

Universidad de las Ciencias Informáticas

Facultad 2



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

**Aplicación web para la planificación y
control del cumplimiento del trabajo
metodológico en el departamento de
Ingeniería de software de la facultad 2**

Autoras: Angela María Lescaille Otaño

Dianelis Ferrer Maestrey

Tutora: MSc. Eliana Bárbara Ril Valentin

Co Tutora: Lic. Lianne Guillen Pérez

Consultante: Yosamy David Bermúdez Pérez

La Habana, junio de 2017
“Año 59 de la Revolución”

Declaración de autoría

Declaramos ser las únicas autoras del trabajo de diploma que tiene como título: aplicación web para la planificación y control del cumplimiento del trabajo metodológico en el departamento de Ingeniería de software de la facultad 2 y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Para que así conste firmamos la presente el 5 de julio del año 2017.

Angela María Lescaille Otaño

Dianelis Ferrer Maestrey

Firma del autor

Firma del autor

MSc. Eliana Bárbara Ril Valentín

Lic. Lianne Guillén Pérez

Firma del tutor

Firma del co-tutor

Ing. Yosamy David Bermúdez Pérez

Firma del consultante

Datos del contacto

Tutora

MSc. Eliana Bárbara Ril Valentin

Centro Laboral: Universidad de las Ciencias Informáticas (UCI)

Formación Académica: Ingeniería en Ciencias Informáticas, Máster en Gestión de Proyectos Informáticos

Correo Electrónico: ebril@uci.cu

Co-Tutora

Lic. Lianne Guillen Pérez

Centro Laboral: Universidad de las Ciencias Informáticas

Formación Académica: Licenciada en Ciencias de la Computación

Correo Electrónico: lianne@uci.cu

Consultante

Ing. Yosamy David Bermúdez Pérez

Centro Laboral: Universidad de las Ciencias Informáticas

Formación Académica: Ingeniería en Ciencias Informáticas

Correo Electrónico: ydbermudez@uci.cu

Autoras

Angela María Lescaille Otaño

Correo Electrónico: amlescaille@estudiantes.uci.cu

Dianelis Ferrer Maestrey

Correo Electrónico: dmaestrey@estudiantes.uci.cu

Dedicatoria

A mi abuela, porque hubiese querido haberme visto graduada.

Angela Maria Lescaille Otaño

A mi madre por ser mi única amiga, a mis hermanos para que vean en mi un ejemplo a seguir.

Dianelis Ferrer Maestrey

Agradecimientos

No debo hacer una lista de agradecimientos porque sería extensa y las omisiones suelen doler. Quiero no obstante señalar especialmente:

A mi madre, mis abuelas y mi padre, por estar siempre, aún lejos.

Sin ellos nunca hubiera llegado hasta aquí.

A mi tutora Baby, por su orientación, ánimo y confianza, por hacerme creer que era posible la realización de un trabajo como este.

Angela María Lescaille Otaño

A mi Gorda, por creer y confiar en mí.

A mis padres por su apoyo incondicional.

A mis abuelas y abuelo, a mis tías, primos y primas en fin a toda mi familia.

A mi novio Andy por ser más que un novio, un amigo en este poco tiempo que llevamos de relación.

A mis suegros, en especial a mi suegra por verme como su hija.

A mi tutora Baby por no matarme.

A mi tutor y amigo Samy por ayudarme siempre que lo necesité.

En fin, a todos los presentes.

Dianelis Ferrer Maestrey

Resumen

En la actualidad, la gestión de la información en el departamento de Ingeniería de software de la facultad 2 afronta nuevas exigencias, lo que hace que dicha gestión esté en constante cambio y transformación. A partir del desarrollo de las Tecnologías de la Información y las Comunicaciones, la Universidad de las Ciencias Informáticas enfocó sus esfuerzos a encontrar nuevas maneras de perfeccionar la planificación y control del cumplimiento del trabajo metodológico de sus profesores, donde la elaboración de sistemas informáticos que contribuyan a llevar la gestión de la información del proceso docente educativo juegue un papel importante. Como parte del perfeccionamiento de la gestión de este proceso, se desarrolló una aplicación web para la planificación y control del cumplimiento del trabajo metodológico en el departamento de Ingeniería de software de la facultad 2, que sirvió de apoyo a la toma de decisiones en el proceso docente educativo. Esta aplicación web posee un aporte práctico, pues brinda las funcionalidades de planificar el plan de control a clase, el plan de trabajo metodológico y el plan de clase para los profesores de dicho departamento. Se desarrolló utilizando como framework de desarrollo Django en su versión 1.8.5 y Bootstrap 3 como framework de desarrollo para el uso del diseño web; PyCharm 4.5 como entorno de desarrollo integrado; SQLite Manager 0.8.3.1 para la administración de la base de datos y como lenguajes Python 2.7.11, HTML 5, CSS 3 y JavaScript. Se documentó el proceso de desarrollo siguiendo la metodología XP.

Palabras claves: control, departamento, planificación, trabajo metodológico.

Índice de contenidos

Introducción	1
Capítulo 1: Fundamentación teórica	6
1.1. Proceso de planificación y control del cumplimiento del trabajo metodológico.....	6
1.1.1 Plan de trabajo metodológico.....	6
1.1.2 Categorías de la didáctica	7
1.1.3 Tipos de clase.....	8
1.1.4 Indicadores de control.....	10
1.1.5 Conceptos fundamentales asociados al tema.....	11
1.2. Aplicaciones web para la planificación y control del cumplimiento del trabajo metodológico	13
1.2.1 Aplicaciones web internacionales	13
1.2.2 Aplicaciones web nacionales	14
1.2.3 Conclusiones del estado del arte	18
1.3. Metodologías, herramientas, lenguajes y tecnologías	19
1.3.1 Metodología de desarrollo	19
1.3.2 Tecnologías y herramientas de desarrollo	20
1.4. Conclusiones del capítulo.....	23
Capítulo 2: Exploración y planificación de la aplicación web propuesta	24
2.1. Descripción del flujo actual de los procesos	24
2.2. Funcionalidades	25
2.3. Propiedades de la aplicación web propuesta.....	27
2.4. Propuesta de la aplicación.....	27
2.5. Historias de usuario.....	29
2.6. Estimación de esfuerzo	39

2.7.	Plan de iteraciones	41
2.8.	Plan de entrega	43
2.9.	Conclusiones del capítulo.....	43
Capítulo 3: Producción y validación de la aplicación web propuesta.....		45
3.1	Patrón arquitectónico.....	45
3.2	Patrones de diseño.....	46
3.3	Diseño de la base de datos	49
3.4	Tareas de ingeniería.....	50
3.5	Clase-Responsabilidad-Creador.....	55
3.6	Pruebas.....	57
3.6.1	Pruebas unitarias.....	57
3.6.2	Pruebas de aceptación	59
3.7	Conclusiones del capítulo.....	66
Conclusiones generales.....		67
Recomendaciones		68
Bibliografía.....		69
Referencias		71

Introducción

El imparable avance de los sistemas informáticos en el mundo, tienen un papel importante en la sociedad actual. Su utilización en las diferentes esferas como la educación, industria, salud y deporte ofrece numerosas ventajas en cuanto a la reducción de tiempo, estandarización de la información y reducción de errores. Son muchos los ejemplos a citar de cuánto se desarrollan las tecnologías y sus disímiles aplicaciones en todos los campos de la existencia humana.

Cuba no está exenta a este desarrollo para todos los sectores de la sociedad, al contrario, el país invierte cada día más recursos para lograr insertar la informatización en todas las esferas sociales. Uno de los grandes pasos realizados para la necesaria incursión del país a la revolución informática fue la creación de la Universidad de las Ciencias Informáticas (UCI). Este centro de la Educación Superior cuenta con 6 facultades y aplica un modelo de integración que vincula los procesos de docencia, producción e investigación. Para el caso de la docencia cada facultad se estructura en varios departamentos docentes, como son: Programación y sistemas digitales, Ciencias básicas, Ciencias sociales y humanidades, Práctica profesional e Ingeniería de software. Estos son los encargados de dirigir y orientar metodológicamente al profesor de acuerdo a la asignatura que imparte. Además “centran su control, fundamentalmente, en la calidad de la labor educativa, docente y metodológica que realizan los profesores y tutores” (Ministerio de Educación Superior, 2007). La presente investigación se centra en el departamento de Ingeniería de software (ISW) de la facultad 2.

Hoy en día, los profesores del departamento de ISW de la facultad 2 realizan varios procesos para la planificación y control del cumplimiento del trabajo metodológico. Entre los procesos se encuentran la gestión del: plan de trabajo mensual, plan de evaluaciones (trimestral y anual), la planificación de las tareas de investigación para cada profesor, la planificación de pruebas, plan de clases, plan de control a clases y plan de trabajo metodológico, tanto de las distintas asignaturas como del departamento, entre otros.

En el caso específico del proceso para la gestión del plan de clase, actualmente en el departamento de ISW no existe una guía de apoyo a los profesores en la forma de planificar sus clases teniendo en cuenta su tipología, ni sugerencias o buenas prácticas en la redacción de las

categorías de la didáctica a partir de los indicadores de control a clases y las resoluciones vigentes que rigen esta planificación. A esto se suma que, en muchos casos, aunque el jefe de departamento archiva las carpetas de asignaturas digitales por semestre, donde se recogen las clases impartidas por semanas, no se guardan dichos planes. Cada profesor guarda su plan de clases de forma independiente, lo cual dificulta la gestión de dicha información de un curso a otro.

Por otra parte, la forma en que se lleva a cabo el proceso relacionado con los controles a clases limita al jefe de departamento de conocer los logros, deficiencias y recomendaciones de las mismas, así como trazar planes con mayor alcance para el tratamiento de deficiencias concretas que puedan presentar los profesores o el colectivo de alguna asignatura. Además, las deficiencias encontradas que puedan ser comunes para varios casos o simplemente los resultados alcanzados por un profesor en los controles a clase, se conocen solamente si se revisan todos y cada uno de los controles a clases realizados, por lo que el trabajo de los decisores involucrados se hace muy difícil y engorroso, así como el empleo de mucho tiempo para obtener un resultado.

Respecto al proceso para la planificación del plan de trabajo metodológico de las asignaturas, este debe nutrirse del plan de trabajo del departamento, sin embargo, en algunos casos no existe una correspondencia entre los mismos. A su vez la verificación del cumplimiento de las actividades planificadas a cada asignatura es engorrosa para el jefe de departamento.

El proceso docente educativo en Cuba es regulado por el Ministerio de Educación Superior, quien resuelve aprobar el “Reglamento para el trabajo docente y metodológico en la Educación Superior” con la Resolución 210/07.

A raíz de lo anteriormente planteado, se define como **problema a resolver**: Las insuficiencias en el proceso de planificación y control del cumplimiento del trabajo metodológico en el departamento de Ingeniería de software de la facultad 2, está afectando la toma de decisiones en el proceso docente educativo. Siendo el **objeto de estudio**: el proceso de planificación y control del cumplimiento del trabajo metodológico. De acuerdo al problema planteado se propone como **objetivo general**: desarrollar una aplicación web para la planificación y control

del cumplimiento del trabajo metodológico en el departamento de Ingeniería de software de la facultad 2, que sirva de apoyo a la toma de decisiones en el proceso docente educativo, enmarcado en el **campo de acción:** aplicaciones web para la planificación y control del cumplimiento del trabajo metodológico.

Para lograr el cumplimiento del objetivo planteado anteriormente se definen las siguientes **tareas de investigación:**

1. Análisis del proceso de planificación y control del cumplimiento del trabajo metodológico y los conceptos asociados al proceso docente educativo y el trabajo metodológico en la Educación Superior, con el propósito de conformar una base teórica para el desarrollo de la aplicación web.
2. Realización del estudio del estado del arte relacionado con el proceso de planificación y control del cumplimiento del trabajo metodológico, así como las aplicaciones web para realizar dicho proceso tanto en el ámbito nacional como internacional.
3. Definición de los procesos involucrados en la planificación y control del cumplimiento del trabajo metodológico.
4. Selección de una metodología de desarrollo que sirva de guía para documentar el proceso de desarrollo de la aplicación web propuesta.
5. Análisis de las tecnologías, lenguajes y herramientas a utilizar para el desarrollo de la aplicación web.
6. Documentación de las funcionalidades de la aplicación web para su posterior implementación.
7. Definición de una arquitectura con el fin de obtener una visión general de la aplicación web que se propone.
8. Análisis de los métodos de pruebas existentes para la validación de la aplicación web a partir de pruebas unitarias y de aceptación.

Para llevar a cabo la investigación se utilizaron diferentes métodos científicos, agrupados en métodos empíricos y teóricos. Estos métodos permiten una mejor recopilación de la información necesaria para realizar el modelado del análisis y diseño del sistema.

Métodos teóricos

Analítico - sintético: facilitó el procesamiento de la información obtenida en la investigación realizada sobre el proceso de planificación y control del cumplimiento del trabajo metodológico. Se realizó un estudio profundo de los conceptos asociados al proceso docente educativo y el trabajo metodológico en la Educación Superior, metodologías y herramientas a utilizar en el desarrollo de la investigación basándose en la revisión de trabajos de diploma, artículos e informes.

Histórico - lógico: permitió realizar un análisis del proceso docente educativo y el trabajo metodológico en la Educación Superior con el propósito de conformar una base teórica para el desarrollo de la aplicación web.

Modelación: permitió la creación de modelos mediante alternativas para la propuesta de solución, facilitó la definición de los componentes de la misma, así como sus relaciones. Además, fue empleado para la realización de los artefactos que requiere la metodología seleccionada: Historias de Usuarios (HU), Tarjetas CRC (Clase, Responsabilidad, Colaborador), Tareas de la ingeniería, Plan de iteraciones y Plan de entregas.

Métodos empíricos

Observación: proporcionó una visión realista y objetiva del proceso de planificación y control del cumplimiento del trabajo metodológico, conceptos y definiciones asociados al proceso docente educativo y el trabajo metodológico en la Educación Superior, relacionado con los controles a clases en un departamento.

Entrevista: permitió obtener la información necesaria relacionada con los problemas presentes en la gestión del proceso de planificación y control del cumplimiento del trabajo metodológico en el departamento de Ingeniería de software de la facultad 2.

El presente documento cuenta con una estructura de tres capítulos, los cuales abordan los siguientes temas:

Capítulo 1. Fundamentación Teórica: En este capítulo se presenta la definición conceptual del marco teórico de la investigación. Se realiza un análisis del estado del arte referente a las aplicaciones web existentes para la planificación y control del cumplimiento del trabajo

metodológico. Además, se exponen la metodología, herramientas, lenguajes y tecnologías que serán utilizadas para el desarrollo de la aplicación web propuesta.

Capítulo 2. Exploración y planificación de la aplicación web propuesta: En este capítulo se describe una propuesta detallada de cómo deberá funcionar la aplicación web propuesta. Se presentan los resultados de las fases de exploración y planificación de la propuesta de solución.

Capítulo 3. Producción y validación de la aplicación web propuesta: Se muestra el diseño del sistema y se describen los artefactos relacionados con la implementación. Se definen las tarjetas CRC (Contenido, Responsabilidad, Colaboración) y se detallan las iteraciones para el desarrollo del sistema, exponiéndose las tareas generadas en cada una de ellas por cada historia de usuario definida. Se describen las pruebas realizadas a la aplicación. Para ello se diseñan los casos de pruebas de aceptación a los que será sometida la aplicación en cada una de las iteraciones.

Capítulo 1: Fundamentación teórica

En este capítulo se analiza el proceso de planificación y control del cumplimiento del trabajo metodológico, se presentan los principales conceptos utilizados y asociados al dominio del problema, preparando la base de la investigación. Se realiza una revisión de trabajos previos vinculados con el proceso docente educativo, además de una descripción de la metodología, tecnologías, lenguajes y herramientas a utilizar para el desarrollo de la aplicación web propuesta.

1.1. Proceso de planificación y control del cumplimiento del trabajo metodológico

En la actualidad el mundo necesita que se desarrollen cada vez más los procesos formativos, en los cuales se garantiza una formación básica de los estudiantes que pasan por los diferentes niveles de enseñanza. En el caso de la educación superior, esto adquiere una importancia significativa, pues es el momento donde los jóvenes se forman como futuros profesionales. Es por ello que las universidades establecen como una prioridad, el adecuado desarrollo del proceso docente, garantizando el cumplimiento de los planes de estudios, así como todo el aseguramiento necesario para ello. Cuba no está ajena a esto, por lo que ha definido en sus diferentes instituciones de educación superior, una estructura departamental, que gestione el proceso de planificación docente y el control del cumplimiento del trabajo metodológico. En este sentido, los diferentes departamentos, gestionan un grupo de procesos (planificación del plan de trabajo individual, planificación del plan de trabajo mensual, planificación del plan de trabajo metodológico) entre ellos el proceso de planificación y control del cumplimiento del trabajo metodológico, que tributan al logro de los objetivos y metas definidas en cada área.

1.1.1 Plan de trabajo metodológico

La lógica del trabajo metodológico, en todos los colectivos y niveles de dirección, se desarrolla atendiendo a los objetivos generales previstos en el modelo del profesional.

Los problemas detectados durante el proceso docente educativo se toman como punto de partida para establecer los objetivos a alcanzar con el trabajo metodológico. Estos objetivos se concretan en acciones específicas a desarrollar, utilizando las formas y tipos que se establecen

en este reglamento u otras que surjan en la dinámica de este trabajo. Las acciones a realizar se plasman en un plan de trabajo metodológico para cada curso académico.

Los planes de trabajo metodológico de cada uno de los colectivos y niveles de dirección deben estar elaborados al inicio de cada curso académico y podrán adecuarse para cada período lectivo, en correspondencia con el diagnóstico y los resultados que se vayan alcanzando.

La planificación de las clases metodológicas aparecerá en el plan de trabajo metodológico elaborado al principio de cada curso en cada una de las instancias correspondientes.

La planificación de las clases abiertas aparecerá en el plan de trabajo metodológico elaborado al principio de cada curso en cada una de las instancias correspondientes.

La planificación de los seminarios científico metodológicos aparecerá en el plan de trabajo metodológico elaborado al principio de cada curso en cada una de las instancias correspondientes.

La planificación de la conferencia científico-metodológica aparecerá en el plan de trabajo metodológico del centro de educación superior elaborado al principio de cada curso.

Los documentos que se conservarán en el departamento para avalar el trabajo metodológico son: el plan de trabajo metodológico aprobado por el decano.

Los documentos que se conservarán en las facultades para avalar el trabajo metodológico son: el plan de trabajo metodológico de la facultad aprobado por la dirección del centro de educación superior (Ministerio de Educación Superior, 2007).

1.1.2 Categorías de la didáctica

La didáctica es una disciplina que abarca los principios más generales de la enseñanza aplicable a todas las asignaturas en su relación con los procesos educativos cuyo objetivo de estudio lo constituye el proceso docente-educativo, también llamado enseñanza-aprendizaje (Colectivo de autores, 2003).

En el enfoque tradicional de la didáctica se introducen otras categorías fundamentales llamadas componentes, estos son los campos con lo que cuenta el plan de trabajo metodológico los cuales se describen a continuación según el libro Preparación pedagógica integral para profesores universitarios:

Objetivos: es la categoría rectora del proceso de enseñanza-aprendizaje, define el encargo que la sociedad le plantea a la educación institucionalizada. Representa el elemento orientador de todo el acto didáctico, la modelación del resultado esperado, sin desconocer el proceso para llegar a este (en una disciplina, una asignatura, un sistema de clases o en una clase).

Contenido: expresa aquella de cual se debe apropiarse el estudiante. Está formado por los conocimientos, habilidades, hábitos, métodos de las ciencias, normas de relación con el mundo y valores que responden a un medio socio-histórico concreto. El contenido cumple funciones instructivas, educativas y desarrolladoras.

Métodos y procedimientos: constituye el sistema de acciones que regula la actividad del profesor y los alumnos, en función del logro de los objetivos. Teniendo en cuenta las exigencias actuales, se debe vincular la utilización de métodos reproductivos con productivos procurando, siempre que sea posible, el predominio de estos últimos.

Medios: están constituidos por objetos naturales o conservados o sus representaciones, instrumentos o equipos que apoyan la actividad de docentes y alumnos en función del cumplimiento del objetivo.

Formas de organización: son el soporte en el cual se desarrolla el proceso de enseñanza-aprendizaje, en ellas intervienen todos los implicados: estudiante, profesor, institución, familia y comunidad. La clase es la forma de organización fundamental.

Evaluación: es el proceso para comprobar y valorar el cumplimiento de los objetivos propuestos y la dirección didáctica de la enseñanza y el aprendizaje en sus momentos de orientación y ejecución. Se deberán propiciar actividades que estimulen la autoevaluación por los estudiantes, así como las acciones de control y valoración del trabajo de los otros.

1.1.3 Tipos de clase

La clase es una de las formas organizativas del proceso docente educativo, que tiene como objetivos la adquisición de conocimientos, el desarrollo de habilidades y la formación de valores e intereses cognoscitivos y profesionales en los estudiantes, mediante la realización de actividades de carácter esencialmente académico. Las clases se clasifican sobre la base de los objetivos que se deben alcanzar y sus tipos principales son: la conferencia, la clase práctica, el

seminario, la clase encuentro, la práctica de laboratorio y el taller (Ministerio de Educación Superior, 2007).

A continuación, se detallan los tipos de clase según la Resolución 210/2007 del Ministerio de Educación Superior:

Conferencia: Es el tipo de clase que tiene como objetivo principal la transmisión a los estudiantes de los fundamentos científico-técnicos más actualizados de una rama del saber con un enfoque dialéctico-materialista, mediante el uso adecuado de métodos científicos y pedagógicos, de modo que les ayude en la integración de los conocimientos adquiridos y en el desarrollo de las habilidades y valores que deberán aplicar en su vida profesional.

Clase práctica: Es el tipo de clase que tiene como objetivos fundamentales que los estudiantes ejecuten, amplíen, profundicen, integren y generalicen métodos de trabajo característicos de las asignaturas y disciplinas que les permitan desarrollar habilidades para utilizar y aplicar, de modo independiente, los conocimientos.

Seminario: Es el tipo de clase que tiene como objetivos fundamentales que los estudiantes consoliden, amplíen, profundicen, discutan, integren y generalicen los contenidos orientados; aborden la resolución de tareas docentes mediante la utilización de los métodos propios de la rama del saber y de la investigación científica; desarrollen su expresión oral, el ordenamiento lógico de los contenidos y las habilidades en la utilización de las diferentes fuentes del conocimiento.

Clase encuentro: Es el tipo de clase que tiene como objetivos aclarar las dudas correspondientes a los contenidos y actividades previamente estudiados por los alumnos; debatir y ejercitar dichos contenidos y evaluar su cumplimiento; así como explicar los aspectos esenciales del nuevo contenido y orientar con claridad y precisión el trabajo independiente que el estudiante debe realizar para alcanzar un adecuado dominio de los mismos. La misión instructiva más importante que tiene el profesor en la clase encuentro es contribuir al desarrollo de la independencia cognoscitiva de los estudiantes. La clase encuentro es la actividad presencial fundamental en la modalidad semipresencial, aunque puede utilizarse también en la presencial.

Práctica de laboratorio: Es el tipo de clase que tiene como objetivos que los estudiantes adquieran las habilidades propias de los métodos y técnicas de trabajo y de la investigación científica; amplíen, profundicen, con-soliden, generalicen y comprueben los fundamentos teóricos de la disciplina mediante la experimentación, empleando para ello los medios necesarios. Las prácticas de laboratorio se realizan en instalaciones propias de las universidades o en las que existen en las unidades docentes u otras entidades laborales. Como norma, en este tipo de clase se deberá garantizar el trabajo individual de los estudiantes en la ejecución de las tareas previstas.

Taller: Es el tipo de clase que tiene como objetivo específico que los estudiantes apliquen los conocimientos adquiridos en las diferentes disciplinas para la resolución de problemas propios de la profesión, a partir del vínculo entre los componentes académico, investigativo y laboral.

El taller contribuye al desarrollo de habilidades para la solución integral de problemas profesionales en grupo, para el grupo y con la ayuda del grupo, donde primen las relaciones interdisciplinarias.

1.1.4 Indicadores de control

Los indicadores de control son los aspectos que los controladores deben tener como guía para dar un criterio positivo (logro) o negativo (deficiencia) del profesor controlado.

Aspectos organizativos de la clase

Correspondencia con el Programa y el Plan Calendario de la asignatura (P1)

Comienzo de la clase a la hora prevista, asistencia y puntualidad de los estudiantes y el profesor.

Tratamiento didáctico-metodológico

Orientación hacia el cumplimiento del objetivo de la actividad.

Organización, tratamiento y dominio de los contenidos. Uso de los métodos de enseñanza:

a) Aseguramiento del nivel de partida de los estudiantes, su auto-preparación y motivación para la actividad.

b) Ambiente que propicia durante el proceso de enseñanza-aprendizaje (orden, atención, interés y participación activa de los estudiantes, trabajo cooperativo).

- c) Actividad centrada en el aprendizaje del estudiante, mediante el empleo de métodos activos y del papel orientador del profesor.
- d) Realización de conclusiones parciales y finales.
- e) Orientación del uso del libro de texto u otros materiales para el estudio independiente.
- f) Atención a las diferencias individuales de los estudiantes.

Uso de formas de control y/o evaluación del aprendizaje.

Orientación del estudio y el trabajo independiente, con indicadores claros y medibles para su ejecución y control.

Uso de los medios de enseñanza aprendizaje.

Aplicación de las estrategias curriculares

Comunicación profesor-estudiante y estudiante-estudiante.

Formación en valores de los estudiantes a través de los contenidos de la clase.

Cumplimiento de la estrategia de Trabajo Político-Ideológico desde la clase.

Cuidado y rigor en la exposición de sus ideas, precisión en el lenguaje oral y escrito.

Vinculación con otras materias del año o de la carrera.

Cumplimiento de los objetivos de la clase

Se cumplieron los objetivos definidos para la clase.

1.1.5 Conceptos fundamentales asociados al tema

Durante el desarrollo de este documento y de la investigación se emplean conceptos que por su importancia se hace necesario reflejar sus definiciones. A continuación, se enuncian los principales conceptos relacionados con la planificación y control del cumplimiento del trabajo metodológico:

Plan de clase: “Es un documento indispensable en el trabajo metodológico del profesor, debe ser elaborado con una semana de antelación, como mínimo. Su estructura y extensión está en dependencia de la experiencia pedagógica y la preparación que posea el docente en el contenido y su didáctica. Se estructura en introducción, desarrollo y conclusiones y no deben faltar los componentes de la didáctica, así como el desarrollo de los elementos fundamentales a

tratar en la clase. En él se incluyen las evaluaciones sistemáticas y parciales que se aplicarán, así como la clave de calificación que corresponda en cada caso” (Ministerio de Educación Superior, 2014).

Control a clases: El control de la actividad docente es aquel que se realiza a una de las formas organizativas del proceso docente educativo, previsto en el horario de clases de los estudiantes, y estará dirigido a comprobar el logro de los objetivos propuestos para dicha actividad. Al finalizar el control, sin la presencia de los estudiantes, el responsable dirigirá el análisis, dará las conclusiones al controlado, señalará los principales logros, las deficiencias y las recomendaciones. Los resultados del control se recogerán en un documento que será firmado por el docente controlado, como constancia de que fue informado de sus resultados. El control de la actividad docente se calificará utilizando las categorías de Excelente, Bien, Regular o Mal. Se exceptuarán de una calificación, los casos que se determinen por el responsable del control (Ministerio de Educación Superior, 2007).

Plan de control a clases: se elabora por cada jefe de departamento docente. Para las carreras que se estudian en las sedes universitarias participará en la elaboración del plan el coordinador de la carrera en cuestión. Los profesores no podrán conocer las actividades docentes en las que serán controlados. Todos los profesores que impartan por primera vez una asignatura deberán ser controlados al menos una vez en el período docente, con el objetivo de verificar su adecuada preparación. Los decanos, vicedecanos y directores de sede elaborarán también su plan de controles a clases en coordinación con el jefe de departamento docente y con el coordinador de la carrera en la sede, según corresponda (Ministerio de Educación Superior, 2007).

Trabajo metodológico: Es la labor que, apoyados en la Didáctica, realizan los sujetos que intervienen en el proceso docente educativo, con el propósito de alcanzar óptimos resultados en dicho proceso, jerarquizando la labor educativa desde la instrucción, para satisfacer plenamente los objetivos formulados en los planes de estudio (Ministerio de Educación Superior, 2007).

1.2. Aplicaciones web para la planificación y control del cumplimiento del trabajo metodológico

Se estudiaron diferentes aplicaciones web internacionales y nacionales para la planificación y control del cumplimiento del trabajo metodológico las cuales se caracterizan a continuación señalando sus principales funcionalidades y limitaciones.

1.2.1 Aplicaciones web internacionales

Sistema Integral de Gestión Académica (CAMS) ¡Error! No se encuentra el origen de la referencia.

Sistema Integral de Gestión Académica (CAMS) haciendo uso de tecnologías de Microsoft, fue desarrollado por Three Rivers Systems Inc. que es un proveedor líder de soluciones de software para la administración superior de educación. Este es altamente escalable, rápido de implementar y fácil de usar, manteniendo siempre la información más reciente y precisa, tratando de satisfacer las necesidades de administradores, personal, profesores y estudiantes. Entre las universidades que usan este sistema se encuentra la William Jessup University (WJU) de California que lo utiliza para gestionar el ciclo de vida estudiantil. Es totalmente integrable, y sigue los estándares establecidos por los sistemas de planificación de recursos empresariales (ERP), se adaptó exclusivamente para la educación superior. Dentro de las funciones que maneja se pueden evidenciar:

Portales.

Registros.

Facturación.

Ayuda financiera.

Gestión de cursos.

Recaudación de fondos.

Gestión de Relaciones con el Cliente (CRM).

Ofrece servicios de implementación, capacitación.

Gestión de proyectos y el apoyo a sus clientes.

Consolida múltiples sistemas independientes, incluidos las admisiones, la información del estudiante, relaciones con ex alumnos y gestión de documentos.

Es una solución integrada, que presenta una profunda inteligencia de negocios, mayor accesibilidad y mejor informatización y comunicación. Brinda un rápido proceso de implementación y de transición al nuevo sistema, con una total integración con la base de datos MySQL. Presenta gran funcionalidad y escalabilidad brindándole una buena gestión de sus procesos y elimina la necesidad de las instituciones académicas y de los usuarios finales tanto para aprender interfaz como para gestionar un conjunto de sistemas que a menudo son incompatibles.

1.2.2 Aplicaciones web nacionales

Akaderos (Sistema de Gestión Universitaria)

Sistema para la gestión de los procesos docentes en la Universidad de las Ciencias Informáticas. La misma cuenta con un grupo de módulos compuesto por un conjunto de funcionalidades de acuerdo al rol asignado al usuario, ejemplo de estas se muestran a continuación:

- Gestionar un control a clase.

- Generar las evaluaciones de un estudiante, por semestre y curso.

- Muestra el plan de estudio a cada estudiante.

- Muestra el registro de asignaturas a cada estudiante.

- Muestra el resumen de evaluaciones a cada estudiante.

- Permite buscar un estudiante por su CI, usuario, nombre y apellidos o solapín.

Está desarrollado con el uso del marco de trabajo GUUD (híbrido entre JQuery y code igniter).

Se utilizó como gestor de base de datos Postgres en su versión 9.4. Para el control de versiones SVN, Pencil Project para el prototipo de interfaces. En el entorno de desarrollo integrado NetBeans, como servidor de aplicaciones Nginx y la máquina virtual de java en su versión 6 para generar PDF.

Sistema para la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software.¡Error! No se encuentra el origen de la referencia.

Es un sistema informático que fue desarrollado por la Facultad 1 en el año 2013. Este cuenta con un grupo de funcionalidades que se detallan a continuación:

Gestionar la planificación de controles a clase por asesores.

Gestionar un control a clase a un asesor.

Generar reporte total de las evaluaciones de control a clases por asignaturas.

Generar reporte total de las evaluaciones de controles a clases.

Generar reporte del estado actual de la planificación de controles a clase por asignaturas.

Generar reporte del estado actual de las evaluaciones de controles a clases.

Gestionar nomenclador.

Gestionar planificación de las preparaciones metodológicas.

Gestionar acta de las preparaciones metodológicas.

Gestionar visitas a las preparaciones metodológicas.

Visualizar la planificación de las visitas a las preparaciones metodológicas.

Emitir valoración de la preparación metodológica.

Gestionar plan de trabajo de los asesores.

Gestionar evaluación del profesor.

Generar reporte de las evaluaciones de los profesores.

Generar corte de resumen de evaluaciones de los profesores.

Gestionar usuario.

Está desarrollado con el uso del marco de trabajo Symfony. Se utilizó como gestor de base de datos PostgreSQL. Como lenguaje del lado del cliente la combinación de HTML5, JavaScript y CSS 3. Del lado del servidor el lenguaje PHP, en el entorno de desarrollo integrado NetBeans y como servidor web Apache.

Sistema de gestión docente metodológica integrado con la plataforma informativa SO3.

(Campos León, y otros, 2013)

Es un sistema de gestión de la Facultad 3 desarrollado en el año 2013 e integrado a la Plataforma de Servicios Integrados de esta facultad. Este cuenta con varias funcionalidades que se detallan a continuación:

- Administrar plan de control a clase.
- Guardar plan de control a clase.
- Administrar planificación de controles a clase.
- Guardar una planificación de control a clases.
- Administrar planilla del control a clase.
- Guardar planilla del control a clase.
- Generar reportes.
- Administrar plan de trabajo metodológico.
- Generar reportes.
- Modificar estado de las actividades.
- Adicionar profesores al listado de la guardia docente.
- Eliminar profesores del listado de la guardia docente.
- Administrar distribución de la guardia docente.
- Guardar distribución de la guardia docente.
- Administrar descripción de la guardia docente.
- Guardar descripción de la guardia docente.
- Generar reportes.
- Administrar notificaciones.
- Administrar departamentos.
- Administrar asignaturas.
- Determinar semanas.

El desarrollo de este sistema fue con el lenguaje de programación PHP en el entorno de desarrollo integrado NetBeans, el marco de trabajo Symfony y el gestor de base de datos PostgreSQL. Además, se empleó la librería JQuery.

Sistema de gestión de la planificación de las actividades de los trabajadores docentes en la Facultad 2 de la Universidad de las Ciencias Informáticas. (Bauza Becerra, 2013)

Es un sistema que gestiona la planificación de las actividades de los trabajadores docentes de la facultad 2. Entre sus funcionalidades se encuentran:

- Autenticar usuario.
- Gestionar usuario.
- Cambiar rol del usuario.
- Gestionar rol.
- Gestionar plan de trabajo mensual de la facultad.
- Publicar plan de trabajo mensual de la facultad.
- Enviar notificación de plan de trabajo metodológico de la facultad.
- Gestionar plan de trabajo mensual del departamento.
- Publicar plan de trabajo mensual del departamento.
- Enviar notificación a profesores del departamento.
- Gestionar plan de trabajo mensual individual.
- Publicar plan de trabajo mensual individual.
- Gestionar plan de trabajo anual.
- Publicar plan de trabajo anual.
- Enviar notificación de plan de trabajo anual.
- Gestionar evaluación trimestral.
- Publicar evaluación trimestral.
- Enviar notificación de evaluación trimestral.
- Gestionar resultados de control a clase.
- Publicar resultados de control a clase.
- Enviar notificación de resultados de control a clase.
- Gestionar área.
- Gestionar artículos.
- Publicar artículos.

La construcción de la base de datos se realizó con MySQL, desarrollado en JavaScript y PHP, con uso del framework Symfony y con el IDE de desarrollo NetBeans.

1.2.3 Conclusiones del estado del arte

Aspectos Sistemas	Mostrar deficiencias comunes	Gestionar plan de clase	Gestionar plan de control a clase	Gestionar plan de trabajo metodológico	Software libre
Sistema Integral de Gestión Académica	NO	NO	NO	NO	NO
Akademos	NO	NO	Parcialmente	NO	SI
Sistema para la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software.	NO	NO	SI	SI	SI
Sistema de gestión docente metodológica integrado con la plataforma informativa SO3.	NO	NO	Parcialmente	SI	SI
Sistema de gestión de la planificación de las actividades de los trabajadores docentes en la facultad 2 de la Universidad de las Ciencias Informáticas.	NO	NO	Parcialmente	Parcialmente	SI

Tabla 1. Estado del arte

Se puede apreciar que los sistemas informáticos anteriormente estudiados ofrecen soluciones parciales a las necesidades del cliente. Sin embargo, ninguno de dichos sistemas centraliza las funcionalidades que requiere el cliente. Es por ello que, se decide desarrollar una aplicación web, que ofrezca una solución completa al problema planteado.

1.3. Metodologías, herramientas, lenguajes y tecnologías

Para guiar el desarrollo de la aplicación web propuesta es necesario el uso de herramientas, lenguajes y tecnologías que estén en correspondencia con la metodología seleccionada.

1.3.1 Metodología de desarrollo

Metodología Programación Extrema (XP)

XP constituye una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. **Error! No se encuentra el origen de la referencia..**

El ciclo de vida de XP consta de 6 fases, exploración, planificación de entrega, iteraciones, producción, mantenimiento y muerte del proyecto (Letelier & Penadés, 2006). Para lograr un correcto desarrollo de la aplicación web, se decide emplear como guía las cuatro fases fundamentales descritas a continuación:

Fase I (exploración): el equipo se familiariza con el proyecto y las tecnologías necesarias. Al mismo tiempo, el cliente desarrolla la mayor cantidad posible de historias de usuarios. La duración de esta fase depende del nivel de conocimientos del equipo sobre las tecnologías a utilizar.

Fase II (planificación de la entrega): el cliente prioriza las historias de usuarios y el equipo hace una estimación de duración para las mismas. Cada entrega debe realizarse en unos pocos meses.

Fase III (iteraciones): cada iteración incluye historias de usuarios que no hayan sido incluidas en la iteración anterior, corrige defectos encontrados y asume tareas anteriormente sin terminar.

Fase IV (producción): al sistema se le realiza una serie de pruebas rigurosas para ser trasladado al entorno del cliente.

Selección de la metodología

Se escoge por parte del equipo de trabajo la metodología XP, ya que es una metodología ágil que busca la satisfacción del cliente y entrega temprana del software incremental, para ello se tuvo en cuenta los 5 criterios de la estrella de Boehm y Turner. Este método caracteriza y estima cuan ágil o prescriptivo debería ser el enfoque a utilizar, estos son: tamaño del equipo, criticidad del producto, dinamismo de los cambios, cultura del equipo y personal con que se cuenta. Cada uno de estos criterios sirvieron de apoyo para afirmar que el desarrollo del producto final debe realizarse en un breve período de tiempo, el equipo de desarrollo está compuesto por cuatro integrantes entre los que se encuentra el cliente. Cuenta con un porcentaje de adaptación alto, al igual que el dinamismo. La experiencia del equipo en el desarrollo de sistemas es media y la vida de desarrollo de la aplicación web es corta. Los bienes tienen un valor medio y existe un alto valor de utilidad de la aplicación web a desarrollar, y se centra casi en su totalidad en las tareas de desarrollo, preocupándose por hacer mejoras continuas al producto.

1.3.2 Tecnologías y herramientas de desarrollo

Para la realización de la aplicación propuesta se cuenta con herramientas de código abierto que permiten una mayor facilidad en su desarrollo y mantenimiento. A continuación, se da una breve descripción de ellas:

Framework de desarrollo

Es un esquema para el desarrollo y la implementación de una aplicación. Permite acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones (Chun, 2012).

Django 1.8.5 es un framework de desarrollo web de código abierto, escrito en Python y que respeta el patrón de diseño arquitectónico Modelo-Vista-Controlador. La meta fundamental de este es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio de “No te repitas”. Es usado en configuraciones, archivos y en los modelos de datos. Se seleccionó por ser un framework web de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático y la experiencia de las autoras con el uso del mismo es alta (Chun, 2012).

Bootstrap 3 es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien. Es Open Source o código abierto, por lo que se puede usar de forma gratuita y sin restricciones. Se seleccionó por ser compatible con la mayoría de los navegadores web y la experiencia de las autoras con el uso del mismo (Cochran, 2012).

Entorno de desarrollo integrado

PyCharm 4.5 es un entorno de desarrollo integrado (IDE, Integrated Development Environment) para el desarrollo de aplicaciones basadas en el lenguaje Python. PyCharm está basado en IntelliJ IDEA, es multiplataforma. Se encuentra disponible para Windows, MAC OS X y Linux. Por estas razones y la experiencia de las autoras con el uso del mismo se decide usarlo como entorno de desarrollo.

Sistema gestor de base de datos

SQLite Manager 0.8.3.1

Es una herramienta que ofrece un gestor o administrador de bases de datos SQLite en Firefox en forma de extensión. Se diseña para crear nuevas tablas, registros e índices, modificarlos y borrarlos, sin olvidar la posibilidad de realizar consultas. Por lo anteriormente descrito y la experiencia de las autoras, se seleccionó esta herramienta.

Lenguajes de programación

Un lenguaje de programación es un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un

conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas (Alegsa, 2010).

Python 2.7.11

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Posee una licencia de código abierto denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores. Se hace selección de este lenguaje debido a la experiencia de las autoras y es el lenguaje integrado al entorno y framework de desarrollo (Martelli, 2008).

HTML 5

El Lenguaje de Marcación de Hipertexto es el principal lenguaje que se utiliza para la creación de páginas web. Tiene como propósito fundamental, definir la apariencia y la estructura básica de las páginas, garantizando que las mismas sean manejadas por el usuario de la red, de una manera fácil y rápida. Se seleccionó por la experiencia de las autoras (Gauchat, 2012).

JavaScript

Usando JavaScript se pueden crear páginas interactivas y con efectos especiales; además, cuenta con muchas posibilidades, con funciones y estructuras de datos complejas. El navegador web es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para lograr la interactividad y los efectos deseados. Este lenguaje cuenta con el navegador web como único recurso para su funcionamiento. Soporta el Modelo de Objetos de Documento (DOM, Document Object Model), lo que nos permite acceder a todos los elementos de una página y a ciertas características específicas del navegador. Se seleccionó por la experiencia de las autoras y las características que presenta (Severance, 2012).

CSS 3

Se utiliza para dar estilo a documentos HTML y XML, además separa el contenido de la presentación. Los estilos definen la forma de mostrar los elementos HTML y XML. CSS permite

a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. Se seleccionó por la experiencia de las autoras y las características anteriormente descritas (Hogan, 2011).

1.4. Conclusiones del capítulo

En este capítulo se realizó un análisis de los principales conceptos asociados al tema, además de un estudio de los sistemas semejantes existentes en el mundo. A raíz de este estudio se llegó a la conclusión de que los mismos no reúnen todas las funcionalidades necesarias para garantizar la gestión y control de los procesos que se llevan a cabo en el departamento de Ingeniería de software de la facultad 2, por lo que se decidió desarrollar una nueva aplicación web que solucionará la problemática anteriormente planteada teniendo en cuenta también como guía algunas de las funcionalidades que muestran estos sistemas. Para el desarrollo del sistema se decidió utilizar como framework de desarrollo Django en su versión 1.8.5 y Bootstrap 3 como framework de desarrollo para el uso del diseño web. PyCharm 4.5 como entorno de desarrollo integrado, para la administración de la base de datos SQLite Manager 3.8.11.1. Como lenguajes se decidió usar Python 2.7.11, HTML 5, CSS 3 para dar estilo y JavaScript para el diseño web. Se decidió trabajar con la metodología de desarrollo XP principalmente por las características que presentan el equipo de desarrollo y el corto período de tiempo que se tiene para la entrega de la aplicación web.

Capítulo 2: Exploración y planificación de la aplicación web propuesta

En este capítulo se presentan los resultados obtenidos de las fases de exploración y planificación de la aplicación web propuesta. Además, se especifican las funcionalidades que debe cumplir para satisfacer las necesidades del cliente, así como los diferentes artefactos que establece la metodología de desarrollo de software utilizada.

2.1. Descripción del flujo actual de los procesos

Es preciso destacar que, de todos los procesos docentes metodológicos, esta investigación solo se centrará, en el Plan de control a clases, Plan de trabajo metodológico y el Plan de clases.

Proceso de gestión de la planificación de los controles a clases

Breve descripción	El proceso inicia cuando el jefe de departamento designa un control a clases a un profesor de su departamento. Para el mismo los controladores deben tener presente un conjunto de indicadores que, finalmente, tributarán al registro de una nota cualitativa. Uno de los elementos más importantes de este proceso es el registro de las deficiencias y principales logros del docente por cada indicador durante la impartición de una clase controlada.
Involucrados	Jefe del departamento
Entradas/salidas	Listado de profesores Listado de controladores Indicadores de control Plan de control a clases

Proceso de gestión del Plan de trabajo metodológico

Breve descripción	El proceso inicia cuando el jefe de departamento elabora el Plan de trabajo metodológico del departamento al inicio de un curso escolar. Luego cada jefe de asignatura, a inicios de semestre, elabora el plan de trabajo metodológico de cada asignatura. Dicho plan debe estar alineado con los objetivos y líneas trazadas por el Plan de trabajo metodológico del departamento. Ambos planes deben poder ser consultados por todos los profesores en el caso del plan de trabajo metodológico del departamento y por todos los profesores que imparten una misma asignatura en caso del plan de trabajo metodológico de una asignatura.
--------------------------	---

Involucrados	Jefe del departamento y jefes de asignaturas
Entradas/salidas	Plan de trabajo metodológico del departamento.pdf Plan de trabajo metodológico de la asignatura.pdf

Proceso de gestión del Plan de clases

Breve descripción	El proceso inicia cuando un profesor crea su plan de clases como parte de su preparación previa antes de cualquier actividad docente. Debe tener en cuenta las categorías de la didáctica para su confección. Dicho plan de clases podrá ser consultado por cualquier profesor que imparta la misma asignatura en el semestre en curso o elaborado en semestres anteriores.
Involucrados	Profesores
Entradas/salidas	Plan de clases.pdf

2.2. Funcionalidades

Para el desarrollo de la aplicación web es muy importante conocer cuáles van a ser las principales funcionalidades que va a presentar. Estas se obtienen con la definición de los principales requisitos que debe cumplir, ya que constituyen la descripción de las necesidades del mismo. A continuación, se muestra una tabla con una lista de las funcionalidades del sistema:

Número	Funcionalidades	Prioridad
1	Gestionar plan de clases Crear plan de clase Modificar plan de clase Eliminar plan de clase Mostrar plan de clase	Alta
2	Gestionar plan de control a clase <ul style="list-style-type: none"> • Crear plan de control a clase • Modificar plan de control a clase • Eliminar plan de control a clase • Mostrar plan de control a clase 	Alta
3	Gestionar plan de trabajo metodológico <ul style="list-style-type: none"> • Crear plan de trabajo metodológico • Modificar plan de trabajo metodológico • Eliminar plan de trabajo metodológico • Mostrar plan de trabajo metodológico 	Alta
4	Gestionar actividad metodológica <ul style="list-style-type: none"> • Añadir actividad metodológica • Modificar actividad metodológica 	Alta

	<ul style="list-style-type: none"> • Eliminar actividad metodológica • Mostrar actividad metodológica 	
5	Mostrar deficiencias comunes	Alta
6	Gestionar claustro	Media
7	Gestionar lineamientos	Media
8	Gestionar indicador	Media
9	Gestionar deficiencia	Media
10	Configurar años para impartir asignaturas	Media
11	Configurar curso docente	Media
13	Autenticar usuario	Media
14	Actualizar perfil	Media
15	Gestionar profesor	Media
16	Notificar actividad metodológica al correo	Baja
17	Mostrar controles a realizar	Baja
18	Buscar profesor visitado	Baja
19	Subir resultados del control a clases	Baja
20	Mostrar resumen de controles realizados	Baja
21	Mostrar evaluaciones	Baja
22	Buscar evaluaciones de cursos anteriores	Baja
23	Mostrar resultados del control a clase realizado	Baja
24	Crear plan de clase a partir de un plan creado	Baja
25	Buscar plan de clase	Baja
26	Buscar planes de clases de cursos anteriores	Baja
27	Listar planes de trabajo metodológicos actuales	Baja
28	Listar planes de trabajo metodológicos creados	Baja
29	Exportar plan de clase a PDF	Baja
30	Imprimir plan de clase	Baja
31	Exportar plan de trabajo metodológico a PDF	Baja
32	Imprimir plan de trabajo metodológico	Baja

Tabla 2. Funcionalidades

2.3. Propiedades de la aplicación web propuesta

Las propiedades son cualidades que el sistema debe tener. Debe pensarse en estas como las características que hacen al sistema atractivo, usable, rápido o confiable (Arias Chaves, 2007).

A continuación se nombran los requisitos no funcionales definidos con los que debe contar el sistema a desarrollar:

Usabilidad: El sistema podrá ser usado por cualquier persona que posee elementos mínimos de informática. El sistema garantizará una conexión segura y eficiente con la base de datos, que tendrá almacenada toda la información del sistema, mejorando esto la gestión de la información y los datos.

Interfaz: El sistema deberá contar con una interfaz fácil, amigable, sencilla, permitiendo que los usuarios finales del mismo sean capaces de interactuar con este teniendo conocimientos básicos.

Seguridad: Solo tendrá acceso al sistema el personal autorizado por el jefe de departamento, el cual le permitirá el acceso a la aplicación web luego que este autenticado.

Soporte: El sistema permitirá ser flexible a modificaciones, permitiendo aumentar su nivel de funcionalidades para brindar mayores prestaciones.

Hardware: Para las PC clientes procesador 2 GHz o superior, RAM: 512 MB o superior, disco duro 20 GB o superior.

Software: Para las PCs clientes navegador Mozilla Firefox, Google Chrome, Opera, Internet Explorer.

Restricciones en el diseño y la implementación: El sistema permitirá ser funcional en diferentes plataformas y su desarrollo estará soportado bajo plataformas libres.

2.4. Propuesta de la aplicación

Se propone la implementación de una aplicación web que brinde las funcionalidades identificadas. La misma contará con varios módulos: Plan de clases, Plan de control a clase, Plan de trabajo metodológico (PTM), Deficiencias y Profesores. Estos permitirán gestionar los planes de clase de cada asignatura del departamento de ISW, los planes de control a clases y el plan de trabajo metodológico de las asignaturas como para el departamento, y mostrará el

listado de deficiencias comunes obtenidas de los controles a clases realizados, así como el listado de profesores del departamento con sus roles correspondientes. Dichos roles se identifican en la aplicación web como jefe de departamento, jefe de asignatura y profesor, quienes serán los encargados de interactuar con la misma, estos a su vez podrán ser controladores o controlados. La aplicación web permitirá al usuario ver sus planes de clases, sus evaluaciones, los controles a realizar y la cantidad de actividades asignadas respecto al estado con que se encuentran las mismas (cumplidas, incumplidas y pendientes). Estas actividades serán las planificadas en el PTM por el jefe departamento, las cuales llegarán a los responsables en forma de notificación a su correo, quienes serán los encargados de modificar el estado de estas.

Para tener acceso a la aplicación web, el usuario tendrá que autenticarse primeramente, así el jefe de departamento le asignará un rol. El usuario con rol de profesor podrá gestionar el plan de clase correspondiente a la asignatura que imparte, podrá ver y actualizar su perfil, buscar los planes de clases de cursos anteriores correspondientes a su asignatura, así como ver los controles a realizar, crear nuevos indicadores y nuevas deficiencias en los controles asignados, subir y mostrar los resultados del control, mostrar evaluaciones de cursos anteriores, buscar profesores visitados, imprimir el plan de clase y el plan de trabajo metodológico y exportar los mismos a formato pdf. Un jefe de asignatura, además de realizar las funcionalidades antes mencionadas podrá gestionar el plan de trabajo metodológico de la asignatura, gestionar una actividad metodológica, crear claustro y listar planes de trabajo metodológico de la asignatura.

Como rol principal, el jefe del departamento tendrá acceso a todas las funcionalidades que realiza un profesor y el jefe e asignatura, incluyendo la gestión de una actividad metodológica del departamento, mostrar resumen de controles realizados, configurar años para impartir asignaturas, configurar curso docente, listar planes de trabajo metodológico del departamento y listar planes de trabajo metodológicos creados. Entre las principales funcionalidades se encuentra la gestión del plan de control a clase, donde el jefe de departamento es el encargado de la misma lo que le permite elegir la cantidad de controles, el controlador o los controles, los controlados y las semanas en que van a realizarse estos controles. La gestión del plan de trabajo metodológico de la asignatura y del departamento será realizada por el jefe de

asignatura y el jefe de departamento respectivamente, para crear un PTM se cuenta con las funcionalidades claustro, donde se caracteriza el claustro de la asignatura o el departamento en dependencia del plan de trabajo metodológico que se esté creando; la funcionalidad lineamiento permite gestionar lineamientos; y la funcionalidad actividades permite gestionar las actividades a través de una tabla que cuenta con los campos: No. (Número), Título, Objetivo, Participan, Planificados para y Responsable. El módulo PTM registra las deficiencias comunes obtenidas de los controles planificados en el sistema, donde el controlador podrá añadirle la evaluación obtenida al profesor controlado, luego de haber discutido y aprobado la misma con el jefe del departamento, el mismo muestra además el plan de trabajo del curso actual y un listado con los planes de trabajo creados anteriormente.

2.5. Historias de usuario

Las historias de usuario (HU) son las técnicas que utiliza la metodología XP para especificar los requisitos del software. Estas son escritas desde la perspectiva del cliente, donde describen de forma breve las características que la aplicación web debe realizar, aunque también los desarrolladores pueden brindar su ayuda en la identificación de las mismas. Cada una de ellas es lo suficientemente comprensible, como para que los programadores puedan estimar con un reducido margen de riesgo, su tiempo de desarrollo. A continuación, se muestran las HU definidas con prioridad alta para el usuario, el resto ver en Anexos 2:

Historia de usuario	
Número: 1	Nombre historia de usuario: Crear plan de clase
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 3
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.6	
Descripción: La presente historia de usuario tiene como objetivo crear un plan de clase. Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello. El usuario deberá llenar los campos: actividad, semana, tipo de clase, unidad didáctica o	

tema, título, objetivo, contenido, bibliografía, métodos, medios, tipo de evaluación, introducción, desarrollo, conclusiones, estudio independiente.

Observaciones: La entrada al sistema.

Interfaz:

Tabla 3: Historia de usuario #1

Historia de usuario	
Número: 2	Nombre historia de usuario: Modificar plan de clase
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 3
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.4	
<p>Descripción: La presente historia de usuario tiene como objetivo modificar un plan de clase.</p> <p>Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello. El usuario podrá modificar cualquiera de los campos del plan de clase: actividad, semana, tipo de clase, unidad didáctica o tema, título, objetivo, contenido, bibliografía, métodos, medios, tipo de evaluación, introducción, desarrollo, conclusiones, estudio independiente.</p>	
Observaciones: La entrada al sistema.	

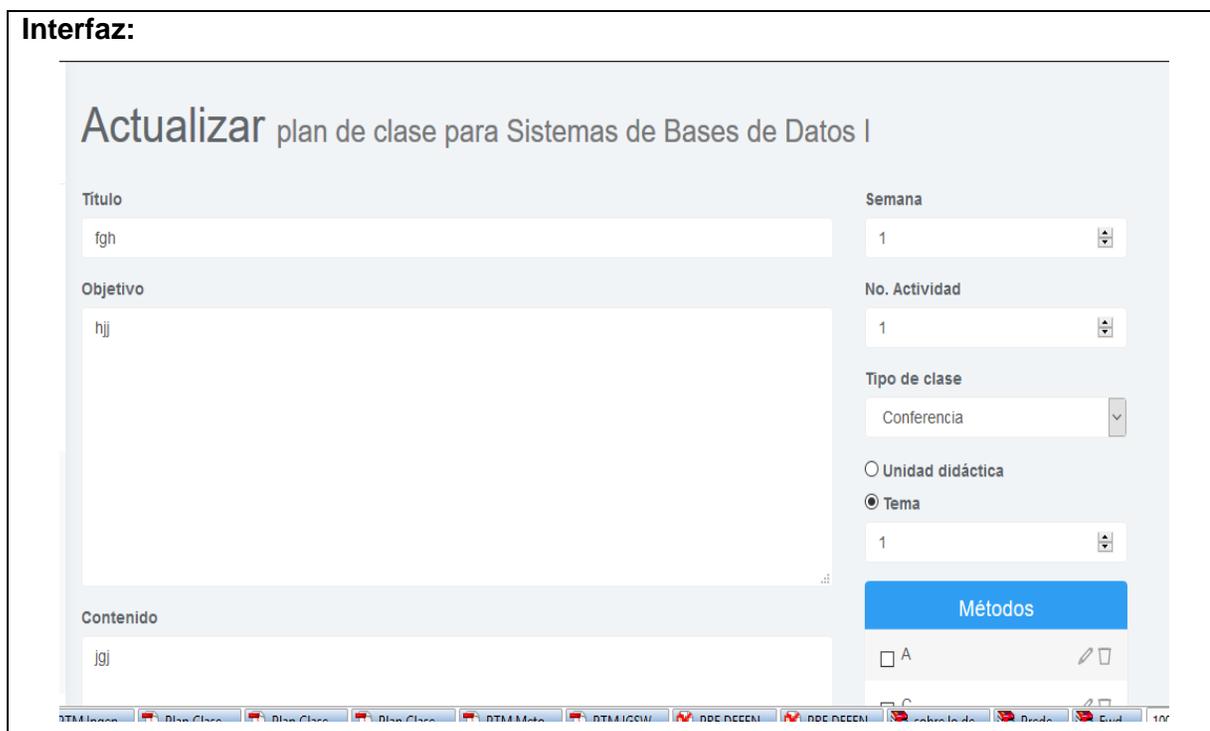


Tabla 4: Historia de usuario #2

Historia de usuario	
Número: 3	Nombre historia de usuario: Eliminar plan de clase
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 3
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.6	
<p>Descripción: La presente historia de usuario tiene como objetivo eliminar un plan de clase. Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello. El usuario deberá eliminar un plan de clase registrado en el sistema.</p>	
Observaciones: La entrada al sistema.	
<p>Interfaz:</p> 	

Tabla 5: Historia de usuario #3

Historia de usuario	
Número: 4	Nombre historia de usuario: Mostrar plan de clase
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 3
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.4	
<p>Descripción: La presente historia de usuario tiene como objetivo mostrar un plan de clase registrado en el sistema.</p> <p>Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello. El usuario podrá listar sus planes de clase.</p>	
Observaciones: La entrada al sistema.	
<p>Interfaz:</p> 	

Tabla 6: Historia de usuario #4

Historia de usuario	
Número: 5	Nombre historia de usuario: Crear plan de control a clase
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 4

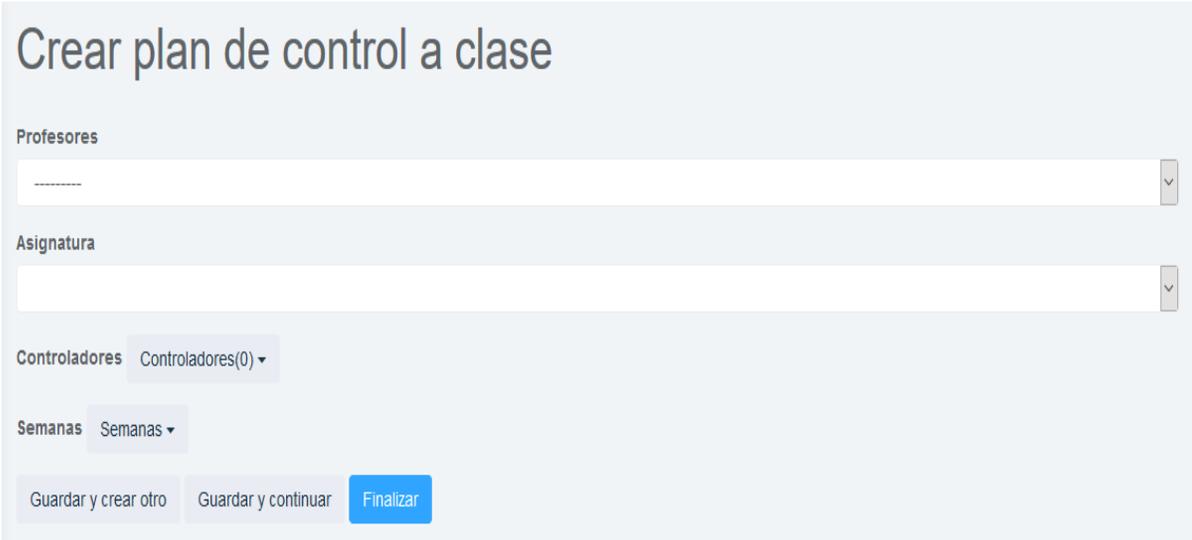
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.6	
<p>Descripción: La presente historia de usuario tiene como objetivo crear un plan de control a clase.</p> <p>Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello. El usuario deberá llenar los siguientes campos para llenar un control: profesor, asignatura, controladores y semana.</p>	
Observaciones: La entrada al sistema.	
<p>Interfaz:</p> 	

Tabla 7: Historia de usuario #5

Historia de usuario	
Número: 6	Nombre historia de usuario: Modificar plan de control a clase
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 4
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.4	
<p>Descripción: La presente historia de usuario tiene como objetivo modificar un control a clase.</p>	

Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello. El usuario podrá modificar algunos de los siguientes campos: profesor, asignatura, controladores y semana.

Observaciones: La entrada al sistema.

Interfaz:

Tabla 8: Historia de usuario #6

Historia de usuario	
Número: 7	Nombre historia de usuario: Eliminar plan de control a clase
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 4
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.6	
Descripción: La presente historia de usuario tiene como objetivo eliminar un control a clase. Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello.	
Observaciones: La entrada al sistema.	
Interfaz:	

Se eliminará el plan de **Control a Andy Reinoso Brito**

Confirmar

Cancelar

Tabla 9: Historia de usuario #7

Historia de usuario																																					
Número: 8	Nombre historia de usuario: Mostrar plan de control a clase																																				
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 4																																				
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio																																				
Tiempo estimado: 0.4																																					
<p>Descripción: La presente historia de usuario tiene como objetivo listar todos los controles a clase creados en el plan de control a clase registrado en el sistema. El usuario podrá listar los controles a clase planificados.</p> <p>Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello.</p>																																					
Observaciones: La entrada al sistema.																																					
Interfaz:																																					
<table border="1"> <thead> <tr> <th>Profesor</th> <th>Asignatura</th> <th>Controladores</th> <th>Semana</th> <th>Resultado</th> <th></th> </tr> </thead> <tbody> <tr> <td>Andy Reinoso Brito</td> <td>Metodología de la Investigación Científica</td> <td>Dianelis Ferrer Maestrey</td> <td>Semana 2</td> <td></td> <td>  </td> </tr> <tr> <td>Andy Reinoso Brito</td> <td>Metodología de la Investigación Científica</td> <td>Dianelis Ferrer Maestrey</td> <td>Semana 2</td> <td></td> <td>  </td> </tr> <tr> <td>Andy Reinoso Brito</td> <td>Metodología de la Investigación Científica</td> <td>Dianelis Ferrer Maestrey</td> <td>Semana 2</td> <td></td> <td>  </td> </tr> <tr> <td>Andy Reinoso Brito</td> <td>Metodología de la Investigación Científica</td> <td>Dianelis Ferrer Maestrey</td> <td>Semana 3</td> <td></td> <td>  </td> </tr> <tr> <td>Andy Reinoso Brito</td> <td>Metodología de la Investigación Científica</td> <td>Dianelis Ferrer Maestrey</td> <td>Semana 4</td> <td></td> <td>  </td> </tr> </tbody> </table>		Profesor	Asignatura	Controladores	Semana	Resultado		Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 2		  	Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 2		  	Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 2		  	Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 3		  	Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 4		  
Profesor	Asignatura	Controladores	Semana	Resultado																																	
Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 2		  																																
Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 2		  																																
Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 2		  																																
Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 3		  																																
Andy Reinoso Brito	Metodología de la Investigación Científica	Dianelis Ferrer Maestrey	Semana 4		  																																

Tabla 10: Historia de usuario #8

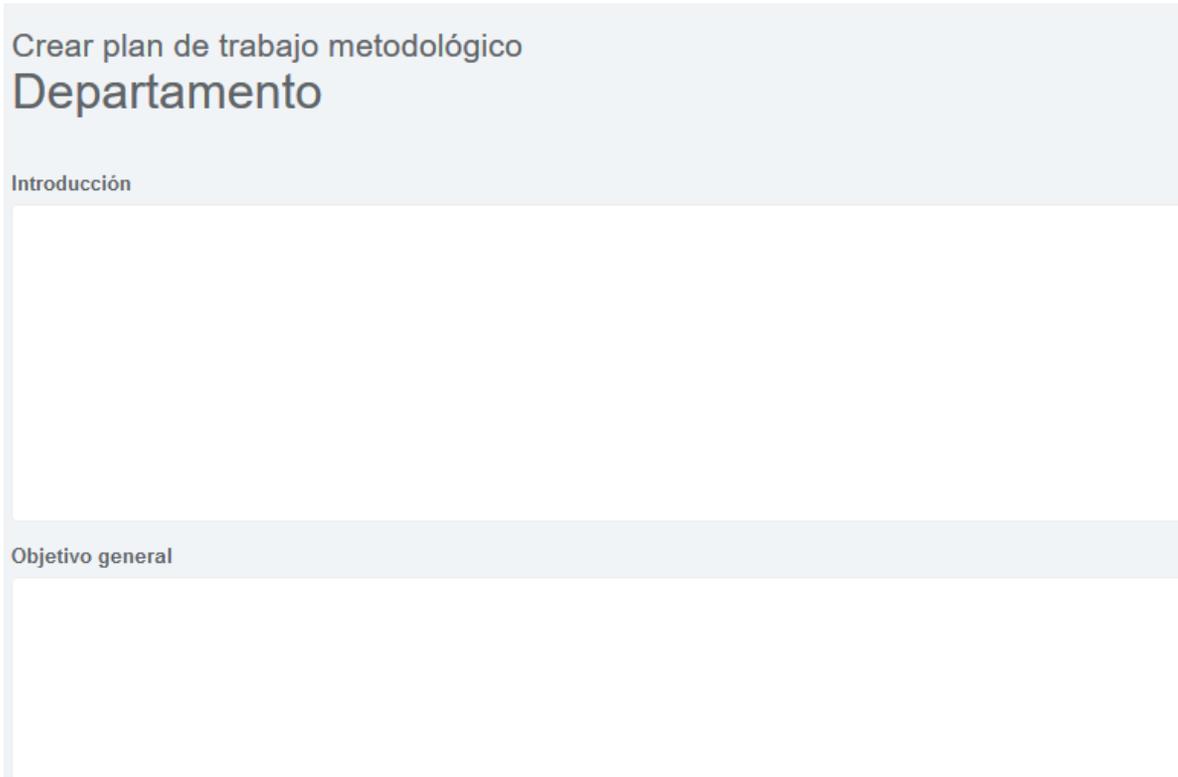
Historia de usuario	
Número: 9	Nombre historia de usuario: Crear plan de trabajo metodológico
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 5
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.6	
<p>Descripción: La presente historia de usuario tiene como objetivo crear un plan de trabajo metodológico.</p> <p>Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello. El usuario deberá llenar los campos de: introducción, caracterización del claustro, fundamentación, principales dificultades detectadas, objetivo general, indicadores, lineamientos de trabajo, sistema de actividades.</p>	
Observaciones: La entrada al sistema.	
<p>Interfaz:</p> 	

Tabla 11: Historia de usuario #9

Historia de usuario	
Número: 10	Nombre historia de usuario: Modificar plan de trabajo metodológico
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 5
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.4	
<p>Descripción: La presente historia de usuario tiene como objetivo modificar un plan de trabajo metodológico.</p> <p>Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello. El usuario podrá modificar algunos de los siguientes campos: introducción, caracterización del claustro, fundamentación, principales dificultades detectadas, objetivo general, indicadores, lineamientos de trabajo, sistema de actividades.</p>	
Observaciones: La entrada al sistema.	
<p>Interfaz:</p> 	

Tabla 12: Historia de usuario #10

Historia de usuario	
Número: 11	Nombre historia de usuario: Eliminar plan de trabajo metodológico
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 5
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio

Tiempo estimado: 0.6
Descripción: La presente historia de usuario tiene como objetivo eliminar un plan de trabajo metodológico. El usuario podrá eliminar un plan de trabajo metodológico registrado en el sistema. Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello.
Observaciones: La entrada en el sistema.
Interfaz: 

Tabla 13: Historia de usuario #11

Historia de usuario	
Número: 12	Nombre historia de usuario: Mostrar plan de trabajo metodológico
Usuario: Angela María Lescaille Otaño Dianelis Ferrer Maestrey	Iteración asignada: 5
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Tiempo estimado: 0.4	
Descripción: La presente historia de usuario tiene como objetivo mostrar el plan de trabajo metodológico. Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello.	
Observaciones: La entrada en el sistema.	
Interfaz:	

PTM IGSW Fac2 - 2016-2017, 2do Semestre

Actualizar Exportar Imprimir Eliminar

Plan de Trabajo Metodológico
 Departamento
 Ingeniería de Software
 Facultad 2
 2016-2017, 2do Semestre
 Elaborado por: Dianelis Ferrer Maestrey
 Aprobado por:

Tabla 14: Historia de usuario #12

2.6. Estimación de esfuerzo

Las estimaciones de esfuerzo asociado a la implementación de las historias de usuario 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. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016).

Número de HU	Nombre de las HU	Puntos de estimación en semanas
HU-1	Crear plan de clase.	0.6
HU-2	Modificar plan de clase.	0.4
HU-3	Eliminar plan de clase.	0.6

HU-4	Mostrar plan de clase.	0.4
HU-5	Crear plan de control a clase.	0.6
HU-6	Modificar plan de control a clase.	0.4
HU-7	Eliminar plan de control a clase.	0.6
HU-8	Mostrar plan de control a clase.	0.4
HU-9	Crear plan de trabajo metodológico.	0.6
HU-10	Modificar plan de trabajo metodológico.	0.4
HU-11	Eliminar plan de trabajo metodológico.	0.6
HU-12	Mostrar plan de trabajo metodológico.	0.4
HU-13	Crear actividad metodológica.	0.4
HU-14	Modificar actividad metodológica.	0.2
HU-15	Eliminar actividad metodológica.	0.2
HU-16	Mostrar actividad metodológica.	0.2
HU-17	Mostrar deficiencias comunes.	1
HU-18	Crear claustro	0.2
HU-19	Modificar claustro	0.2
HU-20	Eliminar claustro	0.2
HU-21	Mostrar claustro	0.2
HU-22	Crear lineamientos	0.2
HU-23	Modificar lineamientos	0.2
HU-24	Eliminar lineamientos	0.2
HU-25	Mostrar lineamientos	0.2
HU-26	Crear indicador	0.2
HU-27	Modificar indicador	0.2
HU-28	Eliminar indicador	0.2
HU-29	Mostrar indicador	0.2
HU-30	Crear deficiencia	0.2
HU-31	Modificar deficiencia	0.2
HU-32	Eliminar deficiencia	0.2
HU-33	Mostrar deficiencia	0.2
HU-34	Configurar años para impartir asignaturas	0.4

HU-35	Configurar curso docente	0.4
HU-36	Autenticar usuario	0.6
HU-37	Actualizar perfil	0.4
HU-38	Modificar profesor	0.2
HU-39	Eliminar profesor	0.2
HU-40	Mostrar profesor	0.2
HU-41	Notificar actividad metodológica al correo	0.4
HU-42	Mostrar controles a realizar	0.4
HU-43	Buscar profesor visitado	0.4
HU-44	Subir resultados del control a clases	0.4
HU-45	Mostrar resumen de controles realizados	0.4
HU-46	Mostrar evaluaciones	0.2
HU-47	Buscar evaluaciones de cursos anteriores	0.4
HU-48	Mostrar resultados del control a clase realizado	0.4
HU-49	Crear plan de clase a partir de un plan creado	0.2
HU-50	Buscar plan de clase	0.2
HU-51	Buscar planes de clases de cursos anteriores	0.4
HU-52	Listar planes de trabajo metodológicos actuales	0.4
HU-53	Listar planes de trabajo metodológicos creados	0.2
HU-54	Exportar plan de clase a PDF	0.4
HU-55	Imprimir plan de clase	0.4
HU-56	Exportar plan de trabajo metodológico a PDF	0.4
HU-57	Imprimir plan de trabajo metodológico	0.4

Tabla 15: Estimación de Esfuerzo

2.7. Plan de iteraciones

Iteración 1: En esta iteración se implementarán las HU de prioridad media, relacionadas con el registro y configuración del sistema.

Iteración 2: En esta iteración se implementarán las HU de prioridad media, relacionadas con los datos que recoge el plan de trabajo metodológico.

Iteración 3: En esta iteración se implementarán las HU de prioridad alta que gestionan el plan de clase.

Iteración 4: En esta iteración se implementarán las HU de prioridad alta relacionadas con la gestión del plan de control a clase y mostrar deficiencias comunes.

Iteración 5: En esta iteración se implementarán las HU de prioridad alta relacionadas con la gestión del plan de trabajo metodológico y de las actividades metodológicas.

Iteración 6: En esta iteración se implementarán las HU de prioridad baja relacionadas con el plan de control a clase como subir resultados del control y/o mostrar controles a realizar, entre otras. Además de la funcionalidad que manda la notificación al correo de la actividad metodológica planificada.

Iteración 7: En esta iteración se implementarán las HU de prioridad baja, relacionadas con buscar los planes de clase, listar los planes de trabajo metodológicos, así como las relacionadas con imprimir y exportar a formato pdf.

Iteración	Historia de usuario por iteraciones	Duración de las iteraciones
Iteración 1	HU-34,HU-35, HU-36, HU-37, HU-38, HU-39, HU-40	3 semanas
Iteración 2	HU-18, HU-19, HU-20, HU-21, HU-22,HU-23, HU-24, HU-25, HU-26, HU-27, HU-28, HU-29, HU-30, HU-31, HU-32, HU-33	3 semanas
Iteración 3	HU-1, HU-2, HU-3, HU-4	2 semanas
Iteración 4	HU-5, HU-6, HU-7, HU-8, HU-17	3 semanas
Iteración 5	HU-9, HU-10, HU-11, HU-12, HU-13, HU-14, HU-15, HU-16	3 semanas
Iteración 6	HU-41, HU-42,HU-43,HU-44, HU-45,HU-46, HU-47, HU-48	3 semanas
Iteración 7	HU-49, HU-50, HU-51, HU-52, HU-53,HU-54, HU-55, HU-56, HU-57	3 semanas

Total	57	20 semanas
--------------	-----------	------------

Tabla 16: Plan de iteraciones

2.8. Plan de entrega

El plan de entrega está compuesto por iteraciones de no más de tres semanas. Basándose en las historias de usuario definidas para el desarrollo de la aplicación web, se elabora el siguiente plan de entrega, el cual muestra las historias de usuario que se llevarán a cabo en cada iteración.

Sistema Gestión Metodológica		
Iteración	Historia de usuario por iteraciones	Fecha de entrega
Iteración 1	HU-34,HU-35, HU-36, HU-37, HU-38, HU-39, HU-40	24 de enero
Iteración 2	HU-18, HU-19, HU-20, HU-21, HU-22,HU-23, HU-24, HU-25, HU-26, HU-27, HU-28, HU-29, HU-30, HU-31, HU-32, HU-33	15 de febrero
Iteración 3	HU-1, HU-2, HU-3, HU-4	28 de febrero
Iteración 4	HU-5, HU-6, HU-7, HU-8, HU-17	21 de marzo
Iteración 5	HU-9, HU-10, HU-11, HU-12, HU-13, HU-14, HU-15, HU-16	11 de abril
Iteración 6	HU-41, HU-42,HU-43,HU-44, HU-45,HU-46, HU-47, HU-48	3 de mayo
Iteración 7	HU-49, HU-50,HU-51, HU-52, HU-53,HU-54, HU-55, HU-56, HU-57	24 de mayo

Tabla 17: Plan de Entrega

2.9. Conclusiones del capítulo

En el capítulo fueron descritas las fases de Exploración y Planificación para la propuesta de solución, lo que permitió la obtención de los artefactos necesarios para la planeación de la aplicación web en correspondencia a la metodología seleccionada. El desarrollo de estas etapas permitió que el proceso fuera dividido en 7 iteraciones en las que se desarrollaron las HU de prioridad alta durante las iteraciones 3, 4 y 5. Se desarrolló las funcionalidades de

prioridad media en las iteraciones 1 y 2, y las de prioridad baja en las iteraciones 6 y 7. Se determinó que el proceso de desarrollo durará un total de 20 semanas, divididas en 7 iteraciones de no más de 3 semanas respectivamente.

Capítulo 3: Producción y validación de la aplicación web propuesta

En este capítulo se realiza el diseño de la base de datos, así como se definen el patrón arquitectónico a utilizar. Se analizan los patrones de diseño empleados en la aplicación web. Además, se especifican los artefactos de la metodología de software utilizada en sus fases de iteraciones y producción, como son las implementaciones de las diferentes iteraciones y las pruebas unitarias y de aceptación.

3.1 Patrón arquitectónico

Para el desarrollo del sistema el patrón a seguir es Modelo-Vista-Plantilla.

Uno de los framework a utilizar para el desarrollo de la solución es Django, el cual usa el Modelo-Vista-Plantilla (MVP) como una modificación del patrón Modelo-Vista-Controlador (MVC), empleado en la implantación de la solución. En Django el modelo sigue siendo modelo, la vista pasa a hacer la plantilla y el controlador la vista (Revista de Información, 2017).

Modelo: la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos mostrando: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tienen, y las relaciones entre ellos.

Vista: la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: es como un puente entre los modelos y las plantillas.

Plantilla: la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.



Fig.1- Patrón arquitectónico Modelo-Vista-Plantilla (MVP)

3.2 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. El catálogo "Design Patterns: Elements of Reusable Object-Oriented Software" impone la línea base en el tema de los patrones (Gamma, Helm, Johnson, & Vlissides, 1994).

Patrones para Asignar Responsabilidades (GRASP)

Alta Cohesión: Define que las responsabilidades asignadas a una misma clase deben guardar relación y estar enfocadas al mismo objetivo. Fomenta de esta forma que las clases no adquieran una complejidad excesiva o realicen un trabajo enorme, evitando así que una clase sea la única responsable de muchas tareas en áreas funcionales muy heterogéneas.

```
def control_done(self):  
    return date.today() > self.fecha  
  
def has_result(self):  
    return Control.objects.filter(plan=self.pk).exists()
```

Bajo Acoplamiento: Define como fomentar una menor dependencia y una mayor reutilización. Plantea mantener la fuerza de las conexiones entre clases lo más bajo posible. Este parámetro es conocido como acoplamiento. No puede verse de forma aislada como Experto o Alta Cohesión. El acoplamiento tal vez no sea muy importante, sino se busca la reutilización. En términos generales, suelen tener escaso acoplamiento las clases muy genéricas y con grandes posibilidades de ser reutilizadas.

```
control_urlpatterns = [  
    # url(r"^list", control_crud.ControlList.as_view(),  
    name='list'),  
    url(r"^create/(?P<plan_id>\d+)$",  
    control_crud.CrearControl.as_view(), name='create'),  
    url(r"^detail/(?P<pk>\d+)$",  
    control_crud.ControlDetail.as_view(), name='detail'),  
    url(r"^delete/(?P<pk>\d+)$",  
    control_crud.ControlDelete.as_view(), name='delete'),  
    url(r"^update/(?P<pk>\d+)$",
```

```
control_crud.ControlUpdate.as_view(), name='update'),  
]
```

Controlador: Define quien debería encargarse de atender un evento del sistema. Asigna la responsabilidad de administrar estos mensajes a una clase que represente una de las siguientes opciones: el negocio o la organización global, el sistema, un controlador de papeles o una clase artificial que represente el caso de uso. Además, sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

```
class ControlUpdate(CreateUpdateMixin, UpdateView):  
    action = 'Actualizar' # usado para informar en la interfaz que se va a hacer, ej: el texto del btn del  
    form  
  
    def get_context_data(self, **kwargs):  
        context = super(ControlUpdate, self).get_context_data(**kwargs)  
        context['key'] = 'value'  
        return context  
  
class ControlDetail(DetailView):  
    model = Control  
  
class ControlDelete(DeleteView):  
    model = Control  
    success_url = reverse_lazy('')
```

Experto: Define el principio fundamental en virtud del cual se asignan las responsabilidades en un diseño orientado a objetos. Plantea que la responsabilidad debe recaer en la clase que cuenta con la información necesaria para cumplir con ella. De forma tal que el sistema sea más fácil de entender, mantener y ampliar, permitiendo reutilizar los componentes creados en futuras aplicaciones.

Creador: Define quien debe ser el responsable de crear una nueva instancia de una clase. Plantea que debe recaer en la clase que agrega, contiene, registra, utiliza o tiene los datos de inicialización de la nueva instancia. La prioridad en caso de que exista más de una posibilidad es en el orden enunciado. En otras palabras, el propósito fundamental de este patrón es asignar responsabilidades relacionadas con la creación de objetos producidos en cualquier evento. Además, facilita un Bajo Acoplamiento, supone menor dependencia respecto al mantenimiento y mejores oportunidades de reutilización.

```
class CreateView(SingleObjectTemplateResponseMixin, BaseCreateView):  
    """  
    View for creating a new object instance,  
    with a response rendered by template.  
    """  
    template_name_suffix = '_form'
```

Patrones GOF

Creación: Son los encargados de “abstraer el proceso de instanciación o creación de objetos, ayudan a que el sistema sea independiente de cómo sus objetos son creados, integrados y representados. Los patrones que forman parte de esta categoría son (Factoría (Factory), Singleton).

Singleton: Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. El sistema posee una instancia única del controlador frontal disponible mediante este patrón para lograr una vía de entrada única a las solicitudes.

Factory: Proporciona una interfaz para la creación de objetos interdependientes o interrelacionados, sin especificar sus clases concretas. El sistema utiliza este patrón en la implementación de un conjunto de clases para el acceso a datos.

Estructurales: separan la interfaz de la implementación y se encargan de guiar la forma en que las clases y los objetos se combinan para formar nuevas estructuras más complejas y proporcionar nuevas funcionalidades. También se encargan de lograr que los cambios en los requisitos de la aplicación no ocasionen cambios en las relaciones existentes entre objetos. Los

patrones pertenecientes a esta categoría son los siguientes: Adaptador, Apoderado, Decorador, Puente, Composición, Fachada y Peso Mosca. Dentro de los patrones Estructurales se utilizó:

Decorador: Es el encargado de asignarle responsabilidades a objetos y añadir funcionalidades de manera dinámica.

Comportamiento: Ayudan a definir la comunicación e interacción de los objetos. Reducir acoplamiento (Observer).

Observador: Define una dependencia de uno a muchos entre objetos, tal que, cuando un objeto cambia de estado todos los que son dependientes de este son actualizados automáticamente.

3.3 Diseño de la base de datos

Una base de datos correctamente diseñada permite obtener acceso a la información exacta y actualizada, por lo que una estructura adecuada es esencial para lograr los objetivos fijados para la aplicación web propuesta.

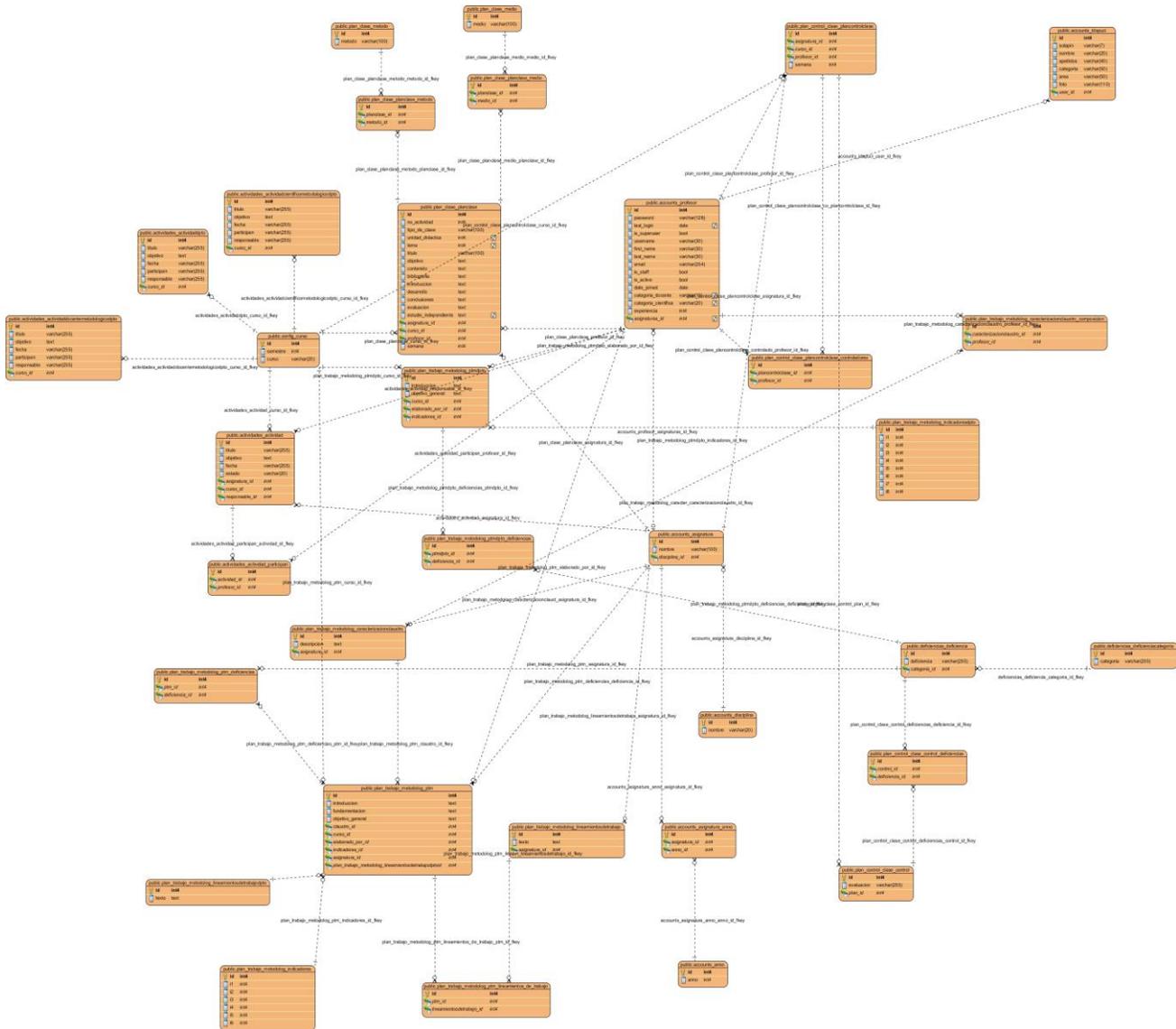


Fig.2- Modelo Entidad-Relación

3.4 Tareas de ingeniería

Componen el conjunto de acciones llevadas a cabo para darle cumplimiento a las distintas historias de usuario, las cuales son realizadas para la especificación de las operaciones ejecutadas por los programadores en cada una de ellas. Estas serán representadas mediante tablas divididas por las siguientes secciones (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016):

Número tarea: los números deben ser consecutivos.

Número historia de usuario: número de la historia de usuario a la que pertenece la tarea.

Nombre tarea: nombre que identifica a la tarea.

Tipo de tarea: las tareas pueden ser de: desarrollo, corrección, mejora, otra (especificar)

Puntos estimados: tiempo estimado en semanas que se le asignará a su desarrollo.

Fecha inicio: fecha en que inicia el desarrollo de la tarea.

Fecha fin: fecha en que finaliza el desarrollo de la tarea.

Programador responsable: nombre y apellidos del programador.

Descripción: breve descripción de la tarea.

A continuación, se muestran las tareas de ingeniería de las HU críticas, para consultar el resto ver Anexos 3.

Tarea de Ingeniería	
Número de tarea: 1	Número de la HU: HU-1
Nombre de tarea: Implementar funcionalidad crear plan de clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 15/2/2017	Fecha fin: 17/2/2017
Programador responsable: Angela María Lescaille Otaño y Danelis Ferrer Maestrey.	
Descripción: Se implementan la funcionalidad que permite obtener los datos del plan de clase a insertar (actividad, semana, tipo de clase, unidad didáctica o tema, título, objetivo, contenido, bibliografía, métodos, medios, tipo de evaluación, introducción, desarrollo, conclusiones, estudio independiente). Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la manera esperada.	

Tabla 18: Tarea de Ingeniería #1

Tarea de Ingeniería	
Número de tarea: 2	Número de la HU: HU-2
Nombre de tarea: Implementar funcionalidad modificar plan de clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 20/2/2017	Fecha fin: 21/2/2017
Programador responsable: Angela María Lescaille Otaño y Danelis Ferrer Maestrey.	

Descripción: Se modifica un plan de clase ya creado. Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la manera esperada.

Tabla 19: Tarea de Ingeniería #2

Tarea de Ingeniería	
Número de tarea: 3	Número de la HU: HU-3
Nombre de tarea: Implementar funcionalidad eliminar plan de clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 22/2/2017	Fecha fin: 24/2/2017
Programador responsable: Angela María Lescaille Otaño y Danelis Ferrer Maestrey.	
Descripción: Se elimina un plan de clase ya creado.	

Tabla 20: Tarea de Ingeniería #3

Tarea de Ingeniería	
Número de tarea: 4	Número de la HU: HU-4
Nombre de tarea: Implementar funcionalidad mostrar plan de clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 27/2/2017	Fecha fin: 28/2/2017
Programador responsable: Angela María Lescaille Otaño y Danelis Ferrer Maestrey.	
Descripción: Muestra un plan de clase creado.	

Tabla 21: Tarea de Ingeniería #4

Tarea de Ingeniería	
Número de tarea: 5	Número de la HU: HU-5
Nombre de tarea: Implementar funcionalidad crear plan de control a clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6

Fecha inicio: 1/3/2017	Fecha fin: 3/3/2017
Programador responsable: Angela María Lescaille Otaño y Dianelis Ferrer Maestrey.	
Descripción: Se implementan las funcionalidades que permiten obtener los datos del plan de control a clase a insertar (profesor, asignatura, controladores y semana). Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la manera esperada, sin dejarse datos nulos.	

Tabla 22: Tarea de Ingeniería #5

Tarea de Ingeniería	
Número de tarea: 6	Número de la HU: HU-6
Nombre de tarea: Implementar funcionalidad modificar plan de control a clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 6/3/2017	Fecha fin: 7/3/2017
Programador responsable: Angela María Lescaille Otaño y Dianelis Ferrer Maestrey.	
Descripción: Permite modificar un plan de control a clase ya creado. Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la manera esperada, sin dejarse datos nulos.	

Tabla 23: Tarea de Ingeniería #6

Tarea de Ingeniería	
Número de tarea: 7	Número de la HU: HU-7
Nombre de tarea: Implementar funcionalidad eliminar un plan de control a clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 8/3/2017	Fecha fin: 10/3/2017
Programador responsable: Angela María Lescaille Otaño y Dianelis Ferrer Maestrey.	
Descripción: Elimina un plan de control a clase ya creado.	

Tabla 24: Tarea de Ingeniería #7

Tarea de Ingeniería	
Número de tarea: 8	Número de la HU: HU-8
Nombre de tarea: Implementar funcionalidad modificar plan de control a clase.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 13/3/2017	Fecha fin: 14/3/2017
Programador responsable: Angela María Lescaille Otaño y Dianelis Ferrer Maestrey.	
Descripción: Muestra el plan de control a clase ya creado.	

Tabla 25: Tarea de Ingeniería #8

Tarea de Ingeniería	
Número de tarea: 9	Número de la HU: HU-9
Nombre de tarea: Implementar funcionalidad crear plan de trabajo metodológico.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 22/3/2017	Fecha fin: 24/3/2017
Programador responsable: Angela María Lescaille Otaño y Dianelis Ferrer Maestrey	
Descripción: Se implementan las funcionalidades que permiten obtener los datos del plan de trabajo metodológico a insertar (introducción, caracterización del claustro, fundamentación, principales dificultades detectadas, objetivo general, indicadores, lineamientos de trabajo, sistema de actividades). Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la manera esperada, sin dejarse datos nulos.	

Tabla 26: Tarea de Ingeniería #9

Tarea de Ingeniería	
Número de tarea: 10	Número de la HU: HU-10
Nombre de tarea: Implementar funcionalidad modificar plan de trabajo metodológico.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4

Fecha inicio: 27/3/2017	Fecha fin: 28/3/2017
Programador responsable: Angela María Lescaille Otaño y Dianelis Ferrer Maestrey	
Descripción: Permite modificar un plan de trabajo metodológico ya creado. Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la manera esperada, sin dejarse datos nulos.	

Tabla 27: Tarea de Ingeniería #10

Tarea de Ingeniería	
Número de tarea: 11	Número de la HU: HU-11
Nombre de tarea: Implementar funcionalidad eliminar plan de trabajo metodológico.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.6
Fecha inicio: 29/3/2017	Fecha fin: 31/3/2017
Programador responsable: Angela María Lescaille Otaño y Dianelis Ferrer Maestrey	
Descripción: Elimina un plan de trabajo metodológico ya creado.	

Tabla 28: Tarea de Ingeniería #11

Tarea de Ingeniería	
Número de tarea: 12	Número de la HU: HU-12
Nombre de tarea: Implementar funcionalidad mostrar plan de trabajo metodológico.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha inicio: 3/4/2017	Fecha fin: 4/4/2017
Programador responsable: Angela María Lescaille Otaño y Dianelis Ferrer Maestrey	
Descripción: Muestra el plan de trabajo metodológico creado.	

Tabla 29: Tarea de Ingeniería #12

3.5 Clase-Responsabilidad-Creador

La metodología de desarrollo XP como parte de la fase de diseño propone el modelado de

Clase-Responsabilidad-Colaborador (CRC), lo que constituye un modelo simple de organizar las clases más relevantes para las funcionalidades del sistema. Este modelado CRC utiliza tarjetas, con el objetivo de desarrollar una representación organizada de las clases, las cuales tienen los siguientes datos (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016):

Nombre: nombre de la clase.

Responsabilidades: Funcionalidades que va a desarrollar esa clase.

Colaboraciones: Clase o clases de las que crea dependencia.

A continuación, se muestran las tarjetas CRC.

Clase plan_clase	
Descripción: Clase que realiza el plan de clases.	
Responsabilidad:	Colaborador:
Crear plan de clase.	PlanClaseForm
Modificar plan de clase.	PlanClaseSearchForm
Eliminar plan de clase.	
Mostrar plan de clase.	

Tabla 30: CRC Clase plan_clase

Clase plan_control_clase	
Descripción: Clase que realiza el plan de control a clases.	
Responsabilidad:	Colaborador:
Crear plan de control a clase.	ControlForm
Modificar plan de control a clase.	PlanControlForm
Eliminar plan de control a clase.	EvaluacionSearchForm
Mostrar plan de control a clase.	

Tabla 31: CRC Clase plan_control_clase

Clase plan_trabajo_metodolog	
Descripción: Clase que realiza el plan de trabajo metodológico.	
Responsabilidad:	Colaborador:
Crear plan de trabajo metodológico.	IndicadoresForm
Modificar plan de trabajo metodológico.	PTMForm

Eliminar plan de trabajo metodológico.	ClaustroDptoForm
Mostrar plan de trabajo metodológico.	IndicadoresDptoForm PTMDptoForm

Tabla 32: CRC Clase plan_trabajo_metodolog

Clase deficiencias	
Descripción: Clase que realiza la gestión de las deficiencias.	
Responsabilidad:	Colaborador:
Crear nuevo indicador.	NuevaDeficienciaForm
Crear nueva deficiencia.	

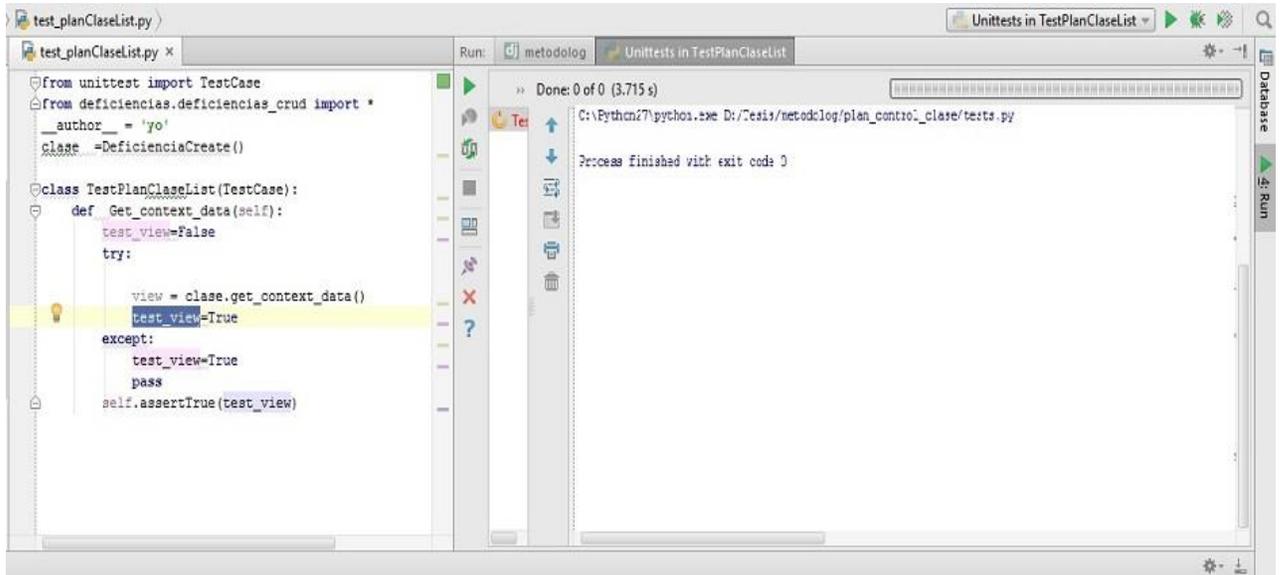
Tabla 33: CRC Clase deficiencias

3.6 Pruebas

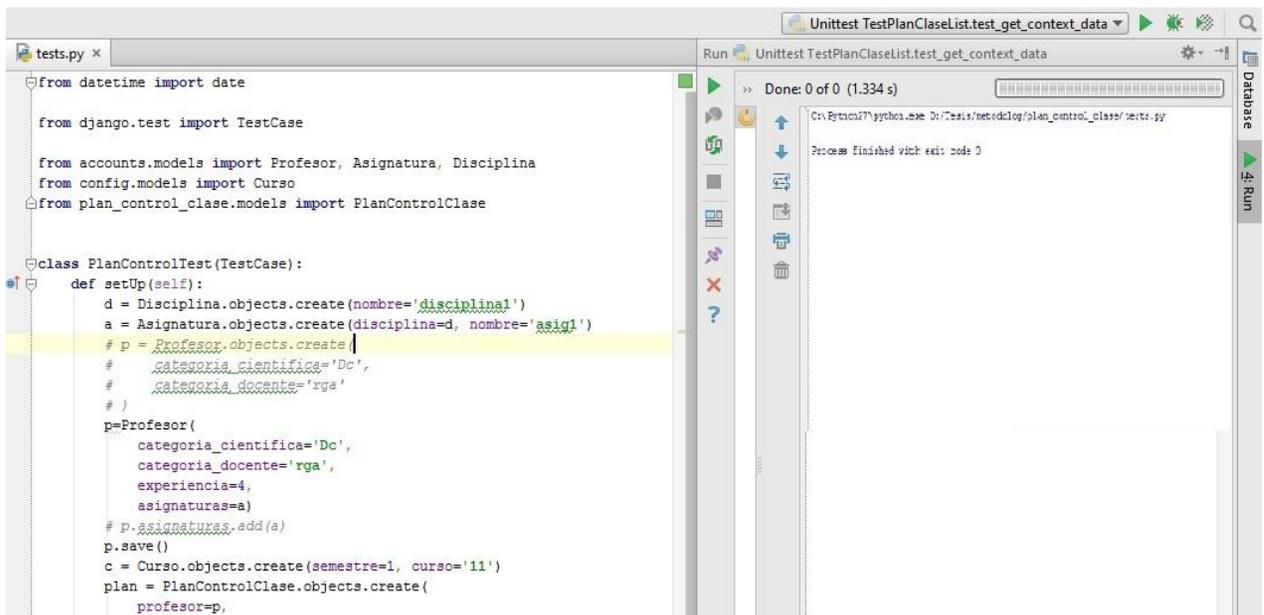
Las pruebas del software permiten validar la calidad del sistema a través de pruebas al diseño y la codificación, entre otras. Prueban si el software hace lo que debe o si se comporta de otra manera. Para ello, se aplican métodos de prueba y técnicas necesarias para poder evaluar el sistema de acuerdo a los parámetros que se quieran probar. Estas validan que el sistema realmente constituye una respuesta de calidad a las necesidades del cliente (Calabria & Píriz, 2003).

3.6.1 Pruebas unitarias

Las pruebas unitarias son pruebas de caja blanca encargadas de verificar el código de forma automática para ello se utilizó la librería instalada en el PyCharm, Unittests. En el presente trabajo se muestran el resultado de las pruebas unitarias aplicada a las funcionalidades con prioridad alta, donde todas las pruebas tuvieron un resultado satisfactorio como se muestra en la siguiente imagen.



Prueba unitaria 1: ListarPlanClase



Prueba unitaria 2: Plan_control_clase

```

# coding=utf-8
"""
Este modulo esta entendido para los ptm de las asignaturas.
donde quiera q se observe la variable ptm significa q es un ptm de alguna
"""
import ...

def get_success_url(user):
    return reverse_lazy('ptm:dpto:index') if user.is_jefe_dpto else \
        reverse_lazy('ptm:asig:index')

@no_curso_configured_yet
@permission_required('plan_trabajo_metodolog.add_ptm')
def ptm_create(request):
    claustrero_qs = CaracterizacionClaustrero.objects.filter(asignatura=request.user.asignatura)
    ltd = LineamientosDeTrabajo.objects.filter(asignatura=request.user.asignatura)
    if request.method == 'POST':
        ptm_form = PTMForm(request.POST)
        indicadores_form = IndicadoresForm(request.POST)
        if ptm_form.is_valid() and indicadores_form.is_valid():
            i = indicadores_form.save()
            ptm = ptm_form.save(commit=False)
            ptm.indicadores = i
            ptm.asignatura = request.user.asignaturas
            ptm.elaborado_por = request.user
            if claustrero_qs.exists():
                ptm.claustrero = claustrero_qs[0] # evitar de forma facil el

```

Prueba unitaria 3: PlanTrabajoMetodológico

```

# coding=utf-8
import ...

class CreateUpdateMixin(LoginRequiredMixin):
    template_name = 'plan_clase/plan_clase_create.html'
    model = Deficiencia
    fields = []
    success_url = reverse_lazy('')

    def get_context_data(self, **kwargs):
        context = super(CreateUpdateMixin, self).get_context_data(**kwargs)
        context['action'] = self.action
        return context

class DeficienciaCreate(CreateUpdateMixin, CreateView):
    action = 'Crear' # usado para informar en la interfaz que se va a hacer

    def get_context_data(self, **kwargs):
        context = super(DeficienciaCreate, self).get_context_data(**kwargs)
        context['key'] = 'value'
        return context

class DeficienciaUpdate(CreateUpdateMixin, UpdateView):
    action = 'Actualizar' # usado para informar en la interfaz que se va a hacer

```

Prueba unitaria 4: Deficiencias

3.6.2 Pruebas de aceptación

Las pruebas de aceptación significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente por esto, el cliente es la persona adecuada para diseñar las pruebas de aceptación. Son pruebas de caja negra que se crean a partir de las HU. En estas serán probadas las funcionalidades exigidas por el cliente, descritas en las HU que se han implementado. Se llevarán a cabo redactando los casos de prueba, teniendo en

cuenta el orden de las HU y la prioridad que ha sido asignada a las funcionalidades (Calabria, et al., 2003).

Las pruebas de aceptación correspondiente a cada una de las funcionalidades serán representadas mediante tablas divididas por las secciones siguientes:

- Código de la prueba de aceptación.
- Número de la historia de usuario a la que se le realiza la prueba.
- Nombre de la funcionalidad.
- Descripción de la funcionalidad.
- Condiciones de ejecución de la funcionalidad.
- Entrada y pasos de ejecución que realiza el usuario con el objetivo de obtener el resultado esperado.
- Resultado esperado.
- Evaluación de la prueba.

A continuación, se muestran los casos de prueba de aceptación de las HU críticas, para consultar el resto ver Anexo 5:

Caso de prueba de aceptación	
Código: HU1-P1	Número de HU: 1
Nombre: Crear plan de clase.	
Descripción: El sistema permitirá crear los planes de clase de los usuarios.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, escoge el módulo Plan de clases y aquí le permite crear planes de clase de la asignatura que imparte el profesor apoyándose en los campos actividad, semana, tipo de clase, unidad didáctica o tema, título, objetivo, contenido, bibliografía, métodos, medios, tipo de evaluación, introducción, desarrollo, conclusiones, estudio independiente. Después de llenado los campos mencionados escoge la opción de crear.	
Resultado esperado: El usuario al introducir los campos requeridos, da en la opción crear y este queda creado mostrándose en un listado de planes de clase.	
Evaluación de la prueba: No satisfactoria. Se crea el plan de clase con campos en blanco.	

Tabla 34: Prueba de aceptación HU1 no satisfactoria

Caso de prueba de aceptación	
Código: HU1-P2	Número de HU: 1
Nombre: Crear plan de clase.	
Descripción: El sistema permitirá crear los planes de clase de los usuarios.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, escoge el módulo Plan de clases y aquí le permite crear planes de clase de la asignatura que imparte el profesor apoyándose en los campos actividad, semana, tipo de clase, unidad didáctica o tema, título, objetivo, contenido, bibliografía, métodos, medios, tipo de evaluación, introducción, desarrollo, conclusiones, estudio independiente. Después de llenado los campos mencionados escoge la opción de crear.	
Resultado esperado: El usuario al introducir los campos requeridos, da en la opción crear y este queda creado mostrándose en un listado de planes de clase.	
Evaluación de la prueba: Satisfactoria.	

Tabla 35: Prueba de aceptación HU1

Caso de prueba de aceptación	
Código: HU2-P1	Número de HU: 2
Nombre: Modificar plan de clase.	
Descripción: El sistema permitirá modificar los planes de clase de los usuarios.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, escoge el módulo Plan de clases y va a la opción actualizar.	
Resultado esperado: Luego de modificado cualquier campo del plan de clase da en actualizar y se guarda el plan de clase actualizado.	
Evaluación de la prueba: Satisfactoria.	

Tabla 36: Prueba de aceptación HU2

Caso de prueba de aceptación	
Código: HU3-P1	Número de HU: 3

Nombre: Eliminar plan de clase.
Descripción: El sistema permitirá eliminar los planes de clase de los usuarios.
Condiciones de ejecución: La entrada al sistema.
Entrada/Pasos: El usuario accede a la interfaz principal, escoge el módulo Plan de clases y aquí le permite ver el listado de los planes de clases creados de las asignaturas que imparte el profesor luego da en la opción eliminar.
Resultado esperado: Se elimina el plan de clase seleccionado.
Evaluación de la prueba: Satisfactoria.

Tabla 37: Prueba de aceptación HU3

Caso de prueba de aceptación	
Código: HU4-P1	Número de HU: 4
Nombre: Mostrar plan de clase.	
Descripción: El sistema permitirá mostrar los planes de clase de los usuarios.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, escoge el módulo Plan de clases y aquí le mostrará un listado con todos los planes de clase creados y al escoger uno se lo mostrará.	
Resultado esperado: El usuario al elegir el módulo Plan de clases le mostrará un listado de planes de clase. Al abrir un plan de clase este se lo mostrará.	
Evaluación de la prueba: Satisfactoria.	

Tabla 38: Prueba de aceptación HU4

Caso de prueba de aceptación	
Código: HU5-P1	Número de HU: 5
Nombre: Crear plan de control a clase.	
Descripción: El sistema permitirá crear un plan de controles a clases.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, escoge el módulo Plan de control a clases y va a la opción crear plan mostrando una interfaz para llenar los campos profesor, asignatura, controladores y semanas.	

Resultado esperado: El usuario al crear un plan de clase introduce los campos requeridos, da en la opción crear y este queda creado mostrándose en un listado de controles creados.

Evaluación de la prueba: No satisfactoria. Se le muestra al usuario en los controladores el nombre del profesor que se va a controlar después de haber seleccionado al profesor controlado.

Tabla 39: Prueba de aceptación HU5 no satisfactoria

Caso de prueba de aceptación	
Código: HU5-P2	Número de HU: 5
Nombre: Crear plan de control a clase.	
Descripción: El sistema permitirá crear un plan de controles a clases.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, escoge el módulo Plan de control a clases y va a la opción crear plan mostrando una interfaz para llenar los campos profesor, asignatura, controladores y semanas.	
Resultado esperado: El usuario al crear un plan de clase introduce los campos requeridos, da en la opción crear y este queda creado mostrándose en un listado de planes de clase.	
Evaluación de la prueba: Satisfactoria.	

Tabla 40: Prueba de aceptación HU5

Caso de prueba de aceptación	
Código: HU6-P1	Número de HU: 6
Nombre: Modificar plan de control a clase.	
Descripción: El sistema permitirá modificar un plan de controles a clases.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, va al módulo Plan de control a clases, en caso de ser jefe de departamento podrá modificar cualquiera de sus campos.	
Resultado esperado: Luego de ser modificado cualquier campo del control a clase da en actualizar y se guarda el control a clase actualizado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 41: Prueba de aceptación HU6

Caso de prueba de aceptación

Código: HU7-P1	Número de HU: 7
Nombre: Eliminar plan de control a clase.	
Descripción: El sistema permitirá eliminar un plan de controles a clases.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, va al módulo Plan de control a clases, en caso de ser jefe de departamento podrá eliminarlo.	
Resultado esperado: El usuario con rol de jefe de departamento elimina un control a clase.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 42: Prueba de aceptación HU7

Caso de prueba de aceptación	
Código: HU8-P1	Número de HU: 8
Nombre: Mostrar plan de control a clase.	
Descripción: El sistema permitirá mostrar un plan de controles a clases.	
Condiciones de ejecución: La entrada al sistema.	
Entrada/Pasos: El usuario accede a la interfaz principal, va al módulo Plan de control a clases, en caso de ser controlador, aparecerá una interfaz con una lista de todos los controles planificados para él. En caso de ser jefe de departamento se le mostrarán todos los controles planificados.	
Resultado esperado: El usuario podrá observar todos los planes de clase planificados.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 43: Prueba de aceptación HU8

Resultado de las pruebas:

El resultado de las pruebas realizadas permitió detectar algunas no conformidades, donde las mismas fueron solucionadas satisfactoriamente. A continuación, en la tabla se muestra un resumen de las no conformidades por cada iteración:

Iteración	Cantidad de no conformidades	No conformidad
1	5	5 Errores de interfaz. 7 Errores de validación. 6 Errores ortográficos.
2	10	4 Errores de interfaz. 5 Errores de validación.

		0 Errores ortográficos.
3	2	1 Errores de interfaz. 1 Errores de validación. 0 Errores ortográficos.
4	3	1 Errores de interfaz. 2 Errores de validación. 0 Errores ortográficos.
5	6	3 Errores de interfaz. 2 Errores de validación. 1 Errores ortográficos.
6	5	1 Errores de interfaz. 2 Errores de validación. 2 Errores ortográficos.
7	4	2 Errores de interfaz. 2 Errores de validación. 0 Errores ortográficos.

Tabla 44: Resumen de las no conformidades

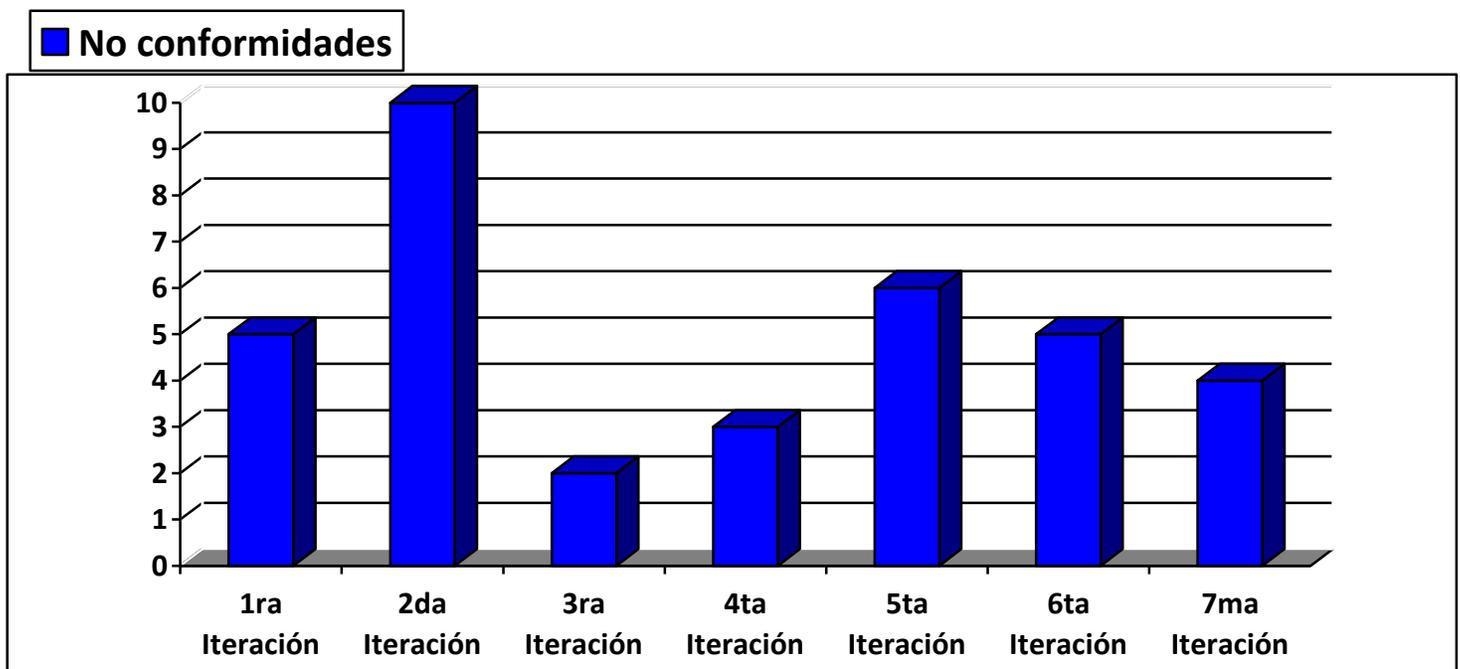


Fig.2- Gráfica del Resultado de las pruebas

3.7 Conclusiones del capítulo

En el capítulo fueron descritas las fases de iteraciones y producción para la propuesta de solución, lo que permitió la obtención de los artefactos necesarios para el desarrollo de la aplicación web. Se realizó el diseño de la base de datos y se desarrollaron las iteraciones planificadas en la fase de planificación. Se permitió definir la estructura del sistema mediante el patrón arquitectónico Modelo-Vista-Plantilla. Se definieron los patrones de diseño con los que se ha trabajado como los patrones GRASP y GoF. Se confeccionaron las tarjetas CRC y Tareas de Ingeniería, para confeccionar las clases más relevantes para las funcionalidades del sistema y componer el conjunto de acciones llevadas a cabo para darle cumplimiento a las distintas historias de usuario respectivamente. Para la verificación y validación del sistema se realizaron pruebas de Aceptación a las distintas historias de usuario algunas con resultado satisfactorio y otras con resultados insatisfactorio, también pruebas unitarias a una parte del código dando éxito la misma.

Conclusiones generales

El desarrollo del presente trabajo de diploma permitió llevar a cabo las tareas de investigación a fin de dar cumplimiento al objetivo general propuesto logrando arribar a las siguientes conclusiones:

- Se realizó un estudio en el ámbito nacional como internacional de las aplicaciones web similares a la propuesta en esta investigación, determinándose las características comunes de las mismas y evidenciando la carencia de una aplicación web capaz de resolver completamente la problemática existente en el departamento de Ingeniería de software de la facultad 2.
- Se justificaron las tecnologías, herramientas y metodología utilizadas en el desarrollo de la investigación, lo que permitió implementar exitosamente la aplicación web.
- Se definieron y describieron las funcionalidades y características de la aplicación web desarrollada, logrando así un mejor entendimiento de la solución a informatizar.
- Se desarrolló una aplicación web que permite el cumplimiento del trabajo metodológico en el departamento de Ingeniería de software de la facultad 2, cumpliendo con las políticas y los estándares de calidad propuestos.
- Se realizaron pruebas unitarias y de aceptación que permitieron la identificación y corrección de no conformidades existentes en la aplicación web desarrollada.

Recomendaciones

Una vez concluido el desarrollo de la aplicación web propuesta y luego de haber cumplido con los objetivos trazados al inicio de la investigación, se recomienda:

Desplegar la aplicación web propuesta al resto de los departamentos docentes de la facultad 2 y de la universidad.

Incorporar nuevas funcionalidades a la aplicación web propuesta utilizando la web semántica, ejemplo: que se propongan habilidades al profesor para redactar el objetivo en el plan de clase, en función de su tipología.

Bibliografía

ADOSSIS, S.A. 2012. KRONOWIN-KGrupos Milenio-6. *Refusiones para el Generador de Horarios y Generador de Grupos Milenio-6.* [En línea] Noviembre de 2012. [Citado el: 10 de Enero de 2017.] <http://www.adossis.es/REFUGHM6.htm..>

Alegsa, Leandro. 2010. Diccionario de informática y tecnologías. *alegsa.com.ar.* [En línea] Mayo de 2010. [Citado el: 13 de Enero de 2017.] <http://www.alegsa.com.ar/Dic/lenguaje%20de%20programacion.php..>

Amaro Calderón, Sarah Dámaris y Valverde Rebaza, Jorge Carlos. 2007. *Metodologías Ágiles.* Facultad de Ciencias Físicas y Matemáticas, Universidad Nacional de Trujillo. Trujillo : s.n., 2007. pág. 37.

Arias Chaves, Michael. 2007. *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software.* 2007.

Bauza Becerra, Jorge Ernesto. 2013. *Sistema de gestión de la planificación de las actividades de los trabajadores docentes en la facultad 2 de la Universidad de las Ciencias Informáticas.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2013. Tesis de grado.

Beck, Kent y Wesley, Addison. 1999. *Extreme Programming Explained.* 1999.

Calabria, Luis y Píriz, Pablo. 2003. *Metodología XP.* Universidad ORT Uruguay. 2003.

Campos León, Arlety y Méndez Méndez, Verónica. 2013. *Sistema de gestión docente metodológica integrado con la plataforma informativa SO3.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2013. Tesis de grado.

Chun, Wesley J. 2012. *Core Python Applications Programming.* 2012.

Cochran, David. 2012. *Twitter Bootstrap Web Development How To.* s.l. : Packt Publishing Ltd, 2012.

Colectivo de autores. 2003. *Preparación pedagógica integral para profesores universitarios.* 2003.

Colegium.com. 2012. Colegium.com. [En línea] 2012. [Citado el: 10 de Enero de 2017.] [http://www.colegium.com/.](http://www.colegium.com/)

Gamma, Erich, y otros. 1994. *Design Patterns.* 1994.

Gauchat, Juan Diego. 2012. *El gran libro de HTML5, CSS3 y Javascript.* Marcombo : s.n., 2012.

Hernández León, Rolando Alfredo y Coello González, Sayda. 2002. *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA*. La Habana : EDUNIV, 2002. pág. 114. ISBN: 959-16-0343-6.

Hogan, Brian P. 2011. *HTML 5 & CSS 3*. 2011.

Martelli, Alex. 2008. *Python. Guía de referencia*. 2008.

Meléndez Valladarez, Sintya Milena, Gaitan, Maria Elizabeth y Pérez Reyes, Neldin Noel. 2016. *METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE*. 2016.

Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). **Letelier, Patricio y Penadés, María del Carmen. 2006.** 2006.

Ministerio de Educación Superior. 2014. Reglamento del trabajo metodológico del ministerio de educación. *Resolución No. 200/2014*. 2014.

—. **2007.** *Reglamento para el Trabajo Docente y Metodológico en la educación superior. Resolución 210/ 07*. Cuba : s.n., 2007.

Ortiz Ocaña, Alexander. 2009. *Temas pedagógicos, didácticos y metodológicos*. Antillas : s.n., 2009. 9583385603.

Osorio, Luis Bernardo Duque. 2008. *Procesos. Guía Didáctica y Módulos*. Colombia : s.n., 2008. s.n.

PostgreSQL. *PostgreSQL*. [En línea] <http://www.postgresql.org.ar/>.

Pressman, Roger S. 2007. *Pressman*. 2007. Vols. IV-V.

Revista de Información, Tecnología y Sociedad. 2017. Revista de Información, Tecnología y Sociedad. [En línea] 10 de mayo de 2017. http://www.revistasbolivianas.org.bo/scielo.php?pid=S1997-40442012000200016&script=sci_arttext..

Severance, Charles. 2012. *Javascript: Designing a language in 10 days*. 2012.

Trejo Quirarte, Rebeca. 2014. *Herramientas de la informática y sus funciones*. 2014.

Vargas Díaz, Osiel Victor y Corzo Guzmán, Taimi Dalila. 2013. *Sistema para la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2013. Tesis de grado.

Referencias

ADOSSIS, S.A. 2012. KRONOWIN-KGrupos Milenio-6. *Refusiones para el Generador de Horarios y Generador de Grupos Milenio-6.* [En línea] Noviembre de 2012. [Citado el: 10 de Enero de 2017.] <http://www.adossis.es/REFUGHM6.htm..>

Alegsa, Leandro. 2010. Diccionario de informática y tecnologías. *alegsa.com.ar.* [En línea] Mayo de 2010. [Citado el: 13 de Enero de 2017.] <http://www.alegsa.com.ar/Dic/lenguaje%20de%20programacion.php..>

Amaro Calderón, Sarah Dámaris y Valverde Rebaza, Jorge Carlos. 2007. *Metodologías Ágiles.* Facultad de Ciencias Físicas y Matemáticas, Universidad Nacional de Trujillo. Trujillo : s.n., 2007. pág. 37.

Arias Chaves, Michael. 2007. *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software.* 2007.

Bauza Becerra, Jorge Ernesto. 2013. *Sistema de gestión de la planificación de las actividades de los trabajadores docentes en la facultad 2 de la Universidad de las Ciencias Informáticas.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2013. Tesis de grado.

Beck, Kent y Wesley, Addison. 1999. *Extreme Programming Explained.* 1999.

Calabria, Luis y Píriz, Pablo. 2003. *Metodología XP.* Universidad ORT Uruguay. 2003.

Campos León, Arlety y Méndez Méndez, Verónica. 2013. *Sistema de gestión docente metodológica integrado con la plataforma informativa SO3.* Universidad de las Ciencias Informáticas. La Habana : s.n., 2013. Tesis de grado.

Chun, Wesley J. 2012. *Core Python Applications Programming.* 2012.

Cochran, David. 2012. *Twitter Bootstrap Web Development How To.* s.l. : Packt Publishing Ltd, 2012.

Colectivo de autores. 2003. *Preparación pedagógica integral para profesores universitarios.* 2003.

Colegium.com. 2012. Colegium.com. [En línea] 2012. [Citado el: 10 de Enero de 2017.] [http://www.colegium.com/.](http://www.colegium.com/)

Gamma, Erich, y otros. 1994. *Desing Patterns.* 1994.

Gauchat, Juan Diego. 2012. *El gran libro de HTML5, CSS3 y Javascript.* Marcombo : s.n., 2012.

Hernández León, Rolando Alfredo y Coello González, Sayda. 2002. *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA*. La Habana : EDUNIV, 2002. pág. 114. ISBN: 959-16-0343-6.

Hogan, Brian P. 2011. *HTML 5 & CSS 3*. 2011.

Martelli, Alex. 2008. *Python. Guía de referencia*. 2008.

Meléndez Valladarez, Sintya Milena, Gaitan, Maria Elizabeth y Pérez Reyes, Neldin Noel. 2016. *METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE*. 2016.

Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). **Letelier, Patricio y Penadés, María del Carmen. 2006.** 2006.

Ministerio de Educación Superior. 2014. Reglamento del trabajo metodológico del ministerio de educación. *Resolución No. 200/2014*. 2014.

—. **2007.** *Reglamento para el Trabajo Docente y Metodológico en la educación superior. Resolución 210/ 07*. Cuba : s.n., 2007.

Ortiz Ocaña, Alexander. 2009. *Temas pedagógicos, didácticos y metodológicos*. Antillas : s.n., 2009. 9583385603.

Osorio, Luis Bernardo Duque. 2008. *Procesos. Guía Didáctica y Módulos*. Colombia : s.n., 2008. s.n.

PostgreSQL. *PostgreSQL*. [En línea] <http://www.postgresql.org.ar/>.

Pressman, Roger S. 2007. *Pressman*. 2007. Vols. IV-V.

Revista de Información, Tecnología y Sociedad. 2017. Revista de Información, Tecnología y Sociedad. [En línea] 10 de mayo de 2017. http://www.revistasbolivianas.org.bo/scielo.php?pid=S1997-40442012000200016&script=sci_arttext..

Severance, Charles. 2012. *Javascript: Designing a language in 10 days*. 2012.

Trejo Quirarte, Rebeca. 2014. *Herramientas de la informática y sus funciones*. 2014.

Vargas Díaz, Osiel Victor y Corzo Guzmán, Taimi Dalila. 2013. *Sistema para la gestión de los procesos que se desarrollan en el Departamento Metodológico Central de Ingeniería y Gestión de Software*. Universidad de las Ciencias Informáticas. La Habana : s.n., 2013. Tesis de grado.