la opción “Eliminar” y aparece mensaje de confirmación: “La actividad ha sido eliminada”.</p>	

Resultado esperado: Los datos son gestionados satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Anexo VIII

Caso de prueba de aceptación	
Código: P5-01	Historia de usuario: Gestionar plan de trabajo de la facultad.
Nombre: Adicionar plan de trabajo de la facultad.	
Descripción: Prueba para la funcionalidad que permite gestionar los planes de trabajo de la facultad.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de asesor. 	
Pasos de ejecución:	
<p>Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de facultad”. Para agregar un plan, la aplicación muestra un formulario el cual permite seleccionar la fecha, compuesta por el mes (no admite valor nulo) y el año (no admite valor nulo), se presiona el botón “Crear” y se muestra un mensaje de confirmación: “El plan ha sido creado”.</p>	
Resultado esperado: El plan es adicionado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P5-02	Historia de usuario: Gestionar plan de trabajo de la facultad.
Nombre: Mostrar plan de trabajo de la facultad.	
Descripción: Prueba para la funcionalidad que permite gestionar los planes de trabajo de la facultad.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un plan de trabajo. 	
Pasos de ejecución:	
<p>Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de facultad”. La aplicación muestra un listado con todos los planes de trabajo de la facultad existentes, el usuario escoge el plan</p>	

que desea, se selecciona la opción “Mostrar” y se muestran todas las actividades pertenecientes a ese plan.
Resultado esperado: El plan es mostrado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P5-03	Historia de usuario: Gestionar plan de trabajo de la facultad.
Nombre: Eliminar plan de trabajo de la facultad.	
Descripción: Prueba para la funcionalidad que permite gestionar los planes de trabajo de la facultad.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de asesor. ✓ Debe existir al menos un plan de trabajo. 	
Pasos de ejecución:	
Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de facultad”. La aplicación muestra un listado con todos los planes de trabajo de la facultad existentes, se selecciona el que va a ser eliminado y se selecciona la opción “Eliminar”. Luego aparece un mensaje: ¿Desea eliminar? y las opciones Aceptar y Cancelar. De escoger Aceptar, se elimina el plan de trabajo de la facultad y se muestra un mensaje de confirmación: “El plan ha sido eliminado” y se muestra un listado con el resto de los planes. De lo contrario se muestra la misma vista.	
Resultado esperado: El plan es eliminado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P6-01	Historia de usuario: Gestionar plan de trabajo por departamentos docentes.
Nombre: Adicionar plan de trabajo por departamentos docentes.	
Descripción: Prueba para la funcionalidad que permite adicionar los planes de trabajo por departamentos docentes.	
Condiciones de ejecución:	

✓ El usuario debe estar autenticado con el rol de jefe de departamento.
Pasos de ejecución: Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de departamento”. Para agregar un plan la aplicación muestra un formulario el cual permite seleccionar la fecha, compuesta por el mes (no admite valor nulo) y el año (no admite valor nulo), se presiona el botón “Crear” y se muestra un mensaje de confirmación: “El plan ha sido creado”.
Resultado esperado: El plan es adicionado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P6-02	Historia de usuario: Gestionar plan de trabajo por departamentos docentes.
Nombre: Mostrar plan de trabajo por departamentos docentes.	
Descripción: Prueba para la funcionalidad que permite mostrar los planes de trabajo por departamentos docentes.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe pertenecer al departamento del que se desea mostrar el plan. ✓ Debe existir al menos un plan de trabajo. 	
Pasos de ejecución: Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de departamento”. La aplicación muestra un listado de los planes de trabajo del departamento docente al que pertenece el usuario. Luego se escoge el plan de trabajo que se desea mostrar.	
Resultado esperado: El plan es mostrado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P6-03	Historia de usuario: Gestionar plan de trabajo por departamentos docentes.
Nombre: Eliminar plan de trabajo por departamentos docentes.	
Descripción: Prueba para la funcionalidad que permite eliminar los planes de trabajo por departamentos	

docentes.
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de jefe de departamento. ✓ Debe existir al menos un plan de trabajo.
Pasos de ejecución: <p>Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de departamento”. La aplicación muestra un listado con todos los planes de trabajo por departamento docente existentes, se selecciona el que va a ser eliminado y se selecciona la opción “Eliminar”. Luego aparece un mensaje: ¿Desea eliminar? y las opciones Aceptar y Cancelar. De escoger Aceptar, se elimina el plan de trabajo del departamento y muestra un mensaje de confirmación: “El plan ha sido eliminado” y se muestra un listado con el resto de los planes. De lo contrario se muestra la misma vista.</p>
Resultado esperado: El plan es eliminado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P6-04	Historia de usuario: Gestionar plan de trabajo por departamentos docentes.
Nombre: Agregar actividad perteneciente al plan de trabajo de la facultad en el plan de trabajo por departamento docente.	
Descripción: Prueba para la funcionalidad que permite adicionar actividad a los planes de trabajo por departamentos docentes.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de jefe de departamento. 	
Pasos de ejecución: <p>Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de departamento”. Se escoge el plan de trabajo deseado y este es mostrado conjuntamente con sus actividades. Se selecciona la opción “Actividad facultad” y se muestran las actividades pertenecientes al plan de la facultad de ese mes, se seleccionan las que van a ser agregadas al plan del departamento y se presiona el botón “Agregar actividades marcadas” y se muestra un mensaje de confirmación: “Las actividades han sido agregadas”.</p>	
Resultado esperado: La actividad es agregada satisfactoriamente.	

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: P6-05

Historia de usuario: Gestionar plan de trabajo por departamentos docentes.

Nombre: Quitar actividad perteneciente al plan de trabajo de la facultad del plan de trabajo por departamento docente.

Descripción: Prueba para la funcionalidad que permite quitar una actividad de los planes de trabajo por departamentos docentes.

Condiciones de ejecución:

- ✓ El usuario debe estar autenticado con el rol de jefe de departamento.
- ✓ Debe existir al menos una actividad del plan de trabajo de la facultad.

Pasos de ejecución:

Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de departamento”. Se escoge el plan de trabajo deseado y este es mostrado conjuntamente con sus actividades. Se selecciona la actividad de la facultad que se desea quitar del plan de trabajo del departamento y se accede a la opción “Quitar”. Se muestra un mensaje de confirmación: “La actividad ha sido quitada”.

Resultado esperado: La actividad es quitada satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: P7-01

Historia de usuario: Gestionar plan de trabajo individual.

Nombre: Adicionar plan de trabajo individual.

Descripción: Prueba para la funcionalidad que permite adicionar los planes de trabajo individuales de cada profesor y especialista.

Condiciones de ejecución:

- ✓ El usuario no debe estar autenticado con el rol de administrador.

Pasos de ejecución:

Se selecciona la opción "Plan de trabajo" del menú principal y se accede a "Plan de trabajo individual". Para agregar un plan la aplicación muestra un formulario el cual permite seleccionar la fecha, compuesta por el mes (no admite valor nulo) y el año (no admite valor nulo), se presiona el botón "Crear" y se muestra un mensaje de confirmación: "El plan ha sido creado".
Resultado esperado: El plan es adicionado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P7-02	Historia de usuario: Gestionar plan de trabajo individual.
Nombre: Mostrar plan de trabajo individual.	
Descripción: Prueba para la funcionalidad que permite mostrar los planes de trabajo individuales de cada profesor y especialista.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un plan de trabajo. 	
Pasos de ejecución: <p>Se selecciona la opción "Plan de trabajo" del menú principal y se accede a "Plan de trabajo individual". La aplicación muestra un listado con todos los planes de trabajo individuales existentes, el usuario selecciona el plan deseado, se selecciona la opción mostrar y se muestran todas las actividades pertenecientes a ese plan.</p>	
Resultado esperado: El plan es mostrado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P7-03	Historia de usuario: Gestionar plan de trabajo individual.
Nombre: Eliminar plan de trabajo individual.	
Descripción: Prueba para la funcionalidad que permite eliminar los planes de trabajo individuales de cada profesor y especialista.	

<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un plan de trabajo.
<p>Pasos de ejecución:</p> <p>Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo individual”. La aplicación muestra un listado con todos los planes de trabajo individuales existentes, se selecciona el que va a ser eliminado y se selecciona la opción “Eliminar”. Luego aparece un mensaje: ¿Desea eliminar? y las opciones Aceptar y Cancelar. De escoger Aceptar, se elimina el plan de trabajo individual y muestra un mensaje de confirmación: “El plan ha sido eliminado” y se muestra un listado con el resto de los planes. De lo contrario se muestra la misma vista.</p>
<p>Resultado esperado: El plan es eliminado satisfactoriamente.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Caso de prueba de aceptación	
Código: P7-04	Historia de usuario: Gestionar plan de trabajo individual.
Nombre: Adicionar actividad perteneciente al plan de trabajo de la facultad y al plan de trabajo por departamento docente en el plan de trabajo individual.	
Descripción: Prueba para la funcionalidad que permite adicionar una actividad a los planes de trabajo individuales de cada profesor y especialista.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador. ✓ Debe existir al menos una actividad del plan de trabajo de la facultad o del plan de trabajo por departamento docente. 	
<p>Pasos de ejecución:</p> <p>Se selecciona la opción “Plan de trabajo” del menú principal y se accede a “Plan de trabajo de individual”. Se escoge el plan de trabajo deseado y este es mostrado conjuntamente con sus actividades. Se selecciona la opción “Actividad facultad” y se muestran las actividades pertenecientes al plan de la facultad de ese mes, se seleccionan las que van a ser agregadas al plan individual y se presiona el botón “Agregar actividades marcadas” mostrándose el mensaje de confirmación: “Las actividades han sido agregadas”. Luego se selecciona la opción “Actividad del departamento” y se muestran las actividades pertenecientes al plan del</p>	

departamento, se escogen las que van a ser agregadas al plan individual y se presiona el botón "Agregar actividades marcadas", se muestra un mensaje de confirmación: "Las actividades han sido agregadas".
Resultado esperado: La actividad es adicionada satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P7-05	Historia de usuario: Gestionar plan de trabajo individual.
Nombre: Quitar actividad perteneciente al plan de trabajo de la facultad y al plan de trabajo por departamento docente del plan de trabajo individual.	
Descripción: Prueba para la funcionalidad que permite quitar una actividad de los planes de trabajo individuales de cada profesor y especialista.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador. ✓ Debe existir al menos una actividad del plan de trabajo de la facultad o del plan de trabajo por departamento docente. 	
Pasos de ejecución:	
Se selecciona la actividad perteneciente al plan de trabajo de la facultad, así como al plan de trabajo del departamento y se selecciona la opción "Quitar". Se muestra el mensaje de confirmación: "La actividad ha sido quitada".	
Resultado esperado: La actividad es quitada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Anexo IX

Caso de prueba de aceptación	
Código: P9-01	Historia de usuario: Gestionar plan de resultados.
Nombre: Crear plan de resultados.	
Descripción: Prueba para la funcionalidad que permite crear los planes de resultados para otorgar la evaluación de los profesores y especialistas.	
Condiciones de ejecución:	

✓ El usuario no debe estar autenticado con el rol de administrador.
Pasos de ejecución: Se selecciona la opción "Evaluación" del menú principal y se escoge la opción "Plan de resultados". Luego se elige el curso (no admite valor nulo) en que se va a realizar dicho plan y se acciona en el botón "Crear". Si ya el plan existe en esa fecha el sistema muestra un mensaje: "El curso seleccionado tiene un plan de resultados" y se muestra la vista anterior.
Resultado esperado: El plan de resultados es creado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P9-02	Historia de usuario: Gestionar plan de resultados.
Nombre: Mostrar plan de resultados.	
Descripción: Prueba para la funcionalidad que permite mostrar los planes de resultados para otorgar la evaluación de los profesores y especialistas.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ Debe existir al menos un plan de resultados. ✓ El usuario no debe estar autenticado con el rol de administrador. 	
Pasos de ejecución: Se selecciona la opción "Evaluación" del menú principal y se escoge la opción "Plan de resultados". Se elige de la lista de planes de resultados el que se desea mostrar y se presiona el botón "Mostrar".	
Resultado esperado: El plan de resultados es mostrado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P9-03	Historia de usuario: Gestionar plan de resultados.
Nombre: Eliminar plan de resultados.	
Descripción: Prueba para la funcionalidad que permite eliminar los planes de resultados para otorgar la evaluación de los profesores y especialistas.	

Condiciones de ejecución: <ul style="list-style-type: none"> ✓ Debe existir al menos un plan de resultados. ✓ El usuario no debe estar autenticado con el rol de administrador.
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se escoge la opción "Plan de resultados". Se elige de la lista de planes de resultados el que se desea eliminar y se presiona el botón "Eliminar".</p>
Resultado esperado: El plan de resultados es eliminado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P10-01	Historia de usuario: Gestionar área.
Nombre: Crear área.	
Descripción: Prueba para la funcionalidad que permite crear las áreas de resultados claves de los planes de resultados para otorgar una evaluación a los profesores y especialistas.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador. 	
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se elige la opción "Área". Luego se escribe el nombre (no puede ser nulo, admite solamente letras y un tamaño máximo de 100 caracteres) y el número del área (no puede ser nulo, admite solamente números entre 0 y 20 donde los números de las áreas son únicos), se acciona en el botón "Crear" y se muestra el mensaje de confirmación "El área ha sido creada". En caso de una entrada de datos no válida se muestra un mensaje de error.</p>	
Resultado esperado: El área es creada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P10-02	Historia de usuario: Gestionar área.
Nombre: Modificar área.	
Descripción: Prueba para la funcionalidad que permite modificar las áreas de resultados claves de los planes	

de resultados para otorgar una evaluación a los profesores y especialistas.
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un área.
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se elige la opción "Área". Se escoge el área que se desea modificar y se escoge la opción "Modificar". El administrador puede modificar el nombre (no puede ser nulo, admite solamente letras y un tamaño máximo de 100 caracteres) y el número del área (no puede ser nulo, admite solamente números entre 0 y 20 donde los números de las áreas son únicos). En caso de una entrada de datos no válida se muestra un mensaje de error.</p>
Resultado esperado: El área es modificada satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P10-03	Historia de usuario: Gestionar área.
Nombre: Eliminar área.	
Descripción: Prueba para la funcionalidad que permite eliminar las áreas de resultados claves de los planes de resultados para otorgar una evaluación a los profesores y especialistas.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un área. 	
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se elige la opción "Área". Se escoge el área que se desea eliminar y se selecciona la opción "Eliminar".</p>	
Resultado esperado: El área es eliminada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P11-01	Historia de usuario: Gestionar parámetros generales.

Nombre: Crear parámetros generales.
Descripción: Prueba para la funcionalidad que permite crear los parámetros generales de los planes de resultados para otorgar una evaluación a los profesores y especialistas.
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador.
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se elige la opción "Parámetros generales". Luego se escribe el nombre (no puede ser nulo, admite letras solamente y un tamaño máximo de 100 caracteres) y se selecciona el número del área a la que va a pertenecer el parámetro general, se acciona en el botón "Crear" y se muestra el mensaje de confirmación "El parámetro ha sido creado." y se incluye al área seleccionada. En caso de una entrada de datos no válida se muestra un mensaje de error.</p>
Resultado esperado: El parámetro general es creado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P11-02	Historia de usuario: Gestionar parámetros generales.
Nombre: Modificar parámetros generales.	
Descripción: Prueba para la funcionalidad que permite modificar los parámetros generales de los planes de resultados para otorgar una evaluación a los profesores y especialistas.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un parámetro general. 	
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se elige la opción "Parámetros generales". Se elige el parámetro general que se desea modificar y se brinda la posibilidad de editar el nombre (no puede ser nulo, admite letras solamente y un tamaño máximo de 100 caracteres) y de cambiar el área a la que pertenece. Por último, se acciona sobre el botón "Aceptar", se muestra el mensaje: "El parámetro ha sido editado" y se guardan los cambios realizados. En caso de una entrada de datos no válida se muestra un mensaje de error.</p>	
Resultado esperado: El parámetro general es modificado satisfactoriamente.	

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P11-03	Historia de usuario: Gestionar parámetros generales.
Nombre: Eliminar parámetros generales.	
Descripción: Prueba para la funcionalidad que permite eliminar los parámetros generales de los planes de resultados para otorgar una evaluación a los profesores y especialistas.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol de administrador. ✓ Debe existir al menos un parámetro general. 	
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se elige la opción "Parámetros generales". Se elige el parámetro general que se desea eliminar y se escoge la opción "Eliminar".</p>	
Resultado esperado: El parámetro general es eliminado satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P12-01	Historia de usuario: Gestionar parámetros.
Nombre: Crear parámetros.	
Descripción: Prueba para la funcionalidad que permite crear los parámetros de los planes de resultados para otorgar una evaluación a los profesores y especialistas.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador o decano. 	
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se escoge "Plan de resultados". Se presiona sobre el botón "Mostrar" del plan de resultados deseado y se escoge la opción "Agregar parámetro". Luego se escribe el nombre (no puede ser nulo, admite solamente letras y un tamaño máximo de 500 caracteres), fecha (no puede ser nulo, admite números y letras y un tamaño máximo de 50 caracteres) del parámetro y se selecciona el parámetro general (no puede ser nulo) en el que va a ser incluido, se acciona en el botón "Crear" y se muestra</p>	

el mensaje de confirmación "El parámetro ha sido creado". Si se acciona en el botón "Cancelar" se muestra la misma vista. En caso de una entrada de datos no válida se muestra un mensaje de error.

Resultado esperado: El parámetro es creado satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: P12-02

Historia de usuario: Gestionar parámetros.

Nombre: Modificar parámetros.

Descripción: Prueba para la funcionalidad que permite modificar los parámetros de los planes de resultados para otorgar una evaluación a los profesores y especialistas.

Condiciones de ejecución:

- ✓ Debe existir al menos un plan de resultados.
- ✓ El usuario no debe estar autenticado con el rol de administrador o decano.

Pasos de ejecución:

Se selecciona la opción "Evaluación" del menú principal y se escoge "Plan de resultados". Se presiona sobre el botón "Mostrar" del plan de resultados deseado, se selecciona la opción "Lista de parámetros del plan" y se escoge la opción "Editar". El usuario tiene la posibilidad de editar el nombre del parámetro (no puede ser nulo, admite solamente letras y un tamaño máximo de 500 caracteres), la fecha (no puede ser nulo, admite números y letras y un tamaño máximo de 50 caracteres) del parámetro y cambiar el parámetro general en el que se va a incluir (no puede ser nulo). El jefe de departamento y el decano tienen la posibilidad de editar los siguientes datos: estado (no puede ser nulo y admite los valores "Cumplido" o "Incumplido") y escribir una observación (admite valor nulo y es una cadena de texto con tamaño máximo de 200 caracteres). Por último se acciona sobre el botón "Aceptar", se muestra el mensaje: "El parámetro ha sido editado" y se guardan los cambios realizados. Si se acciona en el botón "Cancelar" se muestra la misma vista. En caso de una entrada de datos no válida se muestra un mensaje de error.

Resultado esperado: El parámetro es modificado satisfactoriamente.

Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación

Código: P12-03

Historia de usuario: Gestionar parámetros.

Nombre: Eliminar parámetros.
Descripción: Prueba para la funcionalidad que permite eliminar los parámetros de los planes de resultados para otorgar una evaluación a los profesores y especialistas.
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ Debe existir al menos un plan de resultados. ✓ El usuario no debe estar autenticado con el rol de administrador o decano.
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se escoge "Plan de resultados". Se presiona sobre el botón "Mostrar" del plan de resultados deseado, se selecciona la opción "Lista de parámetros del plan", se escoge la opción "Eliminar" y el parámetro se elimina de la lista.</p>
Resultado esperado: El parámetro es eliminado satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P13-01	Historia de usuario: Asignar evaluación anual.
Nombre: Crear evaluación anual.	
Descripción: Prueba para la funcionalidad que permite crear una evaluación anual.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador o decano. 	
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se escoge la opción "Evaluación anual". Luego se selecciona el curso (no admite valor nulo) en que se va a realizar dicha evaluación y se acciona en el botón "Crear". Si en ese curso el profesor ya fue evaluado, el sistema muestra un mensaje: "Su evaluación para el curso seleccionado ya existe" y se muestra la vista anterior.</p>	
Resultado esperado: La evaluación anual es creada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación

Código: P13-02	Historia de usuario: Asignar evaluación anual.
Nombre: Mostrar evaluación anual.	
Descripción: Prueba para la funcionalidad que permite mostrar una evaluación anual.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador o decano. 	
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se escoge la opción "Evaluación anual". Luego se selecciona de la lista de evaluaciones anuales, la que se desea mostrar y se acciona el botón "Mostrar".</p>	
Resultado esperado: La evaluación anual es mostrada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P13-03	Historia de usuario: Asignar evaluación anual.
Nombre: Eliminar evaluación anual.	
Descripción: Prueba para la funcionalidad que permite eliminar una evaluación anual.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador o decano. 	
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se escoge la opción "Evaluación anual". Luego se selecciona de la lista de evaluaciones anuales, la que se desea eliminar y se presiona el botón "Eliminar". El sistema muestra el mensaje "¿Desea eliminar?".</p>	
Resultado esperado: La evaluación anual es eliminada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P13-04	Historia de usuario: Asignar evaluación anual.
Nombre: Otorgar evaluación anual.	

Descripción: Prueba para la funcionalidad que permite otorgar una evaluación anual.
Condiciones de ejecución: ✓ El usuario debe estar autenticado con el rol jefe de departamento o decano.
Pasos de ejecución: Se selecciona la opción "Usuarios" del menú principal y se escoge la opción "Controlar evaluación anual". Se selecciona el usuario a evaluar y luego de mostrada la evaluación anual se accede a la opción "Asignar evaluación". Se brinda la posibilidad de otorgar una evaluación (Excelente, Bien, Regular y Mal) y redactar una recomendación.
Resultado esperado: La evaluación anual es otorgada satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P14-01	Historia de usuario: Asignar evaluación trimestral.
Nombre: Crear evaluación trimestral.	
Descripción: Prueba para la funcionalidad que permite crear una evaluación trimestral.	
Condiciones de ejecución: ✓ El usuario debe estar autenticado con el rol jefe de departamento o decano.	
Pasos de ejecución: Se selecciona la opción "Usuarios" del menú principal y se escoge la opción "Controlar evaluación trimestral". Se selecciona el usuario a evaluar y se muestran las evaluaciones trimestrales que este posee. Para crear una nueva evaluación, se elige el trimestre (no admite valor nulo) en que se va a realizar dicha evaluación y se acciona en el botón "Crear". Si existe la evaluación de ese trimestre, el sistema muestra un mensaje: "El trimestre seleccionado tiene una evaluación" y se muestra la vista anterior.	
Resultado esperado: La evaluación trimestral es creada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P14-02	Historia de usuario: Asignar evaluación trimestral.

Nombre: Mostrar evaluación trimestral.
Descripción: Prueba para la funcionalidad que permite mostrar una evaluación trimestral.
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario no debe estar autenticado con el rol de administrador o decano.
Pasos de ejecución: <p>Se selecciona la opción "Evaluación" del menú principal y se escoge la opción "Evaluación trimestral". Luego se selecciona de la lista de evaluaciones trimestrales, la que se desea mostrar y acciona el botón "Mostrar".</p>
Resultado esperado: La evaluación trimestral es mostrada satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Caso de prueba de aceptación	
Código: P14-03	Historia de usuario: Asignar evaluación trimestral.
Nombre: Eliminar evaluación trimestral.	
Descripción: Prueba para la funcionalidad que permite eliminar una evaluación trimestral.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El usuario debe estar autenticado con el rol jefe de departamento o decano. 	
Pasos de ejecución: <p>Se selecciona la opción "Usuarios" del menú principal y se escoge la opción "Controlar evaluación trimestral". Se selecciona el usuario a evaluar. Luego se escoge de la lista de evaluaciones trimestrales del usuario, la que se desea eliminar y se presiona el botón "Eliminar". El sistema muestra el mensaje "¿Desea eliminar?".</p>	
Resultado esperado: La evaluación trimestral es eliminada satisfactoriamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P14-04	Historia de usuario: Asignar evaluación trimestral.
Nombre: Otorgar evaluación trimestral a los profesores y especialistas.	
Descripción: Prueba para la funcionalidad que permite asignar una evaluación trimestral a los especialistas.	

Condiciones de ejecución: El usuario debe estar autenticado con el rol jefe de departamento o decano.
Pasos de ejecución: Se selecciona la opción "Usuarios" del menú principal y se escoge la opción "Controlar evaluación anual". Se selecciona el usuario a evaluar y luego de mostrada la evaluación anual se accede a la opción "Asignar evaluación trimestral". Se brinda la posibilidad de otorgar una evaluación (Superior, Adecuado y Deficiente).
Resultado esperado: La evaluación trimestral es otorgada satisfactoriamente.
Evaluación de la prueba: Prueba satisfactoria.

Anexo X

Caso de prueba de aceptación	
Código: P15-01	Historia de usuario: Importar plan de trabajo
Nombre: Importar	
Descripción: Prueba para la funcionalidad que permite importar archivos ics.	
Condiciones de ejecución: <ul style="list-style-type: none"> ✓ El archivo tiene que estar en el formato ".ics". ✓ El usuario no debe estar autenticado con el rol de administrador. 	
Pasos de ejecución: Del menú principal se escoge la opción "Archivo" y luego se selecciona "Mis archivos". Se escribe el nombre del mismo (no puede ser nulo, admite números y letras y un tamaño máximo de 50 caracteres), se selecciona el tipo de plan de trabajo que se va a crear con el archivo (no puede ser nulo) y se selecciona la opción "archivo" para subir el mismo (no puede ser nulo, solo admite archivos en formato ".ics" y el archivo seleccionado debe tener 4 <i>mega bytes</i> como tamaño máximo). Se acciona en el botón "Subir".	
Resultado esperado: El archivo es importado satisfactoriamente	
Evaluación de la prueba: Prueba satisfactoria.	

Caso de prueba de aceptación	
Código: P16-01	Historia de usuario: Exportar plan de trabajo
Nombre: Exportar	
Descripción: Prueba para la funcionalidad que permite exportar los planes de trabajo.	

Condiciones de ejecución:

- ✓ Debe existir al menos un plan de trabajo con actividades para que pueda ser exportado.
- ✓ El usuario no debe estar autenticado con el rol de administrador.

Pasos de ejecución:

Se selecciona la opción "Plan de trabajo" del menú principal. Cuando se muestra un plan de trabajo, se acciona en el botón "Exportar a iCalendar" y se muestra el mensaje "El plan ha sido exportado".

Resultado esperado: El archivo es exportado satisfactoriamente

Evaluación de la prueba: Prueba satisfactoria.

Glosario de términos

Artefacto: Un artefacto es un producto tangible resultante del proceso de desarrollo de *software*.

Bootstrap: Framework CSS

CSS: Las hojas de estilo en cascada (en inglés *Cascading Style Sheets*) contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una Web. Se llama “cascada” porque una hoja puede heredar los formatos definidos en otra hoja de tal forma que no hace falta volver a definirlos. Estas hojas permiten la separación entre el contenido y la presentación de un Sitio Web.

Framework: Es una estructura de soporte definida en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros *software*, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de *software* que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

jQuery: Framework para el lenguaje JavaScript.

Software libre: *Software* que puede ser distribuido, modificado, redistribuido, copiado y usado libremente. Se basa en cuatro libertades: libertad para usarlo con cualquier propósito, libertad para modificarlo a las necesidades, libertad para distribuir copias y para mejorarlo. Que un *software* sea libre no quiere decir que sea gratuito, error que viene de la traducción *Free Software*.

Unix: Sistema operativo comercializado en la década de los 70 que alcanzó mucho éxito, sobre todo en las universidades y posteriormente en las empresas. Entre sus principales características tenemos que es: portable, robusto y flexible. Actualmente goza de gran popularidad dentro de la tecnología de Internet.

Web: Es un sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (el Web, un Web) se refiere a un sitio Web entero, en cambio si se utiliza en femenino (la Web, una Web) se refiere a una página Web concreta dentro del sitio Web.

WWW (World Wide Web): O simplemente Web, es el universo de información accesible a través de Internet, una fuente inagotable del conocimiento humano.